

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Інформаційна система для поціновувачів настільно-рольової гри
“Dungeons and Dragons” з інтеграцією AI»**

Виконав студент групи ПЗ-22-4
ЛОМОНОСОВ Ігор Олегович

Керівник роботи:
САВІЦЬКИЙ Роман Святославович

Рецензент:
ЄФРЕМОВ Юрій Миколайович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛОМОНОСОВ Ігор Олегович**

Керівник роботи: **САВІЦЬКИЙ Роман Святославович**

Тема роботи: **«Інформаційна система для поціновувачів настільно-рольової гри “Dungeons and Dragons” з інтеграцією AI»,**

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**

Вихідні дані для роботи: Об'єктом дослідження є процеси проектування та програмної реалізації інформаційних систем підтримки ігрового процесу в настільно-рольових іграх. Предметом дослідження є методи проектування веборієнтованих інформаційних систем.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Савіцький Р.С.	03.03.2026р.	03.03.2026р
2	Савіцький Р.С.	03.03.2026р.	03.03.2026р.
3	Савіцький Р.С.	03.03.2026р.	03.03.2026р.

ЗМІСТ

РЕФЕРАТ.....	5
ABSTRACT	6
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	11
1.1 Постановка задачі.....	11
1.2 Аналіз аналогів програмного продукту	14
1.3 Вибір архітектури інформаційної системи для підтримки ігрового процесу Dungeons & Dragons	21
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	26
Висновки до першого розділу.....	34
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПОЦІНОВУВАЧІВ DUNGEONS & DRAGONS	36
2.1 Визначення функціональних і нефункціональних вимог до системи	36
2.2 Розробка бази даних системи.....	47
2.3 Проєктування та реалізація алгоритмів роботи системи	53
2.4 Реалізація інформаційної системи для поціновувачів Dungeons & Dragons	64
Висновки до другого розділу	70
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНФОРМАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОЦІНОВУВАЧІВ DUNGEONS & DRAGONS.....	73
3.1 Порядок встановлення та налаштування параметрів системи	73
3.2 Структура інтерфейсу інформаційної системи	77
3.3 Тестування інформаційної системи	86
Висновки до третього розділу.....	93
ВИСНОВКИ	95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	98

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

РЕФЕРАТ

Кваліфікаційна робота містить 102 сторінки, з яких текст викладений на 100 сторінках, 28 рисунків, 10 таблиць, 30 використаних джерел та 1 додаток.

Метою кваліфікаційної роботи є проектування та розробка інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента. Система призначена для зручного доступу до довідкових матеріалів, автоматизованого наповнення бази даних, україномовної локалізації контенту та підтримки допоміжних інструментів для гравців.

У першому розділі розглянуто предметну область настільно-рольових ігор, визначено основні проблеми використання матеріалів Dungeons & Dragons та проаналізовано існуючі програмні рішення. У другому розділі виконано проектування системи: визначено ролі користувачів, функціональні вимоги, структуру бази даних MongoDB Atlas та основні алгоритми роботи. У третьому розділі описано порядок встановлення, використання й тестування системи.

У результаті виконання роботи розроблено вебзастосунок на базі Next.js 14, TypeScript, React, MongoDB Atlas, NextAuth.js та Gemini API. Система містить wiki-довідники Dungeons & Dragons, модуль створення й редагування персонажів, будівник сутичок, інструмент кидання дайсів, AI-чат, адміністративну панель, механізм парсингу даних з Open5e REST API та AI-переклад контенту українською мовою.

Практичне значення роботи полягає у створенні програмного продукту, який може бути використаний гравцями та майстрами гри для організації довідкової інформації, підготовки до ігрових сесій, ведення персонажів і спрощення роботи з матеріалами Dungeons & Dragons українською мовою.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, DUNGEONS & DRAGONS, AI-АСИСТЕНТ, ПАРСИНГ ДАНИХ, НАСТІЛЬНО-РОЛЬОВА ГРА.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ABSTRACT

The qualification work contains 102 pages, of which the text is presented on 100 pages, 28 figures, 10 tables, 30 references, and 1 appendix.

The aim of the qualification work is to design and develop an information system for fans of the tabletop role-playing game Dungeons & Dragons with the integration of an AI assistant. The system is intended to provide convenient access to reference materials, automate database population, support Ukrainian localization of content, and provide auxiliary tools for players.

The first section examines the subject area of tabletop role-playing games, identifies the main problems related to the use of Dungeons & Dragons materials, and analyzes existing software solutions. The second section presents the system design: user roles, functional requirements, MongoDB Atlas database structure, and main system algorithms are defined. The third section describes the installation, usage, and testing of the system.

As a result of the work, a web application was developed using Next.js 14, TypeScript, React, MongoDB Atlas, NextAuth.js, and the Gemini API. The system includes Dungeons & Dragons wiki reference sections, a module for creating and editing characters, an encounter builder, a dice roller, an AI chat, an administrative panel, a mechanism for parsing data from the Open5e REST API, and AI-based translation of content into Ukrainian.

The practical value of the work lies in the creation of a software product that can be used by players and game masters to organize reference information, prepare for game sessions, manage characters, and simplify work with Dungeons & Dragons materials in Ukrainian.

Keywords: INFORMATION SYSTEM, DUNGEONS & DRAGONS, AI ASSISTANT, DATA PARSING, TABLETOP ROLE-PLAYING GAME.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI – Artificial Intelligence, штучний інтелект

API – Application Programming Interface, програмний інтерфейс застосунку

CRUD – Create, Read, Update, Delete, базові операції створення, читання, оновлення та видалення даних

D&D – Dungeons & Dragons, настільно-рольова гра у жанрі фентезі

GM – Game Master, ведучий гри, який організовує ігровий процес

HP – Hit Points, очки здоров'я персонажа або монстра

JWT – JSON Web Token, формат токена для передачі даних автентифікації

NPC – Non-Player Character, неігровий персонаж

ODM – Object Document Mapper, інструмент для роботи з документоорієнтованою базою даних через моделі

RPG – Role-Playing Game, рольова гра

SRD – System Reference Document, відкритий довідковий документ із базовими правилами Dungeons & Dragons

UI – User Interface, інтерфейс користувача

UML – Unified Modeling Language, уніфікована мова моделювання

UX – User Experience, користувацький досвід

БД – база даних

НРІ – настільно-рольова гра

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. Dungeons & Dragons є однією з найпопулярніших настільно-рольових ігор, ігровий процес якої потребує постійного звернення до великого обсягу довідкової інформації: правил, характеристик істот, заклинань, класів, рас, спорядження, ігрових станів та інших матеріалів [27]. Для україномовних користувачів додатковою проблемою є мовний бар'єр, оскільки значна частина матеріалів доступна англійською мовою. Використання звичайних засобів перекладу не завжди забезпечує коректну передачу специфічної термінології та ігрових механік. Тому актуальною є розробка спеціалізованої інформаційної системи, яка поєднує довідковий функціонал, автоматизоване наповнення бази даних, AI-переклад і допоміжні інструменти для гравців та майстрів гри.

Окремої уваги потребує організація самих ігрових матеріалів. Під час підготовки до сесії майстру гри або гравцю часто доводиться працювати з кількома джерелами одночасно: правилами, описами монстрів, заклинаннями, аркушами персонажів, нотатками та додатковими таблицями. Через це пошук потрібної інформації може займати більше часу, ніж сама ігрова дія. Централізована система з збереженням персонажів, інструментом кидання дайсів і будівником сутичок дає змогу скоротити кількість розрізнених інструментів і зробити підготовку до гри більш зручною. Використання відкритого джерела даних Open5e дозволяє автоматизувати наповнення довідників і зменшити обсяг ручної роботи під час підтримки контенту [21].

Метою роботи є проектування та розробка інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтегрованим AI-асистентом, яка забезпечує зручний доступ до довідкового контенту, автоматизоване завантаження даних, україномовну локалізацію матеріалів і підтримку допоміжних інструментів для організації ігрового процесу.

Завданням роботи є:

- проаналізувати предметну область настільно-рольових ігор та особливості використання довідкових матеріалів Dungeons & Dragons;

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

- розглянути існуючі програмні рішення для підтримки ігрового процесу та визначити їхні переваги й обмеження;
- сформулювати функціональні та нефункціональні вимоги до інформаційної системи;
- спроектувати архітектуру Full-Stack вебзастосунку на базі Next.js 14;
- розробити структуру бази даних MongoDB Atlas для збереження користувачів, персонажів і довідкового контенту;
- реалізувати модуль автоматизованого завантаження даних з Open5e REST API;
- реалізувати модуль AI-перекладу довідкового контенту за допомогою Gemini API;
- реалізувати AI-асистента для надання користувачам контекстної допомоги;
- розробити користувацькі інструменти для створення та редагування персонажів, кидання дайсів і формування сутичок;
- розробити адміністративну панель для керування контентом, перекладом і користувачами;
- виконати тестування основних функціональних модулів інформаційної системи.

Об’єктом дослідження є процеси проектування та програмної реалізації інформаційних систем підтримки ігрового процесу в настільно-рольових іграх.

Предметом дослідження є методи проектування веборієнтованих інформаційних систем, алгоритми отримання та обробки довідкових даних, засоби інтеграції зовнішніх API, методи організації бази даних і підходи до використання AI-сервісів для локалізації та підтримки користувачів.

Методи дослідження. У роботі використано методи системного аналізу предметної області, об’єктно-орієнтованого проектування із застосуванням UML, методи проектування баз даних, методи вебпрограмування, API-інтеграції, ручного функціонального тестування, а також підходи до використання великих мовних моделей для перекладу та формування відповідей користувачу.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Практичне значення одержаних результатів полягає у створенні програмного продукту, який може бути використаний гравцями та майстрами гри Dungeons & Dragons для організації довідкової інформації, підготовки до ігрових сесій і спрощення роботи з правилами українською мовою. Розроблена система дозволяє централізовано зберігати ігровий контент, автоматизовано наповнювати базу даних матеріалами з відкритого джерела, виконувати AI-переклад довідкових записів, створювати й редагувати персонажів, проводити кидки дайсів, формувати бойові сутички та отримувати контекстні відповіді від AI-асистента.

Апробація результатів дослідження. Основні положення та результати кваліфікаційної роботи були представлені, обговорені та подані до друку у матеріалах науково-практичних конференцій [1, 2].

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1 Постановка задачі

Одним із напрямів, який потребує спеціалізованих програмних рішень, є підтримка настільно-рольових ігор. Настільно-рольові ігри поєднують у собі елементи колективної творчості, стратегічного мислення, рольової взаємодії та роботи з великими обсягами структурованої інформації. Серед таких ігор особливе місце займає Dungeons & Dragons, оскільки вона має розвинену систему правил, велику кількість ігрових сутностей та активну спільноту користувачів.

Ігровий процес Dungeons & Dragons передбачає постійне звернення до правил, характеристик персонажів, описів монстрів, заклинань, магічних предметів, рас, класів, передісторій, здібностей, ігрових станів, спорядження та окремих розділів правил [27]. Значна частина цієї інформації має складну структуру та взаємопов'язана між собою. Наприклад, створення персонажа потребує одночасного врахування раси, класу, передісторії, характеристик, володінь, рятівних кидків, навичок, спорядження та правил розвитку персонажа. Аналогічно, підготовка ігрового зіткнення потребує вибору монстрів, визначення ініціативи, урахування складу групи персонажів, поточних очок здоров'я та можливих станів учасників бою.

Традиційний підхід до організації гри часто передбачає використання паперових книг правил, окремих електронних документів, таблиць, нотаток або різних вебресурсів. Такий спосіб роботи створює низку незручностей. По-перше, користувачеві необхідно витратити значний час на пошук потрібної інформації. По-друге, ігрові дані можуть бути розміщені в різних джерелах, що ускладнює їх систематизацію. По-третє, під час активної ігрової сесії тривалий пошук правил або характеристик істот може порушувати динаміку гри та відволікати учасників від рольової взаємодії. Тому актуальним є створення єдиного веборієнтованого середовища, яке дозволяє зберігати, переглядати, шукати, фільтрувати й

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

використовувати основні матеріали Dungeons & Dragons у зручному цифровому вигляді.

Окрему складність для україномовних користувачів становить мовний бар'єр, оскільки значна частина матеріалів Dungeons & Dragons подана англійською мовою. При цьому проблема полягає не лише в перекладі тексту, а й у збереженні змісту ігрових термінів. У межах правил D&D окремі поняття мають чітко визначене механічне значення, тому неточне тлумачення може призвести до неправильного розуміння дії заклинання, здібності персонажа, характеристики монстра або правила взаємодії [27]. Через це під час локалізації важливо враховувати не тільки мовне значення терміна, а й контекст його використання в ігровій системі.

Для вирішення проблеми доцільним є використання AI-асистента, здатного працювати з текстовими запитамися користувача, пояснювати складні фрагменти правил, допомагати з перекладом і адаптацією ігрового контенту [9, 10]. Інтеграція AI-модуля в інформаційну систему дозволяє не обмежуватися лише статичним довідником, а надати користувачам інструмент для інтерактивної роботи з матеріалами. Такий підхід є особливо корисним для новачків, які ще не володіють усією термінологією, а також для майстрів гри, яким необхідно швидко отримувати пояснення, уточнення або допоміжну інформацію під час підготовки до сесії.

Окремим завданням є наповнення бази даних довідковими матеріалами. Через значну кількість ігрових сутностей ручне внесення таких даних є незручним, повільним і схильним до помилок. До таких сутностей належать монстри, заклинання, раси, класи, передісторії, здібності, магічні предмети, ігрові стани, спорядження та розділи правил. Для зменшення обсягу ручної роботи доцільно реалізувати механізм парсингу даних із зовнішнього відкритого джерела, зокрема Open5e REST API [21]. Це дозволяє автоматизувати завантаження структурованих даних, зберігати їх у базі даних MongoDB Atlas і надалі використовувати у відповідних довідниках системи.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Розроблювана інформаційна система має забезпечувати не лише перегляд довідкової інформації, а й практичну підтримку ігрового процесу. З цією метою до складу системи доцільно включити модулі пошуку та фільтрації, механізм кидання дайсів, інтерактивний лист персонажа, засоби створення персонажа та будівник зіткнень. Наявність таких інструментів дозволяє використовувати систему не лише як електронний довідник, а як комплексну платформу для підготовки та проведення ігрових сесій.

Метою розробки є створення інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента, яка забезпечує централізований доступ до ігрових матеріалів, автоматизує процес наповнення бази даних, підтримує пошук і фільтрацію контенту, надає інструменти для створення персонажів, роботи з ігровими зіткненнями та взаємодії з AI-асистентом.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- створити довідкові розділи для перегляду монстрів, заклинань, рас, класів, передісторій, здібностей, магічних предметів, ігрових станів, спорядження та розділів правил;
- реалізувати адміністративну панель для запуску парсингу, масового й точкового перекладу, редагування та видалення записів у базі даних;
- забезпечити механізм повторного парсингу даних без втрати вже збережених україномовних перекладів;
- реалізувати створення та редагування персонажа з підтримкою характеристик, рятівних кидків, навичок, рівня майстерності та автоматичного розрахунку бонусу майстерності;
- реалізувати допоміжні ігрові інструменти, зокрема механізм кидання дайсів, інтерактивний аркуш персонажа та будівник зіткнень із можливістю додавання персонажів користувача, ручного виставлення ініціативи та зміни НР учасників бою;
- забезпечити відображення зображень монстрів у каталозі та на сторінках детального перегляду, якщо такі зображення доступні у джерелі даних.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

У результаті виконання зазначених задач має бути створено вебзастосунок, який поєднує функції електронного довідника, інструмента підтримки ігрового процесу, адміністративної системи керування даними та AI-асистента. Така система дозволить зменшити час, необхідний для пошуку й опрацювання ігрової інформації, спростити підготовку до сесій Dungeons & Dragons, підвищити зручність роботи з англomовними матеріалами та зробити ігровий контент доступнішим для українськомовних користувачів.

Отже, постановка задачі полягає у проєктуванні та реалізації веборієнтованої інформаційної системи, яка забезпечує централізоване зберігання ігрових матеріалів Dungeons & Dragons, автоматизоване наповнення бази даних, інструменти пошуку, фільтрації та адміністрування даних, а також інтеграцію AI-асистента для підтримки користувачів під час роботи з ігровим контентом.

1.2 Аналіз аналогів програмного продукту

Для обґрунтування доцільності розробки інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons необхідно розглянути наявні цифрові рішення, які використовуються для підтримки ігрового процесу, зберігання довідкових матеріалів, створення персонажів, управління кампаніями та роботи з правилами. Аналіз аналогів дозволяє визначити типовий функціонал програмних продуктів цього класу, їх переваги та недоліки, а також сформулювати вимоги до власної системи.

Для порівняння було обрано такі рішення: D&D Beyond [6, 7], Roll20 [25], Foundry Virtual Tabletop [8] та World Anvil [30]. Вони належать до цифрових інструментів, що використовуються для підтримки настільно-рольових ігор, однак мають різну спеціалізацію. D&D Beyond орієнтований на роботу з офіційними матеріалами Dungeons & Dragons, Roll20 і Foundry Virtual Tabletop – на проведення ігрових сесій у форматі віртуального ігрового столу, а World Anvil – на створення та структурування вигаданих світів і кампаній.

D&D Beyond є офіційним цифровим інструментарієм для Dungeons & Dragons [6]. Платформа надає користувачам доступ до електронних правил,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

довідкових матеріалів, створення персонажів, цифрових листів персонажа, керування кампаніями та інструментів для майстра гри. Однією з ключових переваг D&D Beyond є глибока інтеграція з офіційними матеріалами Dungeons & Dragons, що робить сервіс зручним для гравців, які використовують ліцензійний контент. Система також підтримує функції створення персонажів і роботу з ігровими матеріалами в одному середовищі [7].

На рисунку 1.1. зображено головну сторінку D&D Beyond [6].

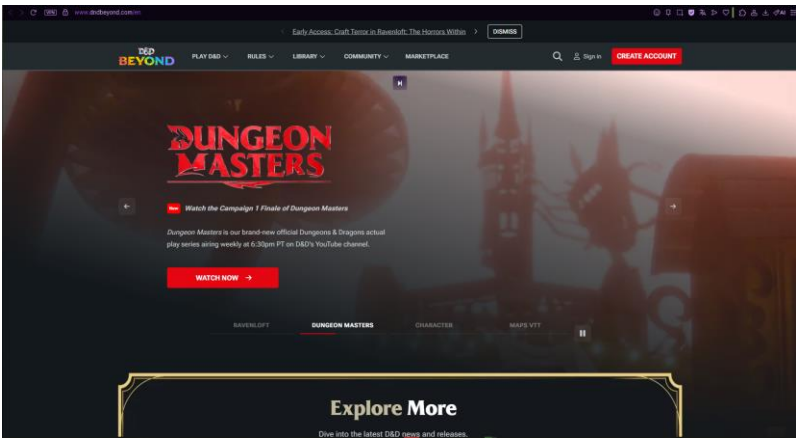


Рис. 1.1. Головна сторінка D&D Beyond

На рисунку 1.2. зображено інтерфейс створення персонажа на сервісі D&D Beyond [7].

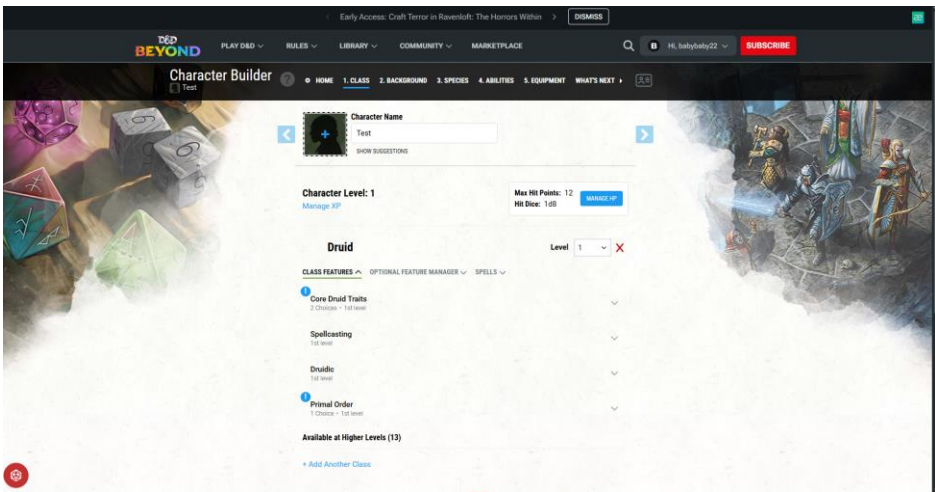


Рис. 1.2. Створення персонажа у сервісі D&D Beyond

Водночас D&D Beyond має низку обмежень у контексті поставленої задачі. Насамперед сервіс орієнтований на англomовну аудиторію та офіційний контент. Для українських користувачів це створює проблему мовного бар'єра, особливо під час роботи з великими текстовими описами правил, заклинань, монстрів або

класових особливостей. Також платформа не орієнтована на автоматизовану локалізацію матеріалів українською мовою за допомогою AI-асистента. Отже, D&D Beyond є функціонально потужним рішенням, однак не вирішує повністю задачу мультимовної підтримки та контекстного пояснення матеріалів для україномовної спільноти.

Roll20 є популярною віртуальною платформою для проведення настільно-рольових ігор онлайн [25]. Основними можливостями Roll20 є віртуальний ігровий стіл, робота з картами, токенами, цифровими дайсами, листами персонажів, довідниками та інструментами для організації кампаній. Платформа дає змогу проводити ігрові сесії дистанційно, забезпечуючи взаємодію між гравцями та майстром гри в режимі реального часу. Завдяки підтримці цифрових листів персонажів і автоматизації частини ігрових обчислень Roll20 є зручним інструментом для онлайн-гри.

Інтерфейс головної сторінки Roll20 зображено на рисунку 1.3 [25].

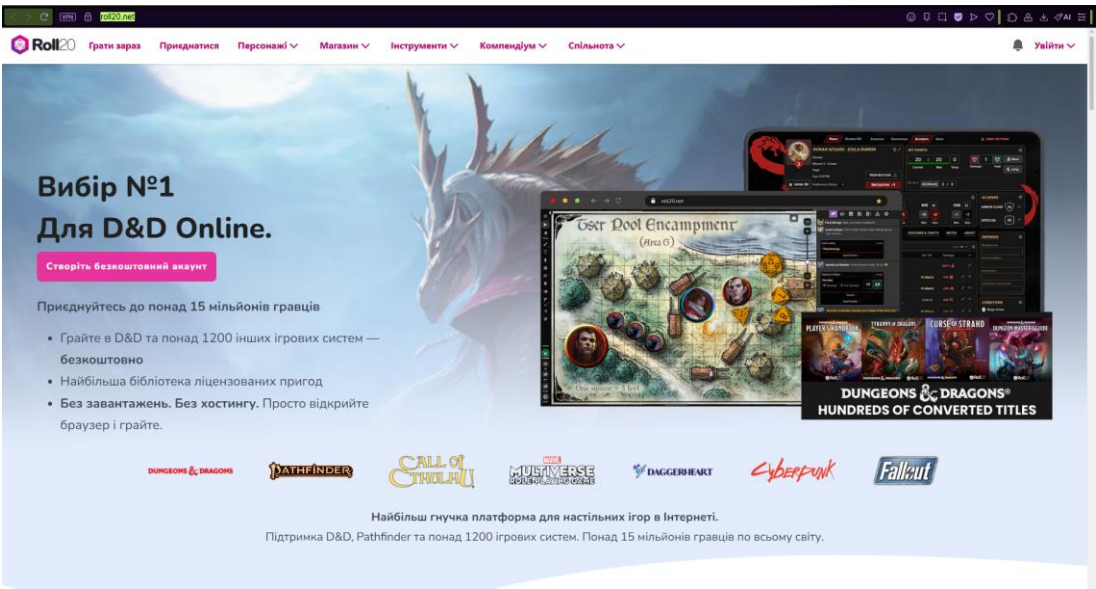


Рис. 1.3. Головна сторінка Roll20

Основним недоліком Roll20 у межах даного дослідження є те, що платформа передусім орієнтована на проведення сесій у форматі віртуального столу, а не на створення локалізованої довідкової системи з AI-асистентом. Хоча Roll20 має інструменти для роботи з правилами та персонажами, його основна функція полягає у забезпеченні ігрового простору. Для задачі автоматизованого

наповнення бази даних, перекладу контенту, редагування входжень через адміністративну панель та інтерактивного пояснення правил такий підхід є недостатнім.

Foundry Virtual Tabletop – це програмне забезпечення для створення самостійно розгорнутого віртуального ігрового столу [8]. Платформа підтримує багато ігрових систем, модулі розширення, карти, токени, персонажні листи, автоматизацію бойових механік та інші інструменти для проведення онлайн-сесій. Важливою перевагою Foundry VTT є можливість самостійного розгортання сервера та широкі можливості розширення за допомогою модулів. Це робить платформу гнучкою для досвідчених користувачів і майстрів гри, які хочуть налаштовувати ігрове середовище під власні потреби.

Головну сторінку сервісу Foundry VTT зображено на рисунку 1.4 [8].

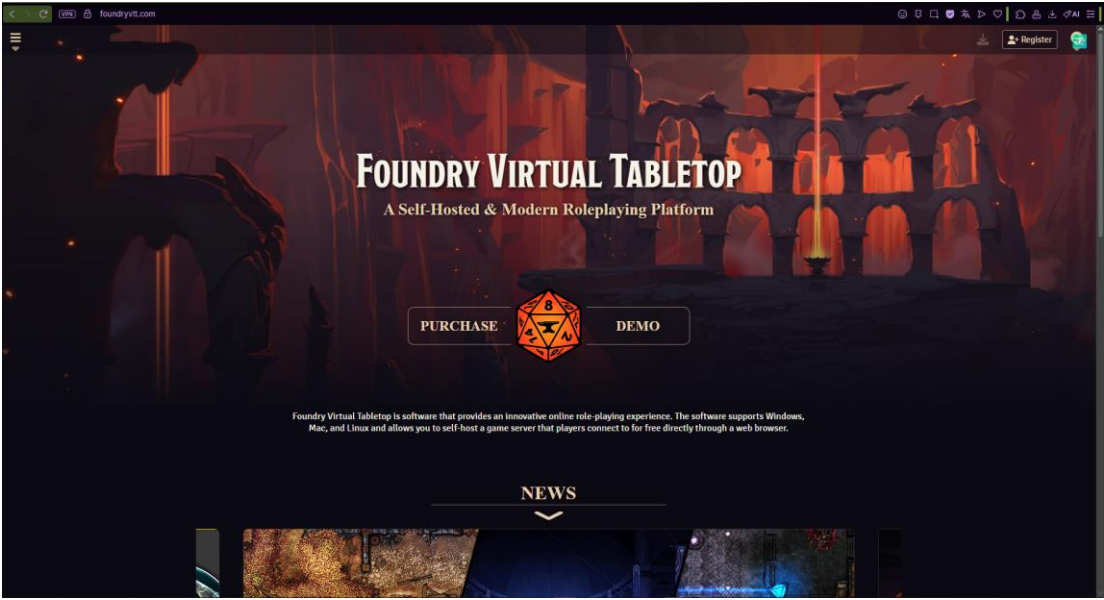


Рис. 1.4. Головна сторінка Foundry Virtual Tabletop

Разом із тим Foundry VTT має вищий поріг входження для нових користувачів, оскільки потребує налаштування середовища, встановлення систем, модулів і додаткових ресурсів. Платформа більше орієнтована на проведення ігрових сесій, ніж на побудову спеціалізованого вебдовідника з автоматизованим парсингом, перекладом та AI-чатом. Для користувачів, яким потрібно швидко отримати структуровану інформацію про монстрів, заклинання, раси, класи, магічні предмети або створити персонажа без складного

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

налаштування VTT-середовища, Foundry може бути надмірно складним рішенням.

World Anvil є вебплатформою для створення, зберігання та організації вигаданих світів [30]. Вона орієнтована на авторів, майстрів гри та розробників кампаній, які потребують інструментів для ведення енциклопедії світу, створення статей, карт, часових ліній, описів персонажів, держав, подій і сюжетних елементів. Для настільно-рольових ігор World Anvil може використовуватися як інструмент підготовки кампанії, зберігання лору та організації наративної інформації.

На рисунку 1.5. зображено головну сторінку сервісу World Anvil [30].



Рис. 1.5. Головна сторінка World Anvil

Однак World Anvil не є спеціалізованим довідником Dungeons & Dragons і не фокусується на автоматизованій роботі з механіками D&D. Його основна цінність полягає у worldbuilding, тобто у створенні та структуризації вигаданого світу. Система не вирішує безпосередньо задачі автоматичного парсингу D&D-даних із відкритого API, локалізації англомовних ігрових матеріалів, створення інтерактивного листа персонажа або будівника зіткнень на основі монстрів і персонажів. Тому World Anvil можна розглядати як корисний інструмент для підготовки сюжетної частини кампанії, але не як повноцінний аналог розроблюваної інформаційної системи.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		18

Порівняльну характеристику розглянутих аналогів наведено у таблиці 1.1.

Таблиця 1.1

Порівняльна характеристика аналогів

Назва рішення	Основне призначення	Основні можливості	Переваги	Обмеження
D&D Beyond	Офіційний цифровий інструментарій для Dungeons & Dragons	Створення персонажів, листи персонажів, довідники правил, кампанії, інструменти для гри	Офіційний контент, зручний character builder, інтеграція з D&D-матеріалами	Орієнтація на англomовний контент, відсутність спеціалізованої AI-локалізації українською мовою
Roll20	Віртуальний ігровий стіл для проведення онлайн-сесій	Карти, токени, дайси, листи персонажів, rulebooks, автоматизація частини ігрових дій	Зручний формат онлайн-гри, підтримка багатьох ігор, інтерактивна взаємодія гравців	Основний акцент на VTT, а не на локалізованому довіднику та AI-асистенті
Foundry Virtual Tabletop	Self-hosted платформа для проведення рольових ігор онлайн	Віртуальний стіл, модулі, карти, токени, підтримка багатьох систем, розширення через API	Гнучкість, самостійне розгортання, широкі можливості модифікації	Вищий поріг входження, потреба в налаштуванні, відсутність вбудованого контекстного AI-перекладу
World Anvil	Платформа для worldbuilding та керування кампаніями	Статті, карти, часові лінії, описи персонажів, організація лору	Зручне структурування вигаданих світів і сюжетної інформації	Не є спеціалізованим D&D-довідником, не вирішує задачі парсингу, перекладу та інтерактивного листа персонажа

Аналіз аналогів показав, що наявні рішення мають значний функціональний потенціал, однак здебільшого орієнтовані або на офіційний

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						19

англомовний контент, або на проведення онлайн-сесій, або на організацію кампаній і вигаданих світів. Типовими функціями систем цього класу є перегляд довідкових матеріалів, створення персонажів, робота з листами персонажів, використання цифрових дайсів, організація кампаній, проведення онлайн-сесій та підтримка майстра гри.

Водночас аналіз показав, що для україномовних користувачів досі залишаються невирішеними окремі питання. Насамперед це стосується доступності матеріалів українською мовою та правильного передавання спеціалізованої термінології Dungeons & Dragons. Оскільки правила гри містять багато понять із конкретним механічним значенням, неточний переклад може ускладнювати розуміння заклинань, здібностей, характеристик монстрів або окремих ігрових ситуацій. Крім того, користувачам потрібне єдине середовище для зберігання довідкового контенту, швидкого пошуку потрібної інформації, автоматизованого наповнення бази даних і отримання контекстної допомоги під час роботи з матеріалами гри.

Важливою особливістю більшості розглянутих аналогів є те, що повний або розширений функціонал доступний лише за умови оформлення платної підписки, придбання окремого цифрового контенту або купівлі ліцензії. Це може створювати додатковий бар'єр для користувачів, які лише починають знайомство з Dungeons & Dragons або не мають можливості оплачувати кілька окремих сервісів для роботи з правилами, персонажами, кампаніями та ігровими матеріалами. У цьому контексті актуальним є створення власної інформаційної системи, яка поєднує довідкові матеріали, інструменти підтримки ігрового процесу, адміністративне керування даними та AI-асистента в межах єдиного вебзастосунку.

Розроблювана інформаційна система відрізняється від розглянутих аналогів тим, що поєднує довідкові матеріали Dungeons & Dragons, автоматизований парсинг даних з Open5e REST API, адміністративну панель для перекладу, AI-асистента на базі Gemini, механізм кидання дайсів, інтерактивний лист персонажа та будівник зіткнень. Таким чином, система орієнтована не лише

на перегляд даних, а й на комплексну підтримку підготовки та проведення ігрового процесу.

Отже, проведений аналіз підтверджує доцільність розробки власного програмного продукту. На відміну від існуючих рішень, запропонована система спрямована на поєднання функцій електронного довідника, інструмента підтримки гравця та майстра гри, адміністративного модуля керування даними й AI-асистента для роботи з ігровим контентом. Це дозволяє зробити матеріали Dungeons & Dragons доступнішими для українськомовної аудиторії та зменшити час, необхідний для пошуку, перекладу й опрацювання правил під час підготовки до гри.

1.3 Вибір архітектури інформаційної системи для підтримки ігрового процесу Dungeons & Dragons

Архітектура визначає загальну структуру системи, спосіб взаємодії її компонентів, принципи обробки даних, масштабування, підтримки та подальшого розвитку. Оскільки розроблювана система має поєднувати довідкові матеріали, механізми автентифікації користувачів, адміністративну панель, парсинг зовнішніх даних, роботу з базою даних, AI-асистента та ігрові інструменти, архітектурне рішення має забезпечувати гнучкість, зрозумілу модульну структуру та можливість інтеграції із зовнішніми сервісами.

Для реалізації подібних інформаційних систем можуть використовуватися різні архітектурні підходи. Одним із найпоширеніших є клієнт-серверна архітектура, у межах якої клієнтська частина відповідає за відображення інтерфейсу та взаємодію з користувачем, а серверна частина – за обробку запитів, роботу з базою даних, бізнес-логіку та інтеграцію із зовнішніми сервісами. Такий підхід є доцільним для вебзастосунків, оскільки дозволяє розділити відповідальність між інтерфейсом користувача та серверною логікою.

Іншим поширеним підходом є монолітна архітектура, у якій основні компоненти системи розміщуються в межах одного програмного застосунку. Перевагою такого підходу є простіше розгортання, легша підтримка на початкових етапах розробки та зручність контролю взаємодії між модулями.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

Водночас надмірне ускладнення монолітної системи може знижувати її гнучкість, якщо не передбачити логічного поділу на окремі модулі.

Мікросервісна архітектура передбачає поділ системи на набір незалежних сервісів, кожен із яких відповідає за окрему частину функціональності. Наприклад, окремими сервісами можуть бути автентифікація, довідники, AI-модуль, обробка перекладів або адміністрування. Такий підхід забезпечує високу масштабованість і незалежність компонентів, однак для бакалаврського дипломного проєкту він є надмірно складним. Мікросервісна архітектура потребує додаткового налаштування взаємодії сервісів, окремого розгортання компонентів, моніторингу, керування помилками та складнішої підтримки.

Також для вебзастосунків часто використовується MVC-підхід, який передбачає поділ логіки системи на модель, представлення та контролер. Модель відповідає за дані та правила їх обробки, представлення – за відображення інформації користувачу, а контролер – за обробку запитів і взаємодію між моделлю та представленням. У сучасних Full-Stack фреймворках цей підхід може реалізовуватися не в класичному вигляді, але його принципи зберігаються через поділ системи на компоненти інтерфейсу, серверні обробники, моделі даних і службові модулі.

Для розроблюваної інформаційної системи обрано веборієнтовану клієнт-серверну архітектуру з елементами модульної Full-Stack побудови. Основою програмного продукту є фреймворк Next.js, який використовується як для реалізації клієнтської частини, так і для організації серверної логіки [17, 18]. Такий підхід дозволяє створити єдиний застосунок, у якому інтерфейс користувача, серверні запити, API-обробники, авторизація та інтеграція з базою даних розміщуються в межах одного технологічного середовища.

Узагальнену архітектуру інформаційної системи наведено на рисунку 1.6.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

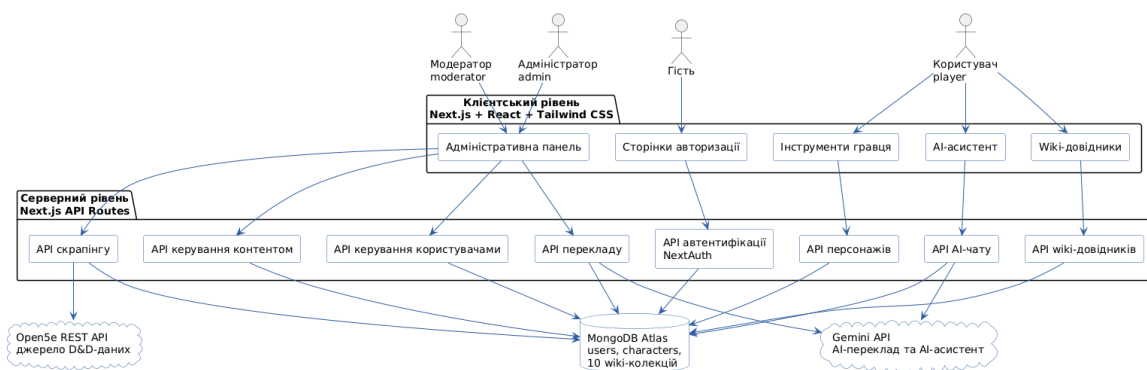


Рис. 1.6. Узагальнена архітектура інформаційної системи для підтримки ігрового процесу Dungeons & Dragons

Як видно з рисунка 1.6, база даних MongoDB Atlas використовується для зберігання користувачів, персонажів і довідкового контенту. Open5e REST API виступає зовнішнім джерелом ігрових даних, а Gemini API використовується для AI-перекладу та роботи AI-асистента. Взаємодія із зовнішніми сервісами виконується через серверну частину, що дозволяє не передавати ключі доступу безпосередньо до клієнтського коду.

Клієнтська частина системи відповідає за відображення сторінок довідників, форм реєстрації та входу, головної сторінки, інтерфейсу AI-асистента, адміністративної панелі, інструмента кидання дайсів, інтерактивного листа персонажа та будівника зіткнень. Для побудови інтерфейсу використовується компонентний підхід. Кожна сторінка або функціональний блок реалізується у вигляді окремих компонентів, що спрощує повторне використання елементів, підтримку коду та подальше розширення системи.

Серверна частина системи відповідає за обробку запитів користувачів, взаємодію з базою даних MongoDB Atlas, виконання операцій створення, читання, оновлення та видалення даних, перевірку прав доступу, роботу з механізмом автентифікації, запуск парсингу, обробку перекладів і звернення до AI-сервісу Gemini. Такий поділ дозволяє приховати критичну логіку від клієнтської частини та забезпечити контрольований доступ до даних.

Для зберігання даних у системі використовується MongoDB Atlas [14]. У базі зберігаються облікові записи користувачів, персонажі, а також довідкові матеріали, отримані з Open5e REST API: монстри, заклинання, магичні предмети,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						23

раси, класи, здібності, передісторії та інші ігрові сутності. Вибір MongoDB пов'язаний з тим, що матеріали Dungeons & Dragons мають неоднорідну структуру. Окремі записи можуть містити вкладені об'єкти, списки характеристик, текстові описи, набори властивостей і додаткові параметри. Документоорієнтований підхід у такому випадку є зручним, оскільки дозволяє зберігати різні за структурою сутності без складного приведення їх до однакової табличної моделі.

За реєстрацію та вхід користувачів у системі відповідає механізм автентифікації, побудований на NextAuth [19]. Після входу користувача система створює JWT-сесію, у якій зберігається необхідна інформація для подальшої перевірки доступу, зокрема роль користувача. У застосунку використовуються три ролі: player, moderator та admin. Звичайний користувач із роллю player працює з основними функціями системи: довідниками, персонажами, ігровими інструментами та AI-асистентом. Роль moderator призначена для керування довідковим контентом через адміністративну панель. Користувач із роллю admin має ширші права і може додатково змінювати ролі інших користувачів. Завдяки такому поділу кожна група користувачів отримує доступ лише до тих функцій, які відповідають її призначенню.

Для роботи з довідковими матеріалами в системі передбачено адміністративну панель. Через неї користувачі з відповідними правами можуть запускати завантаження даних з Open5e REST API, виконувати масовий або окремий переклад записів, редагувати й видаляти матеріали, а також переглядати статистику заповнення колекцій. Панель має три вкладки: «Скрапінг і переклад», «Управління контентом» та «Користувачі». Перші дві використовуються для роботи з ігровими матеріалами й доступні користувачам із ролями moderator та admin. Вкладка «Користувачі» відображається лише для admin і призначена для перегляду зареєстрованих користувачів та зміни їхніх ролей.

Модуль парсингу забезпечує отримання даних із зовнішнього джерела Open5e REST API [21]. Його основним завданням є завантаження структурованих матеріалів Dungeons & Dragons, перетворення їх до формату,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

придатного для збереження у MongoDB Atlas, і подальше використання у довідниках системи. Парсинг охоплює десять типів контенту: монстрів, заклинання, раси, класи, передісторії, здібності, магічні предмети, ігрові стани, спорядження та розділи правил. Для монстрів додатково зберігається поле `img_main`, яке використовується для відображення зображень у каталозі та на сторінці детального перегляду. Повторний запуск парсингу не повинен стирати вже збережені україномовні переклади, тому під час оновлення записів україномовні поля зберігаються окремо від даних, отриманих із зовнішнього API.

AI-модуль системи реалізовано через інтеграцію з Gemini, підключеним за допомогою API-ключа Google AI Studio [12]. Цей компонент використовується для роботи AI-асистента у форматі чату. Користувач може звертатися до асистента із запитаннями щодо ігрових правил, термінології, перекладу або пояснення окремих фрагментів Dungeons & Dragons. Серверна частина системи приймає запит користувача, формує звернення до Gemini API, отримує відповідь і передає її назад до клієнтського інтерфейсу.

У межах цієї системи користувач взаємодіє з вебінтерфейсом, реалізованим на Next.js і Tailwind CSS [17, 28]. Клієнтська частина надсилає запити до серверної логіки Next.js. Серверна частина, залежно від типу запиту, взаємодіє з MongoDB Atlas, NextAuth, Open5e REST API або Gemini API. Дані, отримані із зовнішнього API, зберігаються в базі даних і надалі використовуються у довідниках, модулях пошуку, фільтрації, створення персонажів та будівника зіткнень. Запити до AI-асистента обробляються окремо через серверний маршрут, що дозволяє не передавати API-ключ безпосередньо у клієнтську частину.

Такий архітектурний підхід є зручним для розроблюваної системи. Використання Next.js як Full-Stack фреймворку дає змогу працювати з клієнтською та серверною частинами в межах одного проєкту, що спрощує розробку й подальшу підтримку коду. Клієнт-серверна взаємодія дозволяє відокремити інтерфейс користувача від логіки обробки запитів, роботи з базою

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

даних і перевірки прав доступу. Використання MongoDB Atlas спрощує зберігання даних і не потребує окремого налаштування локального сервера бази даних. Звернення до зовнішніх сервісів через серверні маршрути також є безпечнішим, оскільки API-ключі не передаються до клієнтської частини. Крім того, модульна побудова залишає можливість поступово розширювати систему новими довідниками, ігровими інструментами, функціями перекладу та можливостями AI-асистента.

Таким чином, для інформаційної системи для поціновувачів Dungeons & Dragons доцільно використовувати веборієнтовану клієнт-серверну архітектуру з модульною Full-Stack реалізацією на базі Next.js. Такий підхід відповідає функціональним потребам системи, забезпечує зручну інтеграцію з MongoDB Atlas, NextAuth, Open5e REST API та Gemini API, а також створює основу для подальшого розвитку програмного продукту.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Під час розробки інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons важливим етапом є вибір інструментальних засобів, мов програмування, фреймворків, системи керування базою даних та зовнішніх сервісів. Обрані технології мають відповідати функціональним вимогам системи, забезпечувати зручність розробки, підтримку клієнтської та серверної частин, можливість інтеграції з хмарною базою даних, зовнішнім API для отримання ігрових даних та AI-сервісом.

Розроблювана система є Full-Stack вебзастосунком, тому для її реалізації було обрано стек технологій, орієнтований на створення сучасних веборієнтованих інформаційних систем. Основними інструментальними засобами розробки є Next.js [17, 18], TypeScript [29], Tailwind CSS [28], MongoDB Atlas [14], NextAuth [19], Open5e REST API [21] та Gemini API [9, 10], підключений через Google AI Studio [12].

Основою програмного продукту є фреймворк Next.js. Його вибір обумовлений тим, що він дозволяє реалізовувати як клієнтську частину

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

застосунку, так і серверну логіку в межах одного проєкту. Це особливо важливо для розроблюваної системи, оскільки вона містить значну кількість взаємопов'язаних функціональних модулів: довідники ігрових сутностей, адміністративну панель, модуль автентифікації, AI-чат, інтерактивний лист персонажа, механізм кидання дайсів і будівник зіткнень. Використання Next.js дозволяє організувати структуру сторінок, серверних маршрутів, компонентів інтерфейсу та службової логіки в одному середовищі розробки.

Клієнтська частина системи реалізується засобами Next.js та React-підходу до побудови інтерфейсу [22]. Компонентна структура дозволяє розділяти інтерфейс на окремі логічні частини, наприклад картки монстрів, списки заклинань, форми пошуку, елементи фільтрації, блоки персонажа, панелі адміністратора та вікно AI-чату. Такий підхід спрощує повторне використання компонентів, зменшує дублювання коду та полегшує подальше розширення системи.

Для реалізації інтерфейсу користувача, окрім Tailwind CSS, використано бібліотеку shadcn/ui [26]. Її застосування дозволяє створювати уніфіковані компоненти інтерфейсу, зокрема кнопки, вкладки, випадаючі меню, форми, поля введення, модальні вікна та інші елементи. Це є важливим для розроблюваної системи, оскільки вона містить значну кількість різних інтерфейсних модулів: wiki-довідники, сторінки детального перегляду, форми створення й редагування персонажа, AI-чат, будівник сутичок, інструмент кидання дайсів та адміністративну панель.

Інтерфейс системи реалізовано в темній кольоровій темі з використанням відтінків slate. Такий підхід забезпечує стилістичну цілісність сторінок і відповідає тематичному спрямуванню програмного продукту, пов'язаного з фентезійним світом Dungeons & Dragons. Темна тема застосовується в усіх основних частинах системи: wiki-каталогах, сторінках детального перегляду, інструментах гравця, формах створення та редагування персонажа, AI-чаті й адміністративній панелі.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Мовою програмування проєкту обрано TypeScript. На відміну від JavaScript, TypeScript підтримує статичну типізацію, що дозволяє точніше описувати структуру даних і зменшувати кількість помилок під час розробки. Це особливо важливо для системи, яка працює з великою кількістю різноманітних ігрових сутностей: монстрами, заклинаннями, магічними предметами, расами, класами, feats, передісторіями, персонажами та зіткненнями. Кожна з цих сутностей має власну структуру, набір полів і вкладених параметрів. Використання TypeScript дозволяє формалізувати ці структури у вигляді типів та інтерфейсів, що підвищує надійність коду й полегшує його підтримку.

Для зберігання даних використовується MongoDB Atlas [14]. Вибір саме цієї системи зумовлений особливостями предметної області. Дані Dungeons & Dragons мають документоорієнтований характер: записи можуть містити вкладені об'єкти, масиви властивостей, текстові описи, характеристики, списки дій, посилання на інші сутності та додаткові параметри. Реляційна модель даних у такому випадку вимагала б створення значної кількості пов'язаних таблиць, що ускладнило б збереження та обробку ігрових сутностей. MongoDB дозволяє зберігати такі дані у вигляді документів, що краще відповідає природі контенту Dungeons & Dragons [15].

MongoDB Atlas використовується як хмарна база даних, підключення до якої здійснюється за допомогою URL-з'єднання. Такий підхід спрощує розгортання системи, оскільки база даних не потребує локального встановлення на комп'ютері користувача або окремого ручного адміністрування серверної СКБД. Хмарне зберігання також дозволяє забезпечити доступ до даних із різних середовищ розробки та розгортання.

Для взаємодії з MongoDB Atlas у системі використовується Mongoose ODM [16]. Цей інструмент дозволяє описувати моделі документів, задавати структуру колекцій, виконувати операції створення, читання, оновлення та видалення даних, а також формалізувати роботу з MongoDB у межах TypeScript-застосунку. Використання Mongoose є доцільним, оскільки система працює з

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

великою кількістю колекцій, серед яких users, characters та десять wiki-колекцій Dungeons & Dragons.

Для реалізації реєстрації, входу та керування сесіями користувачів використовується NextAuth [19]. Цей інструмент забезпечує механізми автентифікації в межах Next.js-застосунку та дозволяє контролювати доступ користувачів до окремих частин системи. У контексті розроблюваного програмного продукту це важливо для розмежування можливостей гостя, звичайного користувача та адміністратора. Наприклад, гість може переглядати відкриті сторінки, авторизований користувач – працювати з персонажами та ігровими інструментами, а адміністратор – використовувати панель керування даними, запускати парсинг, переклад і редагування записів у базі даних.

Джерелом ігрових даних для наповнення системи є Open5e REST API [21]. Його використання дозволяє автоматизувати процес отримання структурованої інформації про монстрів, заклинання, предмети, класи та інші сутності Dungeons & Dragons. Завдяки цьому зменшується обсяг ручної роботи під час наповнення бази даних. Дані, отримані через API, обробляються серверною логікою системи, приводяться до потрібної структури та зберігаються в MongoDB Atlas. Надалі ці записи використовуються у довідниках, модулях пошуку, фільтрації, адміністративній панелі та інших функціональних частинах системи.

Для реалізації AI-асистента використовується Gemini API, підключений через Google AI Studio за допомогою API-ключа [12]. AI-асистент реалізований у вигляді чату, через який користувач може ставити запитання, пов'язані з правилами гри, термінологією, перекладом або поясненням окремих фрагментів Dungeons & Dragons. Інтеграція Gemini дозволяє розширити можливості системи порівняно зі звичайним електронним довідником, оскільки користувач отримує не лише статичну інформацію, а й інтерактивну допомогу.

Звернення до Gemini API здійснюється через серверну частину застосунку. Такий підхід є важливим з погляду безпеки, оскільки API-ключ не передається безпосередньо до клієнтської частини та не розміщується у відкритому коді

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

браузера. Серверна логіка приймає запит користувача, формує відповідне звернення до AI-сервісу, отримує відповідь і передає її назад до інтерфейсу чату.

Для розробки програмного продукту також можуть використовуватися допоміжні інструменти, зокрема середовище розробки Visual Studio Code, менеджер пакетів npm, система контролю версій Git та платформа GitHub для зберігання репозиторію. Ці інструменти не є частиною функціоналу кінцевої системи, однак забезпечують зручну організацію процесу розробки, керування залежностями, відстеження змін у коді та підтримку структури проєкту.

Узагальнений перелік обраних інструментальних засобів наведено в таблиці 1.2.

Таблиця 1.2

Перелік обраних інструментальних засобів

Інструментальний засіб	Призначення в системі	Обґрунтування вибору
Next.js	Реалізація клієнтської та серверної частин вебзастосунку	Дозволяє створювати Full-Stack застосунок у межах одного проєкту, організовувати сторінки, серверну логіку та API-маршрути
TypeScript	Основна мова розробки	Забезпечує статичну типізацію, підвищує надійність коду та спрощує роботу зі складними структурами ігрових даних
Tailwind CSS	Стилізація інтерфейсу користувача	Дає змогу швидко створювати адаптивний інтерфейс і підтримувати єдиний стиль оформлення сторінок
MongoDB Atlas	Хмарна база даних	MongoDB Atlas це хмарна база даних для зберігання користувачів, персонажів і wiki-контенту Dungeons & Dragons; обрана через підтримку документоорієнтованої структури, вкладених об'єктів і масивів.

Інструментальний засіб	Призначення в системі	Обґрунтування вибору
NextAuth	Автентифікація та керування сесіями	Забезпечує реєстрацію, вхід користувачів і контроль доступу до функціональних можливостей
Open5e REST API	Джерело структурованих D&D-даних	Дозволяє автоматизувати отримання ігрових матеріалів для наповнення бази даних
Gemini API	Реалізація AI-асистента	Забезпечує інтерактивну роботу користувача з AI-чатом, поясненнями та перекладом ігрового контенту
Google AI Studio	Засіб доступу до Gemini API	Використовується для отримання API-ключа та підключення AI-модуля до системи
Git	Система контролю версій	Дозволяє відстежувати зміни в коді та керувати процесом розробки
Visual Studio Code	Середовище розробки	Забезпечує зручну роботу з TypeScript, Next.js, Tailwind CSS та структурою проєкту
shadcn/ui	Набір компонентів інтерфейсу	Забезпечує створення уніфікованих елементів інтерфейсу: кнопок, форм, вкладок, меню, модальних вікон та інших UI-компонентів
Mongoose ODM	Робота з MongoDB Atlas	Дозволяє описувати моделі даних, виконувати CRUD-операції та структурувати взаємодію застосунку з MongoDB
react-markdown	Відображення Markdown-контенту	Забезпечує коректне відображення форматуваних описів, списків і текстових матеріалів у wiki-довідниках
remark-gfm	Розширення Markdown для таблиць	Дозволяє відображати таблиці та інші елементи GitHub Flavored Markdown, що використовується для правил і передісторій
NextAuth	Автентифікація та керування сесіями	Забезпечує реєстрацію, вхід користувачів і контроль доступу до функціональних можливостей

Для роботи інформаційної системи необхідно врахувати вимоги до апаратного та програмного забезпечення. Оскільки система є веборієнтованою, основна частина навантаження залежить від середовища розгортання серверної частини, доступності хмарної бази даних та зовнішніх API. Користувачеві не потрібно встановлювати спеціалізоване програмне забезпечення для роботи із системою, достатньо сучасного веббраузера та доступу до мережі Інтернет.

Для користувацького пристрою мінімальними вимогами є наявність комп'ютера, ноутбука, планшета або смартфона із сучасним веббраузером. Оскільки інтерфейс системи реалізовано як вебзастосунок, користувач може працювати з ним незалежно від операційної системи, якщо пристрій підтримує сучасні браузери та має стабільне підключення до Інтернету.

Для середовища розробки потрібен персональний комп'ютер або ноутбук із встановленим Node.js, менеджером пакетів npm, середовищем розробки та доступом до MongoDB Atlas, Open5e REST API і Gemini API. Також необхідно мати налаштовані змінні середовища для зберігання URL-підключення до бази даних, параметрів NextAuth та API-ключа Gemini.

Орієнтовні вимоги до апаратного та програмного забезпечення наведено в таблиці 1.3.

Таблиця 1.3

Вимоги до апаратного та програмного забезпечення

Категорія	Мінімальні вимоги	Рекомендовані вимоги
Пристрій користувача	ПК, ноутбук, планшет або смартфон із сучасним браузером	ПК або ноутбук із сучасним браузером для зручної роботи з довідниками, персонажами та адміністративною панеллю
Операційна система користувача	Windows, Linux, macOS, Android або iOS	Будь-яка актуальна ОС із підтримкою сучасних браузерів
Браузер	Google Chrome, Mozilla Firefox, Microsoft Edge, Safari або інший сучасний браузер	Остання стабільна версія браузера

Категорія	Мінімальні вимоги	Рекомендовані вимоги
Підключення до Інтернету	Стабільне підключення для доступу до вебзастосунку	Стабільне широкосмугове підключення для швидкої роботи з довідниками, AI-чатом та зовнішніми сервісами
Пристрій розробника	Комп'ютер або ноутбук із 4 ГБ оперативної пам'яті	Комп'ютер або ноутбук із 8 ГБ оперативної пам'яті або більше
Програмне забезпечення розробника	Node.js, npm, Git, редактор коду	Node.js, npm, Git, Visual Studio Code
База даних	Доступ до MongoDB Atlas	Хмарний кластер MongoDB Atlas із налаштованим URL-підключенням
Зовнішні сервіси	Доступ до Open5e REST API та Gemini API	Налаштовані ключі доступу та змінні середовища для безпечної інтеграції

Обраний набір технологій відповідає вимогам розроблюваної системи. Next.js забезпечує реалізацію клієнтської та серверної частин, TypeScript підвищує надійність програмного коду, Tailwind CSS спрощує створення адаптивного інтерфейсу, MongoDB Atlas дає змогу зберігати складні документоорієнтовані дані, NextAuth забезпечує автентифікацію користувачів, Open5e REST API використовується для автоматизованого наповнення системи, а Gemini API – для реалізації AI-асистента.

Отже, вибір інструментальних засобів є обґрунтованим, оскільки вони забезпечують реалізацію всіх основних функціональних можливостей інформаційної системи: реєстрації та автентифікації користувачів, перегляду довідкових матеріалів, пошуку та фільтрації даних, адміністрування записів, парсингу зовнішніх джерел, роботи з персонажами, кидання дайсів, створення зіткнень і взаємодії з AI-асистентом.

Висновки до першого розділу

У першому розділі було проведено аналіз предметної області та визначено актуальність розробки інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента. Встановлено, що ігровий процес Dungeons & Dragons потребує постійної роботи з великими обсягами структурованої інформації, зокрема описами монстрів, заклинань, магічних предметів, рас, класів, передісторій, умінь та інших ігрових сутностей. Використання розрізнених джерел, паперових матеріалів або окремих електронних документів ускладнює пошук потрібної інформації та знижує зручність підготовки до ігрових сесій.

Окремо було визначено проблему мовного бар'єра для українськомовних користувачів. Значна частина матеріалів Dungeons & Dragons доступна англійською мовою, а звичайні засоби машинного перекладу не завжди коректно передають специфічну ігрову термінологію. Це створює потребу в інструменті, який не лише надає доступ до довідкової інформації, а й допомагає користувачам працювати з контентом за допомогою AI-асистента.

У межах розділу було сформульовано основні задачі, які має вирішувати розроблювана система. До них належать автоматизоване наповнення бази даних ігровими матеріалами, реалізація довідників, пошуку та фільтрації, створення адміністративної панелі, інтеграція AI-асистента, реалізація механізму автентифікації, інтерактивного листа персонажа, кидання дайсів і будівника зіткнень. Таким чином, система має поєднувати функції електронного довідника, інструмента підтримки ігрового процесу та засобу адміністрування даних.

Було проведено аналіз існуючих програмних рішень, зокрема D&D Beyond, Roll20, Foundry Virtual Tabletop та World Anvil. Встановлено, що ці системи мають значний функціональний потенціал, однак здебільшого орієнтовані на окремі аспекти підтримки настільно-рольових ігор. D&D Beyond забезпечує роботу з офіційними матеріалами та персонажами, Roll20 і Foundry Virtual Tabletop зосереджені на проведенні онлайн-сесій у форматі віртуального ігрового столу, а World Anvil призначений переважно для створення й організації

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

вигаданих світів. Водночас більшість аналогів мають обмеження безкоштовного доступу, а повний функціонал часто потребує платної підписки, придбання цифрового контенту або купівлі ліцензії.

За результатами аналізу аналогів визначено, що актуальною є розробка власної інформаційної системи, яка поєднує довідкові матеріали Dungeons & Dragons, автоматизований парсинг даних, хмарне зберігання інформації, пошук і фільтрацію, адміністративне керування записами та інтеграцію AI-асистента. Такий підхід дозволяє створити єдине середовище для гравців і майстрів гри, орієнтоване на зручну роботу з ігровим контентом.

У розділі визначено архітектурний підхід до побудови системи. Для реалізації обрано клієнт-серверний вебзастосунок на базі Next.js. Такий варіант підходить для проєкту, оскільки дозволяє в межах одного застосунку реалізувати інтерфейс користувача, серверні API-маршрути, роботу з базою даних, авторизацію та підключення зовнішніх сервісів. У структурі системи виділено клієнтську частину, серверну логіку, базу даних MongoDB Atlas, а також зовнішні сервіси Open5e REST API та Gemini API.

Також у розділі обґрунтовано вибір основних засобів розробки. Next.js використано як основу Full-Stack застосунку, TypeScript – для типізації програмного коду, Tailwind CSS – для оформлення інтерфейсу, MongoDB Atlas – для зберігання даних, NextAuth – для автентифікації користувачів. Open5e REST API застосовується як джерело довідкових матеріалів Dungeons & Dragons, а Gemini API – для роботи AI-асистента та перекладу контенту. Обрані інструменти відповідають функціональним потребам системи та залишають можливість для її подальшого розширення.

Отже, у результаті виконання першого розділу було визначено теоретичні та практичні передумови створення інформаційної системи, проаналізовано існуючі аналоги, сформульовано задачі розробки, обґрунтовано архітектуру та вибір інструментальних засобів.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПОЦІНОВУВАЧІВ DUNGEONS & DRAGONS

2.1 Визначення функціональних і нефункціональних вимог до системи

Проектування інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons доцільно почати з визначення ролей користувачів, основних сценаріїв взаємодії із системою, функціональних вимог та узагальненої об'єктно-орієнтованої структури програмного продукту [3]. Такий підхід дозволяє формалізувати призначення системи, встановити межі її функціональності та підготувати основу для подальшого проектування бази даних, алгоритмів роботи та програмних модулів.

Розроблювана система є Full-Stack вебзастосунком, що поєднує довідкову базу матеріалів Dungeons & Dragons [27], інструменти гравця, адміністративний модуль керування контентом та AI-асистента. Основними групами функцій системи є: перегляд wiki-довідників, пошук і фільтрація ігрових сутностей [21, 27], створення та ведення персонажів, кидання дайсів, формування ігрових сутичок, робота з AI-асистентом, парсинг даних із зовнішнього джерела та адміністрування записів бази даних.

У системі передбачено чотири логічні типи користувачів: гість, авторизований користувач із роллю player, користувач із розширеними правами moderator та користувач із роллю admin. Гість має доступ лише до сторінок авторизації та реєстрації. Користувач із роллю player може працювати з основними функціями системи: переглядати wiki-довідники, користуватися пошуком і фільтрацією, переглядати детальну інформацію про ігрові сутності, створювати та редагувати персонажів, працювати з інтерактивним аркушем персонажа, кидати дайси, формувати сутички та звертатися до AI-асистента. Користувач із роллю moderator додатково має доступ до адміністративної панелі, де реалізовано парсинг, масовий і точковий AI-переклад, редагування та видалення записів бази даних. Роль admin розширює можливості moderator і

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

надає доступ до керування користувачами, зокрема до перегляду списку користувачів та зміни їхніх ролей.

Перелік ролей користувачів у інформаційній системі представлено у таблиці 2.1.

Таблиця 2.1

Ролі користувачів інформаційної системи

Роль користувача	Характеристика ролі	Основні можливості
Гість	Неавторизований користувач системи	Перехід на сторінки входу та реєстрації
Player	Авторизований користувач, який використовує основні функції системи	Перегляд wiki-довідників, пошук і фільтрація матеріалів, перегляд зображень монстрів, використання AI-асистента, створення та редагування персонажів, робота з інтерактивним аркушем персонажа, кидання дайсів, створення сутичок
Moderator	Авторизований користувач із розширеними правами керування контентом	Усі можливості ролі player, а також доступ до адміністративної панелі, запуск парсингу, масового й точкового AI-перекладу, редагування, видалення записів і перегляд статистики бази даних
Admin	Користувач із найвищим рівнем адміністративного доступу	Усі можливості ролі moderator, а також доступ до вкладки «Користувачі» в адміністративній панелі, перегляд списку користувачів і зміна їхніх ролей

Роль користувача зберігається в базі даних у записі користувача та використовується під час формування сесії. Після входу до системи інформація про роль передається в JWT-токен і надалі застосовується для перевірки доступу до захищених маршрутів. Такий механізм дозволяє відокремити звичайні користувацькі можливості від адміністративних функцій, які впливають на вміст бази даних.

Функціональні вимоги до інформаційної системи визначають перелік дій, які повинна виконувати система для забезпечення потреб користувачів. Основні функціональні вимоги наведено в таблиці 2.2.

Таблиця 2.2

Функціональні вимоги до інформаційної системи

Ідентифікатор	Функціональна вимога	Опис
FR-01	Реєстрація користувача	Система повинна надавати можливість створення нового облікового запису
FR-02	Автентифікація користувача	Система повинна забезпечувати вхід користувача та створення сесії
FR-03	Розмежування прав доступу	Система повинна обмежувати доступ до функцій відповідно до ролей player, moderator та admin
FR-04	Перегляд wiki-довідників	Система повинна надавати доступ до десяти розділів wiki: монстрів, заклинань, рас, класів, передісторій, здібностей, магічних предметів, станів, спорядження та правил ^[21,27]
FR-05	Перегляд детальної інформації	Система повинна відображати окремі сторінки з повним описом ігрових сутностей
FR-06	Пошук матеріалів	Система повинна забезпечувати пошук записів за ключовими словами
FR-07	Фільтрація матеріалів	Система повинна підтримувати фільтрацію за параметрами, зокрема CR, розміром, рівнем заклинання, школою магії, рідкістю предмета або категорією спорядження
FR-08	Відображення зображень монстрів	Система повинна відображати зображення монстрів у каталозі та на сторінці детального перегляду, якщо зображення доступне в джерелі даних
FR-09	Модальний перегляд зображення монстра	Система повинна дозволяти відкрити зображення монстра у збільшеному вигляді при натисканні

Ідентифікатор	Функціональна вимога	Опис
FR-10	Створення персонажа	Система повинна дозволяти користувачу створити персонажа із зазначенням імені, раси, класу, рівня, передісторії, світогляду, характеристик, НР, класу обладунку та швидкості
FR-11	Редагування персонажа	Система повинна надавати можливість редагування створеного персонажа через окрему сторінку
FR-12	Налаштування навичок персонажа	Система повинна дозволяти позначати навички, якими персонаж володіє, а також навички з експертизою
FR-13	Налаштування рятівних кидків	Система повинна дозволяти обирати характеристики, за якими персонаж має володіння рятівними кидками
FR-14	Автоматичний розрахунок бонусу майстерності	Система повинна автоматично визначати бонус майстерності персонажа залежно від його рівня
FR-15	Перегляд аркуша персонажа	Система повинна відображати інтерактивний аркуш створеного персонажа
FR-16	Оновлення НР персонажа	Система повинна дозволяти змінювати поточні НР персонажа з обмеженням у межах від 0 до максимального значення
FR-17	Кидання дайсів	Система повинна підтримувати кидання стандартних кубиків і складних dice-виразів
FR-18	Створення сутички	Система повинна дозволяти формувати бойове зіткнення з монстрами та персонажами користувача
FR-19	Налаштування ініціативи	Система повинна надавати окремий блок для ручного виставлення ініціативи перед початком бою

Ідентифікатор	Функціональна вимога	Опис
FR-20	Керування станом учасників сутички	Система повинна дозволяти змінювати НР довільним значенням і додавати або знімати ігрові стани
FR-21	Використання AI-асистента	Система повинна надавати чат для взаємодії користувача з AI-асистентом ^[9, 10]
FR-22	Пошук контексту для AI-відповіді	Система повинна здійснювати пошук релевантних матеріалів у wiki-колекціях перед формуванням відповіді AI-асистента
FR-23	Парсинг даних з Open5e REST API	Система повинна дозволяти модератору або адміністратору запускати завантаження десяти типів ігрового контенту ^[21]
FR-24	Повторний парсинг без втрати перекладів	Система повинна оновлювати дані з Open5e REST API без стирання вже збережених україномовних полів
FR-25	Масовий AI-переклад контенту	Система повинна дозволяти запускати переклад записів відповідного типу контенту за допомогою Gemini API ^[9, 10]
FR-26	Точковий AI-переклад запису	Система повинна дозволяти перекладати окремий запис через кнопку «Перекласти» в адміністративній панелі ^[9, 10]
FR-27	Керування контентом	Система повинна дозволяти переглядати, редагувати та видаляти записи бази даних через адміністративну панель
FR-28	Перегляд статистики бази даних	Система повинна відображати статистику наповнення всіх десяти wiki-колекцій
FR-29	Керування користувачами	Система повинна дозволяти користувачу з роллю admin переглядати список користувачів і змінювати їхні ролі
FR-30	Захист API-маршрутів	Система повинна перевіряти права доступу під час звернення до захищених API-маршрутів

Нефункціональні вимоги визначають якісні характеристики системи, що впливають на її зручність, безпеку, продуктивність, підтримуваність і можливість подальшого розвитку. Основні нефункціональні вимоги наведено в таблиці 2.3.

Таблиця 2.3

Нефункціональні вимоги до інформаційної системи

Ідентифікатор	Нефункціональна вимога	Опис
NFR-01	Веборієнтованість	Система повинна працювати через сучасний веббраузер без встановлення додаткового клієнтського програмного забезпечення
NFR-02	Адаптивність інтерфейсу	Інтерфейс повинен коректно відображатися на пристроях із різними розмірами екрана
NFR-03	Зручність використання	Навігація між довідниками, інструментами гравця та адміністративною панеллю повинна бути зрозумілою користувачу
NFR-04	Швидкодія	Пошук, фільтрація та відкриття сторінок довідників повинні виконуватися без суттєвих затримок
NFR-05	Захищеність облікових даних	Паролі користувачів повинні зберігатися у хешованому вигляді
NFR-06	Контроль доступу	Доступ до адміністративних функцій повинен надаватися лише користувачам із відповідною роллю
NFR-07	Захист конфігураційних даних	URL підключення до MongoDB Atlas, параметри автентифікації та API-ключ Gemini не повинні передаватися у клієнтську частину
NFR-08	Масштабованість структури даних	База даних повинна дозволяти додавати нові типи ігрових сутностей без повної перебудови системи

NFR-09	Стійкість до помилок зовнішніх сервісів	Система повинна обробляти помилки під час звернення до Open5e REST API та Gemini API
NFR-10	Підтримуваність коду	Програмний код повинен мати модульну структуру, що спрощує його супровід і розширення
NFR-11	Типізація даних	Використання TypeScript повинно забезпечувати контроль структур даних під час розробки
NFR-12	Хмарне зберігання	Дані системи повинні зберігатися у хмарній базі MongoDB Atlas
NFR-13	Підтримка україномовного контенту	Система повинна підтримувати україномовні назви, описи та переклади ігрових матеріалів
NFR-14	Розширюваність AI-модуля	Архітектура AI-модуля повинна дозволяти подальше вдосконалення промптів, перекладу та пошуку контексту
NFR-15	Надійність оновлення даних	Під час парсингу система повинна оновлювати наявні записи або створювати нові без дублювання за унікальним ідентифікатором

На основі визначених вимог сформовано основні варіанти використання інформаційної системи. Для неавторизованого користувача базовими сценаріями є реєстрація та вхід до системи. Після авторизації користувач із роллю player отримує доступ до вікі-довідників, може здійснювати пошук і фільтрацію матеріалів, переглядати детальні сторінки ігрових сутностей, зокрема зображення монстрів, користуватися AI-асистентом, створювати та редагувати персонажів, налаштовувати навички й рятівні кидки, працювати з інтерактивним аркушем персонажа, кидати дайси та формувати бойові сутички.

Користувач із роллю moderator додатково може працювати з адміністративною панеллю, запускати скрапінг десяти типів контенту з Open5e

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

REST API, виконувати масовий і точковий AI-переклад записів, редагувати та видаляти довідкові матеріали, а також переглядати статистику наповнення бази даних. Користувач із роллю admin має всі можливості moderator і додатково отримує доступ до вкладки керування користувачами, де може переглядати їхній список і змінювати ролі.

Побудова варіантів використання виконувалася з урахуванням того, що система має поєднувати кілька різних за призначенням груп функцій. Частина сценаріїв пов'язана з доступом до довідкових матеріалів і призначена для щоденного використання гравцями. Інша частина стосується створення та ведення персонажів, тобто роботи з персональними даними користувача. Окремий блок сценаріїв належить до адміністративних функцій, які не повинні бути доступні всім користувачам, оскільки вони впливають на наповнення бази даних і цілісність довідкового контенту.

Такий поділ дозволяє чітко відокремити загальнодоступні можливості від функцій, що потребують авторизації або підвищеного рівня доступу. Наприклад, звичайний користувач може працювати з персонажами та ігровими інструментами, але не має змоги змінювати довідкові записи або запускати скрапінг. Натомість moderator отримує доступ до керування контентом, а admin додатково відповідає за керування обліковими записами. Це спрощує подальше проєктування інтерфейсу й серверної логіки, оскільки кожен сценарій одразу пов'язується з конкретною роллю користувача.

Діаграму варіантів використання інформаційної системи наведено на рисунку 2.1 [3].

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

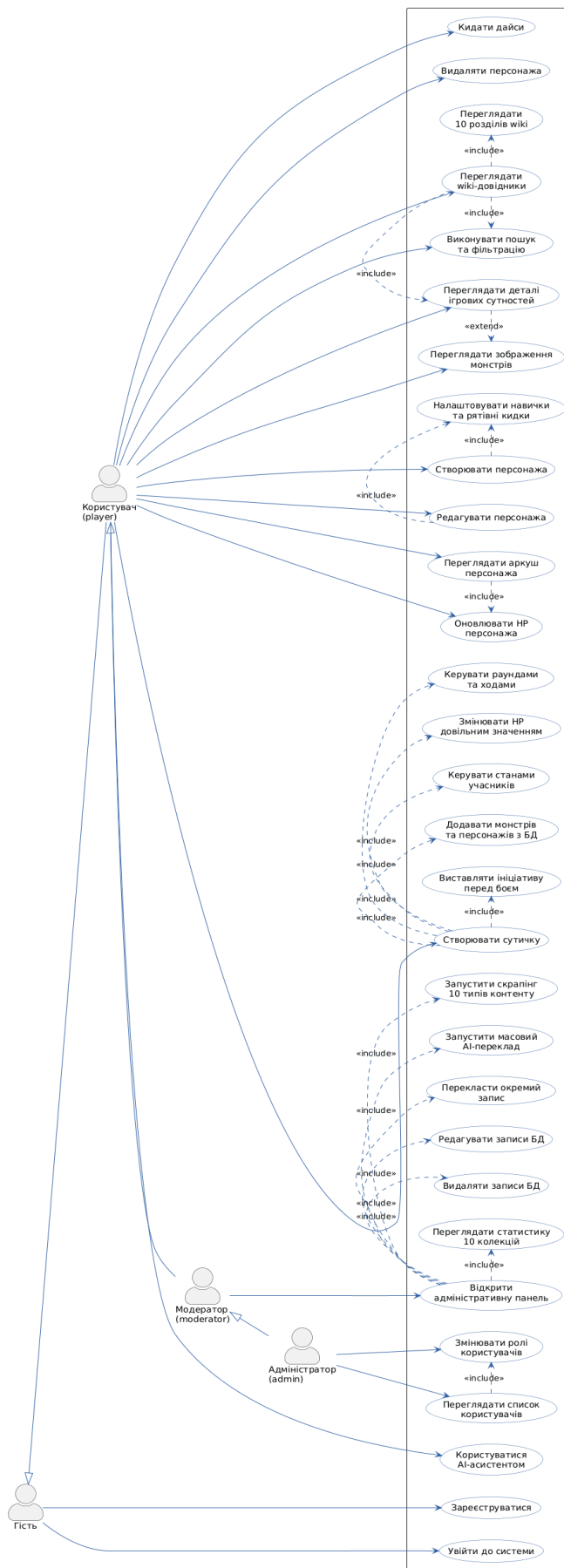


Рис. 2.1. Діаграма варіантів використання інформаційної системи

Діаграма демонструє розмежування функцій між різними типами користувачів. Основна частина функціональності належить авторизованому користувачу з роллю player. Функції керування контентом винесено до сценаріїв ролі moderator, а функції керування користувачами – до сценаріїв ролі admin. Такий підхід забезпечує логічний розподіл відповідальності та зменшує ризик несанкціонованої зміни даних системи.

Для подальшого проєктування системи необхідно також визначити її узагальнену об'єктно-орієнтовану структуру. Оскільки система працює з великою кількістю ігрових сутностей, доцільно виділити базові групи об'єктів: користувачі, довідкові матеріали, персонажі, бойові сутички, повідомлення AI-чату та адміністративні операції.

Основною сутністю користувацького рівня є User. Вона містить дані облікового запису користувача, зокрема ім'я, електронну пошту, хешований пароль, роль і дату створення. Від ролі користувача залежить доступ до окремих функцій системи.

Довідкові матеріали представлені окремими сутностями, що відповідають основним розділам wiki-довідника: Monster, Spell, Race, Class, Background, Feat, MagicItem, Condition, Equipment та Section. Кожна з цих сутностей має англomовну й україномовну назву, короткий або повний опис, службовий ідентифікатор slug, а також додаткові поля, характерні для конкретного типу даних. Наприклад, монстри містять характеристики, клас обладунку, очки здоров'я, рейтинг складності, дії та особливості; заклинання містять рівень, школу магії, час накладання, дистанцію, тривалість і компоненти; магичні предмети містять тип, рідкість і ознаку налаштування.

Сутність Character відповідає за збереження персонажів користувачів. Вона пов'язана з користувачем через user_id і містить ім'я персонажа, расу, клас, рівень, передісторію, світогляд, характеристики, максимальні й поточні HP, клас обладунку, швидкість, навички, інвентар, заклинання та нотатки. Наявність цієї сутності дозволяє реалізувати інтерактивний аркуш персонажа та зберігати його стан між сесіями користувача.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

2.2 Розробка бази даних системи

Одним із ключових етапів проєктування інформаційної системи є розробка структури бази даних. База даних забезпечує збереження облікових записів користувачів, довідкових матеріалів Dungeons & Dragons, персонажів, а також даних, необхідних для роботи адміністративного модуля, пошуку, фільтрації та AI-асистента.

Для реалізації зберігання даних у системі використовується MongoDB Atlas [14]. MongoDB є документоорієнтованою системою керування базами даних, що дозволяє зберігати інформацію у вигляді документів. Такий підхід є доцільним для розроблюваної системи, оскільки ігрові сутності Dungeons & Dragons мають складну та неоднорідну структуру. Наприклад, монстри містять характеристики, дії, реакції, особливості, імунітети, мови, середовища існування та показники складності; заклинання містять рівень, школу магії, компоненти, тривалість, класи та опис; персонажі містять характеристики, інвентар, навички, заклинання і нотатки.

У разі використання реляційної бази даних така структура вимагала б створення значної кількості пов'язаних таблиць і проміжних зв'язків. Документоорієнтована модель MongoDB дозволяє зберігати вкладені об'єкти та масиви безпосередньо в межах одного документа, що спрощує роботу з ігровим контентом та відповідає природі даних Dungeons & Dragons.

У проєкті для взаємодії з MongoDB Atlas використовується Mongoose ODM [16]. Mongoose дозволяє описувати схеми документів, задавати типи полів, реалізовувати моделі даних та виконувати операції створення, читання, оновлення і видалення записів. Завдяки цьому забезпечується більш формалізована робота з колекціями MongoDB у межах TypeScript/Next.js-застосунку.

У базі даних системи передбачено групи колекцій користувачів та автентифікації, колекції wiki-довідників Dungeons & Dragons, колекції ігрових інструментів користувача, колекції для збереження персонажів, колекції, що

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

використовуються адміністративним модулем під час парсингу, перекладу й оновлення даних.

Для збереження облікових записів використовується колекція `users`. Вона містить ідентифікатор користувача, ім'я, `email`, хешований пароль, роль і дату створення запису. Роль визначає, які можливості системи доступні конкретному користувачу. Для звичайної роботи з довідниками, персонажами та ігровими інструментами використовується роль `player`. Керування контентом через адміністративну панель доступне користувачам із роллю `moderator`, а користувачі з роллю `admin` можуть додатково переглядати список користувачів і змінювати їхні ролі.

Колекції `wiki`-довідників зберігають основний ігровий контент `Dungeons & Dragons` [27]. До них належать `monsters`, `spells`, `races`, `classes`, `backgrounds`, `feats`, `magicitems`, `conditions`, `equipment` та `sections`. Таким чином, база даних містить десять основних колекцій довідкових матеріалів. Ці колекції наповнюються за допомогою парсингу даних з `Open5e REST API` та можуть редагуватися через адміністративну панель [21]. Для довідкових сутностей передбачено збереження англійської та української версій назв і описів, що дозволяє працювати як з оригінальним, так і з локалізованим контентом.

Колекція `monsters` призначена для збереження інформації про монстрів [21, 27]. Вона містить службовий ідентифікатор `slug`, англійську та українську назву, поле `img_main` для зображення монстра, розмір, тип, світогляд, клас обладунку, очки здоров'я, швидкість, основні характеристики, рятівні кидки, навички, опори, імунітети, чуття, мови, рейтинг складності, досвід, опис, дії, реакції, особливі здібності та легендарні дії. Така структура дозволяє відображати повну картку монстра, використовувати його у будівнику сутичок, а також показувати зображення в каталозі та на сторінці детального перегляду.

Колекція `spells` містить дані про заклинання [21, 27]. Основними полями є `slug`, `name_en`, `name_uk`, `level`, `school`, `casting_time`, `range`, `duration`, `components`, `material`, `ritual`, `concentration`, `classes`, `description_en`, `description_uk`,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

higher_levels_en та higher_levels_uk. Наявність цих полів дає змогу фільтрувати заклинання за рівнем, школою магії, класами та іншими параметрами.

У колекції races зберігаються дані про раси персонажів і пов'язані з ними підраси. Для кожного запису передбачено назву, опис, інформацію про бонуси до характеристик, вік, світогляд, розмір, швидкість, мови, зір та расові риси. Підраси зберігаються у вкладеному масиві subraces, тому для них не потрібно створювати окрему колекцію.

Колекція classes призначена для опису класів персонажів. Вона містить назву класу, кістку хітів, правила визначення очок здоров'я на першому та наступних рівнях, а також відомості про володіння обладунками, зброєю, інструментами, рятівними кидками й навичками. Крім того, у цій колекції зберігаються початкове спорядження та текстовий опис класу.

Дані про передісторії персонажів розміщуються в колекції backgrounds. У ній зберігаються опис передісторії, володіння навичками й інструментами, мови, спорядження, особлива риса передісторії та рекомендовані характеристики персонажа. Оскільки частина матеріалів може містити списки або таблиці, для їх коректного відображення в інтерфейсі використовується підтримка Markdown і таблиць GitHub Flavored Markdown.

Колекція feats містить інформацію про здібності персонажів. Для кожної здібності зберігаються службовий ідентифікатор slug, англomовна та україномовна назви, опис, умови отримання здібності та службові дані документа. Ці записи використовуються у відповідному wiki-розділі, де користувач може переглядати доступні здібності та їхні описи [11].

Колекція magicitems використовується для збереження магічних предметів. Вона містить назву, тип, рідкість, ознаку необхідності налаштування та опис. За цими параметрами реалізовано пошук і фільтрацію магічних предметів.

Колекція conditions містить ігрові стани, які можуть застосовуватися до персонажів або монстрів під час сутички. Основними полями є slug, name_en,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

name_uk, desc_en та desc_uk. Ця інформація може використовуватися як у довіднику, так і в бойових інструментах системи.

Колекція equipment призначена для збереження спорядження, зокрема зброї та обладунків. Вона містить категорію предмета, вартість, вагу, опис, показники пошкодження, тип пошкодження, властивості зброї, дальність, клас обладунку, категорію обладунку, вимоги до сили та ознаку перешкоди для прихованості.

Колекція sections зберігає розділи правил Dungeons & Dragons. Вона містить назву, опис, батьківський розділ та службові поля документа. Ця колекція використовується для формування довідника правил.

Окремою важливою колекцією є characters. Вона зберігає створених користувачами персонажів. Кожен персонаж пов'язаний із конкретним користувачем через поле user_id. До основних полів персонажа належать name, race, class, level, background, alignment, experience_points, шість основних характеристик, max_hit_points, current_hit_points, armor_class, speed, inspiration, proficiency_bonus, saving_throws, skills, inventory, spells, notes, createdAt та updatedAt. Поле saving_throws використовується для збереження володіння рятівними кидками за окремими характеристиками. Поле skills містить інформацію про навички персонажа, зокрема ознаки володіння навичкою та експертизи. Поле proficiency_bonus зберігає бонус майстерності, який автоматично визначається залежно від рівня персонажа. Завдяки цій колекції система може зберігати стан персонажа, забезпечувати його редагування та підтримувати роботу інтерактивного аркуша.

Узагальнену структуру основних колекцій бази даних наведено в таблиці 2.4.

Таблиця 2.4

Основні колекції бази даних MongoDB

Назва колекції	Призначення	Основні дані
users	Збереження облікових записів користувачів	Ім'я користувача, email, хешований пароль, роль player/moderator/admin, дата створення

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

Назва колекції	Призначення	Основні дані
monsters	Збереження даних про монстрів	Назва, зображення img_main, розмір, тип, характеристики, HP, AC, CR, дії, здібності, опис
spells	Збереження заклинань	Назва, рівень, школа магії, час накладання, дистанція, компоненти, опис
races	Збереження рас і підрас	Назва, опис, розмір, швидкість, мови, риси, підраси
classes	Збереження класів персонажів	Назва, кістка хітів, володіння, спорядження, опис
backgrounds	Збереження передісторій	Опис, володіння, мови, спорядження, особливість, характеристики
feats	Збереження здібностей	Назва, опис, передумови
magicitems	Збереження магичних предметів	Назва, тип, рідкість, налаштування, опис
conditions	Збереження ігрових станів	Назва, опис стану, україномовний переклад
equipment	Збереження спорядження	Категорія weapon/armor, вартість, вага, пошкодження, броня, властивості
sections	Збереження розділів правил	Назва, опис, батьківський розділ, україномовний переклад
characters	Збереження персонажів користувачів	Користувач, ім'я, раса, клас, рівень, світогляд, характеристики, HP, AC, бонус майстерності, рятівні кидки, навички, інвентар, заклинання

Зв'язки між колекціями в MongoDB мають переважно логічний характер. Основний явний зв'язок реалізовано між users та characters: один користувач може мати багато персонажів, а кожен персонаж належить одному користувачу. У базі даних це реалізується через поле user_id у документі персонажа.

Довідкові колекції не мають жорстких зовнішніх ключів, як у реляційних базах даних. Зв'язки між ними встановлюються на рівні логіки застосунку.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						51

Наприклад, персонаж може містити назву раси та класу, які відповідають записам у колекціях `races` і `classes`. Будівник сутичок використовує дані з колекцій `monsters` та `characters`, але сам процес формування сутички виконується на рівні клієнтської і серверної логіки застосунку.

Для унікальної ідентифікації більшості довідкових записів використовується поле `slug`. Воно застосовується для формування маршрутів детальних сторінок, пошуку записів, оновлення даних під час парсингу та запобігання дублюванню записів. Під час завантаження даних з Open5e REST API система виконує операцію оновлення або вставки запису за значенням `slug`. Якщо запис уже існує, він оновлюється; якщо запис відсутній, створюється новий документ.

Під час повторного парсингу важливо не втрачати вже збережені україномовні переклади [21]. Для цього під час оновлення записів англomовні та службові поля можуть оновлюватися відповідно до актуальних даних із Open5e REST API, тоді як україномовні поля не повинні перезаписуватися порожніми значеннями. Такий підхід дозволяє безпечно оновлювати довідкові матеріали та зберігати результати попереднього AI-перекладу або ручного редагування.

Використання полів `name_en`, `name_uk`, `description_en` та `description_uk` дозволяє зберігати як оригінальний англomовний контент, так і його україномовний переклад. Це важливо для реалізації функцій локалізації та AI-перекладу. Адміністративна панель дає змогу запускати переклад окремих типів контенту та редагувати результати в разі потреби.

На рисунку 2.3 наведено узагальнену структуру колекцій бази даних MongoDB.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

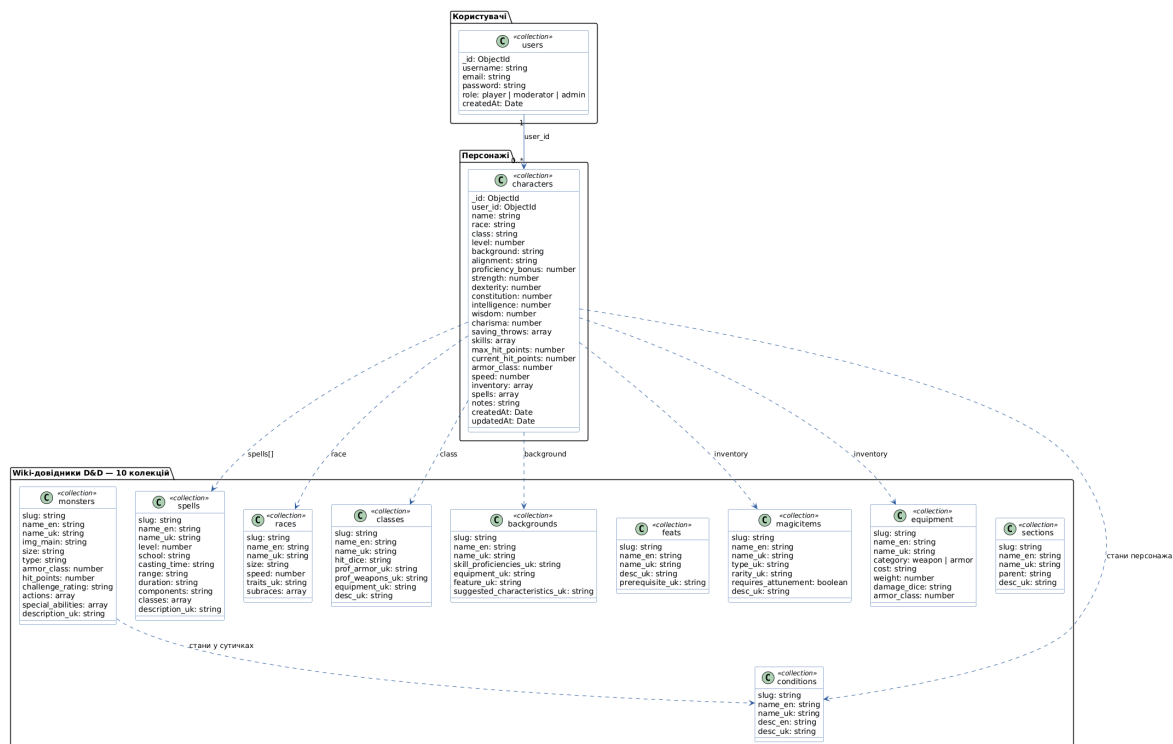


Рис. 2.3. Структура колекцій бази даних MongoDB

Отже, розроблена структура бази даних відповідає функціональним потребам інформаційної системи. Використання MongoDB Atlas забезпечує гнучке зберігання складних ігрових сутностей, підтримку вкладених структур і зручну інтеграцію з Next.js-застосунком через Mongoose ODM. Наявність окремих колекцій для користувачів, десяти типів вікі-контенту та персонажів дозволяє логічно розділити дані системи. Додаткові поля, зокрема `img_main` у монстрів, `saving_throws` і `proficiency_bonus` у персонажів, а також розширена рольова модель `users`, забезпечують підтримку оновлених функцій системи: відображення зображень, редагування персонажів, керування навичками, рятівними кидками та ролями користувачів.

2.3 Проєктування та реалізація алгоритмів роботи системи

Після визначення варіантів використання та розробки структури бази даних необхідно описати алгоритми роботи основних модулів інформаційної системи. Алгоритми визначають послідовність дій, які виконує система під час обробки запитів користувача, взаємодії з базою даних, зовнішніми API та внутрішніми програмними модулями.

У розроблюваній інформаційній системі основними алгоритмами є: алгоритм парсингу даних з Open5e REST API [21], алгоритм AI-перекладу контенту, алгоритм роботи AI-асистента [9, 10], алгоритм створення та редагування персонажа, алгоритм кидання дайсів, алгоритм роботи будівника сутичок і алгоритм керування ролями користувачів. Кожен із цих алгоритмів відповідає за окрему функціональну частину системи та забезпечує взаємодію між клієнтським інтерфейсом, серверною логікою, базою даних MongoDB Atlas [14] і зовнішніми сервісами.

Одним із ключових алгоритмів системи є алгоритм парсингу даних з Open5e REST API [21]. Його призначення полягає в автоматизованому отриманні структурованих матеріалів Dungeons & Dragons і збереженні їх у базі даних. Парсинг запускається з адміністративної панелі користувачем із роллю moderator або admin. Перед запуском модератор обирає тип контенту та кількість сторінок, які необхідно обробити.

У системі реалізовано парсинг десяти типів контенту: монстрів, заклинань, рас, класів, передісторій, здібностей, магічних предметів, ігрових станів, спорядження та розділів правил [21, 27]. Після натискання відповідної кнопки клієнтська частина надсилає POST-запит до серверного маршруту /api/admin/scrape із параметрами типу контенту та максимальної кількості сторінок для обробки.

На серверному рівні спочатку виконується перевірка прав доступу користувача. Якщо користувач не має ролі moderator або admin, система припиняє виконання операції та повертає помилку доступу. Якщо права підтверджено, система запускає цикл обробки сторінок зовнішнього API. Для кожної сторінки виконується запит до відповідного endpoint Open5e REST API за допомогою Axios [5, 21]. Отримані дані проходять етап трансформації, під час якого структура зовнішнього запису приводиться до внутрішнього формату системи.

Для записів монстрів додатково обробляється поле `img_main`, яке містить посилання на зображення монстра, якщо воно доступне у джерелі даних. Це поле

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

надалі використовується для відображення зображень у каталозі монстрів і на сторінці детального перегляду.

Після нормалізації даних система виконує операцію вставки або оновлення запису в MongoDB Atlas. Для цього використовується унікальне поле slug. Якщо запис із таким slug уже існує, він оновлюється; якщо не існує – створюється новий документ. Особливу увагу приділено збереженню україномовних полів. Під час повторного парсингу вже наявні переклади не повинні стиратися порожніми значеннями з зовнішнього API, тому для україномовних полів використовується підхід, за якого вони встановлюються лише під час створення нового запису. Це дозволяє повторно оновлювати дані з Open5e REST API без втрати результатів попереднього AI-перекладу або ручного редагування.

Алгоритм завершується після обробки заданої кількості сторінок або після отримання ознаки відсутності наступної сторінки у відповіді API. Алгоритм парсингу даних з Open5e REST API наведено на рисунку 2.4. (Додаток А).

Наступним важливим алгоритмом є алгоритм AI-перекладу контенту. Він використовується в адміністративній панелі для перекладу довідкових матеріалів українською мовою. На початковому етапі переклад підтримувався лише для окремих типів контенту, однак у поточній версії системи його розширено на всі десять wiki-розділів: монстрів, заклинання, раси, класи, передісторії, здібності, магічні предмети, стани, спорядження та правила.

У системі передбачено два режими перекладу: масовий і точковий. У масовому режимі модератор або адміністратор обирає тип контенту та кількість записів, які потрібно перекласти. Після цього клієнтська частина надсилає POST-запит до маршруту /api/admin/translate. Серверна частина знаходить записи відповідного типу, які ще не мають україномовного перекладу, і послідовно обробляє їх за допомогою Gemini API [9, 10].

У точковому режимі біля окремого запису в адміністративній панелі доступна кнопка «Перекласти». Після її натискання система надсилає запит на переклад конкретного документа. Такий підхід є зручним у випадках, коли

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

необхідно перекласти або повторно обробити лише один запис без запуску масового перекладу для всієї колекції.

Для кожного запису формується промпт до Gemini API ^[9, 10]. У промпті враховується контекст Dungeons & Dragons і спеціалізована термінологія, що дозволяє підвищити точність локалізації. Відповідь AI-моделі очікується у структурованому JSON-форматі. Після отримання відповіді система очищує її від службових markdown-маркерів, виконує JSON.parse [13] та оновлює відповідні україномовні поля в базі даних.

Для зменшення ризику перевищення лімітів запитів між зверненнями до AI-сервісу використовується затримка. Також передбачена retry-логіка: якщо Gemini API тимчасово недоступний або повертає помилку, система може повторити запит після паузи. Це підвищує надійність роботи модуля перекладу.

Під час збереження результату перекладу система не замінює початковий англomовний текст, а записує україномовний варіант в окремі поля документа. Це дозволяє зберігати вихідні дані з Open5e REST API та одночасно працювати з локалізованим контентом у користувацькому інтерфейсі. Такий підхід є зручним і для подальшого оновлення матеріалів: якщо запис повторно завантажується з відкритого джерела, уже наявний переклад можна залишити без змін і не виконувати зайву обробку через AI-сервіс.

Також у модулі перекладу враховано, що різні типи контенту мають неоднакову структуру. Наприклад, для монстрів важливими є назви дій, характеристики й описи здібностей, для заклинань — рівень, школа магії та текст ефекту, а для передісторій або правил — довші текстові блоки з можливими списками чи таблицями. Через це після отримання відповіді система оновлює тільки ті поля, які стосуються конкретного типу запису. Це дає змогу використовувати один загальний механізм перекладу для всіх wiki-розділів, не втрачаючи особливостей окремих колекцій.

Алгоритм AI-перекладу контенту наведено на рисунку 2.5.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

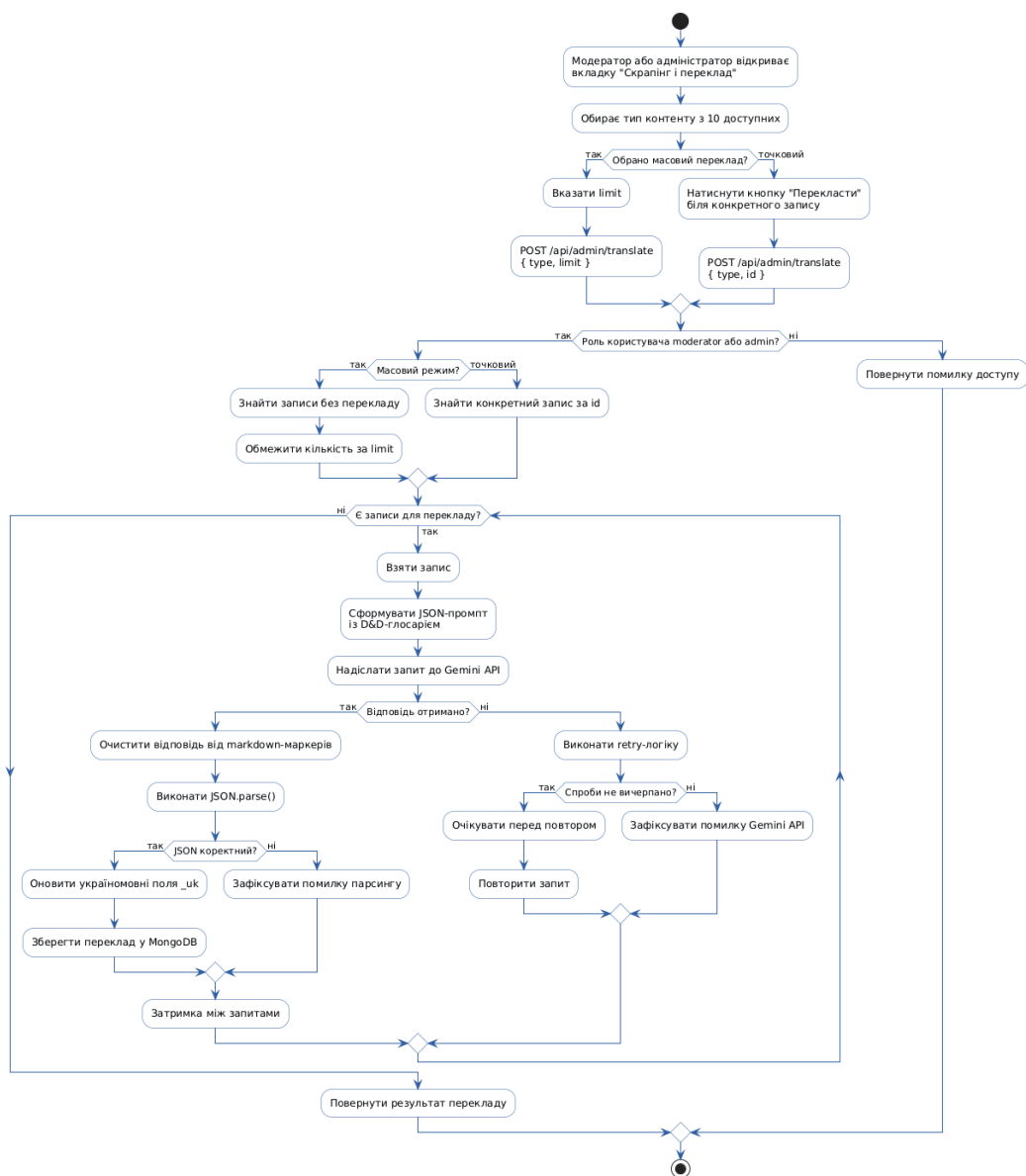


Рис. 2.5. Алгоритм AI-перекладу контенту за допомогою Gemini API

Окремим функціональним модулем є AI-асистент у вигляді чату. Його робота базується на підході RAG, тобто перед формуванням відповіді система шукає релевантний контекст у власній базі даних. Користувач надсилає повідомлення через інтерфейс чату, після чого запит передається до серверного маршруту `/api/chat`. На сервері перевіряється наявність активної сесії користувача.

Після перевірки сесії система виконує попередню обробку тексту запиту: видаляє зайву пунктуацію, розбиває повідомлення на ключові слова та відбирає найбільш значущі з них. Далі виконується паралельний пошук у кількох колекціях MongoDB, зокрема серед монстрів, заклинань, рас, класів,

передісторій, магічних предметів та інших довідкових матеріалів. Пошук здійснюється за допомогою регулярних виразів по англомовних та україномовних назвах.

На основі знайдених результатів формується контекстний блок, який містить назви релевантних сутностей і короткі фрагменти описів. Цей блок додається до повідомлення користувача та передається до Gemini API разом із системним промптом і частиною історії діалогу. Після отримання відповіді від AI-моделі система повертає її у клієнтський інтерфейс чату.

Алгоритм роботи AI-асистента наведено на рисунку 2.6.

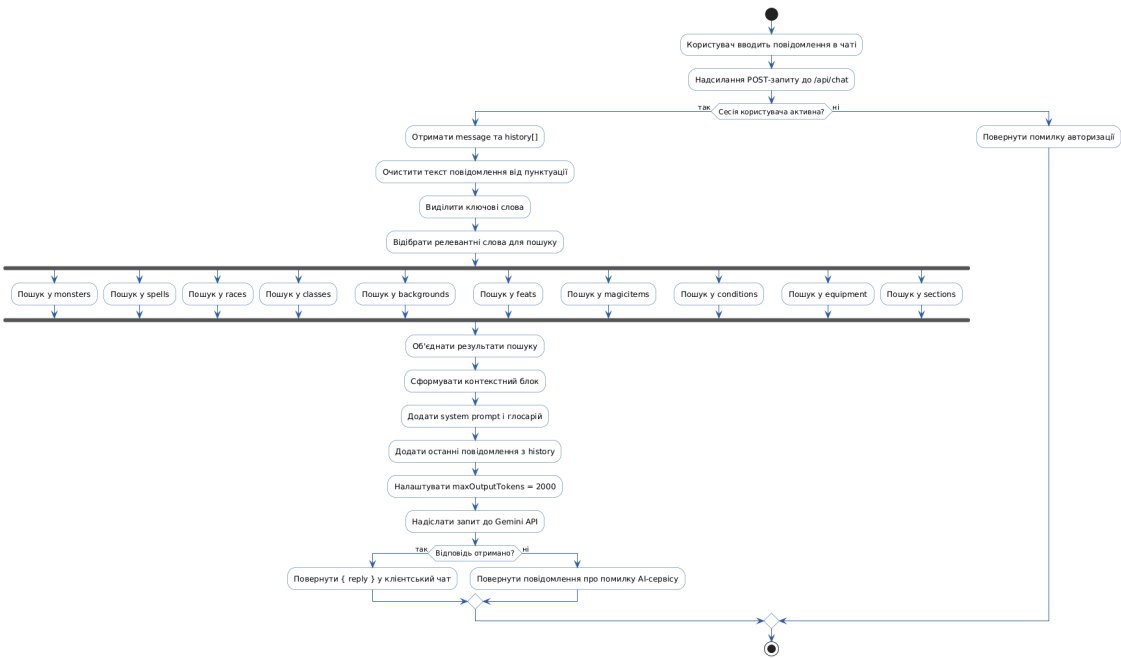


Рис. 2.6. Алгоритм роботи AI-асистента Gemini з використанням RAG-пошуку

Важливим користувацьким сценарієм є створення персонажа. Для цього система спочатку завантажує з бази даних списки доступних рас, класів і передісторій. Користувач заповнює форму створення персонажа, вказуючи ім'я, расу, клас, рівень, передісторію, світогляд, основні характеристики, очки здоров'я, клас обладунку та швидкість. Під час введення характеристик система може розраховувати модифікатори за стандартною формулою Dungeons & Dragons.

Після заповнення форми дані надсилаються POST-запитом до маршруту `/api/characters`. Серверна частина перевіряє сесію користувача, отримує `user_id` з JWT-сесії та створює новий документ у колекції `characters`. Після успішного збереження система перенаправляє користувача на сторінку аркуша створеного персонажа. Окрім створення нового персонажа, система підтримує редагування вже наявного персонажа через сторінку `/character/[id]/edit`. Під час редагування користувач може змінити основні параметри персонажа, характеристики, навички, рятівні кидки, НР, клас обладунку та інші дані. Для навичок передбачено вибір рівня володіння: звичайна майстерність або експертиза. Рятівні кидки зберігаються в окремому полі `saving_throws`. Бонус майстерності розраховується автоматично за формулою $\text{Math.ceil}(\text{level} / 4) + 1$.

На аркуші персонажа користувач може змінювати поточні НР. Для цього вводиться значення зміни, після чого система обчислює нове значення `current_hit_points`. Щоб уникнути некоректних значень, результат обмежується діапазоном від 0 до `max_hit_points`. Оновлені дані передаються PATCH-запитом до маршруту `/api/characters/[id]` і зберігаються в базі даних.

Під час створення персонажа важливо не лише зберегти введені користувачем дані, а й підготувати їх до подальшого використання в інтерактивному аркуші. Саме тому частина полів має числовий формат і використовується для розрахунків: характеристики впливають на модифікатори, рівень персонажа — на бонус майстерності, а значення НР застосовуються під час роботи з поточним станом персонажа. Така структура дає змогу не дублювати обчислення вручну й забезпечує узгодженість між формою створення, сторінкою редагування та аркушем персонажа.

Оновлення поточних НР реалізовано окремо від редагування основних даних персонажа, оскільки ця дія може виконуватися значно частіше під час ігрової сесії. Користувач змінює не повний запис персонажа, а лише значення, пов'язане з поточним станом здоров'я. Це спрощує взаємодію з аркушем і зменшує ризик випадково змінити інші параметри. Обмеження значення в межах

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

від 0 до максимального НР дозволяє уникнути некоректних станів, наприклад від'ємного здоров'я або перевищення максимально допустимого значення.

Алгоритм створення персонажа та оновлення НР наведено на рисунку 2.7.

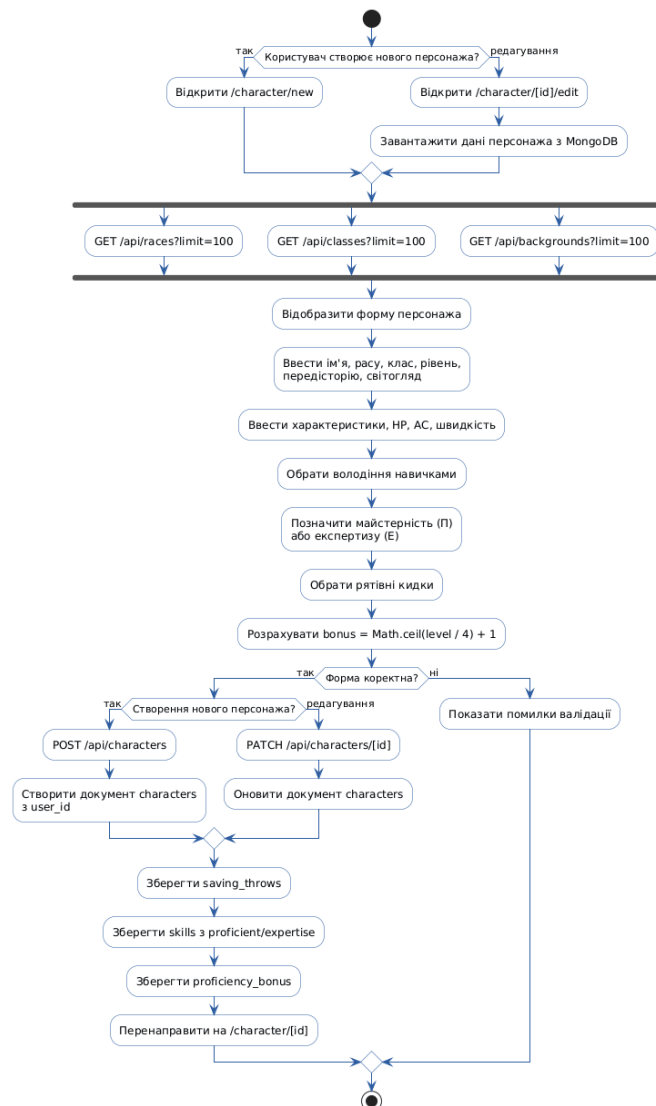


Рис. 2.7. Алгоритм створення персонажа та оновлення його поточних НР

Для підтримки ігрового процесу в системі реалізовано модуль кидання дайсів. Він дозволяє користувачу виконувати як стандартні кидки окремих кубиків, так і складні дісе-вирази. Алгоритм починається з введення користувачем виразу, наприклад 1d20+5 або 2d6+3. Система нормалізує рядок, видаляє пробіли, переводить текст до нижнього регістру та визначає наявність модифікатора.

Після цього виконується розбір d-нотації, тобто визначення кількості кубиків і кількості граней. Якщо вказано перевагу або недолік, система генерує

два значення кидка d20 і вибирає відповідно більше або менше значення. Для кожного кубика результат генерується випадково за допомогою Math.random. Після виконання всіх кидків система додає модифікатор, визначає підсумкове значення та тип результату. Для d20 може визначатися критичний успіх у разі випадіння 20 або критичний провал у разі випадіння 1.

Алгоритм кидання дайсів наведено на рисунку 2.8.

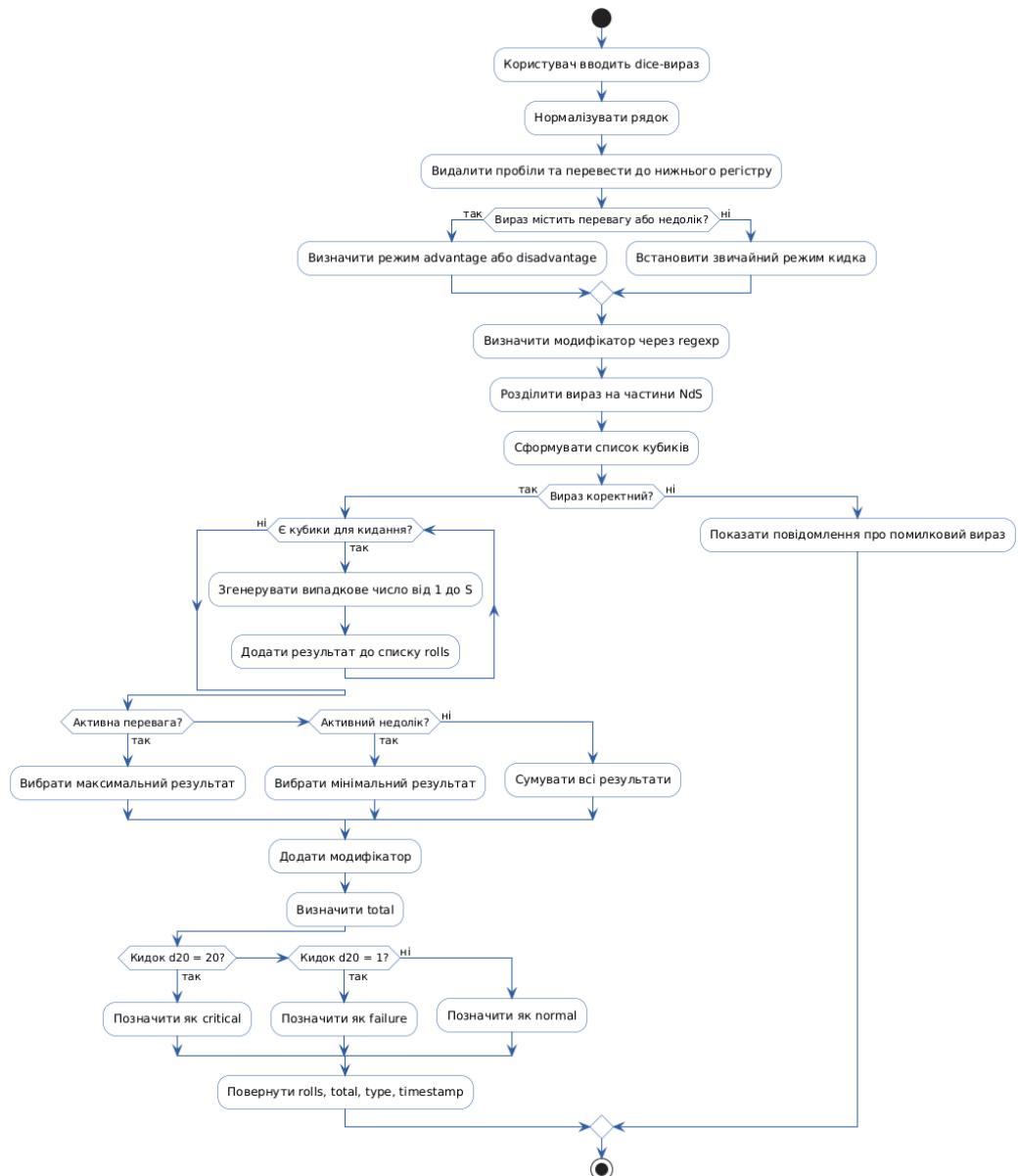


Рис. 2.8. Алгоритм кидання дайсів

Ще одним важливим модулем системи є будівник сутичок. Його призначення полягає у формуванні бойового зіткнення з персонажами та монстрами, визначенні порядку ініціативи, відстеженні раундів, НР і станів

учасників. Користувач може додавати до сутички монстрів із бази даних та персонажів, створених у системі.

Після додавання учасників система формує список бойових одиниць. Для монстрів ініціатива може генеруватися автоматично, а для персонажів користувач може ввести її вручну. Далі учасники сортуються за спаданням ініціативи. Під час бою система зберігає індекс активного учасника та номер поточного раунду. Після завершення ходу викликається функція nextTurn, яка переходить до наступного учасника. Якщо список учасників завершено, індекс повертається на початок, а номер раунду збільшується.

Модуль також дозволяє змінювати НР учасників і додавати або знімати ігрові стани. Для зміни НР використовується обмеження значення в межах від 0 до максимального НР учасника. Скидання сутички відновлює НР до максимального значення та очищує список станів.

Під час формування сутички важливо, щоб дані про учасників мали єдину структуру незалежно від того, чи це монстр із довідника, чи персонаж користувача. Для цього під час додавання до списку бою система приводить потрібні характеристики до спільного вигляду: назву, максимальні та поточні НР, ініціативу, активні стани й тип учасника. Завдяки цьому подальша логіка бою може працювати з одним списком учасників, не розділяючи окремо монстрів і персонажів під час переходу ходів або зміни їхнього стану.

Будівник сутичок розрахований на використання безпосередньо під час ігрової сесії, тому основні дії виконуються швидко й без переходу між додатковими сторінками. Користувач може змінити НР, додати стан, перейти до наступного ходу або скинути сутичку в межах одного інтерфейсу. Це зменшує кількість ручних записів і допомагає майстру гри зосередитися на перебігу бою, а не на постійному веденні допоміжних таблиць чи нотаток.

Алгоритм роботи будівника сутичок наведено на рисунку 2.9.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

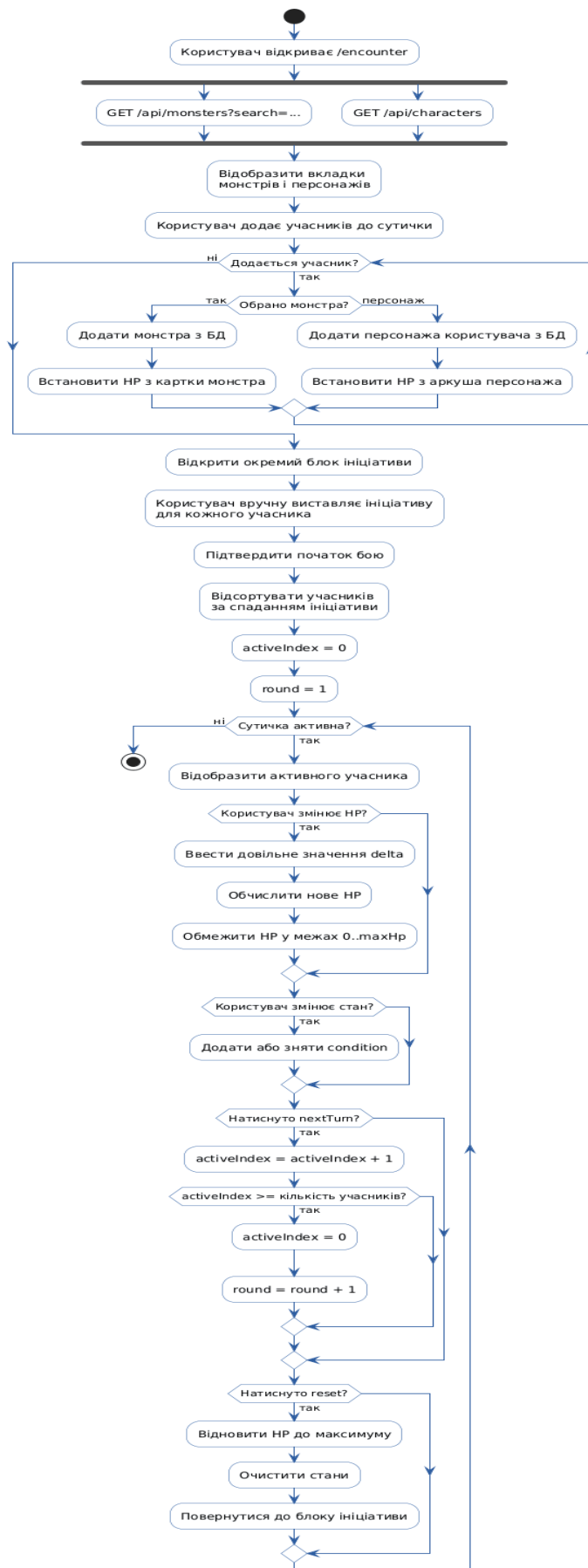


Рис. 2.9. Алгоритм роботи будівника сутичок

Окремим адміністративним процесом є керування ролями користувачів. Ця функція доступна лише користувачу з роллю admin. Після відкриття вкладки «Користувачі» адміністративна панель надсилає GET-запит до маршруту `/api/admin/users` і отримує список зареєстрованих користувачів. Для кожного користувача відображається поточна роль і елемент select для її зміни. Після вибору нової ролі система надсилає PATCH-запит до `/api/admin/users`, перевіряє права поточного користувача та оновлює поле `role` у колекції `users`. Такий алгоритм дозволяє централізовано керувати доступом до функцій системи.

Отже, у цьому підрозділі було описано основні алгоритми роботи інформаційної системи. Розглянуті алгоритми охоплюють як адміністративні процеси, зокрема парсинг і переклад даних, так і користувацькі сценарії: роботу з AI-асистентом, створення персонажа, кидання дайсів та формування бойових сутичок. Формалізація цих алгоритмів дозволяє перейти до опису програмної реалізації основних модулів системи.

2.4 Реалізація інформаційної системи для поціновувачів **Dungeons & Dragons**

Після проєктування варіантів використання, структури бази даних та алгоритмів роботи основних модулів було виконано програмну реалізацію інформаційної системи. Розроблений програмний продукт є вебзастосунком, побудованим на базі Next.js 14 з використанням App Router [17, 18]. Такий підхід дозволяє поєднати клієнтську частину, серверну логіку, API-маршрути, модулі автентифікації, роботу з базою даних і зовнішніми сервісами в межах одного репозиторію.

Застосунок реалізовано з використанням мови TypeScript [29], що забезпечує статичну типізацію та підвищує надійність роботи з даними. Для побудови інтерфейсу використано React-компоненти [22], Tailwind CSS [28] та shadcn/ui [26]. Інтерфейс системи оформлено в темній темі з використанням відтінків slate, що забезпечує єдиний візуальний стиль для wiki-довідників, сторінок деталей, форм створення й редагування персонажа, AI-чату, будівника сутичок, інструмента кидання дайсів та адміністративної панелі.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

Серверна частина реалізована засобами Next.js API Routes. Для збереження даних використовується MongoDB Atlas у поєднанні з Mongoose ODM [14, 16]. Автентифікацію користувачів реалізовано за допомогою NextAuth.js v5 із JWT-сесіями [4, 19]. AI-функціональність і переклад контенту реалізовано через Google Gemini API [9, 10]. Для відображення форматowanego тексту у довідниках використано react-markdown та remark-gfm [23, 24].

Структура проєкту організована таким чином, щоб відокремити сторінки застосунку, серверні маршрути, службові бібліотеки, моделі бази даних, модулі парсингу та AI-модулі. Основні директорії проєкту наведено в таблиці 2.5.

Таблиця 2.5

Структура директорій програмного проєкту

Директорія або файл	Призначення
app/	Основна директорія Next.js App Router, що містить сторінки застосунку та API-маршрути
app/(auth)/	Публічні сторінки авторизації та реєстрації
app/(main)/	Захищені сторінки основної частини застосунку
app/api/	Серверні API-маршрути для роботи з даними, автентифікацією, AI-чатом та адміністративними функціями
lib/auth.ts	Конфігурація NextAuth, налаштування credentials-провайдера та callback-функцій JWT
lib/db/	Модулі підключення до MongoDB Atlas і Mongoose-моделі
lib/ai/	Модулі для роботи з Gemini API та перекладом контенту
lib/scrapers/	Модулі для отримання, трансформації та збереження даних з Open5e REST API
lib/utils/	Допоміжні функції та словники перекладів
proxu.ts	Middleware для захисту маршрутів, перевірки авторизації та ролі користувача

Користувацький інтерфейс системи реалізовано у вигляді набору сторінок. Публічна частина містить сторінки входу та реєстрації. Основна частина застосунку є захищеною та доступна лише авторизованим користувачам. Вона

містить головну сторінку, wiki-довідники, сторінки детального перегляду сутностей, інструменти гравця, AI-чат та адміністративну панель.

Реалізація wiki-довідників побудована за спільним принципом. Для кожного типу контенту створено сторінку списку та, за потреби, сторінку детального перегляду. У системі реалізовано десять wiki-розділів: монстри, заклинання, раси, класи, передісторії, здібності, магічні предмети, стани, спорядження та правила. Сторінки списків використовують API-маршрути для отримання даних із MongoDB Atlas, підтримують пагінацію, пошук і фільтрацію за параметрами. Наприклад, для монстрів реалізовано фільтрацію за рейтингом складності та розміром, для заклинань – за рівнем і школою магії, для магічних предметів – за рідкістю та необхідністю налаштування, а для спорядження – за категорією.

Для монстрів додатково реалізовано відображення зображень, отриманих із поля `img_main`. Зображення показуються як у каталозі, так і на сторінці детального перегляду монстра. Для цього використовується компонент `MonsterImage.tsx`, який підтримує модальне відкриття зображення при натисканні. Якщо зображення недоступне або не завантажується, обробник `onError` приховує його, що дозволяє уникнути некоректного відображення інтерфейсу.

Сторінки детального перегляду використовують динамічний параметр `slug`. Це дозволяє формувати зрозумілі URL-адреси та відкривати конкретний запис довідника. Наприклад, маршрут `/wiki/monsters/[slug]` використовується для перегляду окремого монстра, а `/wiki/spells/[slug]` – для перегляду окремого заклинання. Такий підхід спрощує навігацію, дозволяє використовувати унікальний ідентифікатор запису та відповідає структурі даних, отриманих із Open5e REST API.

Серверна частина застосунку реалізована через API-маршрути. Вони відповідають за отримання довідкових матеріалів, роботу з персонажами, реєстрацію користувачів, взаємодію з AI-асистентом і виконання

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

адміністративних операцій. Основні групи API-маршрутів наведено в таблиці 2.6.

Таблиця 2.6

Основні API-маршрути інформаційної системи

Група маршрутів	Приклади маршрутів	Призначення
Довідники	/api/monsters, /api/spells, /api/races, /api/classes, /api/backgrounds, /api/feats, /api/magic-items, /api/conditions, /api/equipment, /api/sections	Отримання довідкових матеріалів із підтримкою пошуку, пагінації та фільтрації
Детальний перегляд	/api/monsters/[slug], /api/spells/[slug], /api/races/[slug], /api/classes/[slug], /api/backgrounds/[slug], /api/feats/[slug], /api/magic-items/[slug], /api/sections/[slug]	Отримання одного запису за slug
Персонажі	/api/characters, /api/characters/[id]	Створення, перегляд, оновлення, редагування та видалення персонажів поточного користувача
Автентифікація	/api/auth/[...nextauth], /api/auth/register	Обробка входу, сесій та реєстрації користувачів
AI-чат	/api/chat	Пошук контексту в базі даних і формування відповіді через Gemini API
Адміністрування контенту	/api/admin/scrape, /api/admin/translate, /api/admin/content/[type]/[id]	Скрапінг десяти типів контенту, масовий і точковий AI-переклад, оновлення та видалення записів бази даних
Керування користувачами	/api/admin/users	Отримання списку користувачів і зміна їхніх ролей користувачем із роллю admin

Модуль автентифікації реалізовано за допомогою NextAuth.js v5. Конфігурація автентифікації розміщена у файлі lib/auth.ts. Для входу користувача використовується credentials-провайдер. Під час реєстрації пароль користувача хешується, після чого дані облікового запису зберігаються в колекції users. Роль користувача передається в JWT-токен через callback-функції NextAuth, що дозволяє перевіряти доступ до захищених сторінок і API-маршрутів. У поточній реалізації передбачено ролі player, moderator та admin.

Захист маршрутів реалізовано у файлі proxy.ts. Middleware перевіряє наявність активної сесії користувача, виконує перенаправлення неавторизованих користувачів на сторінку входу та обмежує доступ до адміністративної панелі лише для користувачів із роллю moderator або admin. Таким чином, адміністративні функції, пов'язані з парсингом, перекладом, редагуванням і видаленням даних, відокремлені від звичайної користувацької функціональності.

Робота з базою даних винесена в директорію lib/db. Підключення до MongoDB Atlas описано у файлі lib/db/mongoose.ts. Щоб під час роботи застосунку не створювалися зайві з'єднання з базою даних, використовується повторне використання вже відкритого підключення. Моделі даних розміщені в директорії lib/db/models. Вони описують структуру колекцій користувачів, довідкових матеріалів і персонажів.

Логіка отримання даних із зовнішнього джерела розміщена в директорії lib/scrapers. У файлі lib/scrapers/open5e.ts описано клієнт для роботи з Open5e REST API, типи даних і функції отримання записів для різних розділів контенту. Файл lib/scrapers/scrapper.ts використовується для перетворення отриманих записів до структури, прийнятої в системі, та подальшого збереження в MongoDB Atlas. Під час збереження застосовується upsert-логіка: якщо запис уже існує, він оновлюється, а якщо ні — створюється новий документ.

AI-модуль системи складається з двох основних частин: модуля перекладу та модуля AI-чату. Модуль перекладу реалізовано у файлі lib/ai/translator.ts і використовується для перекладу всіх десяти типів wiki-контенту. Він підтримує як масовий переклад записів відповідного типу, так і точковий переклад

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

окремого запису з адміністративної панелі. Модуль AI-чату реалізований через маршрут `app/api/chat/route.ts`. Він виконує пошук релевантного контексту в MongoDB і формує відповідь користувачу за допомогою Gemini API.

Адміністративна частина системи розміщена на сторінці `/admin`. Доступ до неї мають користувачі з ролями `moderator` та `admin`, але набір доступних дій залежить від конкретної ролі. Панель поділена на три вкладки: «Скрапінг і переклад», «Управління контентом» та «Користувачі». У вкладці «Скрапінг і переклад» можна запускати завантаження матеріалів з Open5e REST API, обирати кількість сторінок для обробки та виконувати масовий переклад записів. Вкладка «Управління контентом» використовується для пошуку, перегляду, редагування, видалення й окремого перекладу записів. Розділ «Користувачі» доступний тільки для ролі `admin`. У ньому адміністратор переглядає список зареєстрованих користувачів і може змінювати їхні ролі.

Ігрові інструменти винесені в окремі сторінки застосунку. На сторінці `/character` користувач переглядає власних персонажів, на `/character/new` створює нового персонажа, на `/character/[id]` працює з його інтерактивним аркушем, а сторінка `/character/[id]/edit` призначена для редагування вже збережених даних. У формах створення та редагування можна вказати основні параметри персонажа: ім'я, расу, клас, рівень, передісторію, світогляд, характеристики, HP, клас обладунку, швидкість, навички та рятівні кидки. Для навичок передбачено позначення майстерності або експертизи, а бонус майстерності визначається автоматично відповідно до рівня персонажа.

Будівник сутичок на сторінці `/encounter` дозволяє додавати до бою як монстрів із бази даних, так і власних персонажів користувача. Перед початком бою передбачено окремий блок налаштування ініціативи, у якому користувач вручну виставляє значення для кожного учасника. Під час бою система дозволяє переходити між ходами, відстежувати раунди, змінювати HP учасників довільним значенням і додавати або знімати ігрові стани.

Для підтримки форматowanego тексту у довідкових матеріалах використовується `react-markdown` разом із `remark-gfm`. Це дає змогу коректно

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

відображати описи, списки та таблиці, зокрема для передісторій персонажів і правил Dungeons & Dragons.

Компонентна структура реалізованого застосунку наведена на рисунку 2.10.

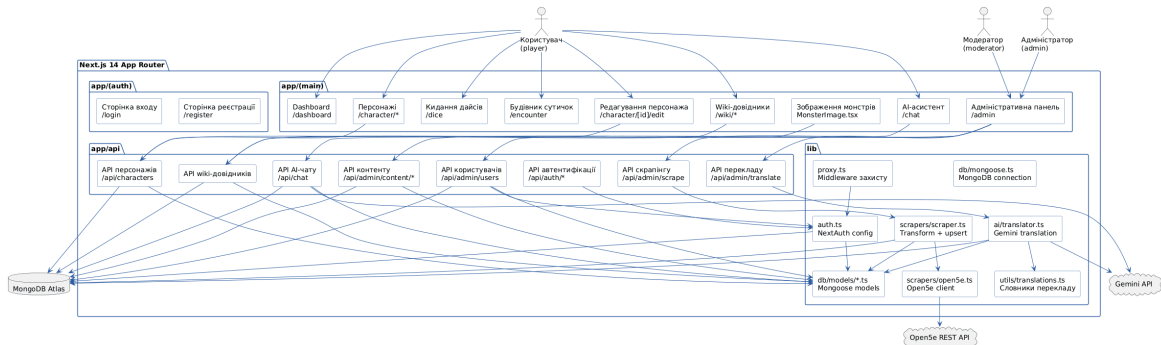


Рис. 2.10. Компонентна структура реалізованого застосунку

Отже, реалізація інформаційної системи побудована за модульним принципом. Користувацький інтерфейс, серверні маршрути, модулі автентифікації, база даних, парсинг, AI-асистент, адміністративні функції та інструменти гравця відокремлені логічно, але працюють у межах єдиного Next.js-застосунку. Такий підхід спрощує підтримку програмного продукту, забезпечує контроль доступу до функцій системи та створює основу для подальшого розширення функціональності.

Висновки до другого розділу

У другому розділі було виконано проектування інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента. На основі визначених функціональних і нефункціональних вимог було сформовано основні варіанти використання системи, виділено ролі користувачів і визначено їхні можливості. У системі передбачено ролі player, moderator та admin, що дозволяє розмежувати доступ до основних користувацьких функцій, інструментів керування контентом і функцій адміністрування користувачів.

Було спроектовано структуру бази даних MongoDB Atlas. Вона включає колекцію users для збереження облікових записів користувачів, колекцію characters для збереження персонажів, а також десять wiki-колекцій для

довідкового контенту Dungeons & Dragons: monsters, spells, races, classes, backgrounds, feats, magic items, conditions, equipment та sections. Така структура дає змогу зберігати як основні текстові дані, так і складні вкладені об'єкти, масиви характеристик, дії, здібності, навички, рятівні кидки та інші елементи, необхідні для роботи системи.

Окрему увагу було приділено алгоритмам роботи системи. Було описано алгоритм парсингу даних з Open5e REST API, який забезпечує автоматизоване наповнення бази даних десятьма типами ігрового контенту. Під час повторного парсингу передбачено збереження вже наявних україномовних перекладів, що дозволяє оновлювати вихідні дані без втрати результатів локалізації. Також було розглянуто алгоритм AI-перекладу контенту за допомогою Gemini API, який підтримує як масовий переклад записів, так і точковий переклад окремої сутності через адміністративну панель.

Було спроектовано та реалізовано алгоритм роботи AI-асистента, який використовує пошук релевантного контексту в базі даних перед формуванням відповіді користувачу. Це дозволяє AI-асистенту відповідати не лише на загальні питання, а й враховувати матеріали, збережені у wiki-довідниках системи. Для отримання повніших відповідей збільшено параметр maxOutputTokens, що зменшує ризик обрізання пояснень.

Також було описано реалізацію користувацьких інструментів. Модуль персонажів підтримує створення, перегляд і редагування персонажа, налаштування характеристик, навичок, експертизи, рятівних кидків, НР, класу обладунку та інших параметрів. Бонус майстерності персонажа розраховується автоматично відповідно до його рівня. Будівник сутичок дозволяє додавати до бою не лише монстрів, а й персонажів користувача з бази даних, вручну виставляти ініціативу перед початком бою, змінювати НР довільним значенням і керувати станами учасників.

У межах реалізації адміністративної частини було передбачено панель керування зі вкладками для скрапінгу й перекладу, управління контентом і керування користувачами. Користувачі з роллю moderator можуть виконувати

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

операції з довідковим контентом, а користувачі з роллю admin додатково мають можливість переглядати список користувачів і змінювати їхні ролі. Це забезпечує гнучке адміністрування системи та контроль доступу до критичних функцій.

Отже, у другому розділі було сформовано проєктну основу інформаційної системи, визначено її функціональні сценарії, структуру бази даних, основні алгоритми та компоненти реалізації. Запропоновані рішення забезпечують підтримку довідкових матеріалів Dungeons & Dragons, автоматизованого наповнення бази даних, україномовної локалізації, інтерактивних інструментів гравця, AI-асистента та адміністративного керування системою.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ІНФОРМАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОЦІНОВУВАЧІВ DUNGEONS & DRAGONS

3.1 Порядок встановлення та налаштування параметрів системи

Розроблена інформаційна система для поціновувачів настільно-рольової гри Dungeons & Dragons є веборієнтованим Full-Stack застосунком, реалізованим на базі Next.js 14 з використанням App Router. Для роботи системи необхідне серверне середовище з підтримкою Node.js [20], доступ до хмарної бази даних MongoDB Atlas, налаштований механізм автентифікації NextAuth та API-ключ для взаємодії з Google Gemini API.

Оскільки система є вебзастосунком, кінцевому користувачу не потрібно встановлювати додаткове програмне забезпечення. Для роботи достатньо сучасного веббраузера та підключення до мережі Інтернет. Усі основні операції виконуються через вебінтерфейс системи: перегляд довідників, створення персонажів, кидання дайсів, робота з AI-асистентом, формування сутичок та адміністрування контенту.

Для встановлення та запуску системи в середовищі розробника необхідно встановити Node.js актуальної стабільної версії, встановити менеджер пакетів npm, який постачається разом із Node.js, клонувати або скопіювати репозиторій програмного проєкту на локальний комп'ютер, встановити залежності проєкту за допомогою команди `npm install`, створити файл змінних середовища для зберігання конфігураційних параметрів, додати URL підключення до MongoDB Atlas, додати параметри, необхідні для роботи NextAuth, додати API-ключ Google Gemini, отриманий через Google AI Studio, запустити застосунок у режимі розробки, відкрити систему у браузері за локальною адресою.

Приклад послідовності команд для локального запуску застосунку [18]:

```
npm install
```

```
npm run dev
```

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

Після запуску застосунків стає доступним у браузері за адресою локального сервера розробки. Зазвичай для Next.js-застосунків використовується адреса `http://localhost:3000`, якщо інший порт не задано в налаштуваннях середовища.

Для коректної роботи системи необхідно налаштувати змінні середовища. Вони використовуються для безпечного зберігання конфігураційних параметрів, які не повинні передаватися у клієнтську частину застосунку. До таких параметрів належать URL підключення до MongoDB Atlas [14], секретні ключі автентифікації та API-ключ Gemini. Зберігання цих параметрів у змінних середовища дозволяє підвищити безпеку системи та уникнути розміщення конфіденційних даних у відкритому програмному коді.

Орієнтовна структура файлу змінних середовища може містити такі параметри:

`MONGODB_URI=рядок_підключення_до_MongoDB_Atlas`

`NEXTAUTH_SECRET=секретний_ключ_автентифікації`

`NEXTAUTH_URL=адреса_застосунку`

`GEMINI_API_KEY=api_ключ_Google_Gemini`

Назви змінних можуть відрізнятися залежно від фактичної реалізації проєкту, однак їх призначення залишається однаковим: забезпечити підключення до бази даних, роботу механізму автентифікації та доступ до AI-сервісу.

Для роботи з базою даних використовується MongoDB Atlas. Перед запуском системи необхідно створити хмарний кластер MongoDB Atlas, сформувати користувача бази даних, налаштувати права доступу та отримати URL-підключення. Цей URL додається до змінних середовища та використовується серверною частиною застосунку для встановлення з'єднання з базою даних. Підключення реалізовано через Mongoose ODM [16], що дозволяє працювати з колекціями MongoDB за допомогою моделей.

Для автентифікації користувачів використовується NextAuth [4, 19]. У системі реалізовано реєстрацію, вхід, підтримку JWT-сесій і передачу ролі користувача в токені. Захищені сторінки основної частини застосунку доступні

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

лише після входу до системи. Адміністративна панель додатково захищена перевіркою ролі користувача, тому доступ до неї мають лише користувачі з роллю moderator та admin.

Для роботи AI-асистента використовується Gemini API [9, 12]. API-ключ отримується через Google AI Studio та зберігається у змінних середовища. Звернення до Gemini API виконується на серверному рівні, що дозволяє не передавати ключ доступу в клієнтську частину застосунку. AI-модуль використовується для роботи чату та перекладу окремих типів ігрового контенту.

Після налаштування підключення до бази даних, автентифікації та AI-сервісу система може бути запущена. Початкове наповнення бази даних виконується через адміністративну панель. Користувач із роллю moderator або admin може запустити парсинг матеріалів з Open5e REST API [21], після чого отримані записи зберігаються в MongoDB Atlas. Надалі через адміністративну панель можна запускати AI-переклад контенту, переглядати записи, редагувати їх або видаляти.

Для розгортання системи у робочому середовищі потрібно забезпечити сервер або хмарну платформу, яка підтримує запуск Next.js-застосунків. У робочому середовищі також необхідно налаштувати змінні середовища, вказати адресу MongoDB Atlas, параметри NextAuth та API-ключ Gemini. Після цього виконується збірка застосунку та запуск у production-режимі.

Приклад команд для production-запуску:

npm run build

npm run start

На рисунку 3.1 подано діаграму розгортання системи, яка відображає взаємодію користувача, вебзастосунку, MongoDB Atlas, Open5e REST API та Gemini API.

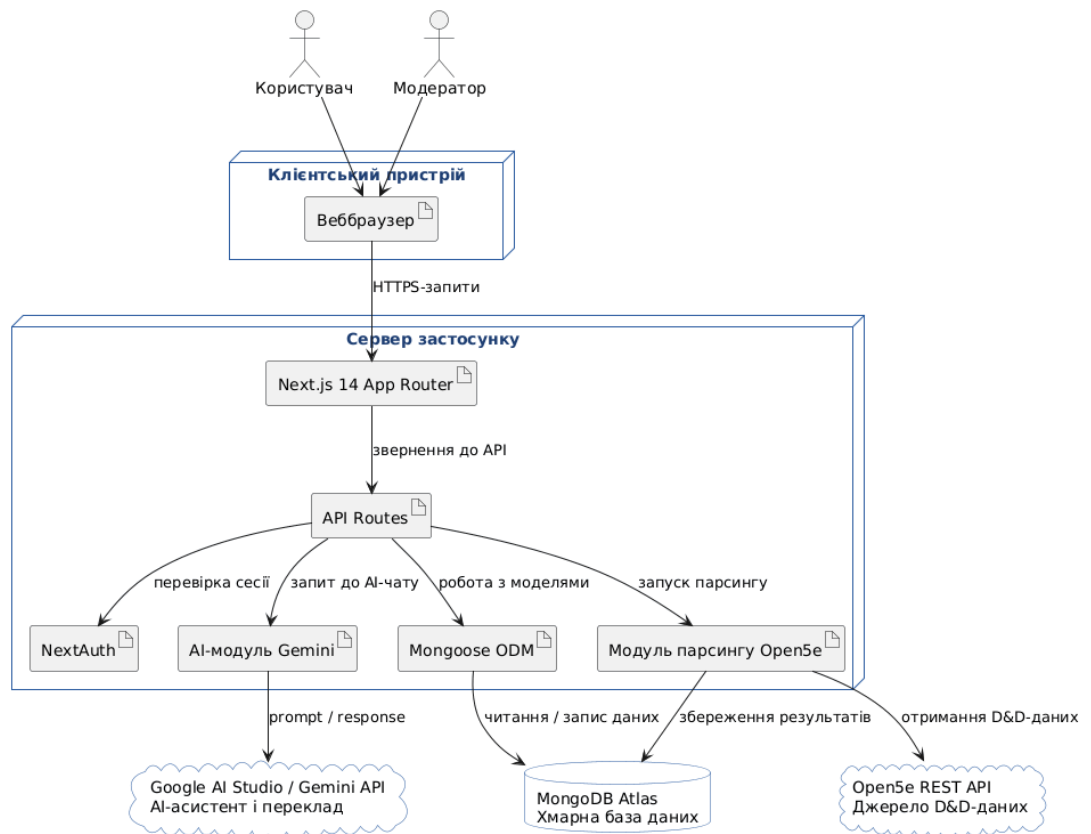


Рис. 3.1. Діаграма розгортання інформаційної системи

Після запуску системи користувач може перейти на сторінку реєстрації, створити обліковий запис і виконати вхід до системи. За замовчуванням новий користувач отримує роль `player`, яка надає доступ до основних функцій: перегляду `wiki`-довідників, створення персонажів, використання `AI`-асистента, будівника сутичок і дайсів. Для доступу до адміністративної панелі користувач повинен мати роль `moderator` або `admin`. Роль `admin` додатково відкриває можливість керування іншими користувачами.

Налаштування `MongoDB Atlas` виконується шляхом створення бази даних у хмарному середовищі, отримання `connection string` і додавання його до змінної `MONGODB_URI`. Після першого запуску система може автоматично створювати необхідні колекції під час виконання операцій запису через `Mongoose ODM`. Основними колекціями системи є `users`, `characters`, `monsters`, `spells`, `races`, `classes`, `backgrounds`, `feats`, `magicitems`, `conditions`, `equipment` та `sections`.

Для роботи `AI`-функцій необхідно отримати `API`-ключ `Gemini API` через `Google AI Studio` та додати його до файлу `.env.local`. Цей ключ використовується

у двох основних частинах системи: для AI-перекладу довідкового контенту в адміністративній панелі та для роботи AI-асистента, який формує відповіді користувачам на основі контексту, знайденого в базі даних.

Отже, процес встановлення та налаштування системи складається з підготовки середовища Node.js, встановлення залежностей, створення конфігураційного файлу, підключення MongoDB Atlas, додавання ключа Gemini API та запуску Next.js-застосунку. Такий порядок дозволяє швидко розгорнути систему як у локальному середовищі розробки, так і в production-середовищі.

3.2 Структура інтерфейсу інформаційної системи

Інтерфейс інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons реалізовано як веборієнтований користувацький інтерфейс, доступний через сучасний браузер. Основним завданням інтерфейсу є забезпечення зручного доступу до довідкових матеріалів, інструментів гравця, AI-асистента та адміністративних функцій. Під час реалізації інтерфейсу було використано компонентний підхід, що дозволяє розділити сторінки системи на окремі функціональні блоки та забезпечити повторне використання елементів.

Інтерфейс системи побудовано з використанням Next.js [17], React-компонентів [22], Tailwind CSS [28] та shadcn/ui [26]. Такий набір інструментів дозволяє створити адаптивний і візуально узгоджений інтерфейс, який коректно відображається на різних пристроях. Основні елементи інтерфейсу мають єдиний стиль оформлення, що спрощує навігацію та робить роботу користувача із системою більш послідовною.

Сторінки авторизації та реєстрації призначені для створення облікового запису та входу користувача до системи. Сторінка реєстрації містить форму введення даних нового користувача, після чого система створює обліковий запис і зберігає пароль у захищеному хешованому вигляді. Сторінка входу дозволяє користувачу пройти автентифікацію та отримати доступ до захищеної частини застосунку. Після успішного входу система формує JWT-сесію, у якій зберігається інформація про користувача та його роль.

На рисунку 3.2. зображена сторінка для входу користувача в акаунт.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

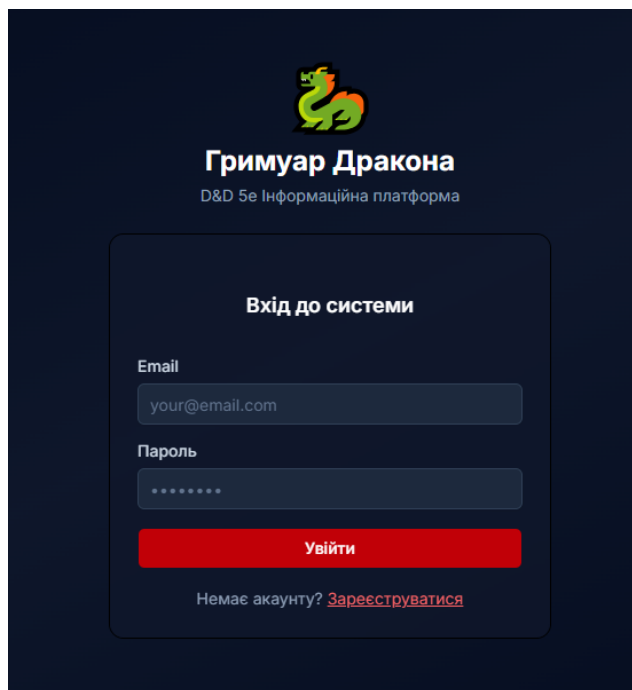


Рис. 3.2. Сторінка входу користувача до системи

Після входу користувач переходить на головну сторінку системи. Головна сторінка виконує роль початкового навігаційного центру, з якого користувач може перейти до вікі-довідників, інструментів гравця, AI-асистента або інших функціональних модулів. На цій сторінці розміщено основні категорії контенту та швидкі переходи до найважливіших розділів.

На рисунку 3.3. зображена головна сторінка системи.

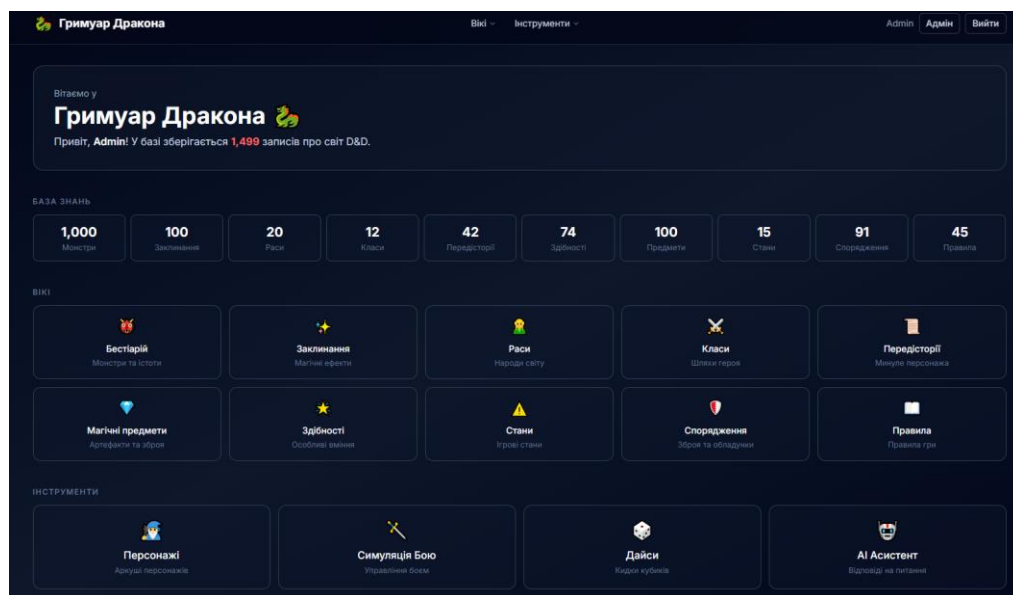


Рис. 3.3. Головна сторінка інформаційної системи

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

Однією з основних частин інтерфейсу є wiki-довідники. Вони забезпечують доступ до структурованих матеріалів Dungeons & Dragons. У системі реалізовано довідники монстрів, заклинань, рас, класів, передісторій, магічних предметів, здібностей, станів, спорядження та правил. Кожен довідник має сторінку списку записів, на якій користувач може переглядати основні відомості, здійснювати пошук і застосовувати фільтри.

Наприклад, у довіднику монстрів користувач може виконувати пошук за назвою, фільтрувати записи за рейтингом складності та розміром. У довіднику заклинань передбачено фільтрацію за рівнем і школою магії. Для магічних предметів реалізовано фільтрацію за рідкістю та ознакою налаштування. Такий підхід дозволяє швидко знаходити потрібні ігрові матеріали без перегляду великої кількості записів вручну.

На рисунку 3.4. представлено головну сторінку перегляду монстрів.

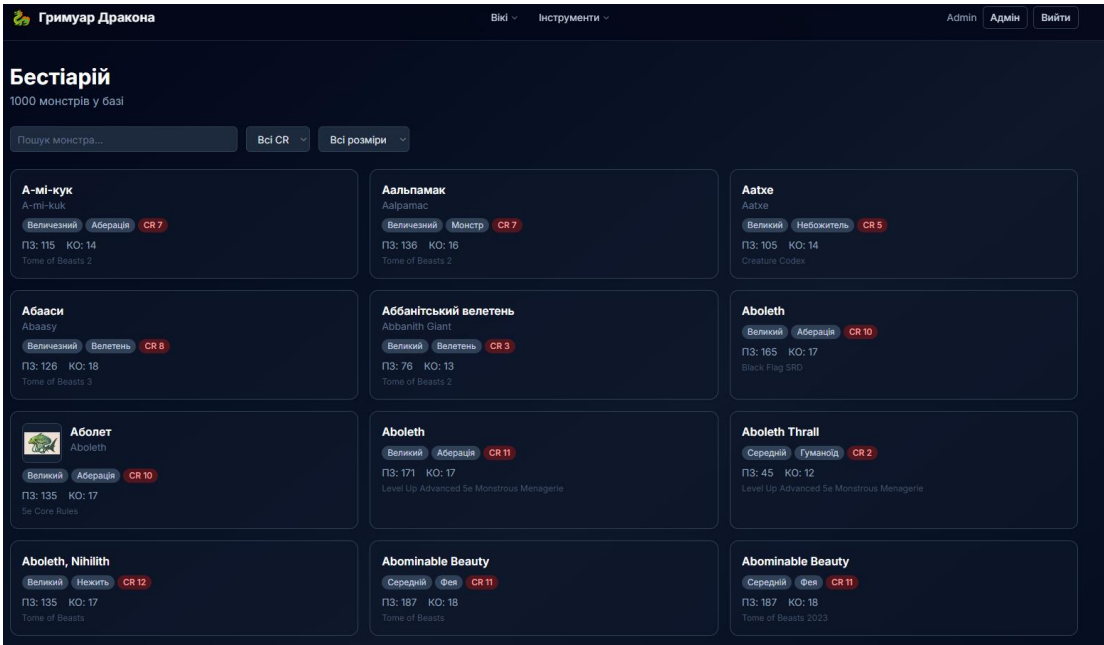


Рис. 3.4. Сторінка wiki-довідника монстрів із пошуком і фільтрацією

Для кожної ігрової сутності передбачено сторінку детального перегляду. Вона відкривається за динамічним маршрутом із використанням параметра slug. На детальній сторінці відображається повний опис запису, його основні характеристики, україномовні та англійськомовні поля, а також додаткова інформація залежно від типу сутності. Наприклад, сторінка монстра містить клас обладунку, очки здоров'я, швидкість, характеристики, рятівні килки, навички, імунітети, дії.

реакції та особливі здібності. Сторінка заклинання містить рівень, школу магії, час накладання, дистанцію, компоненти, тривалість, класи та опис ефекту.

Сторінка детальної інформації про певного монстра представлена на рисунку 3.5.

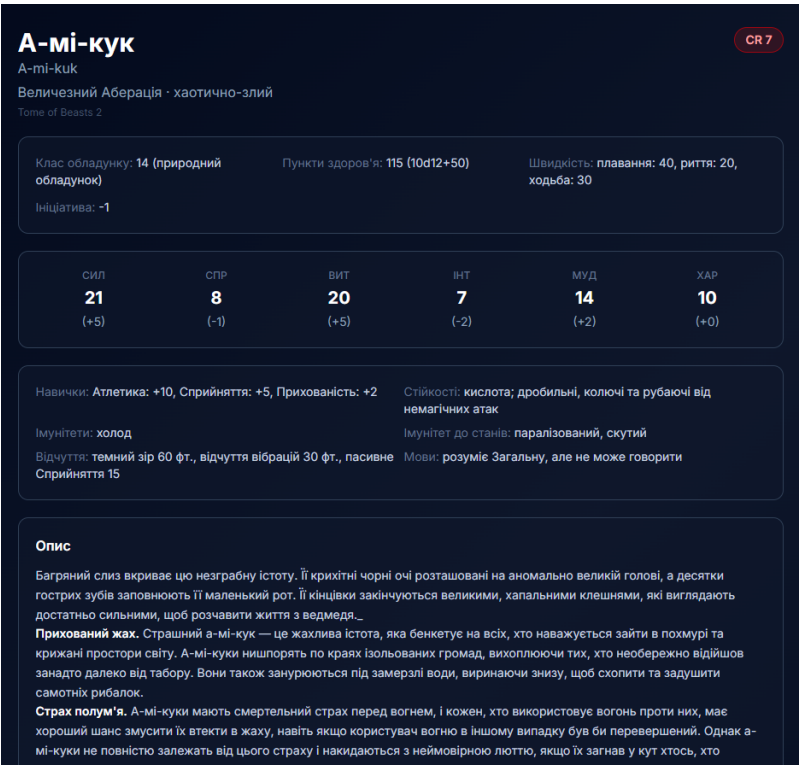


Рис. 3.5. Сторінка детального перегляду монстра

Окремий блок інтерфейсу становлять інструменти гравця. До них належать модуль створення персонажа, інтерактивний аркуш персонажа, механізм кидання дайсів і будівник сутичок. Ці інструменти призначені для практичної підтримки ігрового процесу та можуть використовуватися як під час підготовки до сесії, так і безпосередньо під час гри.

Сторінка створення персонажа містить форму, у якій користувач задає ім'я персонажа, расу, клас, рівень, передісторію, світогляд, основні характеристики, максимальні НР, клас обладунку та швидкість. Значення рас, класів і передісторій завантажуються з бази даних, що дозволяє користувачу обирати їх із наявних довідкових матеріалів. Після створення персонажа система зберігає його в базі даних і пов'язує з поточним користувачем.

На рисунку 3.6. представлено сторінку створення нового персонажа.

Новий персонаж

Основна інформація

Ім'я персонажа *

Раса

Обери расу

Клас

Обери клас

Рівень

1

Передісторія

Обери передісторію

Світогляд

Обери Світогляд

Бойові характеристики

Макс. ПЗ

8

Поточні ПЗ

8

Клас обладунку

10

Швидкість (фут.)

30

Бонус майстерності для 1 рівня: +2

Характеристики

Сила	Спритність	Витривалість	Інтелект	Мудрість	Харизма
10	10	10	10	10	10
+0	+0	+0	+0	+0	+0

Рятівні кидки

Рис. 3.6. Сторінка створення персонажа

Інтерактивний аркуш персонажа відображає основні параметри створеного персонажа. На цій сторінці користувач може переглядати характеристики, рівень, расу, клас, передісторію, поточні та максимальні НР, клас обладунку, швидкість, навички, інвентар, заклинання та нотатки. Однією з важливих функцій є можливість змінювати поточні НР персонажа в реальному часі. Це дозволяє використовувати аркуш персонажа під час ігрової сесії для відстеження отриманих пошкоджень або лікування.

На рисунку 3.7. зображено інтерактивний аркуш створеного персонажа, з можливістю зміни поточної кількості здоров'я, та можливістю редагування характеристик та умінь персонажа.

Тестовий персонаж

Альсайд · Варвар 12 рівня · Послушник
Законно-добрий

Редагувати

13

Клас обладунку

30 фут.

Швидкість

+4

Бонус майстерності

0

Досвід

Пункти здоров'я

24 / 52

0

+ Зцілення

- Шкода

Повне зцілення

Характеристики

СИЛ

12

+1

СПР

18

+4

ВИТ

15

+2

ІНТ

14

+2

МУД

8

-1

ХАР

12

+1

Рятівні кидки

СИЛ

+1

СПР

+8

ВИТ

+2

ІНТ

+6

МУД

+3

ХАР

+5

Навички

Акробатика (СПР)

+12

Догляд за тваринами (МУД)

-1

Таємні знання (ІНТ)

+2

Атлетика (СИЛ)

+1

Обман (ХАР)

+1

Історія (ІНТ)

+6

Проникливість (МУД)

+3

Залюкування (ХАР)

+9

Рис. 3.7. Інтерактивний аркуш персонажа

Сторінка кидання дайсів реалізує цифровий інструмент для виконання ігрових кидків. Користувач може обрати стандартний кубик або ввести дісе-вираз, наприклад 1d20+5 чи 2d6+3. Система обробляє вираз, генерує випадкові значення, додає модифікатор і виводить результат. Також передбачена підтримка переваги та недоліку, що є важливою механікою Dungeons & Dragons.

На рисунку 3.8. представлений інтерфейс кидання кубиків.

Кидки кубиків

Кидай кубики для атак, перевірок та рятівних кидків

Стандартні кубики

Модифікатор: 0

- 1 +

d4

- 1 +

d6

- 1 +

d8

- 1 +

d10

- 1 +

d12

- 1 +

d20

- 1 +

d100

Всі

Швидкі кидки

Атака

+модифікатор атаки

Перевага

вищий результат

Недолік

нижчий результат

Ініціатива

+модифікатор СПР

Смерть

рятівний кидок

Відновлення

кості ПЗ + CON

Довільний вираз

напр. 2d6+3, 4d8, 1d20+5

Назва (необов'язково)

Кинути

Підтримується: 2d6+3, 1d20adv (перевага), 1d20dis (недолік)

Рис. 3.8. Інтерфейс кидання дайсів

Будівник сутичок призначений для формування бойових зіткнень із монстрами та персонажами. Користувач може додавати до сутички монстрів із бази даних і створених персонажів, задавати або генерувати ініціативу, відстежувати порядок ходів, номер раунду, поточні НР та стани учасників. Такий інструмент є корисним для майстра гри, оскільки дозволяє зменшити кількість ручних обчислень під час бою.

На рисунку 3.9. представлено будівник сутичок, з можливістю обрання існуючих персонажів та монстрів, або створення нового персонажа.

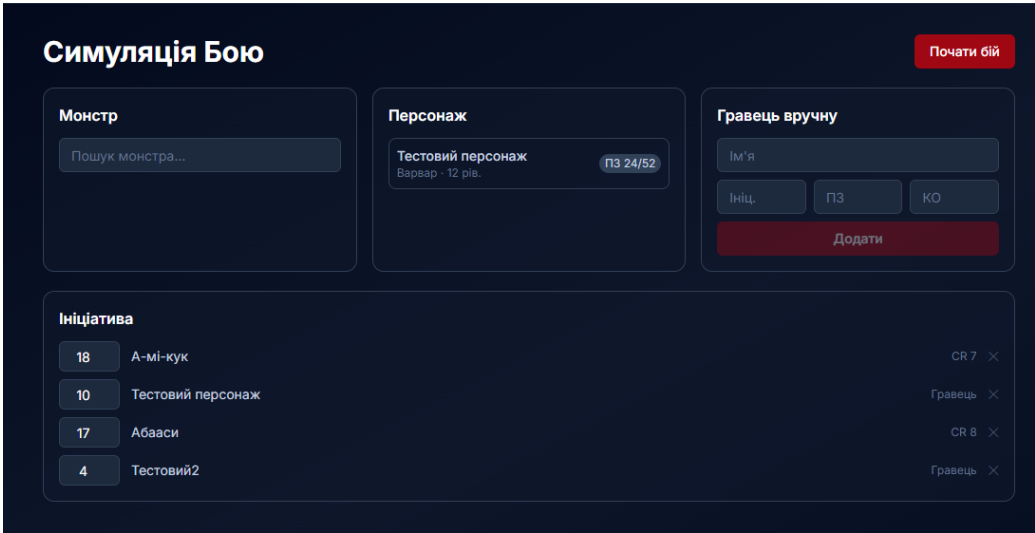


Рис. 3.9. Інтерфейс будівника сутичок

AI-асистент реалізований у вигляді чату. Користувач може ввести запитання, пов’язане з правилами гри, перекладом, термінологією або поясненням ігрових механік. Після надсилання повідомлення система виконує пошук релевантного контексту в базі даних і передає запит до Gemini API. Відповідь AI-асистента відображається у чаті. Для зручного форматування відповіді може використовуватися Markdown, що дозволяє відображати списки, таблиці та структуровані пояснення.

На рисунку 3.10. представлений інтерфейс чату AI-асистента.

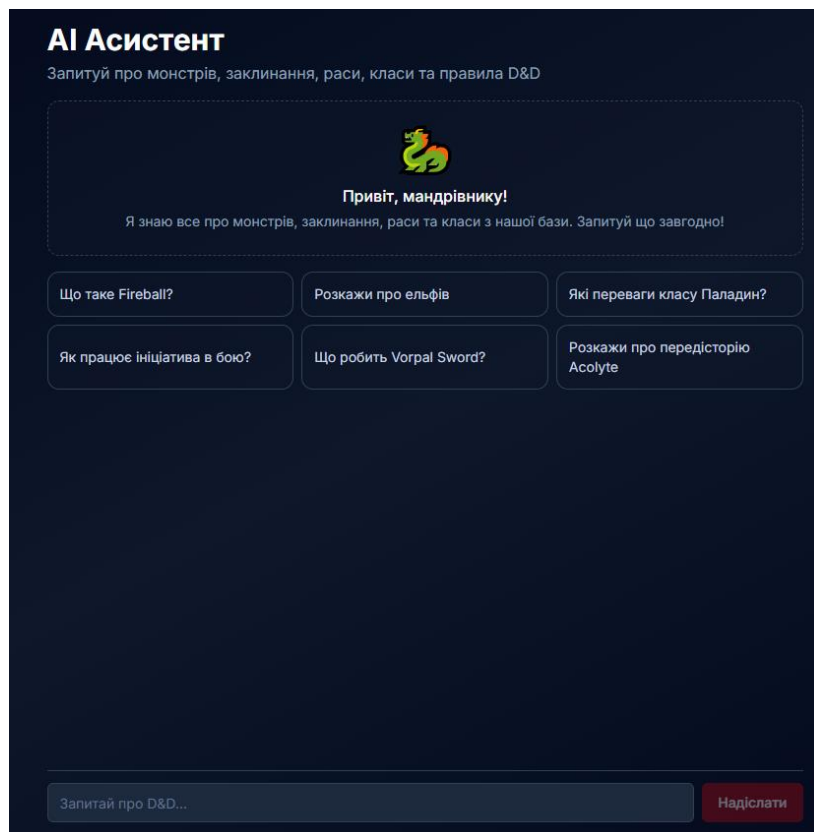


Рис. 3.10. Інтерфейс AI-асистента у вигляді чату

Адміністративна панель доступна лише користувачам із роллю moderator або admin. Вона призначена для керування наповненням системи. Через адміністративний інтерфейс модератор може запускати парсинг даних з Open5e REST API, виконувати AI-переклад контенту, переглядати записи бази даних, здійснювати пошук, редагувати або видаляти окремі документи. Також адміністративна панель відображає статистику наповнення бази даних у реальному часі.

На рисунку 3.11. зображена адміністративна панель системи, з можливістю оновлення входжень у базу даних, їх переклад, та управління. А також управління користувачами для присвоєння їм функціональних ролей, або управління самими акаунтами.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

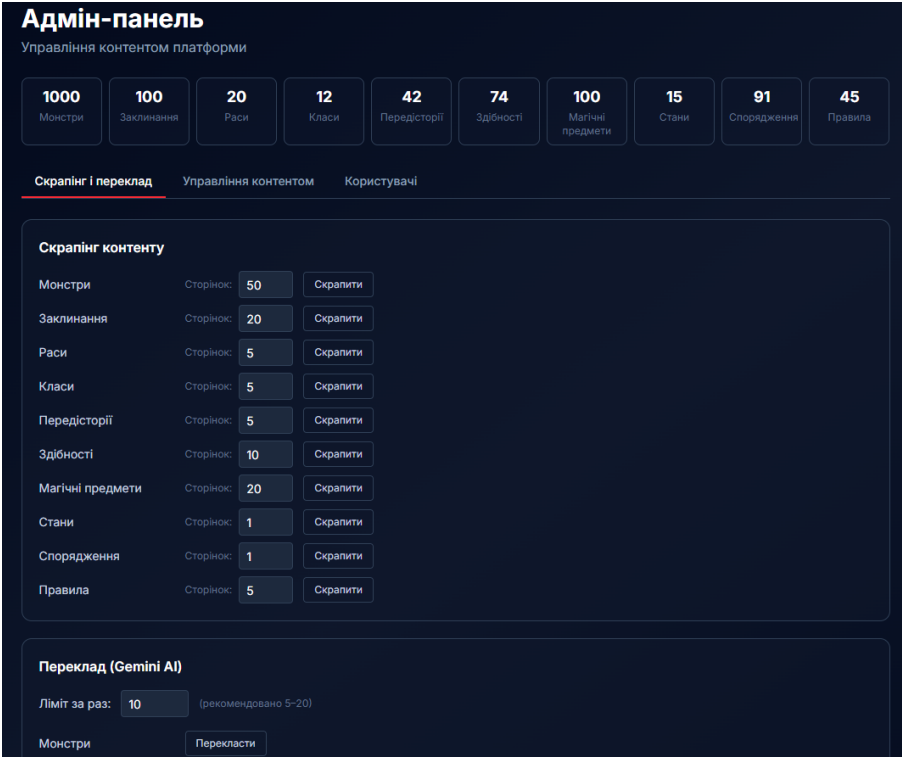


Рис. 3.11. Адміністративна панель системи

Загальну структуру навігації інтерфейсу наведено на рисунку 3.12.

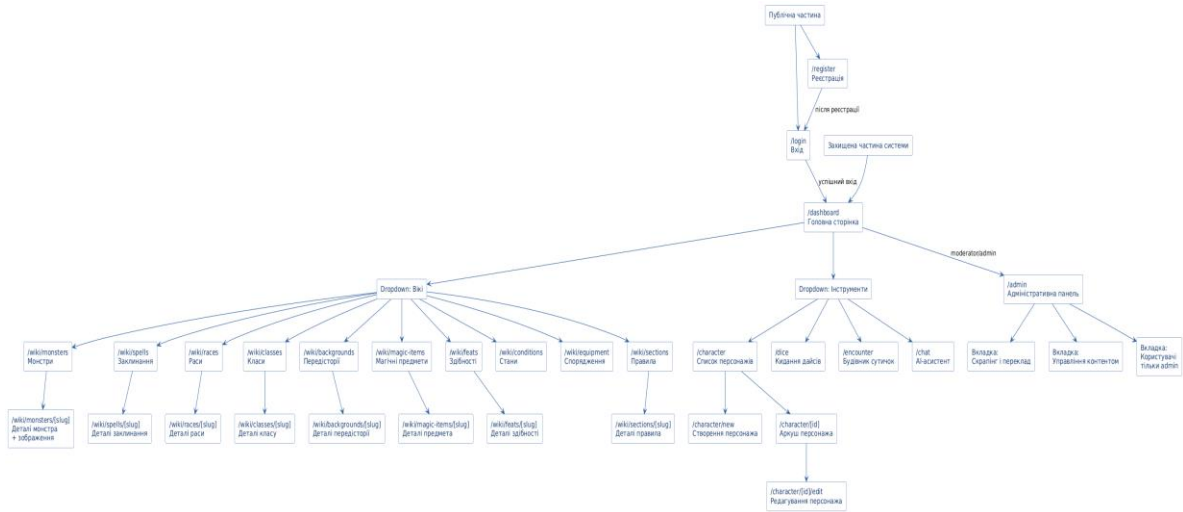


Рис. 3.12. Структура навігації інформаційної системи

Отже, структура інтерфейсу інформаційної системи побудована відповідно до основних сценаріїв її використання. Користувач отримує доступ до довідкових матеріалів, інструментів гравця та AI-асистента, а модератор – до засобів адміністрування даних. Такий підхід дозволяє забезпечити зручну навігацію, розділення функціональних зон і практичну підтримку ігрового процесу Dungeons & Dragons.

3.3 Тестування інформаційної системи

Після реалізації основних модулів інформаційної системи було проведено функціональне тестування з метою перевірки коректності роботи користувацьких, адміністративних та AI-функцій [3]. Основна увага під час тестування приділялася перевірці авторизації, розмежування доступу за ролями, роботи wiki-довідників, створення та редагування персонажів, функціонування інструментів гравця, адміністративної панелі, модуля скрапінгу, AI-перекладу та AI-асистента

Тестування виконувалося вручну шляхом послідовного проходження основних сценаріїв роботи користувача в інтерфейсі системи. Такий підхід є доцільним для перевірки вебзастосунку, оскільки дозволяє оцінити не лише правильність виконання серверної логіки, а й зручність взаємодії з інтерфейсом, коректність відображення даних, роботу форм, кнопок, випадаючих списків, модальних вікон і повідомлень про помилки.

Перевірка виконувалася для трьох основних рівнів доступу: player, moderator та admin. Для ролі player перевірялися основні користувацькі функції: перегляд wiki-довідників, пошук і фільтрація матеріалів, створення та редагування персонажа, робота з аркушем персонажа, кидання дайсів, створення сутички та використання AI-асистента. Для ролей moderator та admin перевірялися функції адміністративної панелі, пов'язані зі скрапінгом, перекладом і керуванням контентом. Для ролі admin додатково перевірялася можливість перегляду користувачів і зміни їхніх ролей.

Окремо перевірялася взаємодія системи із зовнішніми сервісами. Для Open5e REST API тестувалося отримання даних різних типів контенту та їх збереження в MongoDB Atlas. Для Gemini API перевірялися два сценарії використання: AI-переклад довідкових матеріалів та формування відповідей AI-асистента на основі контексту, знайденого в базі даних.

Результати функціонального тестування наведено в таблиці 3.1.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

Таблиця 3.1

Результати функціонального тестування інформаційної системи

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
1	Реєстрація користувача	Заповнити форму реєстрації коректними даними та створити обліковий запис	Користувач створюється в колекції users, пароль зберігається у хешованому вигляді	Пройдено
2	Вхід до системи	Авторизуватися за допомогою email і пароля	Користувач потрапляє до захищеної частини системи	Пройдено
3	JWT-сесія	Оновити сторінку після входу до системи	Сесія користувача зберігається, повторний вхід не потрібен	Пройдено
4	Розмежування доступу	Увійти під користувачем із роллю player і спробувати відкрити адміністративну панель	Доступ до адміністративної панелі обмежено	Пройдено
5	Доступ moderator	Увійти під користувачем із роллю moderator	Відображається адміністративна панель без вкладки керування користувачами	Пройдено
6	Доступ admin	Увійти під користувачем із роллю admin	Відображається адміністративна панель із вкладкою «Користувачі»	Пройдено
7	Керування користувачами	Змінити роль користувача через select у вкладці «Користувачі»	Поле role у колекції users оновлюється	Пройдено

продовж. табл.3.1

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
8	Wiki-довідники	Відкрити сторінки monsters, spells, races, classes, backgrounds, feats, magic-items, conditions, equipment та sections	Усі десять розділів відкриваються та відображають відповідні дані	Пройдено
9	Пошук у довідниках	Виконати пошук за ключовим словом у wiki-каталозі	Система відображає записи, що відповідають пошуковому запиту	Пройдено
10	Фільтрація монстрів	Відфільтрувати монстрів за рейтингом складності або розміром	У списку залишаються лише записи, що відповідають обраним параметрам	Пройдено
11	Фільтрація заклинань	Відфільтрувати заклинання за рівнем або школою магії	Відображаються лише заклинання з відповідними параметрами	Пройдено
12	Фільтрація спорядження	Обрати категорію weapon або armor	Відображається спорядження відповідної категорії	Пройдено
13	Детальна сторінка сутності	Перейти зі списку до сторінки детального перегляду	Відображається повна інформація про обраний запис	Пройдено
14	Зображення монстра	Відкрити каталог або сторінку детального перегляду монстра з полем img_main	Зображення монстра коректно відображається в інтерфейсі	Пройдено
15	Модальне зображення	Натиснути на зображення монстра	Зображення відкривається у модальному вікні	Пройдено

продовж. табл.3.1

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
16	Обробка недоступного зображення	Відкрити запис із некоректним або недоступним посиланням на зображення	Пошкоджене зображення не відображається завдяки обробнику onError	Пройдено
17	Створення персонажа	Заповнити форму створення персонажа та зберегти дані	У колекції characters створюється новий документ, пов'язаний із поточним користувачем	Пройдено
18	Редагування персонажа	Відкрити /character/[id]/edit, змінити дані персонажа та зберегти	Дані персонажа оновлюються в базі даних	Пройдено
19	Навички персонажа	Позначити навички з майстерністю та експертизою	У полі skills зберігаються значення proficient та expertise	Пройдено
20	Рятівні кидки	Обрати характеристики, за якими персонаж має володіння рятівними кидками	Дані зберігаються у полі saving_throws	Пройдено
21	Бонус майстерності	Змінити рівень персонажа	Бонус майстерності автоматично розраховується за формулою $\text{Math.ceil}(\text{level} / 4) + 1$	Пройдено
22	Аркуш персонажа	Відкрити сторінку /character/[id]	Відображаються характеристики, HP, навички, рятівні кидки, інвентар і нотатки персонажа	Пройдено

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

продовж. табл.3.1

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
23	Зміна НР персонажа	Змінити поточні НР персонажа	Значення НР оновлюється та не виходить за межі від 0 до максимального НР	Пройдено
24	Кидання дайсів	Виконати стандартний кидок d20	Система генерує випадкове значення в межах від 1 до 20	Пройдено
25	Dice-вираз	Ввести складний вираз, наприклад 2d6+3	Система обчислює результат кидків і додає модифікатор	Пройдено
26	Будівник сутичок	Додати до сутички монстрів із бази даних	Учасники додаються до списку сутички з відповідними НР та характеристиками	Пройдено
27	Додавання персонажів до сутички	Додати до сутички персонажа користувача з колекції characters	Персонаж додається до списку учасників бою	Пройдено
28	Налаштування ініціативи	Вручну задати ініціативу для учасників перед початком бою	Учасники сортуються відповідно до значень ініціативи	Пройдено
29	Ходи та раунди	Натискати кнопку переходу до наступного ходу	Активний учасник змінюється, після завершення кола збільшується номер раунду	Пройдено
30	Зміна НР у сутичці	Змінити НР учасника довільним значенням	Поточні НР оновлюються з урахуванням мінімальної та максимальної межі	Пройдено

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		90

продовж. табл.3.1

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
31	Стани учасників	Додати або зняти стан учасника сутички	Список станів учасника оновлюється	Пройдено
32	Скрапінг контенту	Запустити скрапінг одного з десяти типів контенту	Дані завантажуються з Open5e REST API і зберігаються в MongoDB Atlas	Пройдено
33	Вибір кількості сторінок	Вказати кількість сторінок для скрапінгу	Система обробляє не більше заданої кількості сторінок	Пройдено
34	Повторний скрапінг	Повторно запустити скрапінг для вже наявних записів	Дані оновлюються без втрати україномовних перекладів	Пройдено
35	Масовий AI-переклад	Запустити переклад для обраного типу контенту	Записи отримують україномовні поля, сформовані за допомогою Gemini API	Пройдено
36	Точковий AI-переклад	Натиснути кнопку «Перекласти» біля окремого запису	Перекладається лише вибраний запис	Пройдено
37	Керування контентом	Відредагувати запис через вкладку «Управління контентом»	Зміни зберігаються в MongoDB Atlas	Пройдено
38	Видалення контенту	Видалити тестовий запис через адміністративну панель	Запис видаляється з відповідної колекції	Пройдено
39	Статистика бази даних	Відкрити адміністративну панель	Відображається статистика наповнення десяти wiki-колекцій	Пройдено

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		91

продовж. табл.3.1

№	Об'єкт тестування	Сценарій перевірки	Очікуваний результат	Результат
40	AI-асистент	Надіслати запитання щодо правил або ігрової сутності	Система знаходить релевантний контекст у базі даних і повертає відповідь Gemini API	Пройдено
41	Довгі відповіді AI	Поставити запитання, що потребує розгорнутого пояснення	Відповідь не обрізається завдяки збільшенню maxOutputTokens до 2000	Пройдено
42	Адаптивність інтерфейсу	Відкрити сторінки системи на екранах різної ширини	Основні елементи інтерфейсу залишаються доступними та коректно відображаються	Пройдено

Проведене тестування показало, що основні функції інформаційної системи працюють відповідно до визначених вимог. Система коректно обробляє реєстрацію та вхід користувачів, підтримує JWT-сесії, розмежовує доступ відповідно до ролей player, moderator та admin. Wiki-довідники відображають дані з десяти колекцій, підтримують пошук, фільтрацію та перехід до сторінок детального перегляду.

Модуль персонажів успішно забезпечує створення, редагування та перегляд інтерактивного аркуша персонажа. Під час тестування підтверджено коректність збереження навичок, рятівних кидків, автоматичного розрахунку бонусу майстерності та зміни поточних очок здоров'я. Інструмент кидання дайсів правильно обробляє як стандартні кидки, так і складні дісе-вирази. Будівник сутичок коректно додає монстрів і персонажів користувача, дозволяє налаштовувати ініціативу, керувати ходами, раундами, НР та станами учасників.

Адміністративна панель забезпечує запуск скрапінгу, масового й точкового AI-перекладу, редагування, видалення записів і перегляд статистики бази даних. Окремо підтверджено, що повторний скрапінг не призводить до

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		92

втрати вже збережених україномовних перекладів. Для ролі admin успішно протестовано функцію керування користувачами та зміни їхніх ролей.

AI-функції системи також працюють коректно: AI-асистент отримує запит користувача, виконує пошук релевантного контексту в MongoDB Atlas і формує відповідь за допомогою Gemini API. Збільшення параметра maxOutputTokens до 2000 дозволило отримувати повніші відповіді без передчасного обрізання тексту.

Отже, результати тестування підтверджують працездатність основних модулів інформаційної системи та відповідність реалізованого функціоналу поставленим вимогам. Виявлені під час розробки незначні недоліки інтерфейсу та логіки обробки даних були усунені в процесі перевірки, після чого система продемонструвала стабільну роботу основних сценаріїв використання.

Висновки до третього розділу

У третьому розділі було розглянуто порядок встановлення, налаштування, використання та тестування інформаційної системи для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента. Було визначено основні етапи підготовки середовища для запуску застосунку, зокрема встановлення залежностей, налаштування змінних середовища, підключення до MongoDB Atlas, додавання ключа Gemini API та запуск Next.js-застосунку.

У розділі розглянуто порядок взаємодії користувача із системою. Описано загальну структуру інтерфейсу, використання темної візуальної теми, навігацію через згруповані пункти «Вікі» та «Інструменти», а також роботу з основними сторінками застосунку. Зокрема, було показано використання wiki-довідників, сторінок детального перегляду, перегляд зображень монстрів, роботу з персонажами, кидання дайсів, створення сутичок та взаємодію з AI-асистентом. Також описано адміністративну панель, через яку виконуються скрапінг, переклад, редагування контенту, перегляд статистики та керування користувачами для ролі admin.

Під час тестування було перевірено ключові сценарії роботи інформаційної системи. Перевірка охопила реєстрацію й авторизацію користувачів, роботу JWT-сесій, розмежування доступу між ролями player, moderator та admin,

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		93

перегляд wiki-довідників, пошук і фільтрацію записів, відображення зображень монстрів, створення та редагування персонажів. Також було протестовано налаштування навичок і рятівних кидків, автоматичний розрахунок бонусу майстерності, роботу інтерактивного аркуша персонажа, інструмент кидання дайсів і будівник сутичок.

Також було перевірено адміністративні можливості системи: скрапінг десяти типів контенту з Open5e REST API, вибір кількості сторінок для обробки, повторний скрапінг без втрати україномовних перекладів, масовий і точковий AI-переклад, редагування та видалення записів бази даних, перегляд статистики колекцій і зміну ролей користувачів. Окремо підтверджено працездатність AI-асистента, який формує відповіді з використанням контексту, знайденого в MongoDB Atlas, і Gemini API.

Результати тестування показали, що реалізована інформаційна система працює відповідно до поставлених вимог. Основні функціональні модулі взаємодіють коректно, дані зберігаються в базі MongoDB Atlas, рольова модель забезпечує розмежування доступу, а AI-функції доповнюють систему можливостями перекладу та контекстної допомоги користувачам. Отже, розроблений програмний продукт є придатним до практичного використання для організації довідкового, ігрового та адміністративного функціоналу, пов'язаного з настільно-рольовою грою Dungeons & Dragons.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		94

ВИСНОВКИ

У кваліфікаційній роботі було розроблено інформаційну систему для поціновувачів настільно-рольової гри Dungeons & Dragons з інтеграцією AI-асистента. Розроблений програмний продукт поєднує довідкову систему, інструменти для гравців, адміністративну панель, модуль автоматизованого наповнення бази даних, AI-переклад контенту та чат-асистента для допомоги користувачам.

У першому розділі було проаналізовано предметну область і визначено основні проблеми, пов'язані з використанням матеріалів Dungeons & Dragons. До таких проблем належать великий обсяг довідкової інформації, складність швидкого пошуку потрібних правил або характеристик, наявність мовного бар'єра для україномовних користувачів і потреба в додаткових інструментах для ведення персонажів та організації ігрових ситуацій. Також було розглянуто існуючі програмні рішення, що використовуються гравцями та майстрами гри, визначено їхні переваги й обмеження. На основі цього було обґрунтовано доцільність створення власної інформаційної системи, орієнтованої на довідковий контент, україномовну локалізацію та інтеграцію AI-функцій.

Було обрано архітектуру вебзастосунку, реалізованого на основі Next.js 14 з використанням App Router. Такий підхід дозволив об'єднати клієнтську частину, серверну логіку та API-маршрути в межах одного Full-Stack-застосунку. Для зберігання даних використано MongoDB Atlas, для роботи з базою даних – Mongoose ODM, для автентифікації – NextAuth.js, а для AI-функцій – Gemini API. Інтерфейс реалізовано з використанням React, TypeScript, Tailwind CSS і shadcn/ui.

У другому розділі було виконано проєктування інформаційної системи. Було визначено ролі користувачів: player, moderator та admin. Для кожної ролі сформовано набір доступних функцій. Користувач із роллю player може працювати з wiki-довідниками, створювати та редагувати персонажів, користуватися інтерактивним аркушем персонажа, кидати дайси, створювати сутички та звертатися до AI-асистента. Користувач із роллю moderator отримує

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		95

доступ до адміністративних функцій керування контентом, а користувач із роллю admin додатково може керувати ролями інших користувачів.

Було спроектовано структуру бази даних MongoDB Atlas. Вона містить колекцію users для збереження облікових записів, колекцію characters для збереження персонажів користувачів і десять wiki-колекцій для довідкового контенту Dungeons & Dragons: monsters, spells, races, classes, backgrounds, feats, magicitems, conditions, equipment та sections. Така структура дозволяє зберігати складні вкладені дані, характеристики, описи, дії, навички, рятівні кидки, інвентар, заклинання та інші елементи, необхідні для роботи системи.

Окрему увагу було приділено алгоритмам роботи системи. Реалізовано алгоритм парсингу даних з Open5e REST API, який дозволяє автоматизовано наповнювати базу даних довідковими матеріалами. Під час повторного парсингу передбачено збереження вже наявних україномовних перекладів, що дозволяє оновлювати вихідні дані без втрати результатів локалізації. Також реалізовано алгоритм AI-перекладу контенту за допомогою Gemini API, який підтримує як масовий переклад, так і точковий переклад окремих записів.

Було реалізовано AI-асистента, який працює за принципом пошуку релевантного контексту в базі даних перед формуванням відповіді користувачу. Завдяки цьому асистент може надавати відповіді з урахуванням матеріалів, збережених у wiki-довідниках системи. Для отримання повніших відповідей було збільшено ліміт формування тексту, що зменшило ризик обрізання відповідей.

У третьому розділі було описано порядок встановлення, налаштування та використання системи. Було визначено необхідні змінні середовища. Також описано порядок запуску застосунку в режимі розробки та production-режимі.

Було розглянуто інтерфейс системи та порядок роботи користувача. Інтерфейс реалізовано в темній темі та структуровано за функціональними групами. Wiki-довідники забезпечують перегляд десяти типів контенту, пошук, фільтрацію та перехід до сторінок детального перегляду. Для монстрів реалізовано відображення зображень і модальний перегляд. Модуль персонажів

підтримує створення, редагування, перегляд аркуша, налаштування навичок, експертизи, рятівних кидків і автоматичний розрахунок бонусу майстерності. Будівник сутичок дозволяє додавати монстрів і персонажів, виставляти ініціативу, керувати ходами, НР та станами учасників.

Адміністративна панель забезпечує скрапінг десяти типів контенту, масовий і точковий AI-переклад, редагування й видалення записів, перегляд статистики бази даних, а також керування користувачами для ролі admin. Такий розподіл функцій дозволяє зручно адмініструвати систему та контролювати доступ до критичних можливостей.

Під час тестування було перевірено основні сценарії роботи системи: реєстрацію, авторизацію, JWT-сесії, розмежування доступу за ролями, роботу wiki-довідників, пошук і фільтрацію, створення та редагування персонажів, роботу аркуша персонажа, кидання дайсів, будівник сутичок, скрапінг, AI-переклад, керування контентом, керування користувачами та роботу AI-асистента. Результати тестування підтвердили, що основні модулі системи працюють коректно та відповідають поставленим вимогам.

Отже, у результаті виконання кваліфікаційної роботи було досягнуто поставленої мети – розроблено інформаційну систему для поціновувачів Dungeons & Dragons з інтеграцією AI-асистента. Розроблений застосунок забезпечує зручний доступ до довідкових матеріалів, підтримує україномовну локалізацію, надає інструменти для роботи з персонажами та сутичками, дозволяє автоматизовано наповнювати базу даних і використовує AI-функції для перекладу та допомоги користувачам. Система є придатною для практичного використання гравцями, майстрами гри та адміністраторами контенту.

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		97

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ломоносов І. О., Савіцький Р.С., Толстой І.А. Використання штучного інтелекту для локалізації та генерації контенту в системах підтримки НРІ. Тези Всеукраїнської науково-практичної конференції «Тенденції розвитку технологій в автоматизації, електроенергетиці, приладобудуванні та робототехніці», Житомир: "Житомирська політехніка", 13-15 травня 2026 р. [подано до друку].
2. Ломоносов І. О. , Савіцький Р.С., Толстой І.А. Проектування інформаційної системи для гравців з модулем автоматизованого збору. Тези Всеукраїнської науково-практичної конференції «Тенденції розвитку технологій в автоматизації, електроенергетиці, приладобудуванні та робототехніці», Житомир: "Житомирська політехніка", 13-15 травня 2026 р. [подано до друку].
3. Методичні рекомендації до написання кваліфікаційної роботи для здобувачів вищої освіти освітнього ступеня «бакалавр» спеціальності 121 «Інженерія програмного забезпечення» освітньо-професійної програми «Інженерія програмного забезпечення» / А. Морозов, Т. Вакалюк, Ю. Єфремов, Р. Савіцький. Житомир: Державний університет «Житомирська політехніка», 2024–2025. 24 с.
4. Auth.js. Migrating to v5 / NextAuth.js Documentation. URL: <https://authjs.dev/getting-started/migrating-to-v5> Дата звернення: 22.03.2026.
5. Axios Documentation. URL: <https://axios-http.com/docs/intro> (дата звернення: 10.04.2026).
6. D&D Beyond. Official Digital Toolset for Dungeons & Dragons. URL: <https://www.dndbeyond.com/> (дата звернення: 05.02.2026).
7. D&D Beyond. The Official D&D Character Builder. URL: <https://www.dndbeyond.com/characters/builder> (дата звернення: 07.02.2026).
8. Foundry Virtual Tabletop. A Self-Hosted & Modern Roleplaying Platform. URL: <https://foundryvtt.com/> (дата звернення: 10.02.2026).

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Gemini API Documentation. Google AI for Developers. URL: <https://ai.google.dev/gemini-api/docs> (дата звернення: 18.04.2026).
10. Gemini API Reference. Google AI for Developers. URL: <https://ai.google.dev/api> (дата звернення: 19.04.2026).
11. GitHub Flavored Markdown Spec. URL: <https://github.github.com/gfm/> (дата звернення: 03.05.2026)
12. Google AI Studio quickstart. Gemini API. URL: <https://ai.google.dev/gemini-api/docs/ai-studio-quickstart> (дата звернення: 20.04.2026).
13. JavaScript Guide. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide> (дата звернення: 06.04.2026).
14. MongoDB Atlas Documentation. MongoDB, Inc. URL: <https://www.mongodb.com/docs/atlas/> (дата звернення: 25.03.2026).
15. MongoDB Node.js Driver Documentation. MongoDB, Inc. URL: <https://www.mongodb.com/docs/drivers/node/current/> (дата звернення: 26.03.2026).
16. Mongoose ODM Documentation. URL: <https://mongoosejs.com/docs/> (дата звернення: 28.03.2026).
17. Next.js Documentation. App Router. Vercel. URL: <https://nextjs.org/docs/app> (дата звернення: 12.03.2026).
18. Next.js Documentation. Getting Started. Vercel. URL: <https://nextjs.org/docs/app/getting-started> (дата звернення: 13.03.2026).
19. NextAuth.js Documentation. URL: <https://next-auth.js.org/> (дата звернення: 23.03.2026).
20. Node.js Documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 08.03.2026).
21. Open5e API Documentation. URL: <https://open5e.com/api-docs> (дата звернення: 18.02.2026).

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

22. React Documentation. Quick Start. URL: <https://react.dev/learn> (дата звернення: 14.03.2026).
23. react-markdown Documentation. URL: <https://github.com/remarkjs/react-markdown> (дата звернення: 01.04.2026).
24. remark-gfm Documentation. URL: <https://github.com/remarkjs/remark-gfm> (дата звернення: 02.04.2026).
25. Roll20. Online Virtual Tabletop for Pen and Paper RPGs. URL: <https://roll20.net/> (дата звернення: 08.02.2026).
26. shadcn/ui Documentation. URL: <https://ui.shadcn.com/docs> (дата звернення: 17.03.2026).
27. System Reference Document. Dungeons & Dragons. Wizards of the Coast / D&D Beyond. URL: <https://www.dndbeyond.com/srd> (дата звернення: 04.02.2026).
28. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 16.03.2026).
29. TypeScript Handbook. Microsoft. URL: <https://www.typescriptlang.org/docs/handbook/intro.html> (дата звернення: 15.03.2026).
30. World Anvil. Worldbuilding Tools & RPG Campaign Manager. URL: <https://www.worldanvil.com/> (дата звернення: 11.02.2026).

ДОДАТКИ

		Ломоносов І.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Савіцький Р.С.				101
Змн.	Арк.	№ докум.	Підпис	Дата		

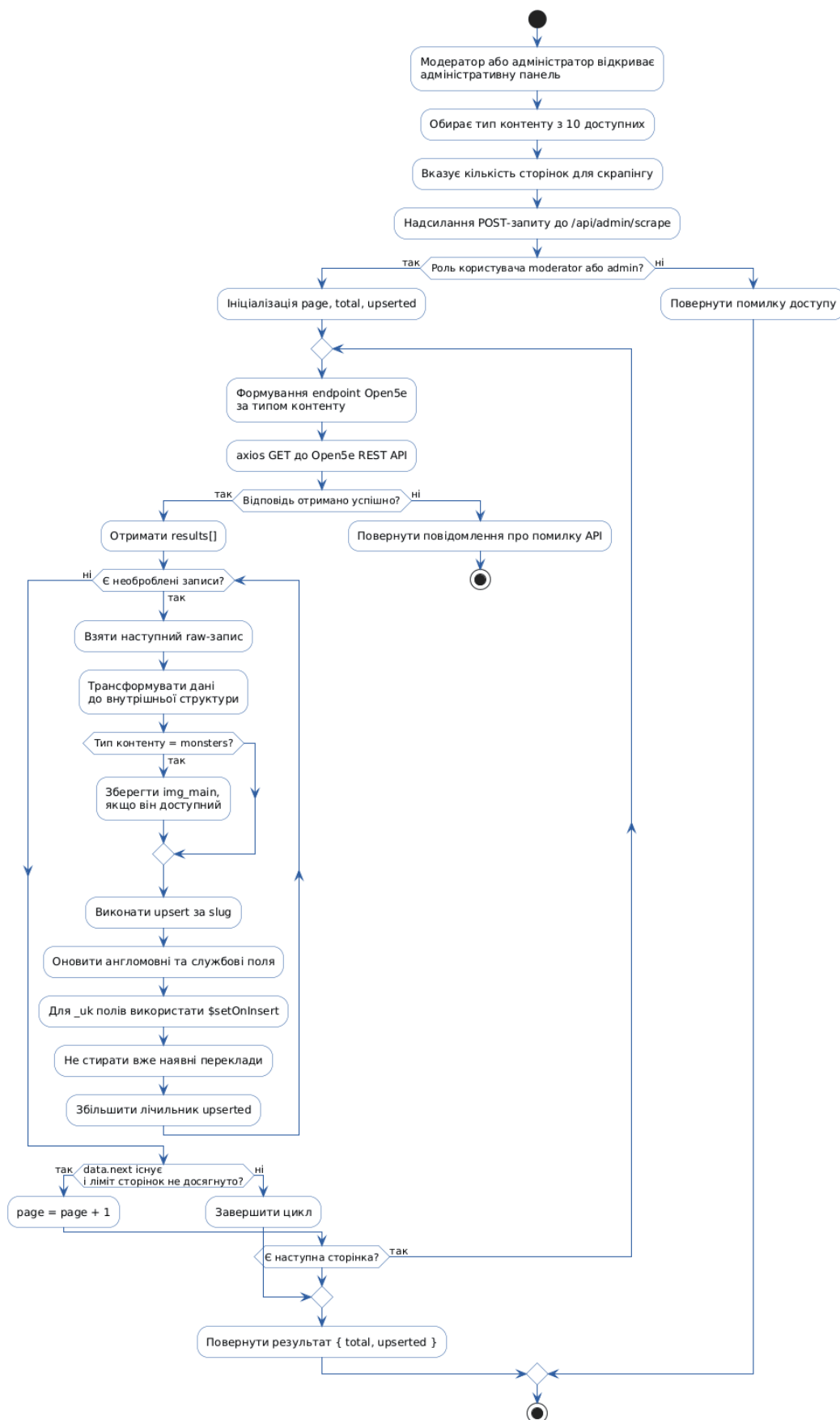


Рис. 2.4. Алгоритм парсингу даних з Open5e REST API