

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект інформаційної системи Центру
підтримки академій Cisco Державного університету
"Житомирська політехніка"»

Виконав студент 4-го курсу, групи КІ-3
спеціальності 123 «Комп'ютерна інженерія»

 В.Ю. Войналович

Керівник к.т.н., доцент кафедри КІ та КБ

 О.С. Головня

Рецензент к.т.н., завідувач кафедри КІ та КБ

 А.А. Єфіменко

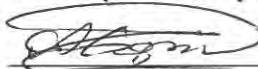
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ

СПЕЦІАЛЬНІСТЬ 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної
інженерії та кібербезпеки

 Єфіменко А.А.

«12» 04 2021 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу

Студента Войналовича Владислава Юрійовича

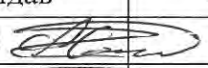
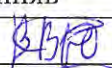
Тема роботи: Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"

Затверджена Наказом університету від «12» квітня 2021 р. № 133\С

Термін здачі студентом закінченої роботи 25.04.2021р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження) процес проектування, розробки та тестування інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Єфіменко А.А.	12.04.21 	15.06.21 
2	Єфіменко А.А.	12.04.21 	15.06.21 
3	Єфіменко А.А.	12.04.21 	15.06.21 

Керівник  Головня О.С.

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	12.04.21	
2	Розробка плану роботи	15.04.21 – 17.04.21	
3	Розробка розгорнутого плану	18.04.21 – 19.04.21	
4	Подача чернетки керівнику для перевірки	25.04.21	
5	Доопрацювання роботи та перевірка керівником	26.04.21 – 30.05.21	
6	Затвердження графічних матеріалів у керівника	01.06.21 – 04.06.21	
7	Проведення попереднього захисту	05.06.21	
8	Отримання відгуку керівника	16.06.21	
9	Допуск завідувача кафедри	16.06.21	
10	Подання роботи на рецензування	16.06.21	
11	Захист		

Студент  Войналович В.Ю.

Керівник  Головна О.С.

РЕФЕРАТ

Пояснювальна записка до випускної кваліфікаційної роботи на тему «Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 52 сторінках друкованого тексту.

Пояснювальна записка має 4 сторінки додатків. Список використаних джерел містить 15 найменувань і займає 2 сторінки. В роботі наведено 39 рисунків. Загальний обсяг роботи – 64 сторінки.

У першому розділі було обґрунтовано створення сайту на тему «Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"».

У другому розділі проведено проектування і розробку сайту.

У третьому розділі проведено тестування програмного продукту та описана робота з додатком.

У додатку представлений лістинг окремих компонентів розробленого програмного продукту.

Ключові слова: VUE.JS, VUEX, NOSQL, JAVASCRIPT, WEB-САЙТ, MVVM, MODEL, VIEW, VIEWMODEL, ЗАПИТИ, ЗАПИСИ, КЛІЄНТ, АДМІНІСТАТОР, КОРИСТУВАЧ.

					КІ.КР.Б – 123 – 02 – ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата	Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка" Пояснювальна записка Лім. Арк. Аркуші 4 64 Житомирська політехніка, група КІ-3		
Розроб.		Войналович В.Ю.		29.06.21			
Керівник		Головня О.С.		29.06.21			
Реценз.		Єфіменко А.А.		29.06.21			
Зав. каф.		Єфіменко А.А.		29.06.21			

ABSTRACT

The explanatory note to the final qualification work on the topic "Project of the information system of the Cisco Academy Support Center of Zhytomyr Polytechnic State University" consists of a list of abbreviations, introduction, three sections, conclusions, list of references and appendix.

The text is set out on 52 pages of printed text.

The explanatory note has 4 pages of appendices. The list of used sources contains 15 names and occupies 2 pages. The paper contains 39 illustrations. The total volume of the work is 64 pages.

The first section justified the creation of a site on the topic "Web-based application for project management."

The second section is the design and development of the site.

The third section tests the software product and describes the work with the application.

The appendix lists the individual components of the developed software product.

Keywords: VUE.JS, VUEX, NOSQL, JAVASCRIPT, WEB SITE, MVVM, MODEL, VIEW, VIEWMODEL, REQUESTS, RECORDS, CLIENT, ADMINISTRATOR, USER.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ	9
1.1 Аналіз задачі, засобів та методів її вирішення	9
1.2 Аналіз вимог до програмного продукту.....	9
1.3 Аналіз існуючого програмного забезпечення за тематикою кваліфікаційної роботи	19
1.4 Технічне завдання на кваліфікаційну роботу.....	22
Висновки до першого розділу	27
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.1 Алгоритм роботи та стани програмної системи	28
2.2 Об'єктно-орієнтована модель системи	29
2.3 Комунікації і послідовність взаємодії об'єктів системи.....	33
2.4 Розробка функціональних алгоритмів роботи програми.....	36
Висновки до другого розділу	45
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	46
3.1 Архітектура програмного комплексу та його розгортання.....	46
3.2 Опис роботи з додатком.....	46
3.3 Тестування веб-додатку.....	48
Висновки до третього розділу.....	58
ВИСНОВКИ	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТКИ.....	62

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

MVVM – Модель–представлення–контролер (Model-View-ViewModel, MVVM) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

ПЗ – Програмне забезпечення.

AJAX — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Автоматизовані системи, призначені для вирішення завдань оптимізації вирішення задач, які постають перед людством, представляють собою послідовність кроків, на кожному з яких людина співпрацює з системою. Такі системи мають велике як теоретичне, так і практичне значення: допомагаючи нам знайти рішення для будь-якої конкретної задачі й отримати бажаний результат. Системи допомагають командам зручно працювати з проектами, наприклад керування процесом над розвитком проекту впродовж всього його циклу розробки, витрачаючи при цьому мінімум своїх ресурсів. У таких завданнях може бути безліч різних рішень; їхню якість визначають значенням параметра, і потрібно вибрати оптимальне рішення, за якого значення параметра буде мінімальним або максимальним (залежно від постановки задачі).

На даний час існує безліч автоматизованих систем управління, які відрізняється за своїм методом використання, сферою використання та ефективністю. Для кожної поставленої задачі існує свій оптимальний алгоритм, який призведе до вирішення цієї задачі найефективніше.

Об'єкт дослідження є процес обробки інформації наданої користувачами для здійснення статистичного зрізу чи надання інформації відповідно до запиту користувача.

Предметом дослідження є розробка веб-сайту, який зберігатиме дані для подальшої обробки, надаватиме користувачеві інформацію за запитом.

Метою роботи є розроблення програмного продукту, за допомогою якого можна отримати інформацію щодо академій Cisco.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1 Аналіз задачі, засобів та методів її вирішення

Широке застосування сучасних засобів обчислювальної техніки, впровадження в економічну діяльність методів оптимізації та формалізації ситуацій значно змінили технологію інформаційного забезпечення управління.

У сучасних умовах інформаційне забезпечення управління здійснюється за допомогою автоматизованих систем управління (АСУ).

Автоматизована система управління - інформаційна система, призначена для автоматизованого здійснення управлінських процесів.

Впровадження АСУ має бути виправданим, тобто повинне призводити до корисних техніко-економічних, соціальних або інших результатів. Зокрема, використання автоматизованої системи управління дозволяє домогтися зниження чисельності управлінського персоналу, підвищення якості функціонування об'єкта управління і самого управління і т.д.

До АСУ висувається ряд загальних вимог.

У першу чергу, повинна бути забезпечена сумісність елементів один з одним, а також з автоматизованими системами, взаємопов'язаними з даною АСУ.

Автоматизована система повинна бути пристосована до модернізації, розвитку і розширенню з урахуванням майбутніх перспектив.

АСУ повинна мати достатній ступінь надійності для досягнення встановлених цілей функціонування системи при заздалегідь заданих умовах її застосування.

Автоматизована система управління повинна володіти достатньою адаптивністю до змін умов її використання. При цьому ступінь зміни умов застосування системи, як правило, спеціально обговорюється заздалегідь.

В АСУ повинні бути передбачені контроль правильності виконання автоматизуються функцій і діагностування із зазначенням місця, виду та причини виникнення порушень правильності функціонування системи.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				9
Змн.	Арк.	№ докум.	Підпис	Дата		

В автоматизованій системі управління повинні бути передбачені заходи захисту від неправильних дій персоналу, які призводять до аварійного стану об'єкта або системи управління, від випадкових змін і руйнування інформації і програм, а також від несанкціонованого втручання і витоку інформації.

АСУ, як і будь-яку сучасну інформаційну систему, яка має складну багатопланову структуру, можна розділити на дві складові - функціональну і забезпечувальну.

Функціональна частина вирішує ті завдання, заради виконання яких створюється кожна окремо взята система. Ці завдання перетворюються у відповідні функції АСУ.

Будь-яка АСУ в процесі своєї роботи повинна виконувати наступні функції:

- збір, обробку та аналіз інформації (сигналів, повідомлень, документів тощо) про стан об'єкта управління;
- вироблення керуючих дій (програм, планів і т.д.);
- передачу керуючих впливів (сигналів, вказівок, документів) на виконання і контроль їх передачі;
- реалізацію і контроль виконання керівних впливів;
- обмін інформацією (документами, повідомленнями тощо) з іншими пов'язаними з нею автоматизованими системами.

Склад автоматизованих функцій АСУ і ступінь їх автоматизації визначаються відповідно до техніко-економічних показників, а також з урахуванням необхідності звільнення персоналу від виконання повторюваних дій і створення умов для використання його творчих здібностей в процесі роботи.

До забезпечувальної частини АСУ входить наступне забезпечення:

- програмно-математичне;
- інформаційне;
- технічне;
- методико-організаційне;
- лінгвістичне;

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				10
Змн.	Арк.	№ докум.	Підпис	Дата		

- кадрове.

Програмно-математичне забезпечення є однією з найбільш важливих складових сучасної інформаційної системи. У програмне забезпечення входять всі програмні засоби, що використовуються як безпосередньо для виконання поставлених перед системою завдань, так і для здійснення нормального функціонування всього комплексу використовуваних технічних засобів. Математичне забезпечення являє собою сукупність математичних алгоритмів, методів і моделей, які використовуються в роботі інформаційної системи.

Програмне забезпечення АСУ повинно бути достатнім для виконання всіх її функцій, реалізованих із застосуванням засобів обчислювальної техніки. Крім того, повинні бути наявні засоби організації всіх необхідних процесів обробки даних, що дозволяють своєчасно виконувати всі автоматизовані функції у всіх режимах функціонування АСУ.

Програмне забезпечення АСУ має володіти такими властивостями:

- функціональною достатністю (повнотою);
- надійністю (у тому числі відновлюваністю і наявністю засобів виявлення помилок);
- адаптивністю до змінних умов;
- можливістю модифікації системи - при необхідності;
- модульністю побудови;
- зручністю експлуатації.

Як правило, програмне забезпечення АСУ будується на базі вже наявних пакетів прикладних програм. Таке програмне забезпечення допускає завантаження і перевірку по частинах і дозволяє проводити заміну одних програм без корекції інших.

До програмного забезпечення АСУ пред'являється ряд вимог, які дозволяють домогтися надійності роботи системи в цілому. Зокрема, програмне забезпечення підбирається і налаштовується таким чином, щоб відсутність окремих даних не позначалася на виконанні функцій АСУ, при реалізації яких ці дані не використовуються. В обов'язковому порядку здійснюються заходи щодо

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		11

захисту від помилок при введенні і обробці інформації, що забезпечують задану якість виконання функцій автоматизованої системи.

Використовуване програмне забезпечення повинне мати засоби діагностики технічних засобів АСУ та контролю достовірності вхідної інформації.

Загальне програмне забезпечення АСУ повинно дозволяти здійснювати настройку окремих компонентів спеціального програмного забезпечення і подальший розвиток програмного забезпечення системи без переривання процесу її функціонування. Всі програми спеціального програмного забезпечення конкретної АСУ повинні бути сумісні як між собою, так і з її загальним програмним забезпеченням. Крім того, необхідно забезпечити захист вже згенерованої і завантаженої частини програмного забезпечення від випадкових змін.

Інформаційне забезпечення включає всю сукупність інформації, на основі якої буде функціонувати АСУ, у тому числі дані за змістом, системою кодування, методами адресування, форматами даних і формою представлення інформації, яку одержує і видає АСУ.

Сукупність інформаційних масивів АСУ представляється у вигляді баз даних на машинних носіях. Інформація, яка міститься у базах даних повинна постійно оновлюватися відповідно до періодичності її використання при функціонуванні системи. Передбачаються необхідні заходи по відновленню інформаційних масивів при відмовах будь-яких технічних засобів АСУ, а також заходи з контролю ідентичності однойменної інформації в базах даних.

Технічне забезпечення складає комплекс всіх технічних засобів, що використовуються при роботі інформаційної системи. Сучасні технічні засоби відрізняються великою різноманітністю і дозволяють вирішувати широкий спектр завдань.

Можна виділити наступні групи технічних засобів, що забезпечують функціонування сучасних інформаційних систем:

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- засоби обчислювальної техніки (ЕОМ різної продуктивності і призначення);
- засоби комунікації;
- засоби організаційної техніки.

Засоби обчислювальної техніки використовуються на всіх стадіях обробки та зберігання інформації і є основою для інтеграції всіх технічних засобів в єдину автоматизовану систему.

Засоби комунікації призначені, в першу чергу, для передачі інформації і в ряді випадків функціонують спільно із засобами обчислювальної техніки.

Оргтехніка дозволяє здійснювати з інформацією різні дії (наприклад, подання в різних формах, копіювання і т.п.), а також допоміжні операції в рамках різних завдань інформаційного забезпечення управлінської діяльності.

Технічні засоби АСУ при взаємодії з іншими системами повинні бути сумісні по інтерфейсах з відповідними технічними засобами цих систем і використовуваних систем зв'язку.

Будь-яке технічний засіб АСУ повинно допускати заміну його аналогічним технічним засобом без регулювання або будь-яких конструктивних змін в інших технічних засобах АСУ.

Методико-організаційне забезпечення являє собою сукупність методів, засобів і спеціальних документів, що встановлюють порядок спільної роботи технічних засобів АСУ та обслуговуючого персоналу, а також взаємодію персоналу між собою в процесі роботи з системою. До цього виду забезпечення також відносять різні методи і засоби організації та проведення навчання персоналу прийомів роботи з даною інформаційною системою (наприклад, методики навчання, програми курсів і практичних занять, технічні засоби навчання тощо).

Основною метою методико-організаційного забезпечення є підтримання працездатності інформаційної системи і можливості подальшого її розвитку. Інструкції, що входять до методико-організаційне забезпечення АСУ, повинні чітко визначати дії персоналу при виконанні всіх функцій системи у всіх

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		13

режимах функціонування. Крім того, інструкції повинні містити конкретні вказівки про дії у разі виникнення аварійних ситуацій або порушення нормальних умов функціонування АСУ.

Процес автоматизації управління може здійснюватися різними шляхами. В одному випадку організація може встановити і використовувати комп'ютерні засоби обробки інформації лише для спрощення деяких рутинних операцій процесу роботи з документами. При цьому загальні принципи і методи роботи з інформацією залишаються незмінними. Такий шлях не є ефективним, оскільки не повною мірою використовує можливості сучасних інформаційних технологій.

Принципово інший підхід полягає у створенні комплексних систем автоматизації управлінської діяльності. Такі системи включають не тільки засоби обробки документів, а й системи управління базами даних, експертні системи, сучасні засоби телекомунікацій та багато іншого. Створення таких систем дозволяє значно підвищити ефективність управління організацією.

У сучасних умовах функціонування організацій значно підвищуються вимоги до оперативності доставки інформації споживачу і до швидкості обробки інформації. Тому створюються багаторівневі, розподілені АСУ. Прикладами таких систем є банківські, податкові, статистичні, постачальницькі та інші служби, інформаційне забезпечення яких здійснюється шляхом створення електронних баз і банків даних, побудованих з урахуванням організаційної, функціональної та інформаційної структур об'єкта. Для їх реалізації використовуються засоби і системи розподіленої обробки інформації, побудовані на основі локальних автоматизованих робочих місць, з'єднаних високопродуктивними каналами зв'язку.

Подібні багаторівневі розподілені АСУ дозволяють ефективно вирішувати проблеми оперативної обробки інформації та роботи з документами, допомагають в аналізі ринкових ситуацій і виробленню управлінських рішень.

Застосування автоматизованої системи управління сприяє підвищенню продуктивності конкретної організації і забезпечує певний рівень якості

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				14
Змн.	Арк.	№ докум.	Підпис	Дата		

управління. Найбільша ефективність АСУ досягається шляхом оптимізації планів робіт підприємств і галузей в цілому.

Велике значення мають швидка вироблення оперативних рішень, чітке маневрування матеріальними, фінансовими та іншими ресурсами, інші фактори. Тому процес управління з використанням автоматизованих систем ґрунтується на економіко-організаційних моделях, які більшою чи меншою мірою відображають структурні та динамічні властивості об'єкта (підприємства, організації, фірми). Адекватність моделі є неодмінною умовою її застосування. Під адекватністю розуміється, передусім, її відповідність об'єкту з погляду поведінки в умовах, що імітують реальну ситуацію. Крім того, модель повинна відображати поведінку модельованого об'єкта в частині його характеристик і властивостей, істотних для вирішення поставленого завдання. Очевидно, що повного повторення об'єкта в моделі домогтися неможливо, проте можна знехтувати деякими деталями, що не є істотними для аналізу ситуації і прийняття відповідного управлінського рішення.

Основною задачею АСУ є обробка, подання та зберігання даних, але перед тим як починати роботу над ПЗ, потрібно ознайомитись з методом програмування, основними поняттями та способами зберігання та обробки інформації.

При розробці даного сайту ми будемо використовувати компонентний підхід, адже він має безліч переваг над іншими методами програмування.

Компонент це автономний код (підсистема), який може бути повторно використаний за рахунок його незалежного розгортання. Компоненти не обов'язково повинні бути великими за обсягом коду, але в основному вони більше за обсягом окремого класу або групи слабо пов'язаних класів. Тому для розробки був вибраний фреймворк Vue.js, який включає в себе всі основні принципи компонентного підходу.

Vue.js - JavaScript-фреймворк що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних [10]. Переваги Vue.js:

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				15
Змн.	Арк.	№ докум.	Підпис	Дата		

- Фреймворк досить простий й функціональний. Для того щоб розібратися в ньому, потрібен мінімальний багаж знань.
- Вимоги до стека відсутні, тому Vue.JS можна використовувати на будь-якому проекті.
- Досить висока швидкість розробки. Завдяки використанню будь-яких шаблонів і доступності документації, більшість проблем вирішуються досить швидко. У тому числі в порівнянні з React, так як в більшості додатків, які не мають складних інтерфейсів, вся міць цього фреймворка є трохи надлишковою.

Важливою частиною даного проекту є зберігання та обробка інформації.

Зберігання інформації - комплекс заходів, спрямований на забезпечення збереження повноти і цілісності сформованих даних, створення і підтримання належних умов для їх використання, а також запобігання несанкціонованому доступу, поширенню і використанню.

Найзручнішим способом зберігання інформації є база даних.

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів. Для зручної роботи з базою слід використати фреймворк Mongoose.[9]

Для зручного, гнучкого та інтерактивного написання модулів до сайту використовується MVVM паттерн.

Добре упорядкований і, можливо, самий повторно використовуваний спосіб організації коду - це використання шаблону MVVM. Model, View, ViewModel (шаблон MVVM) - це паттерн, що допомагає вам організувати і структурувати код для написання підтримуваних, тестуваних і розширюваних додатків. Model - зберігає дані і не має нічого спільного з будь-якою бізнес-логікою. ViewModel - діє як сполучна ланка між моделлю і представленням і

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		16

робить речі красивими. View - містить відформатовані дані і по суті делегує все в модель.

Шаблон MVVM, в кінцевому підсумку, є сучасною структурою шаблону MVC, тому основна мета полягає в тому, щоб забезпечити чіткий розподіл між предметної логікою і рівнем подання. Ось деякі з переваг і недоліків шаблону MVVM. Ключовою перевагою є можливість істинного поділу між поданням і моделлю, крім досягнення поділу та ефективності, яку ви отримуєте від цього. У реальному вираженні це означає, що коли ваша модель потребує зміни, її можна легко змінити, не звертаючись до подання, і навпаки.

1.2 Аналіз вимог до програмного продукту

Вимоги до програмного забезпечення:

1. Використання MVVM системи.

MVVM використовується для відокремлення моделі та її відображення. Це пов'язано з необхідністю змінювати їх незалежно одну від одної. Наприклад, розробник працює над логікою роботи з даними, а дизайнер — з користувацьким інтерфейсом.

2. Інтерфейс.

Користувацький інтерфейс повинен бути простим та зрозумілим для користувача, задля полегшення роботи з додатком.

Для того, щоб детальніше зрозуміти вимоги для розробки ми повинні створити діаграму варіантів використання (Use case diagram).

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				17
Змн.	Арк.	№ докум.	Підпис	Дата		

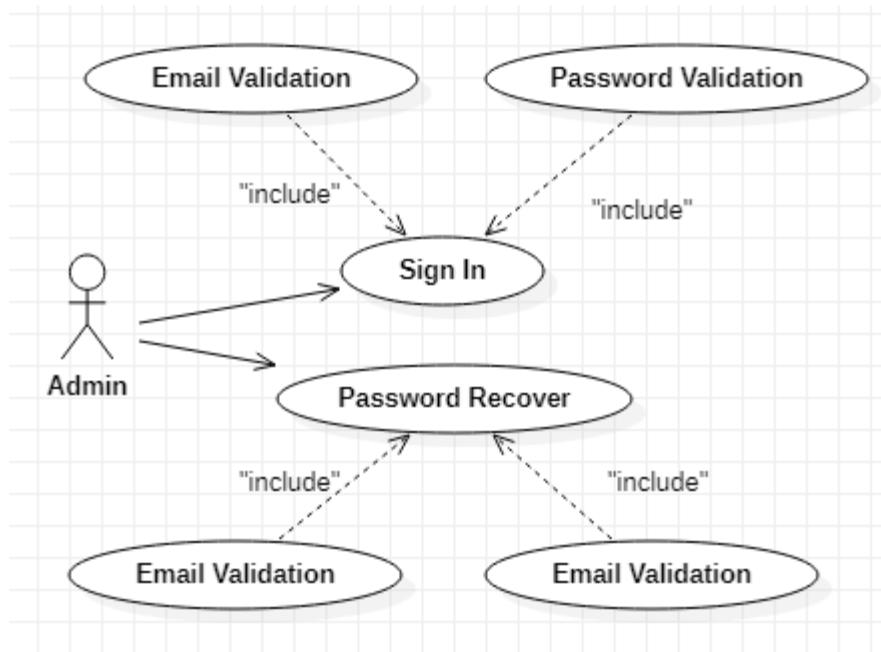


Рисунок 1.1 – Діаграма варіантів використання авторизації та відновлення паролю

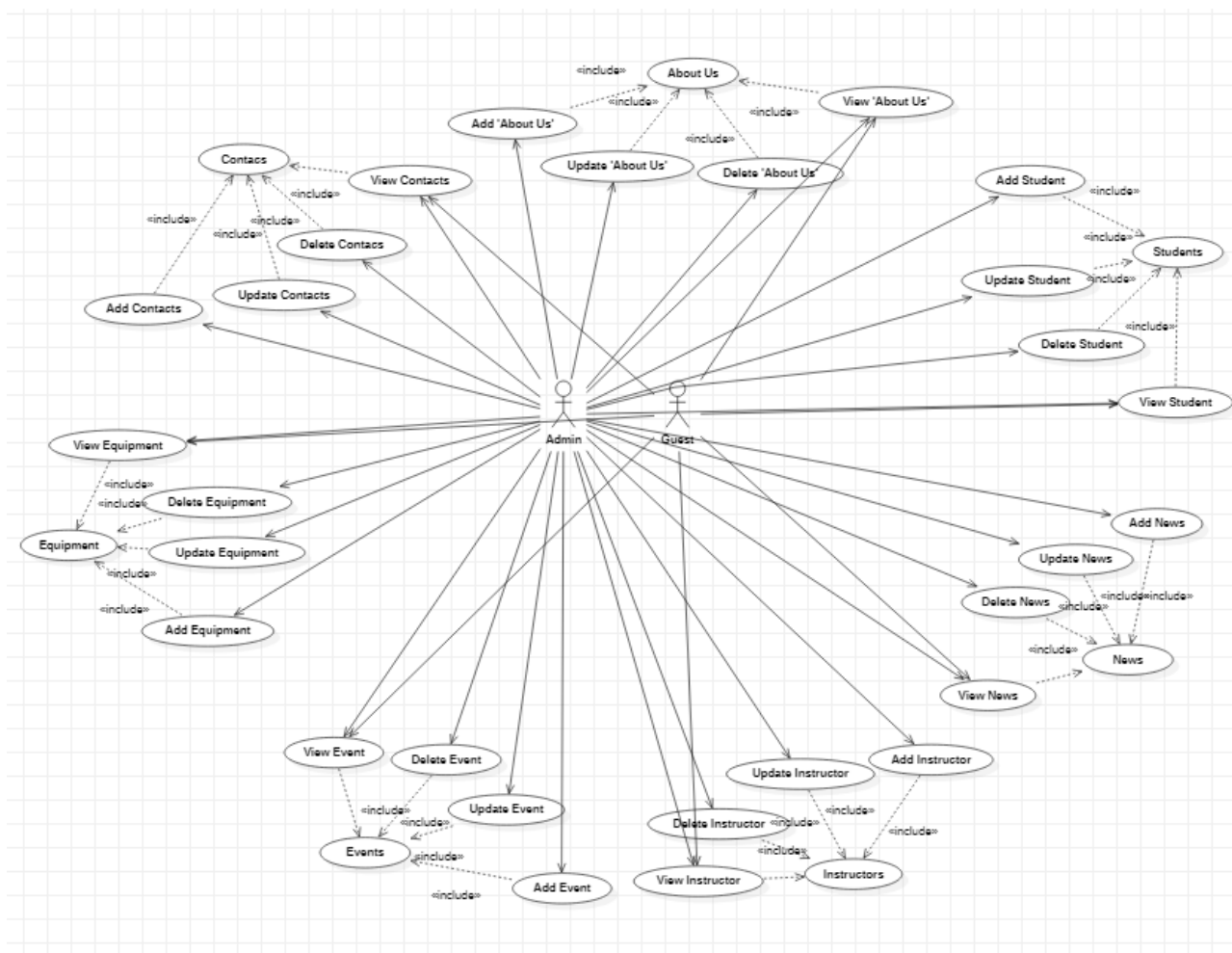


Рисунок 1.2 – Діаграма керування проектом

На рисунку 1.1 зображено діаграму варіантів використання авторизації та відновлення пароллю. Показана діаграма показує, що авторизуватись має можливість тільки адміністратор, наявність інших користувачів не передбачено. Гостьовий доступ не вимагає авторизації.

На рисунку 1.2 зображено керування проектом. Дана діаграма показує, що керувати модулями проекту може тільки адміністратор, а користувач (гість) має можливість тільки переглядати.

1.3 Аналіз наявного програмного забезпечення за тематикою кваліфікаційної роботи

Академії Cisco, як мережні, так і інших спрямувань зараз досить широко поширені й затребувані. Велика кількість українських ЗВО проводить курси при вишах, тому знайти схожі програмні продукти було не складно.

Розглянемо перший схожий програмний продукт “Центр підтримки академій Циско” Запорізького Національного Університету [16]. Структура даного програмного продукту включає в себе інформацію про сам портал “Про нас”, інформацію про курси “Наші курси”, інформацію для вчителів “Вчителю”, модулі подій та контактів. Даний портал має такі переваги:

- Наявність модуля “Студенту”, де студент може переглянути записити вебінарів для школярів та студентів.
- Наявність модуля “Вчителю”, де вчитель / викладач може переглянути корисні для вчителя матеріали.
- Наявність календаря, в якому відзначені події пов’язані з мережевими та іншими ІТ – новинками та подіями.

Також даний портал має такі недоліки:

- Відсутність можливості переглянути працюючих інструкторів та їхніх курсів.
- Перегляд подій захований в календарі і не зрозумілий інтуїтивно, для того, щоб зрозуміти, що це архів подій, потрібно погортати календар.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Відсутня можливість переглянути успішних студентів, які вже закінчили якісь курси й отримали сертифікати.

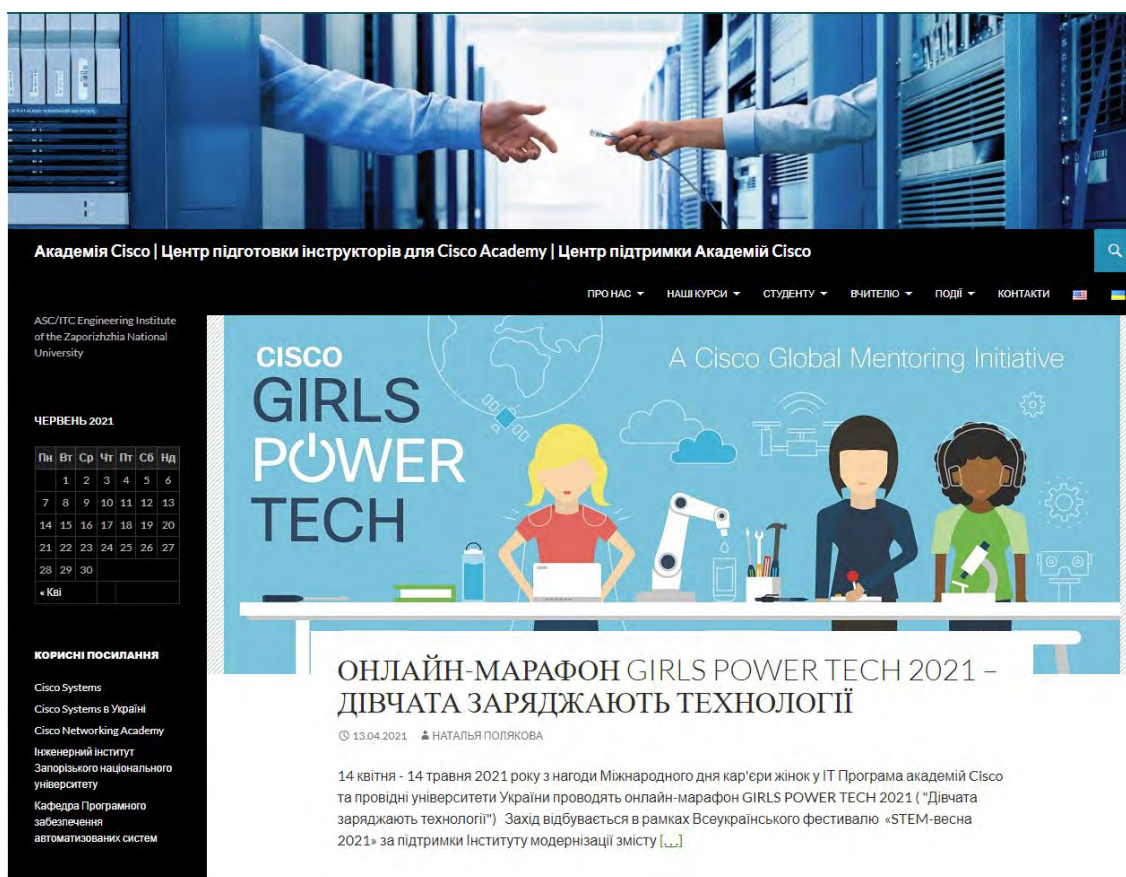


Рисунок 1.3 – Онлайн портал “Центр підтримки академій Cisco”

Наступним буде сайт Черкаського Державного Технологічного Університету [17]. Тут мережева академія Cisco є лише модулем сайту університету, але має мінімальний корисний функціонал потрібний для порталу такого роду.

Модуль має такі переваги:

- Перелік курсів за напрямками, які надає університет.
- Вартість та тривалість курсів, які викладають на комерційній основі.
- Можливість переглянути профілі інструкторів, які викладають курси. В профілях можна переглянути ПІБ інструкторів, їх посади, та акредитовані курси.

Також даний модуль має такі недоліки:

- Новини та події об’єднані в один модуль.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				20
Змн.	Арк.	№ докум.	Підпис	Дата		

- Відсутня можливість переглянути успішних студентів, які вже закінчили якісь курси й отримали сертифікати.



Рисунок 1.4 – Модуль “Мережева академія CISCO” сайту Черкаського державного технологічного університету

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 – Підсумкова таблиця результатів

Назва сайту/Критерій	Перегляд інструкторів	Перегляд курсів	Перегляд подій	Перегляд новин	Вартість курсів	Перегляд корисних матеріалів для вчителів
Центр підтримки академій Циско	-	+	+	+	-	+
Модуль “Мережева академія CISCO” сайту Черкаського Державного Технологічного Університету	+	+	+	+	+	-

В результаті можемо зробити певні підсумки, другий сайт має більший функціонал в плані опису курсів, їх вартості, та відображено інструкторів та їх акредитовані курси. Перший сайт має більш зручний користувацький інтерфейс, події та новини розташовані в окремих модулях, також їх можна переглянути в архіві за допомогою календаря.

1.4 Технічне завдання на кваліфікаційну роботу

1. Загальне положення

1.1. Найменування програмного засобу

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Повне найменування програмної системи: «Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка» (надалі «програма»).

1.2. Призначення розробки та область застосування

Програмна система «Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"» призначена для збору, обробки та відображення даних академії Циско.

1.3. Мета

Програмна система «Проект інформаційної системи Центру підтримки академій Cisco Державного університету "Житомирська політехніка"» дозволить переглядати інформацію про академії Циско, їх студентів, інструкторів тощо.

1.4. Найменування розробника та замовника

Розробник даного продукту – студент групи КІ-3 Войналович Владислав (надалі «розробник»).

Замовник програмного продукту – кафедра комп’ютерної інженерії та кібербезпеки (надалі «замовник»).

2. Підстава для розробки

2.1. Документ на підставі якого ведеться розробка

Робота ведеться на підставі навчального плану за спеціальністю «Комп’ютерна інженерія», наказу про закріплення тем кваліфікаційних робіт за студентами.

3. Вимоги до програми

3.1. Вимоги до функціональних характеристик

3.1.1. Загальні вимоги

Програмна система має забезпечувати:

- Незалежний від операційної системи користувацький інтерфейс;
- надійний доступ до системи;
- виключення можливості несанкціонованого доступу до системи;
- надійне збереження та коректна обробка даних;

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		23

- можливість оновлення системи при зміні функціональних потреб користувача.

3.1.2. Склад виконуваних функцій

Програма має виконувати наступні функції:

- Основні цілі: метою створення проекту є реалізація порталу, який візуалізує корисну інформацію щодо академій Циско, та пов'язаної з академіями інформації.
- Представлення проекту: проект буде реалізовано у вигляді сайту, що містить структуровану інформацію по даній тематичній області.
- Візуалізація робочого процесу: сайт міститиме окремі модулі для кожної області, які можна буде оновлювати, редагувати та видаляти.

3.1.3. Організація вхідних і вихідних даних

Вхідними даними є інформація академії, інструкторів, студентів, обладнання та ін.

3.1.4. Часові характеристики і розмір пам'яті, необхідної для роботи програми

Час реакції програми на дії користувача (маніпуляції в пристроях введення даних) не повинен перевищувати 0,25 с.

Час виконання команд не більше 1 с.

Відображення масивів даних за запитами не більше 3 хвилин.

Доступність БД – 90% цілодобово.

Операції з'єднання з БД - не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми - не менше 2Гб.

Дисковий простір, необхідний для збереження програми і файлів даних - не більше 3.24 Мбайт.

Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

3.2. Вимоги до надійності

3.2.1. Вимоги до надійного функціонування

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Програма повинна нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі 30 користувачів.

При апаратних збоях, відновлення нормальної роботи програми повинне виконуватися після:

а) апаратні збої сервера – перезавантаження ОС сервера, запуск сервера БД (запуску резервного сервера, використання технологій RAID для збереження даних);

б) апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми;

При збоях програмного забезпечення:

а) в системі має бути реалізована коректна обробка виняткових ситуацій.

3.2.2. Контроль вхідної та вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- Перевірка відповідності даних доменам інформаційних атрибутів;
- Використанням механізму автозаповнення та вибору за переліком для зв'язаних даних;
- Захист від помилок оператора (залипань, випадкових символів тощо).

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування.

Для вихідної інформації передбачити:

- відсутність блокування даних через багатокористувальницький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД.

3.2.3. Час відновлення після відмови

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезавантаження операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.3. Умови експлуатації і збереження

Програма використовується у багатокористувацькому середовищі. Регламентні операції проводити за графіком:

- оновлення системи ідентифікації та аутентифікації користувачів – 1 раз на місяць та за потребою.

3.4. Вимоги до інформаційної і програмної сумісності.

Формат відображення даних має дозволяти імпорт даних в додатки MS Office для редагування та перегляду.

3.4.1. Вимоги до інформаційних структур на вході і виході

Вибір методів рішення здійснюється розробником без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п.3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.4.3. Вимоги до системи програмних засобів

Вимоги до програмного забезпечення сервера:

ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником).

Вимоги до програмного забезпечення робочої станції:

ОС - родина Windows не нижче Windows XP.

Драйвери периферійних пристроїв - введення/виведення визначаються та встановлюються при встановленні ОС в залежності від конфігурації робочої станції.

Пакет додатків MS Office.

Робоча станція є клієнтом мереж Microsoft.

3.5. Вимоги до складу і параметрів технічних засобів

Вимоги до складу технічних засобів:

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				26
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- знімний запам'ятовуючий пристрій для архівування даних;
- принтер для друку;
- засоби для під'єднання до локальної мережі.

2. Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- засоби для під'єднання до локальної мережі.
- Intel i5 тактовою частотою 2,7 ГГц, або аналогічний за продуктивністю AMD;
- оперативна пам'ять – 4Гб;
- об'єм дискової пам'яті - 128Гб.

Висновки до першого розділу

Під час написання першого розділу було проаналізовано схоже ПЗ, та повністю сформовано уявлення про кінцевий вигляд розроблюваного програмного продукту. Отримано теоретичний фундамент для подальшої реалізації програмної та візуальної частин продукту.

Наступним кроком було визначено необхідний для реалізації інструментарій.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				27
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Алгоритм роботи та стани програмної системи

Для чіткого розуміння станів та алгоритмів роботи системи розглянемо діаграми діяльності та діаграми класів. Насамперед розглянемо теоретичну основу, а саме визначення діаграми діяльності.

Діаграма діяльності (activity diagram) - візуальне представлення графу діяльностей у мові UML. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. [11]

Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів - та перетворює їх на множину вихідних сигналів. Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності. Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку. Діаграми активностей будуються з обмеженої кількості фігур, з'єднаних стрілочками. Найважливіші типи фігур:

- скруглені прямокутники позначають дії;
- ромби позначають рішення;
- риски позначають початок (розподіл) чи кінець (об'єднання) паралельних активностей;
- чорний кружечок позначає старт (початковий стан) процесу;
- чорний кружечок в колі позначає кінець (кінцевий стан).

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Представлена діаграми активності на рисунку 2.1.

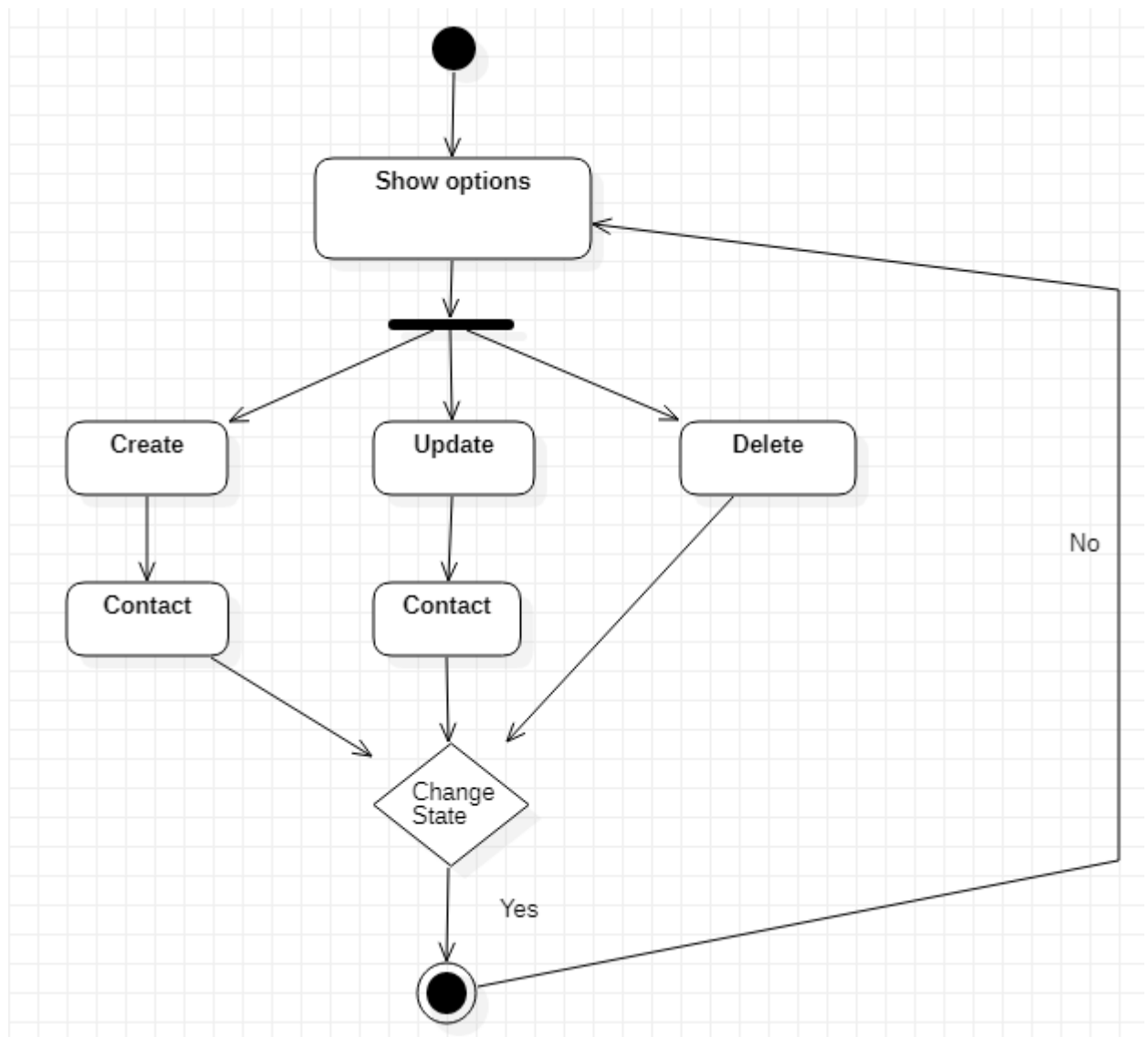


Рисунок 2.1 – Діаграма активності керування контактами

На діаграмі 2.1 видно, що при керуванні контактами адміністратор може створити, оновити, або видалити контакти. При створенні або редагуванні обирається опція вміст контакту. При видаленні контакту користувач лише виконує дію видалення.

2.2 Об'єктно-орієнтована модель системи

Діаграма класів (англ. Static Structure diagram)– статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів також може містити

позначення для пакетів та позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Клас — це спеціальна конструкція, яка використовується для групування пов'язаних змінних та функцій. При цьому, згідно з термінологією ООП, глобальні змінні класу (члени-змінні) називаються полями даних (також властивостями або атрибутами), а члени-функції називають методами класу. Створений та ініціалізований екземпляр класу називають об'єктом класу. На основі одного класу можна створити безліч об'єктів, що відрізнятимуться один від одного своїм станом (значеннями полів).

Виходячи з даних відомостей, було побудовано діаграму класів, зображену на рисунку 2.2.

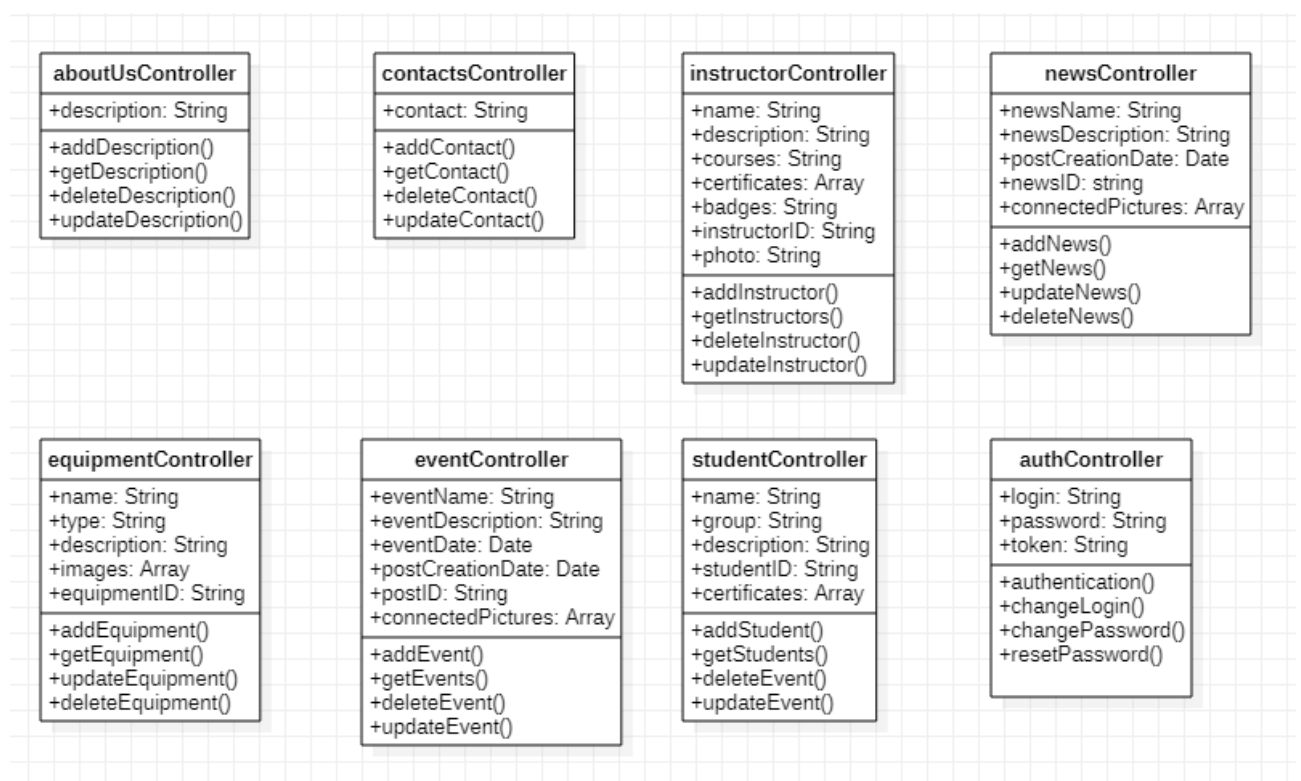


Рисунок 2.2 – Діаграма класів

Опис класів, а також їх полів та методів (у класах використовується англomовна назва полів та методів з використанням верблужого регістру -

camelCase). В даному випадку класи представляють собою збірки обробників запитів, які клієнт надсилає серверу, тому явних зв'язків між собою класи не мають.

Клас `aboutUsController` має поле `description`, яке й містить в собі опис, й методи які цим полем маніпулюють, а саме: `addDescription` – додає інформацію в поле `description`, `getDescription` – отримує з бази даних інформацію поля `description`, `deleteDescription` – видаляє з бази даних значення поля `description`, `updateDescription` – оновлює значення поля `description`, та перезаписує його значення в базі даних.

Клас `contactsController` має поле `contact`, яке містить в собі інформацію про контакти, й методи які цим полем маніпулюють, а саме: `addContact` – додає інформацію в поле `contact`, `getContact` – отримує контакти з бази даних, `deleteContact` – видаляє контакти, `updateContact` – оновлює значення поля `contact`, та перезаписує його значення в базі даних.

Клас `instructorController` містить поля які описують інструктора, й методи які маніпулюють значеннями цих полів, а саме: `addInstructor` – додає нового інструктора, `getInstructors` – отримує з бази даних інформацію про всіх інструкторів, `deleteInstructor` – видаляє інструктора, `updateInstructor` – оновлює значення полів класу та перезаписує ці значення для вказаного інструктора в базі даних.

Клас `newsController` містить поля які описують новину, й методи які маніпулюють значеннями цих полів, а саме: `addNews` – додає нову новину, `getNews` – отримує дані всіх новин з бази даних, `updateNews` – оновлює значення полів новини й перезаписує ці значення для вказаної новини в базі даних, `deleteNews` – видаляє новину.

Клас `equipmentController` містить поля які описують обладнання, й методи які маніпулюють значеннями цих полів, а саме: `addEquipment` – додає нове обладнання, `getEquipment` – отримує дані обладнання з бази даних, `updateEquipment` – оновлює значення полів обладнання й перезаписує ці

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				31
Змн.	Арк.	№ докум.	Підпис	Дата		

значення для вказаного обладнання в базі даних, deleteEquipment – видаляє обладнання.

Клас eventController містить поля які описують новину, й методи які маніпулюють значеннями цих полів, а саме: addEvent – додає нову подію, getEvents – отримує список всіх подій збережених в базі даних, updateEvent – оновлює значення полів події та перезаписує ці значення для вказаної події в базі даних, deleteEvent – видаляє подію.

Клас studentController містить поля, які описують студента, й методи які маніпулюють значеннями цих полів, а саме: addStudent – додає нового студента, getStudents – отримує список всіх студентів з бази даних, updateStudent – оновлює значення полів студента й перезаписує ці значення для вказаного студента в базі даних, deleteStudent – видаляє студента.

Клас authController містить поля які описують користувача, й методи які маніпулюють значеннями цих полів, а саме: authorization – авторизує користувача в системі, changeLogin – змінює логін користувача, changePassword – змінює пароль користувача, resetPassword – надсилає тимчасовий випадково згенерований пароль користувачеві на пошту, пароль можна змінити після авторизації.

Робота всієї клієнтської частини реалізована за допомогою фреймворка Vue.js, в основі якого лежить патерн MVVM. [12] MVVM полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес-логіки (бек-енд логіки), відомої як модель. Схема роботи MVVM представлена на рисунку 2.3.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				32
Змн.	Арк.	№ докум.	Підпис	Дата		

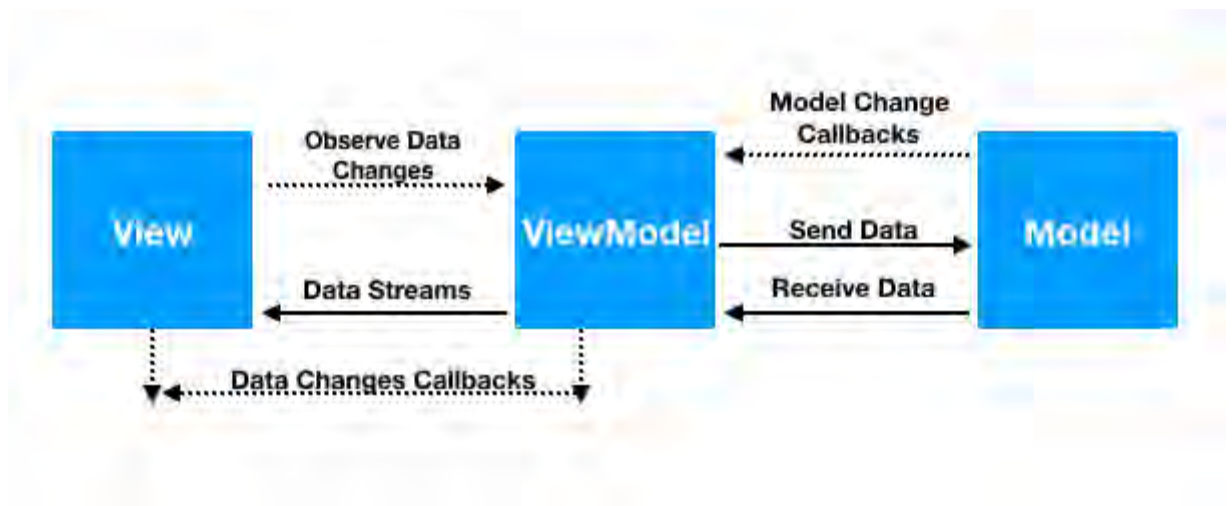


Рисунок 2.3 - Модель MVVM

MVVM зручно використовувати замість класичного MVC та йому подібних у тих випадках, коли на платформі, де ведеться розробка, присутнє «зв'язування даних».

В MVC/MVP зміни у користувацькому інтерфейсі не впливають безпосередньо на модель, а йдуть через Контролер/Presenter. У таких технологіях, як WPF та Silverlight, присутня концепція «зв'язування даних», що дозволяє зв'язувати дані із візуальними елементами в обидві сторони.

Архітектура MVVM вирішує цю проблему чітким розмежовуванням відповідальності:

- Розробка користувацького інтерфейсу здійснюється дизайнером інтерфейсів за допомогою технології, більш-менш природної для такої роботи (XML)
- Логіка користувацького інтерфейсу реалізується розробником як компонент ViewModel
- Функціональні зв'язки між користувацьким інтерфейсом та ViewModel реалізуються через біндинги (bindings), які, по суті, є правилами типу «якщо кнопка А була натиснута, повинен бути викликаний метод `onButtonAClick()` з ViewModel». Біндинги можуть бути написані в коді або визначені декларативним шляхом (Android використовує обидва типи).

Архітектура MVVM використовується в тому чи іншому вигляді усіма сучасними технологіями, наприклад Microsoft WPF і Silverlight, Oracle JavaFX, Adobe Flex, AJAX.

2.3 Комунікації і послідовність взаємодії об'єктів системи

Діаграма послідовності (sequence diagram) — різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів, впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями.

На діаграмі послідовності зображаються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії.

Лінія життя об'єкта зображується пунктирною вертикальною лінією, асоційованою з єдиним об'єктом на діаграмі послідовності. Лінія життя служить для позначення періоду часу, протягом якого об'єкт існує в системі і, отже, може потенційно брати участь у всіх її взаємодіях. Якщо об'єкт існує в системі постійно, то і його лінія життя повинна продовжуватися по всій площині діаграми послідовності від самої верхньої її частини до самої нижньої.

У процесі функціонування об'єктно-орієнтованих систем одні об'єкти можуть перебувати в активному стані, безпосередньо виконуючи певні дії, або стані пасивного очікування повідомлень від інших об'єктів. Щоб явно виділити подібну активність об'єктів, в мові UML застосовується спеціальне поняття, що отримало назву фокуса управління. Фокус управління зображується у формі витягнутого вузького прямокутника, верхня сторона якого позначає початок отримання фокусу управління об'єкта, а його нижня сторона - закінчення фокусу управління. Прямокутник розташовується нижче позначення відповідного

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		34

об'єкта і може замінювати його лінію життя, якщо на всьому її протязі він є активним.

В UML кожна взаємодія описується сукупністю повідомлень, якими ті об'єкти, що беруть участь у ньому обмінюються між собою. Повідомлення є закінченим фрагментом інформації, який відправляється одним об'єктом іншому. Прийом повідомлення ініціює виконання певних дій, спрямованих на вирішення окремого завдання тим об'єктом, якому це повідомлення відправлено.

Логіка виконання дій для кожного із процесів, зображених на діаграмах послідовності, описана раніше, у діаграмах кооперації.

Позначення :

- об'єкти позначаються прямокутниками з підкресленими іменами (щоб відрізнити їх від класів);
- повідомлення (виклики методів) - лініями зі стрілками;
- повідомлення які повертають результати - пунктирними лініями зі стрілками;
- прямокутники на вертикальних лініях під кожним з об'єктів показують "час життя" (фокус) об'єктів;
- лінія життя - вертикальна лінія, яка представляє послідовність подій, що відбуваються підчас взаємодії.

Нижче на рисунку 2.4 наведено діаграму послідовності процесів роботи в системі при авторизації.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				35
Змн.	Арк.	№ докум.	Підпис	Дата		

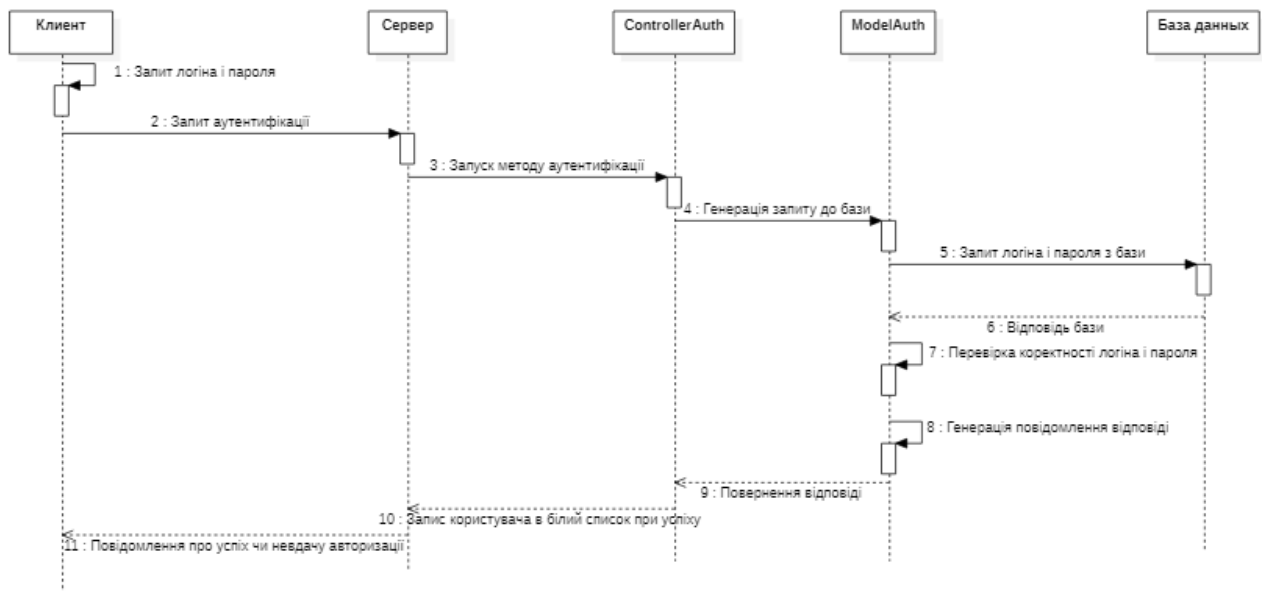


Рис. 2.4 - Діаграма послідовності для авторизації

Бачимо що авторизація користувачів є достатньо складним процесом, при цьому обробка інформації покладається тільки на один клас AuthModel.

2.4 Розробка функціональних алгоритмів роботи програми

Після опису загального алгоритму роботи додатку, реалізованого в попередній розділах можна приступити до безпосередньої роботи над проектом.

Почнемо з програмної реалізації модуля авторизації. Для цього потрібно підключити до проекту базу даних, MongoDB. Це відбувається за допомогою створення нового проекту в обліковому записі на сайті

<https://cloud.mongodb.com/>. Процес створення зображено на рисунках 2.5 – 2.6.

Create Database

DATABASE NAME ?

ciscoAcademySupportCenter

COLLECTION NAME ?

users

☐ Capped Collection

Before MongoDB can save your new database, a collection name must be specified at the time of creation.

Cancel Create

Рис. 2.5 – Інтерфейс вибору назви нового проекту

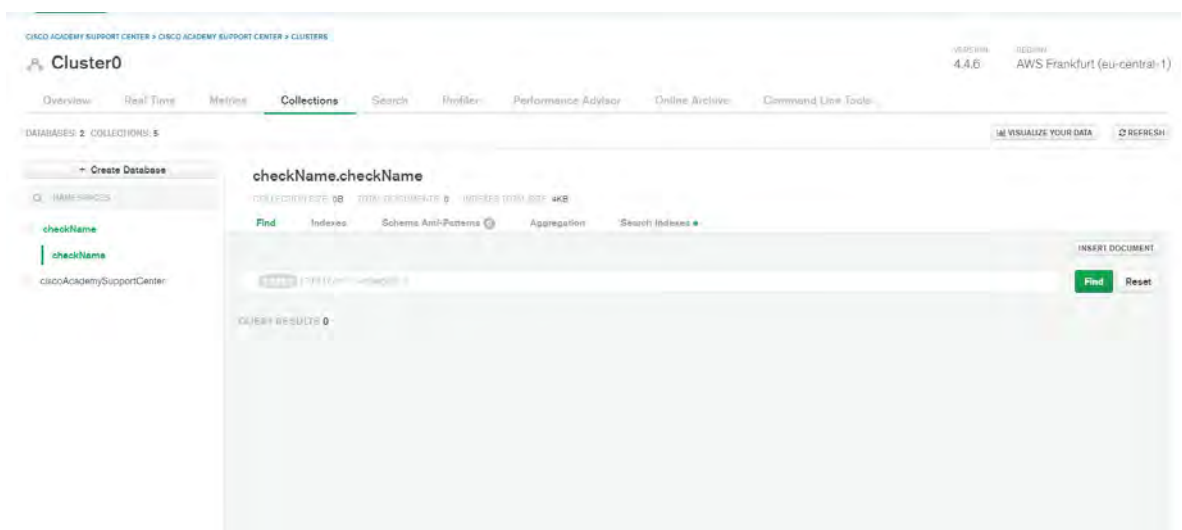


Рис. 2.6 – Відображення бази даних

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				37
Змн.	Арк.	№ докум.	Підпис	Дата		

Після налаштування бази даних потрібно налаштувати підключення веб-додатку до створеної нами бази даних. Для цього потрібно додати в проект бібліотеку, яка допоможе комфортно працювати з базою даних.

Після реалізації зв'язку з базою даних потрібно реалізувати компоненти, які зв'язуватимуться з базою через шину даних. Потрібно реалізувати два компоненти: авторизація та відновлення паролю за допомогою електронної пошти. Код компонентів можна переглянути в Додатку А. Після розробки компонентів потрібно реалізувати обробку інформації й повернення результату обробки користувачеві. Скористаємось бібліотекою Vuex [13] для фреймворку Vue.js. Бібліотека представляє собою шину даних, через яку компоненти можуть надсилати інформації на сервер для подальшої її обробки. Для роботи з бібліотекою потрібно створити екземпляр об'єкту бібліотеки. Так як ми будемо використовувати модульний підхід для роботи з бібліотекою, для кожного окремого модуля проекту будемо використовувати окремий модуль для Vuex. Основні функції реалізованого модуля можна переглянути в таблиці 2.1.

Таблиця 2.1 – Опис функцій модуля авторизації та їх реалізація

Таблиця 1

Опис функції	Програмна реалізація
Авторизація (серверна частина)	<pre> async authentication(req, res) { try { let { login, password } = req.body; //check is there some user with this name const user = await User.findOne({ login }); if (!user) { return res.status(400).json({ message: `Користувача з логіном \${login} не знайдено` }); } //compare password from frontend with encrypted password from db </pre>

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				38
Змн.	Арк.	№ докум.	Підпис	Дата		

	<pre> const validPassword = bcrypt.compareSync(password, user.password) if (!validPassword) { return res.status(400).json({ message: `Введено невірний пароль`}); } const token = generateAccessToken(user._id); return res.json({ token }); } catch (e) { console.log(e) } } </pre>
Авторизація (клієнтська частина)	<pre> async auth({ commit }, authOptions) { let { token } = await request(`\${url.domen}/auth`, 'POST', authOptions); commit('setAdminToken', token); localStorage.setItem("adminToken", JSON.stringify(token)); return token; } </pre>
Відновлення паролю (Клієнтська частина)	<pre> async resetPassword({ commit }, data) { let response = await request(`\${url.domen}/auth/reset`, "POST", data, true); console.log(commit) return response; } </pre>

Продовження таблиці 2.1.

Відновлення паролю (Сервер)	<pre> async resetPassword(req, res) { const { email } = req.fields; let user = await User.findOne({ login: email }).exec(); console.log(user) let randomPass = Math.random().toString(36).substr(2, 9); await bcrypt.hash(randomPass, 10, (err, hash) => { if (!err) { const message = { to: email, subject: "Відновлення паролю", text: `Вам надано тимчасовий пароль \${randomPass} для авторизації.` </pre>
-----------------------------	--

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				39
Змн.	Арк.	№ докум.	Підпис	Дата		

	<pre> } mailer(message) user.password = hash; user.save(); res.json({ success: true }); } else { console.log(err) } }); } </pre>
--	---

Вище описані функції реалізують основний функціонал авторизації та відновлення паролю користувача.

Наступним кроком є реалізація адміністраторської частини проекту. Для об'єднання всіх модулів адміністраторської панелі створимо компонент загального макету для панелі адміністратора AdminLayout.vue. Підключення всіх підкомпонентів панелі адміністрування реалізовано за допомогою роутера. В кожному окремому підкомпоненті отримуватиметься інформація з бази даних за допомогою запиту до сервера, реалізованого в відповідному модулі Vuex. Загальна схема взаємодії компонента представлена на рисунку 2.7.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				40
Змн.	Арк.	№ докум.	Підпис	Дата		

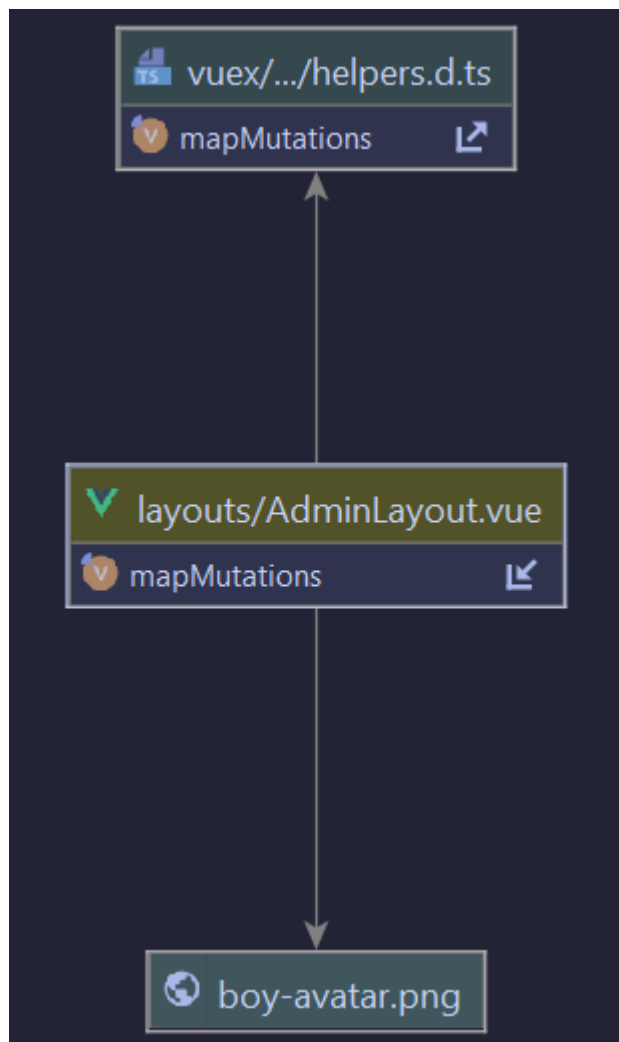


Рисунок 2.7 – Схема взаємодії компонента `AdminLayout.vue` з допоміжними модулями й медіаресурсами системи

Для всіх подулів панелі адміністратора створимо окремі компоненти які будуть рендеритись всередині `AdminLayout.vue`.

Компонент для модуля “Про нас” називатиметься `AdminAboutUs.vue`, в ньому виводитиметься текст модуля, буде наявна можливість додати, видалити або редагувати текст модуля.

Компонент для модуля “Контакти” називатиметься `AdminContacts.vue`, в ньому виводитиметься текст модуля, буде наявна можливість додати, видалити або редагувати текст модуля.

Компонент для модуля “Обладнання” називатиметься `AdminEquipment.vue`, в ньому виводитиметься все наявне обладнання, буде наявна можливість додавати нове обладнання, видаляти, та редагувати наявне обладнання.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Компонент для модуля “Події” називатиметься AdminEvents.vue, в ньому виводитимуться всі наявні події, буде наявна можливість додавати нові події, видаляти, та редагувати вже існуючі події.

Компонент для модуля “Інструктори” називатиметься AdminInstructor.vue, в ньому виводитимуться всі наявні інструктори, буде наявна можливість додавати інструктора, видаляти, та редагувати вже існуючих інструкторів.

Компонент для модуля “Новини” називатиметься AdminNews.vue, в ньому виводитимуться всі наявні новини, буде наявна можливість додавати новини, видаляти, та редагувати вже існуючі новини.

Компонент для модуля “Студенти” називатиметься AdminStudent.vue, в ньому виводитимуться всі наявні студенти, буде наявна можливість додавати нових студентів, видаляти, та редагувати вже наявних студентів.

Всі компоненти взаємодіють між собою та іншими модулями системи, які представлені на рисунку 2.8.

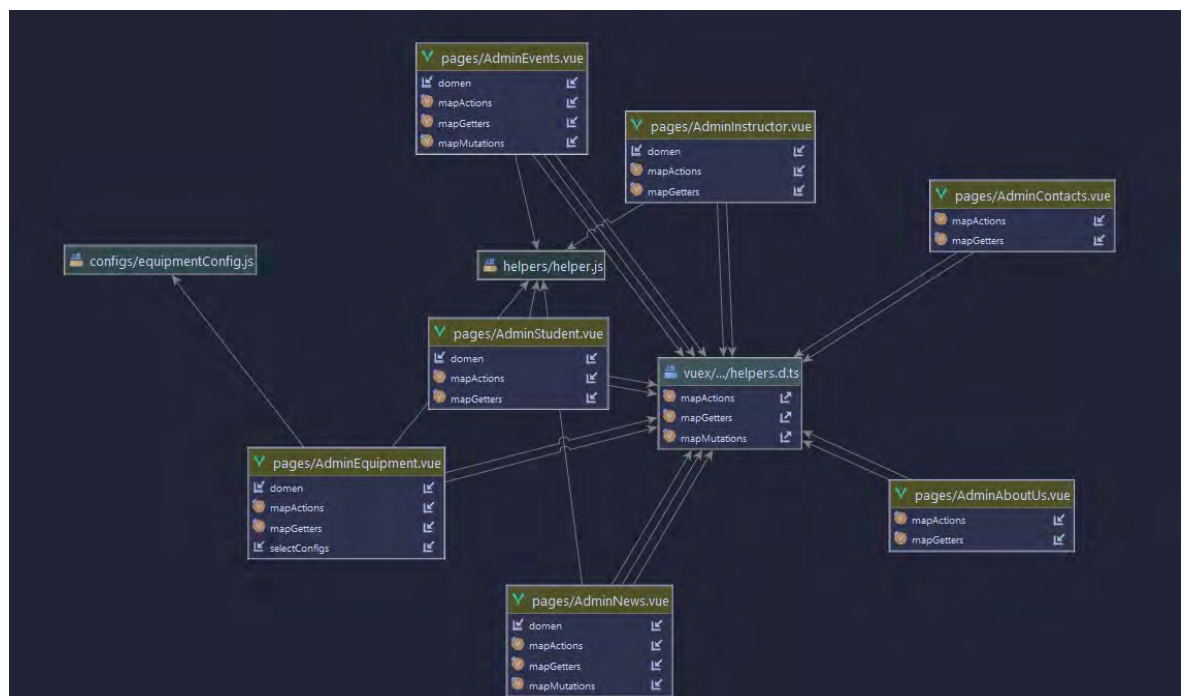


Рисунок 2.8 – Схема взаємодії компонентів системи з допоміжними модулями системи

На даному етапі, коли створено інтерфейс панелі адміністратора, та описаний весь функціонал, який мають компоненти системи, приступимо до створення бази даних. Так як MongoDB являється документо-орієнтованою базою даних, то всі дані будуть зберігатись у форматі JSON [14]. Тому опишемо основні поля, які будуть зберігати дані користувачів.

Дані модуля “Про нас” зберігатимуться в сутності «aboutUs», структура сутності представлена в таблиці 2.2.

Дані модуля “Контакти” зберігатимуться в сутності «contacts», структура сутності представлена в таблиці 2.3.

Дані модуля “Обладнання” зберігатимуться в сутності «equipment», структура сутності представлена в таблиці 2.4.

Дані модуля “Події” зберігатимуться в сутності «events», структура сутності представлена в таблиці 2.5.

Дані модуля “Інструктори” зберігатимуться в сутності «instructors», структура сутності представлена в таблиці 2.6.

Дані модуля “Новини” зберігатимуться в сутності «news», структура сутності представлена в таблиці 2.7.

Дані модуля “Студенти” зберігатимуться в сутності «students», структура сутності представлена в таблиці 2.8.

Таблиця 2.2 – Сутність «aboutUs»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
description	String	Текст опису

Таблиця 2.3 – Сутність «contacts»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
contact	String	Текст контакту

Таблиця 2.4 – Сутність «equipment»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
name	String	Назва обладнання
type	String	Тип обладнання
description	String	Опис обладнання
images	Array	Зображення обладнання
equipmentID	String	Id обладнання

Таблиця 2.5 – Сутність «events»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
eventName	String	Назва події
eventDescription	String	Опис події
eventDate	String	Дата події
postCreationDate	String	Дата створення запису події
postID	String	Id події
connectedPictures	Array	Прикріплені зображення

Таблиця 2.6 – Сутність «instructors»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
name	String	Ім'я інструктора
description	String	Опис інструктора
courses	String	Курси
certificates	Array	Сертифікати
badges	String	Електронні бейджі

Продовження таблиці 2.6

instructorID	String	Id інструктора
photo	String	Фото інструктора

Таблиця 2.7 – Сутність «news»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
newsName	String	Назва новини
newsDescription	String	Опис новини
postCreationDate	String	Дата створення новини
newsID	String	Id новини
connectedPictures	String	Приєднані зображення

Таблиця 2.8 – Сутність «students»

Назва поля	Тип даних	Опис поля
_id	Int(11)	Автоматично згенероване id сутності
name	String	Ім'я студента
group	String	Група
description	String	Опис студента
studentID	String	Id студента
certificates	Array	Сертифікати

Висновки до другого розділу

В даному розділі було проаналізовано та розроблено загальні алгоритми сайту, описано основні принципи його роботи.

Реалізовано користувацький інтерфейс, змодельовано вікна сайту, та реалізовано програмний код сайту.

В результаті було отримано готовий програмний продукт з функціональністю, який виконує всі поставлені задачі та вимоги.

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Архітектура програмного комплексу та його розгортання

Діаграма розгортання (deployment diagram) — діаграма в UML [15], на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми, на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

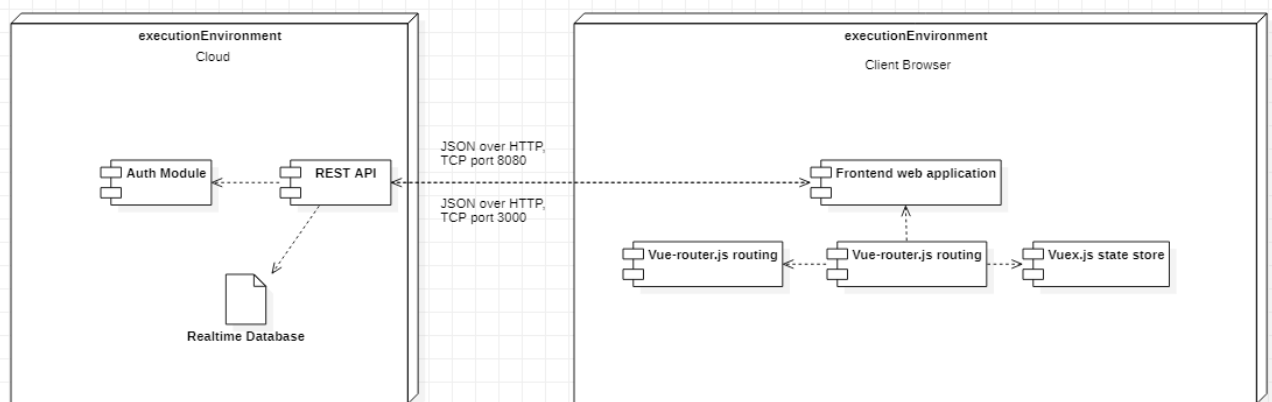


Рисунок 3.1 – Діаграма розгортання

3.2 Опис роботи з додатком

На даному етапі сайт вже повністю реалізовано, тому зараз опишемо його роботу. Спочатку розглянемо форму авторизації до адміністраторської панелі. Користувач повинен ввести свій логін та пароль.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				
Змн.	Арк.	№ докум.	Підпис	Дата		46

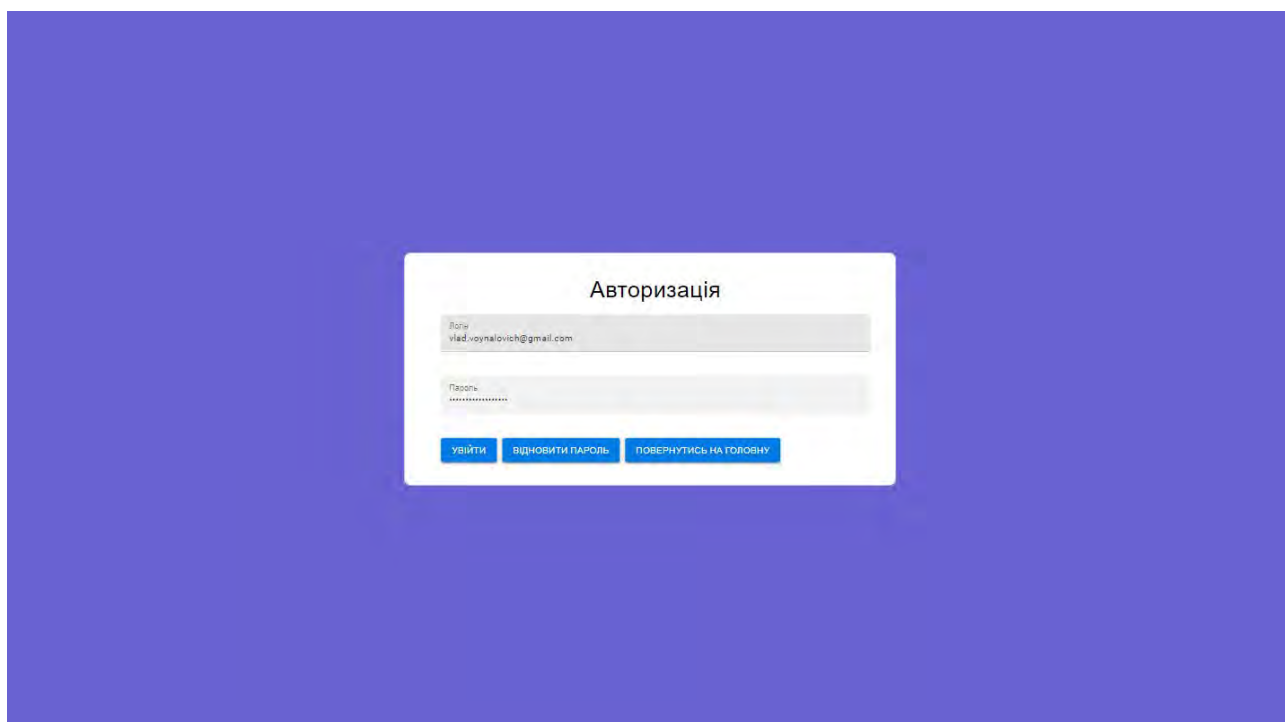


Рис. 3.2 – Форма авторизації

У випадку якщо користувач забув пароль, можна отримати тимчасовий пароль, який буде надісланий на пошту користувача.

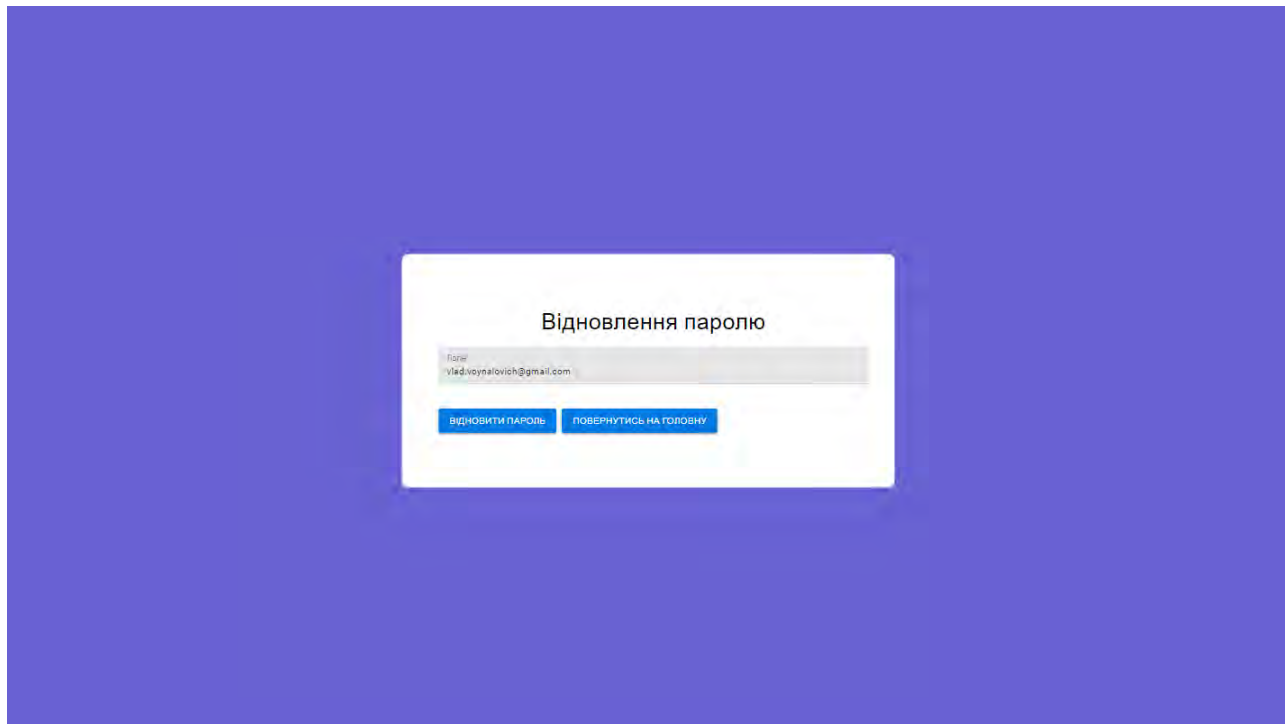


Рис. 3.3 – Форма зміни пароля за допомогою електронної пошти

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				47
Змн.	Арк.	№ докум.	Підпис	Дата		

Після авторизації користувача буде допущено до адміністраторської панелі на сторінку подій.

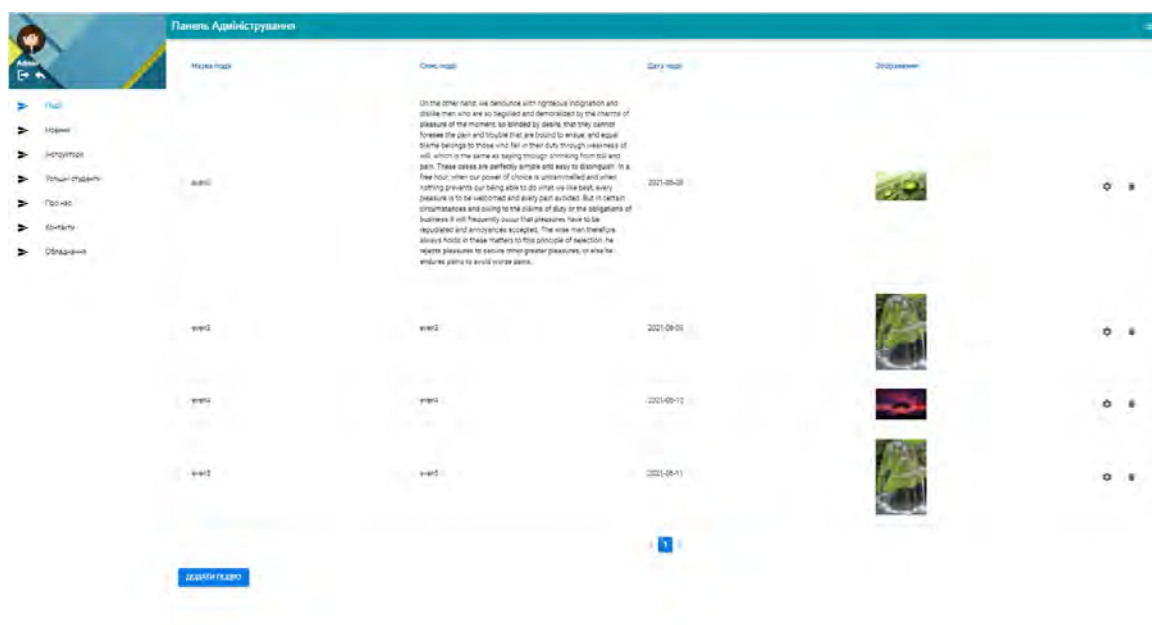


Рис. 3.4 – Адміністраторська панель

3.3 Тестування веб-додатку

На кожному з модулів адміністраторської панелі користувач має можливість додавати нові записи, оновити окремий запис, або ж видалити запис. Розглянемо кожну з функціональностей окремо. Спочатку спробуємо додати подію в модулі подій.

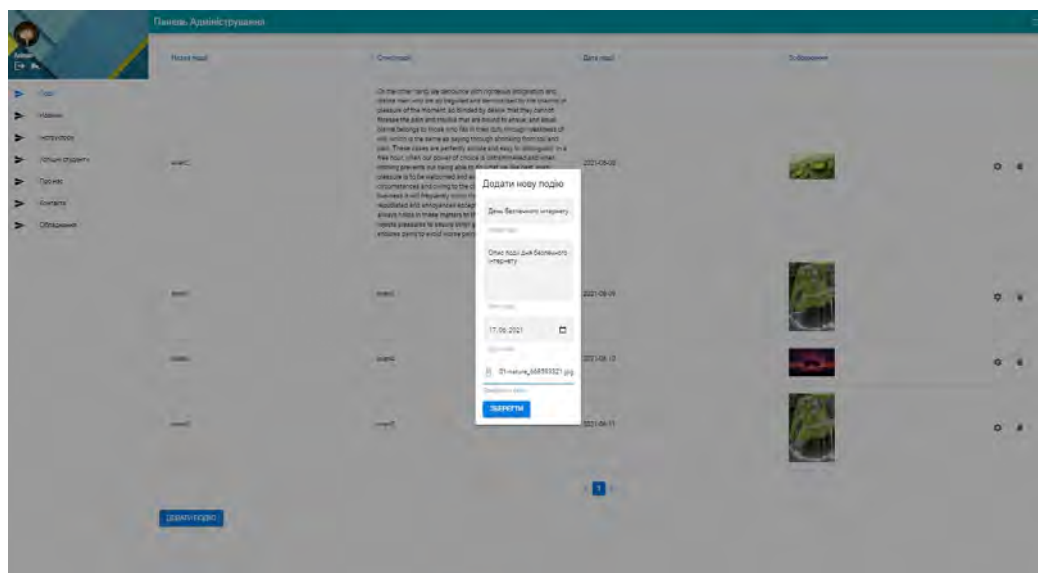


Рис. 3.5 – Заповнена форма нової події

		Войналович В.Ю							Арк.
		Головня О.С							48
Змн.	Арк.	№ докум.	Підпис	Дата					

Тепер відправимо форму, й переглянемо результат

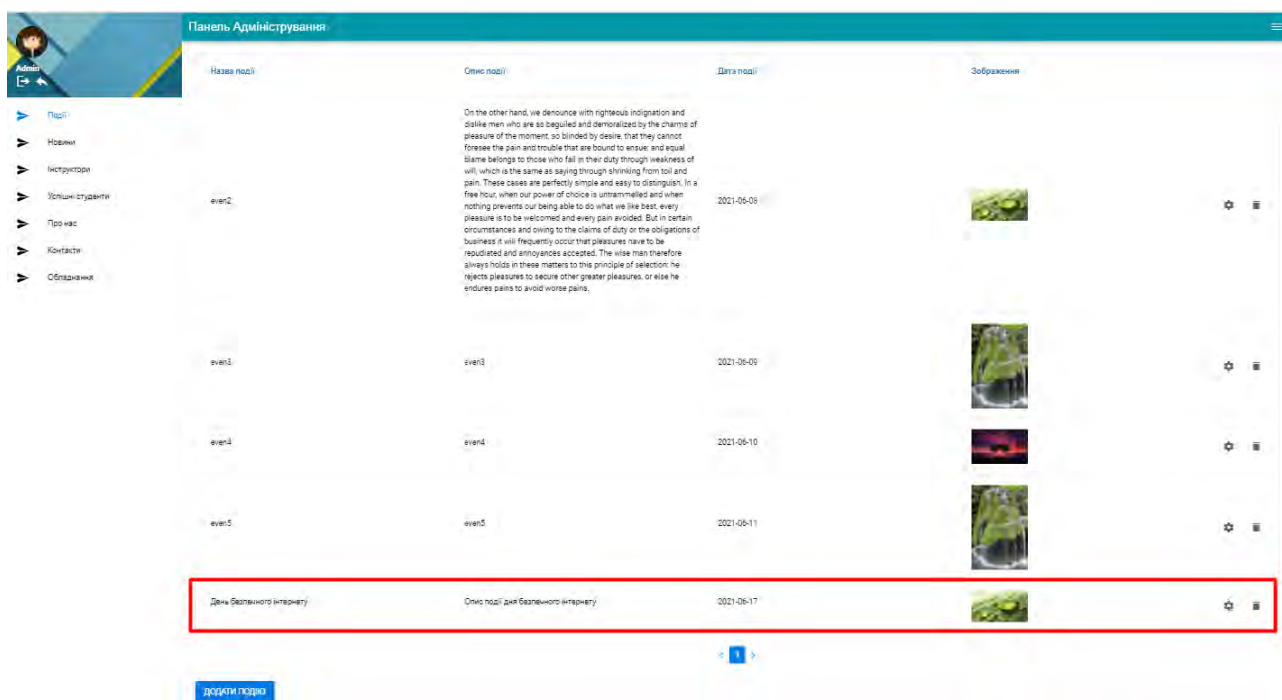


Рис. 3.6 – Нова подія в списку подій

Тепер спробуємо редагувати цю подію

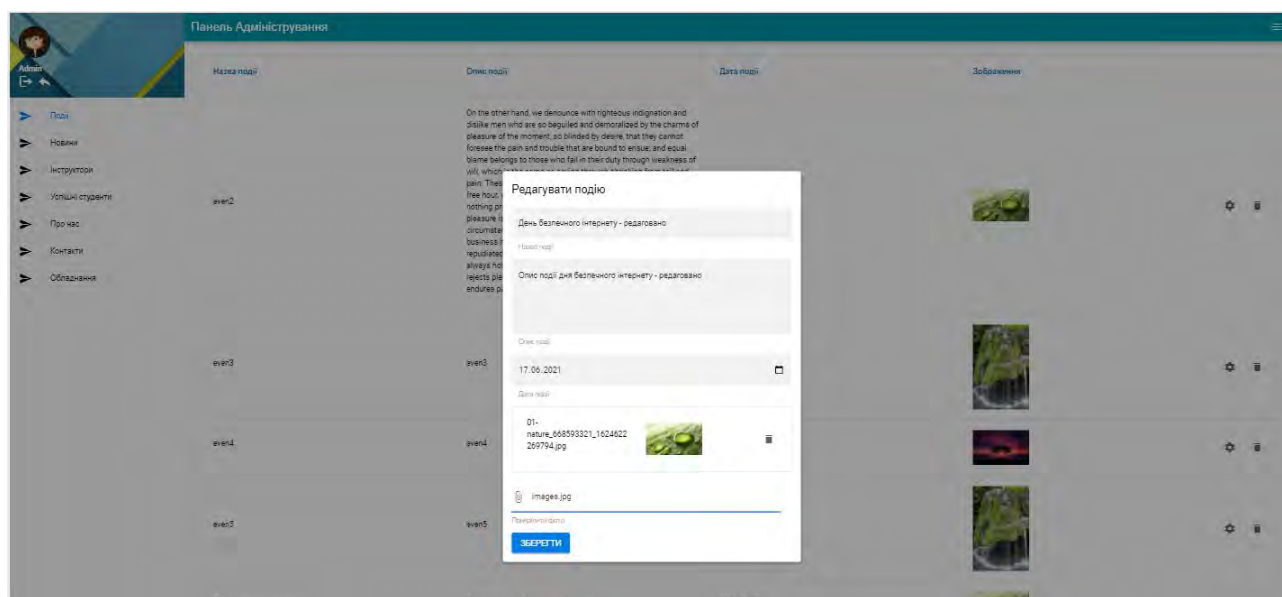


Рис. 3.7 – Форма редагування події

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				49
Змн.	Арк.	№ докум.	Підпис	Дата		

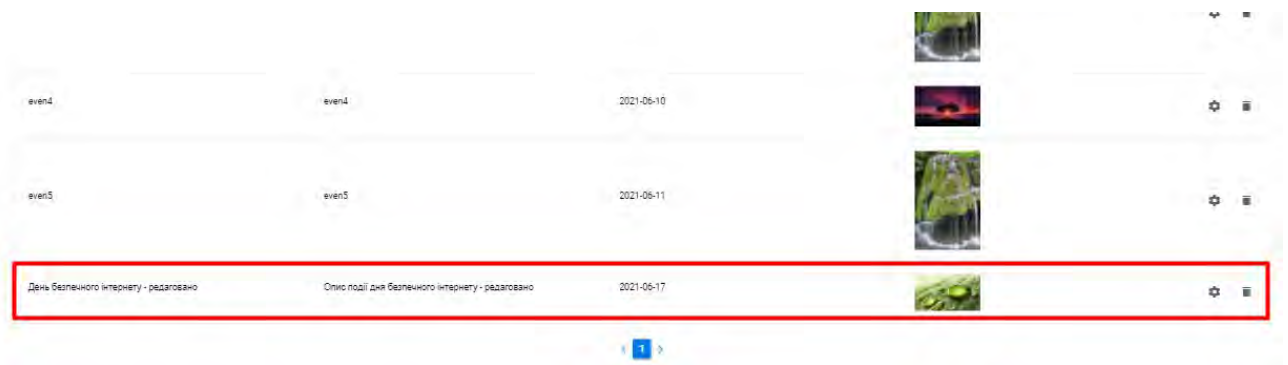


Рис. 3.8 – Відредагована форма події

Тепер переглянемо прикріплені зображення цієї події в режимі редагування

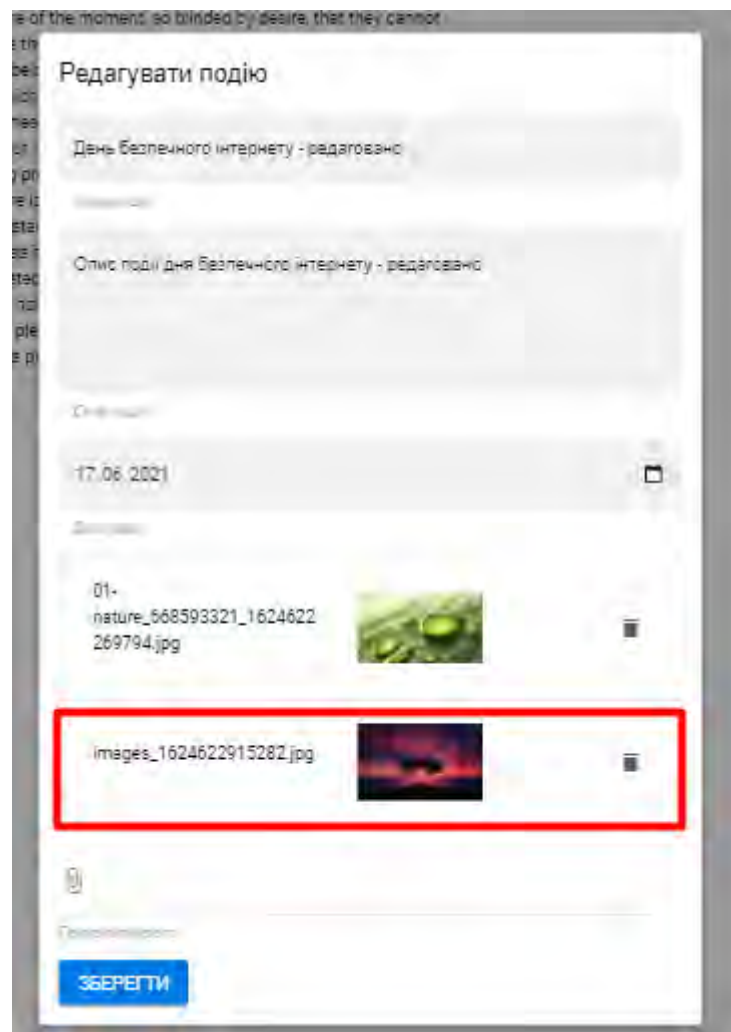


Рис. 3.9 – Відображення нового зображення прикріпленого до події

Тепер спробуємо видалити одну з подій

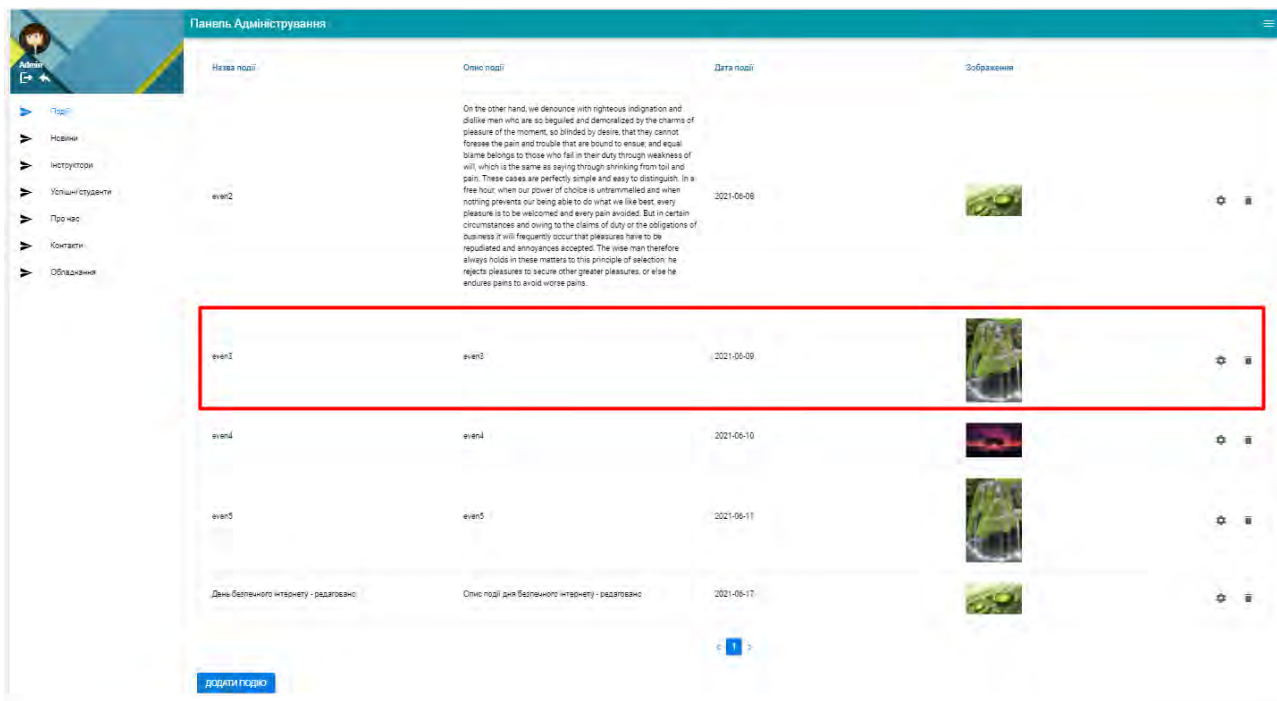


Рис. 3.10 – Подія, яку будемо видаляти

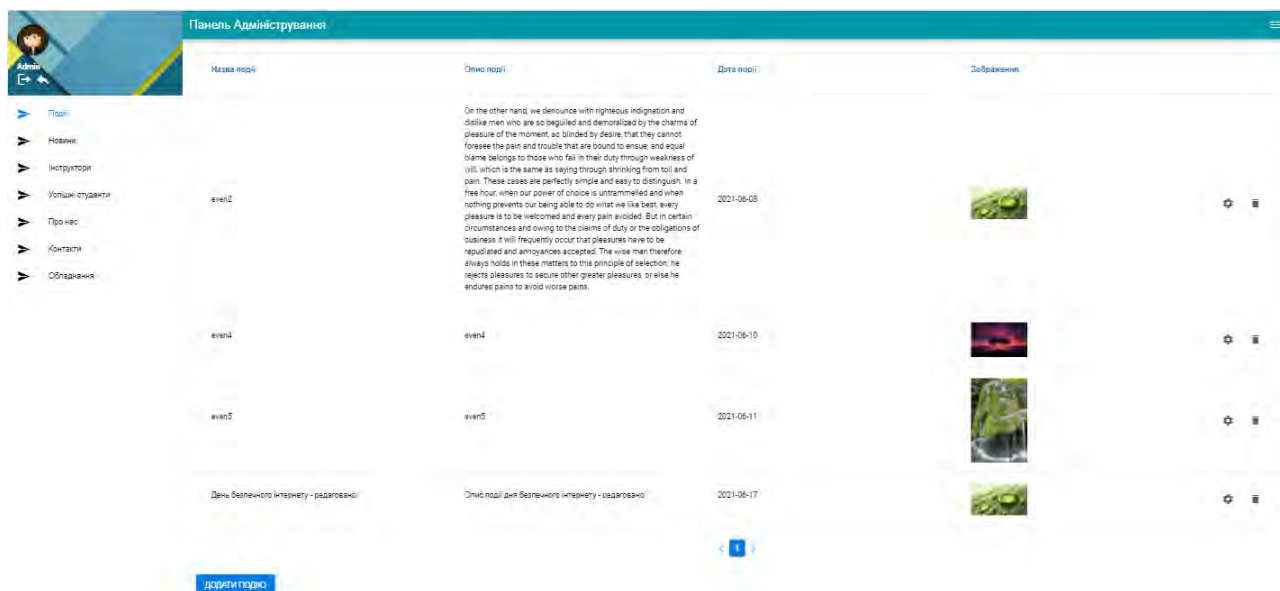


Рис. 3.11 – Список подій після видалення однієї події

Модулі “Про нас” й “Контакти” не дозволяють додавати більше ніж один запис, їх так само можна редагувати й видаляти. Кнопка додавання контакту чи тексту “Про нас” з’являється тільки тоді, коли дописів в цих модулях ще не існує. Спочатку переглянемо підготовлений опис, й спробуємо його відредагувати.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				51
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.12 – Опис в модулі “Про нас”

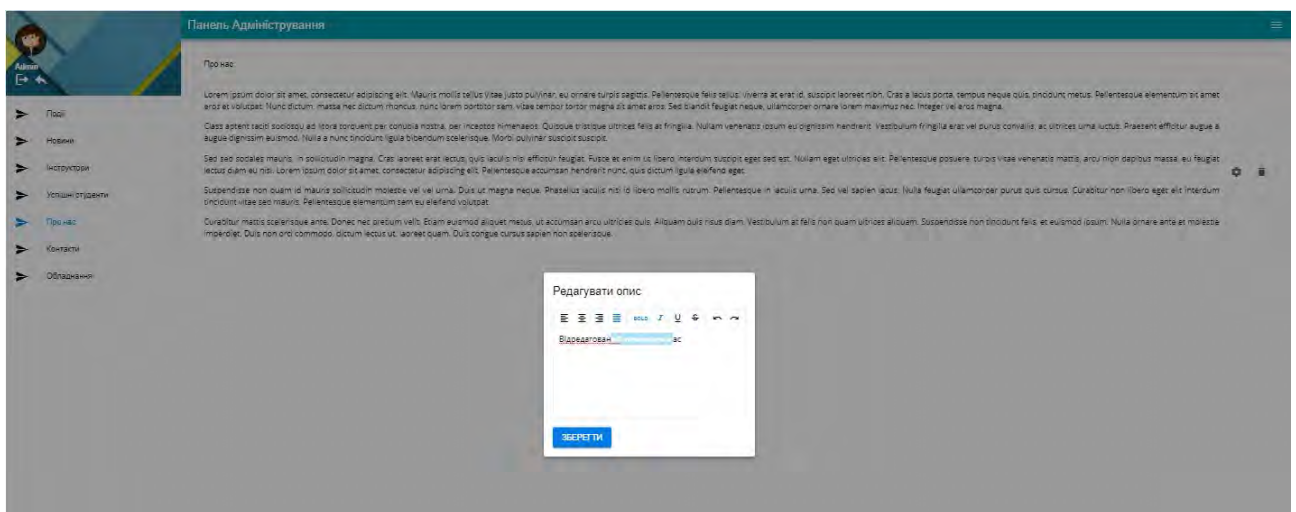


Рис. 3.13 – Редагування опису в модулі “Про нас”

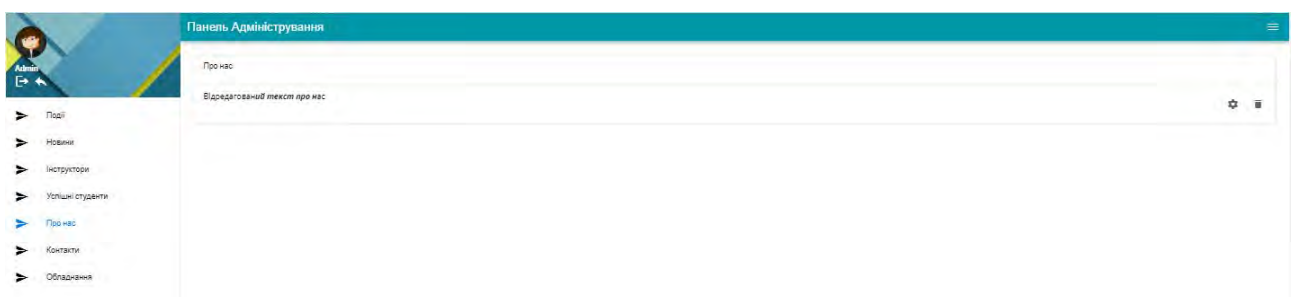


Рис. 3.14 – Відредагований опис в модулі “Про нас”

Тепер видалимо опис, й побачимо, що з’явиться кнопка додавання опису

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				52
Змн.	Арк.	№ докум.	Підпис	Дата		

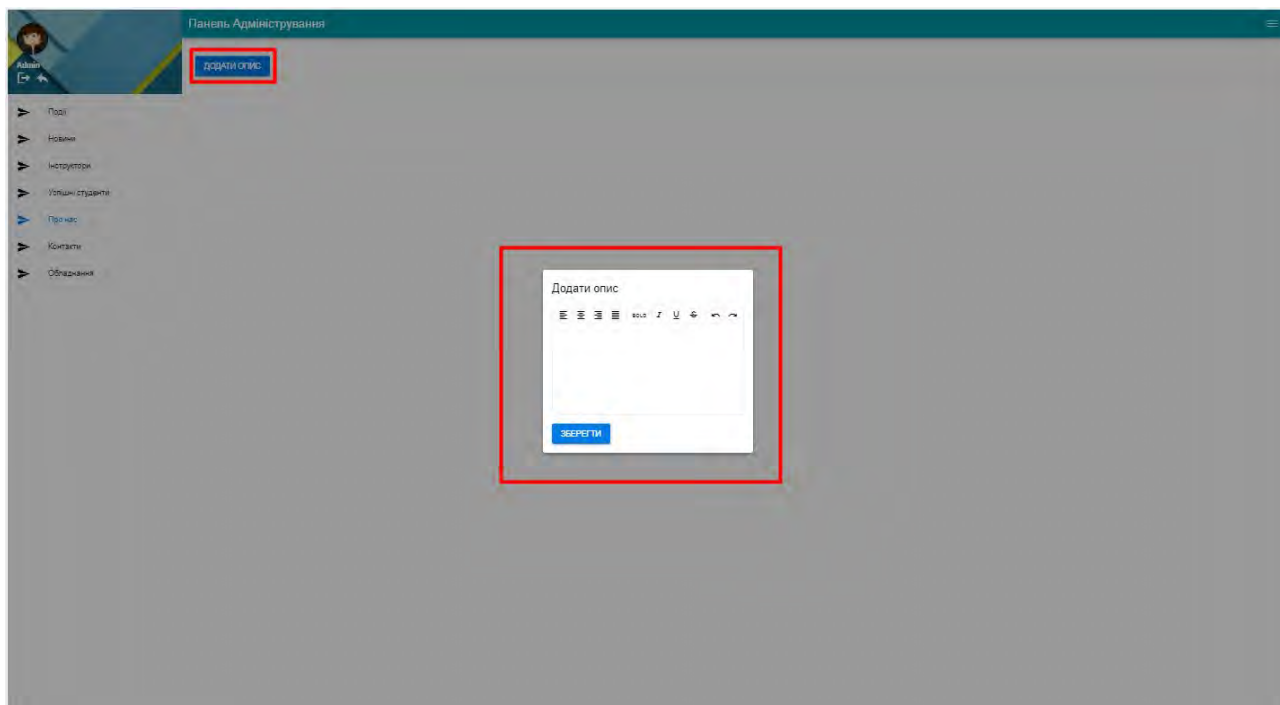


Рис. 3.15 – Додавання нового опису в модулі “Про нас”

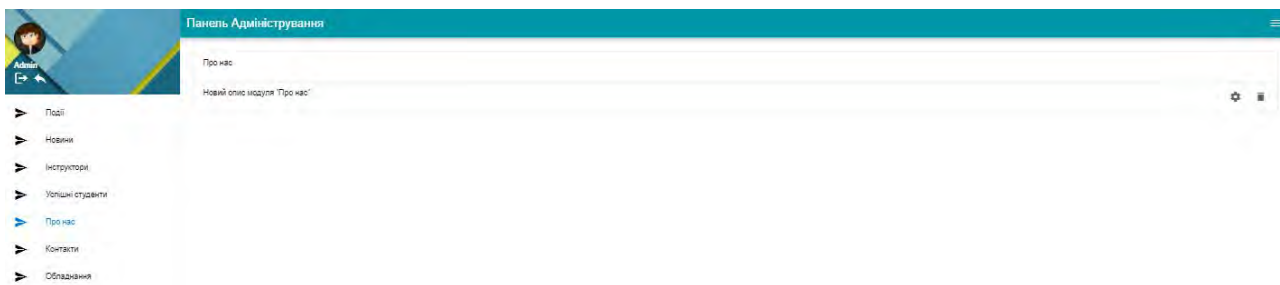


Рис. 3.16 – Новий опис в модулі “Про нас”

Решта модулів мають подібний функціонал, за винятком того, що форми додавання й редагування мають різні поля. Тому зараз розглянемо решту можливостей адміністраторської панелі, а саме: зміна логіну, зміна пароля в кабінеті адміністратора.



Рис. 3.17 – Кабінет адміністратора

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				53
Змн.	Арк.	№ докум.	Підпис	Дата		

Для того, щоб змінити логін потрібно ввести новий логін в поле Email, та натиснути кнопку зберегти логін

Рис. 3.18 – Редагування логіна

Рис. 3.19 – Редагування пароля

Тепер авторизуємось під оновленими даними користувача

Рис. 3.20 – Авторизація під оновленими даними користувача

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				54
Змн.	Арк.	№ докум.	Підпис	Дата		

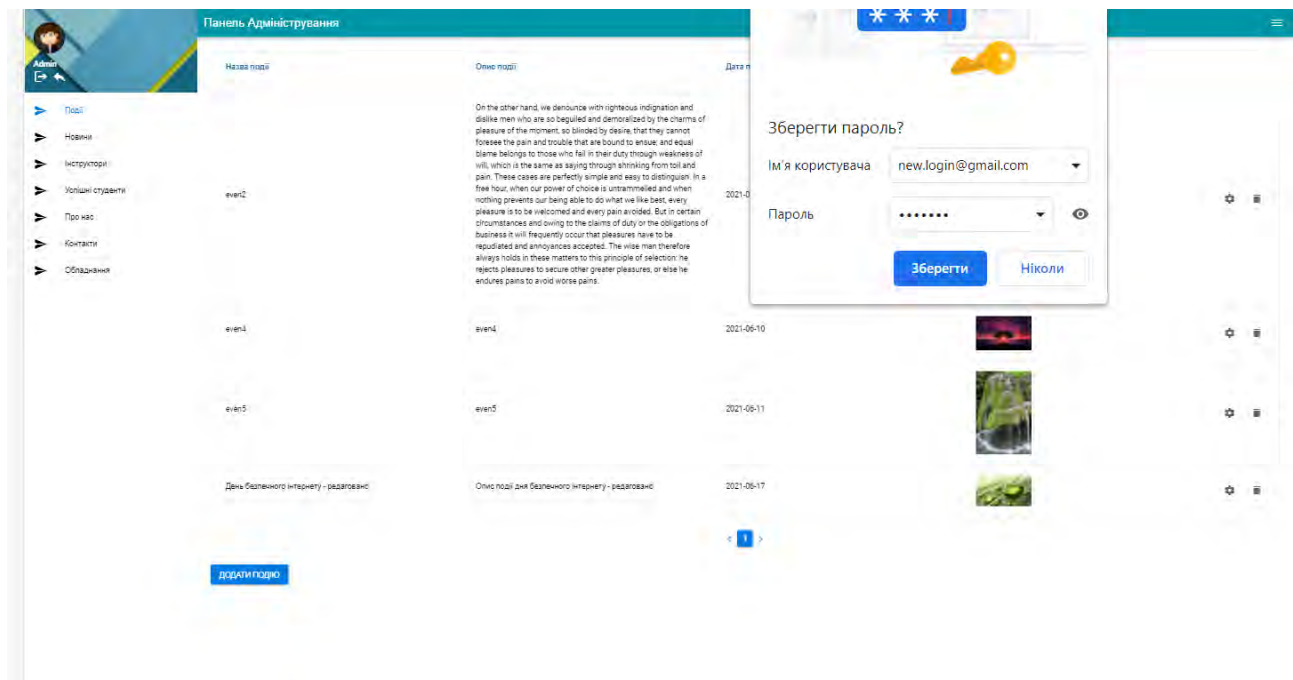


Рис. 3.21 – Успішний вхід під оновленими даними користувача

Тепер переглянемо користувацьку частину сайту

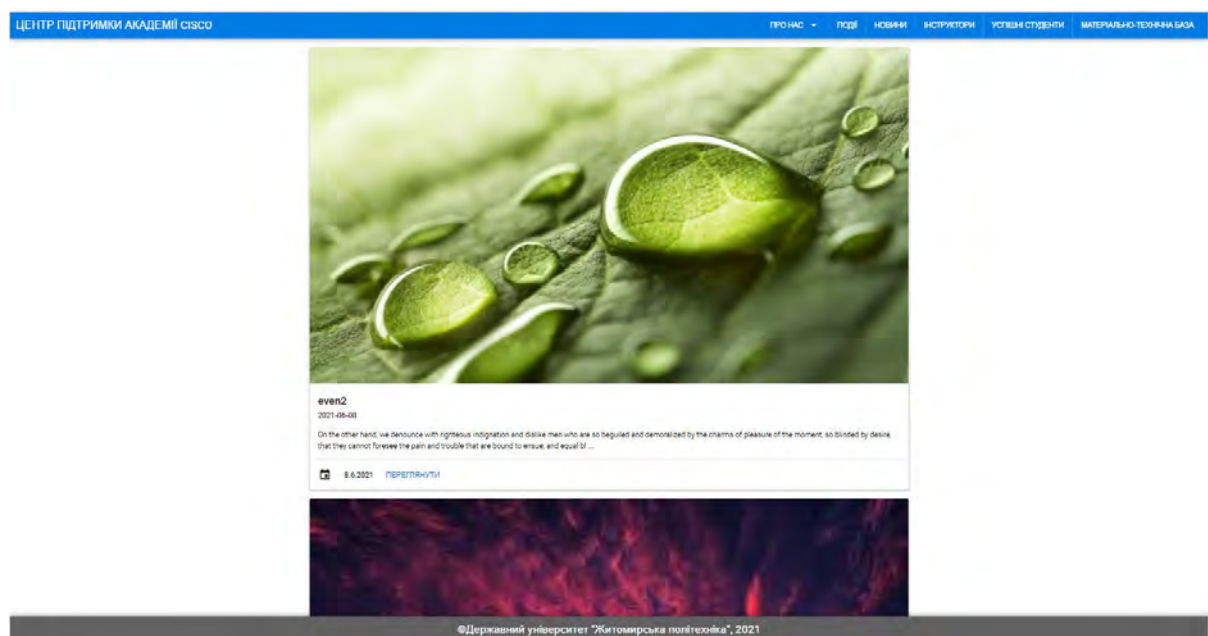


Рис. 3.22 – Домашня сторінка

Домашньою сторінкою користувацької частини є сторінка подій. Тепер переглянемо подію на окремій сторінці.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				55
Змн.	Арк.	№ докум.	Підпис	Дата		

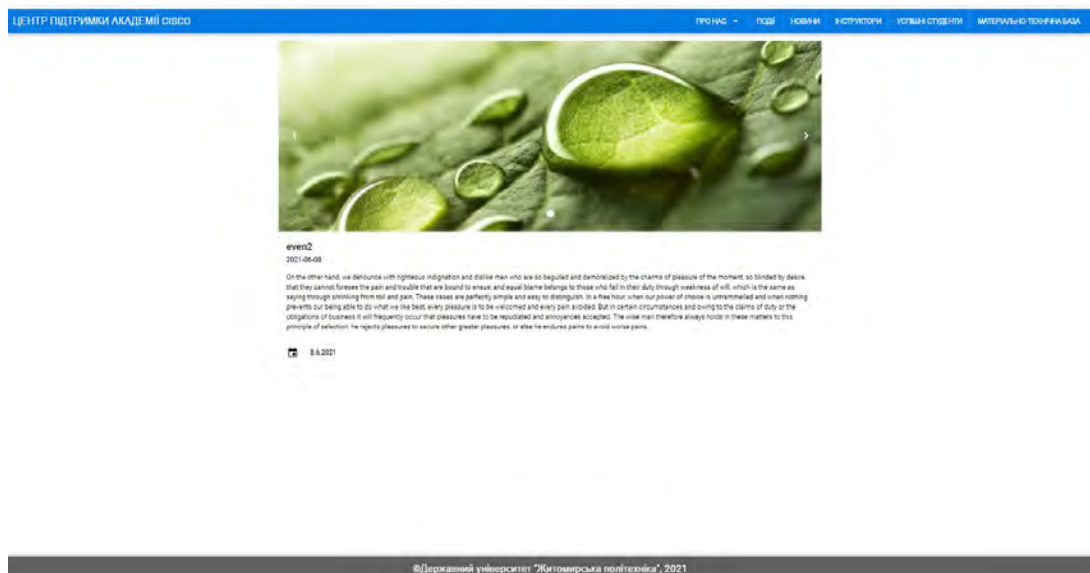


Рис. 3.23 – Сторінка події

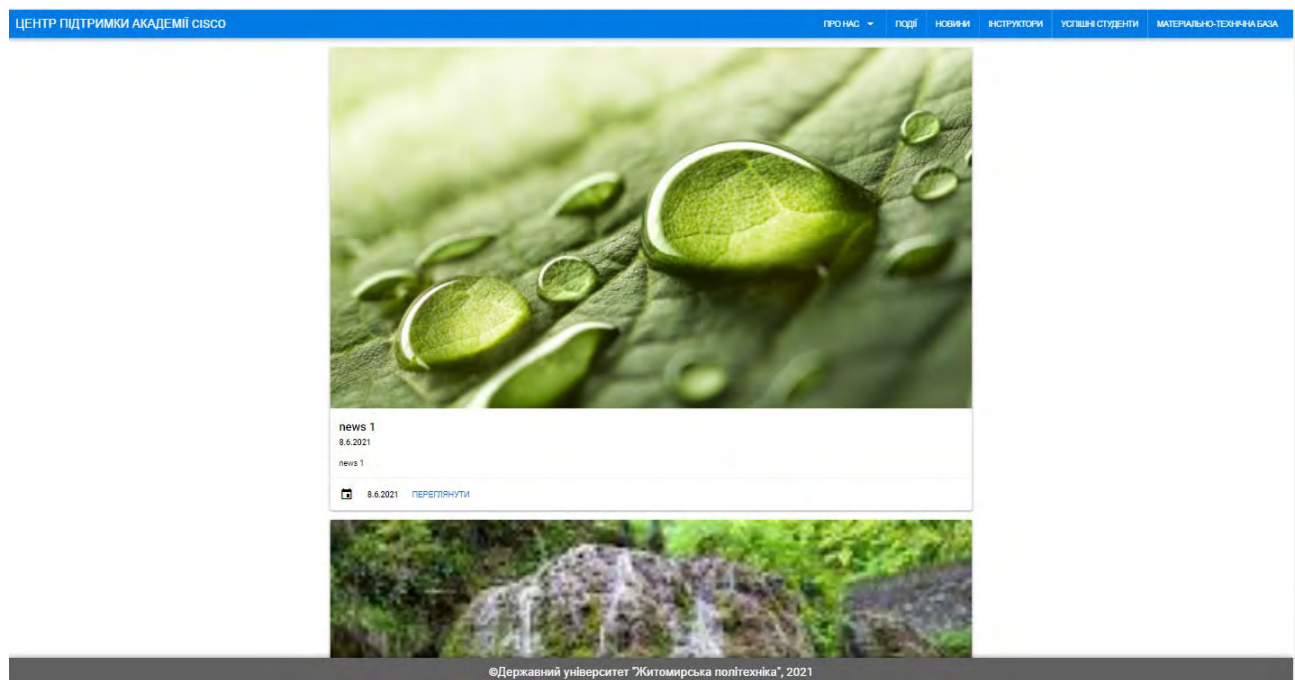


Рис. 3.24 – Сторінка новин

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				56
Змн.	Арк.	№ докум.	Підпис	Дата		

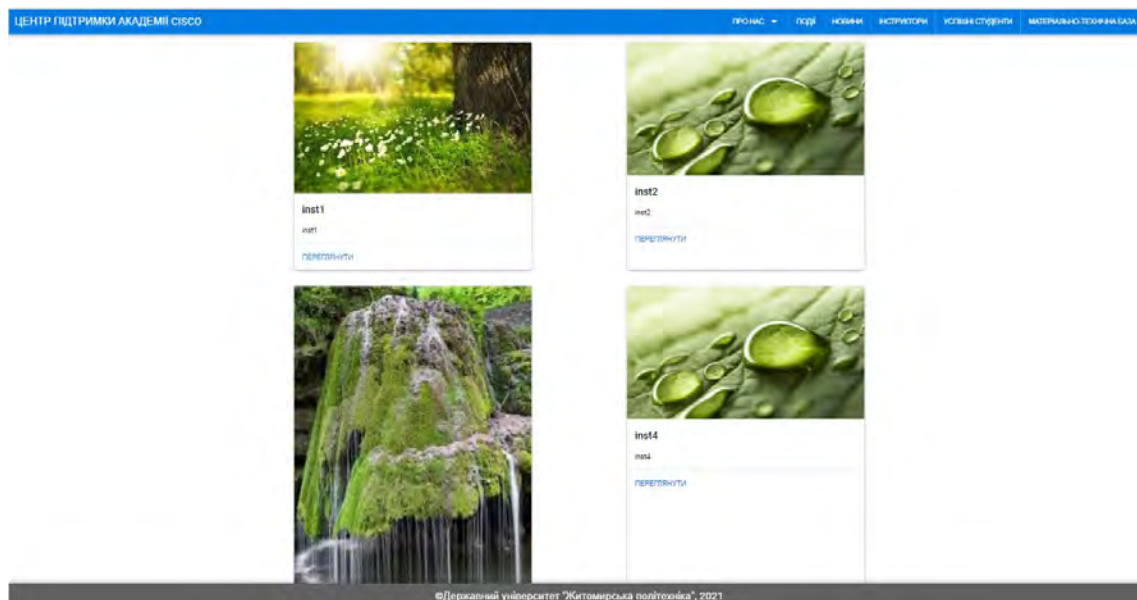


Рис. 3.25 – Сторінка інструкторів

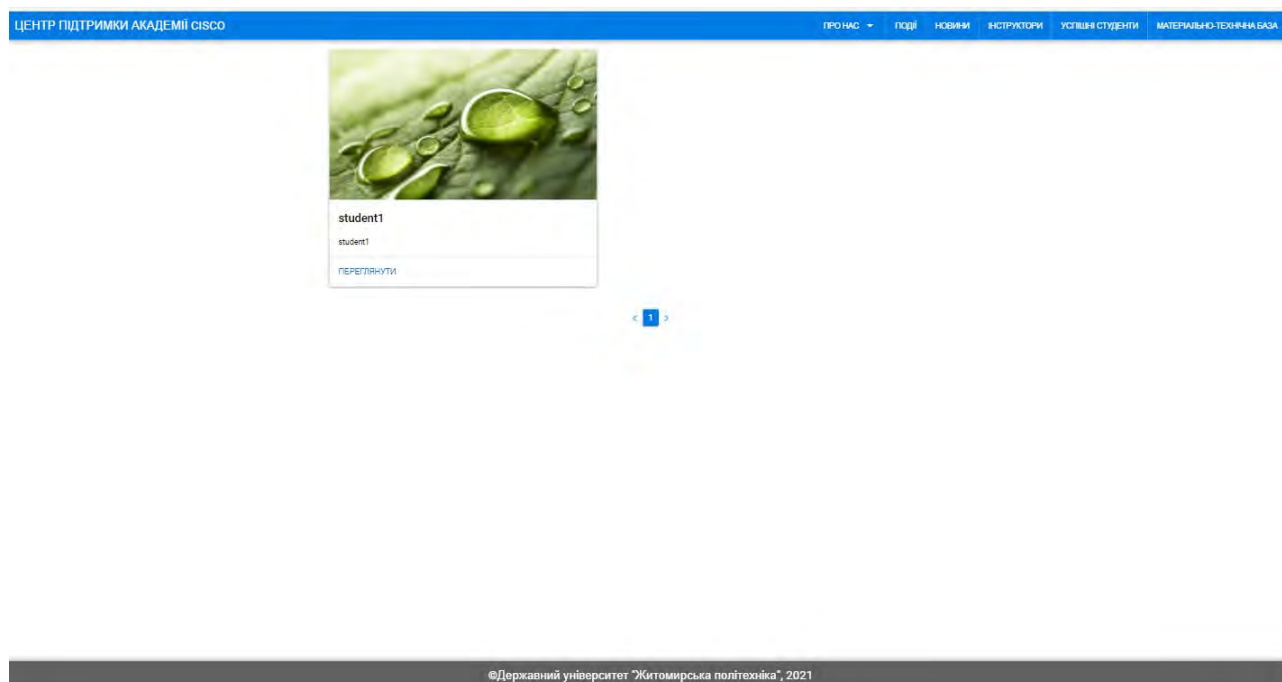


Рис. 3.26 – Сторінка успішних студентів

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				57
Змн.	Арк.	№ докум.	Підпис	Дата		

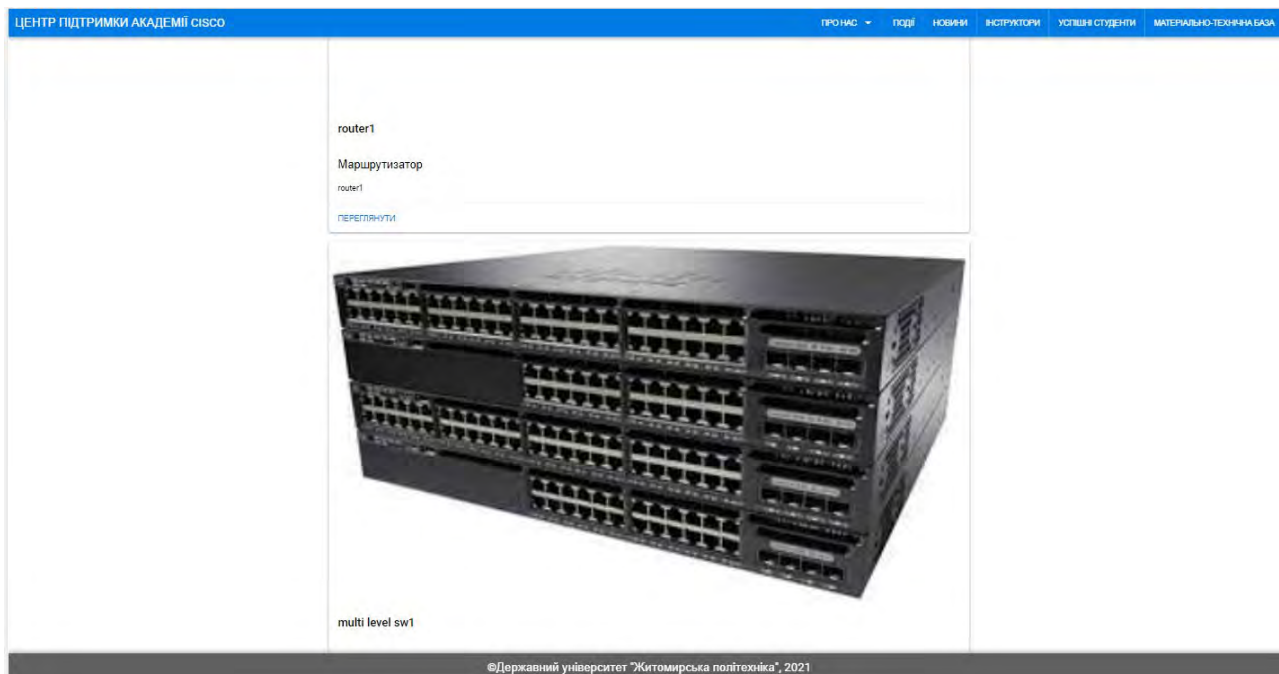


Рис. 3.27 – Сторінка обладнання

Висновки до третього розділу

В розділі було описано методологію використання створеного програмного додатку, та проведено його тестування

Опис роботи з додатком включає в себе текстову та візуальні частини, які пояснюють основні принципи роботи додатку, та опис елементів його інтерфейсу.

При проведенні тестування додаток був запущений на виконання у реальних умовах, при проведенні тестування було знайдено декілька помилок, які були виправлені.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				58
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В першому розділі було проаналізовано схожі за функціоналом програмні додатки, на основі порівняння з якими було обрано функціонал, та отримано теоритичний фундамент для подальшої розробки програмного додатку.

Відповідно до функціональних потреб було обрано інструментарій для розробки програмного додатку.

В другому розділі було проведено розробку загального і функціонального алгоритмів сайту, та описано принцип виконання основних методів роботи. Також представлено деталізовану розробку посторінково.

Розроблено загальний алгоритм функціонування роботи сайту. Була розроблена інтерфейсна частина, змодельовані вкладки сайту, та спроектовано програмний код сайту. В результаті було отримано готовий програмний продукт з повною функціональністю, який виконує всі поставлені задачі та вимоги.

В третьому розділі було описано методологію використання створеного програмного додатку, та проведено його тестування

Опис роботи з додатком включає в себе текстову та візуальні частини, які пояснюють основні принципи роботи додатку, та опис елементів його інтерфейсу.

При проведенні тестування додаток був запущений на виконання у реальних умовах, при проведенні тестування було знайдено декілька помилок, які були виправлені.

В результаті виконання кваліфікаційної роботи було розроблено сайт, що повністю відповідає поставленим вимогам.

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				59
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плюси і мінуси об'єктно-орієнтованого програмування [електронний ресурс] - 2015. Режим доступу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogo-programuvannya/>
2. Тим Кедлек. “Адаптивный дизайн. Делаем сайты для любых устройств” [електронний ресурс] - 2013. Режим доступу: <https://www.ozon.ru/context/detail/id/21098804/>
3. Выразительный JavaScript [електронний ресурс] - 2017. Режим доступу: https://legacy.gitbook.com/book/karmazzin/eloquentjavascript_ru/details
4. Володимир Г. Бази даних в інформаційних системах / Г. Володимир, І. Ігор., 2018. – 418 с.
5. Энсор Oracle проектирование баз данных / Энсор, Стивенсон Дейв; , Йен. - М.: ВНУ Киев, 2015. - 560 с.
6. Мюллер, Р.Дж. Базы данных и UML. Проектирование / Р.Дж. Мюллер. - М.: ЛОРИ, 2017. - 420 с.
7. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. Слинкин А.А. - Учебное пособие - М.: ДМК Пресс, 2000. - 432 с.
8. Никольский А. П. JAVASCRIPT НА ПРИМЕРАХ - СПб.: Наука и Техника, 2017. - 272 с.,ил.
9. Документація по фреймворку Mongoose. Режим доступу: <https://mongoosejs.com/docs/guide.html>
10. Документація по фреймворку Vue.js. Режим доступу: <https://vuejs.org/v2/guide/index.html>
11. UML діаграми. Режим доступу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
12. Паттерн MVVM. Режим доступу: https://professorweb.ru/my/WPF/documents_WPF/level36/36_5.php
13. Документація по бібліотеці Vuex.js. Режим доступу: <https://vuex.vuejs.org/guide/>

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				60
Змн.	Арк.	№ докум.	Підпис	Дата		

14. Документація по СУБД MongoDB. Режим доступу:
<https://docs.mongodb.com/>
15. Діаграми розгортання в UML. Режим доступу: <https://uk.photo-555.com/3332326-uml-deployment-diagram>
16. ASC/ITC Engineering Institute of the Zaporizhzhia National University.
Режим доступу: <https://netacad.znu.edu.ua/uk/>
17. Черкаський Державний Технологічний Університет. Режим доступу:
<https://chdtu.edu.ua/structure/cisco-academy>

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				61
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				62
Змн.	Арк.	№ докум.	Підпис	Дата		

Програмний код

Лістинг компоненту Login:

```

<template>
  <q-form
    @submit.prevent="authenticate"
    class="q-gutter-md full-width"
  >
    <p class="text-h4 text-center">Авторизація</p>
    <q-input
      filled
      label="Логін"
      v-model="$v.login.$model"
      :error="(!$v.login.email || !$v.login.required) && $v.login.$dirty"
      :error-message="checkLogin"
    />

    <q-input
      filled
      type="password"
      v-model="password"
      label="Пароль"
      lazy-rules
      :rules="[
        val => val !== null && val !== '' || 'Поле обов'язкове для
заповнення',
      ]"
    />

    <div>
      <q-btn class="q-mr-sm" label="Увійти" type="submit" color="primary"/>
      <q-btn
        class="q-mr-sm"
        label="Відновити пароль"
        color="primary"
        @click="$router.push('auth/reset')"
      />

      <q-btn
        label="Повернутись на головну"
        color="primary"
        @click="$router.push('/')"
      />
    </div>
  </q-form>
</template>

<script>
import {email, required} from 'vuelidate/lib/validators';
import {mapActions} from 'vuex';

export default {
  name: "Login",
  data() {
    return {
      login: null,
      password: null,

      accept: false,
    }
  }
}

```

```

    },
    validations: {
      login: {required, email},
      password: {required}
    },
    computed: {
      checkLogin() {
        if (!this.$v.login.required) {
          return 'Поле обов\'язкове для заповнення';
        } else if (!this.$v.login.email) {
          return 'Некоректний email';
        } else {
          return "Все коректно";
        }
      }
    },
    methods: {
      ...mapActions("authentication", ["auth"]),
      async authenticate() {
        let formData = new FormData();
        formData.set('login', this.login);
        formData.set('password', this.password);

        const { token } = await this.auth(formData);

        if (token) {
          this.$router.push("/admin");
        }
      }
    },
  },
</script>

<style scoped>
  p.text-h4 {
    font-family: "Roboto", sans-serif;
  }
</style>

```

Лістинг компоненти Reset Password:

```

<template>
  <q-form
    @submit.prevent="onResetPassword"
    class="q-gutter-md full-width"
  >
    <p class="text-h4 text-center">Відновлення паролю</p>
    <q-input
      filled
      label="Логін"
      v-model="$v.login.$model"
      :error="(!$v.login.email || !$v.login.required) && $v.login.$dirty"
      :error-message="checkLogin"
    />

    <div>
      <q-btn class="q-mr-sm" label="Відновити пароль" type="submit"
        color="primary"/>

      <q-btn
        label="Повернутись на головну"
        color="primary"
      />
    </div>
  </q-form>

```

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        @click="$router.push('/')"/>
    </div>
</q-form>
</template>

<script>
import {email, required} from 'vuelidate/lib/validators';
import {mapActions} from 'vuex';

export default {
  name: "Login",
  data() {
    return {
      login: null,
      accept: false,
    }
  },
  validations: {
    login: {required, email},
    password: {required}
  },
  computed: {
    checkLogin() {
      if (!this.$v.login.required) {
        return 'Поле обов\'язкове для заповнення';
      } else if (!this.$v.login.email) {
        return 'Некоректний email';
      } else {
        return "Все коректно";
      }
    }
  },
  methods: {
    ...mapActions('authentication', ['resetPassword']),
    async onResetPassword() {
      let formData = new FormData();
      formData.set('email', this.login);

      let { success } = await this.resetPassword(formData);

      if (success) {
        this.$router.push("/auth");
      }
    }
  },
}
</script>

<style scoped>
p.text-h4 {
  font-family: "Roboto", sans-serif;
}
</style>

```

		Войналович В.Ю			КІ.КР.Б – 123 – 02 – ПЗ	Арк.
		Головня О.С				65
Змн.	Арк.	№ докум.	Підпис	Дата		