

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 «магістр» за
спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення») на
тему:

**«Розробка web-орієнтованої інформаційно-
аналітичної системи для управління та аналізу
фінансових внесків у благодійність»**

Виконала студентка групи ІПЗм-23-2
ГЛОБІНА Анастасія Володимирівна

Керівник роботи:
ВАКАЛЮК Тетяна Анатоліївна

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2024

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

«01» жовтня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **Глобіна Анастасія Володимиріна**

Керівник роботи: **ВАКАЛЮК Тетяна Анатоліївна**

Тема роботи: **«Розробка web-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність»**, затверджена Наказом закладу вищої освіти від «01» жовтня 2024 р. № 520/с.

Вихідні дані для роботи: аналіз даних про фінансові внески у благодійність; оптимізація процесів у фінансовій діяльності благодійних фондів.

Предметом дослідження є процеси управління, аналізу та оптимізації фінансових внесків у благодійність за допомогою web-орієнтованих інформаційно-аналітичних систем, які забезпечують автоматизацію збору, обробки та аналізу даних, а також інструменти для виявлення аномалій і прийняття обґрунтованих управлінських рішень у благодійній діяльності. Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Вакалюк Т.А.	03.10.2024р.	03.10.2024р.
2	Вакалюк Т.А.	03.10.2024р.	03.10.2024р.
3	Вакалюк Т.А.	03.10.2024р.	03.10.2024р.

РЕФЕРАТ

Кваліфікаційна робота містить 80 сторінок, 10 таблиці, 26 рисунків, список використаних джерел із 17 найменувань, 3 додатки.

Метою випускної кваліфікаційної роботи є розробка web-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність ЗСУ, що дозволить прогнозувати та оцінювати ситуацію з потребами для наших військових. В роботі розроблено базу даних, що містить інформацію про потреби військових відфільтровані за категоріями, та інформацію про грошові надходження у якості адресної допомоги. Реалізовано можливість аналізувати динаміку надходжень та потреб за різними категоріями та країнами. Інтерфейс програми користувача дозволяє формувати web-орієнтований інтерфейс для збору грошової допомоги.

В роботі запропоновано підхід до аналізу потреб та надходжень. Система реалізована за допомогою мови програмування PHP.

Ключові слова: МАТИМАТИЧНА СТАТИСТИКА, АНАЛІТИКА, БЛАГОДІЙНІСТЬ, ДОНАТИ, БАЗА ДАНИХ.

ABSTRACT

The master's thesis contains 67 pages, 10 tables, 26 figures, a list of used sources of 25 items, 2 appendices.

The purpose of the final qualification work is to develop a web-oriented information and analytical system for managing and analyzing financial contributions to the charity of the Armed Forces of Ukraine, which will allow predicting and assessing the situation with the needs of our military. The work has developed a database containing information on the needs of the military filtered by categories, and information on cash receipts as targeted assistance. The ability to analyze the dynamics of receipts and needs by different categories and countries has been implemented. The user interface of the program allows you to form a web-oriented interface for collecting cash assistance.

The work proposes an approach to analyzing needs and receipts. The system is implemented using the PHP programming language.

Keywords: MATHEMATICAL STATISTICS, ANALYTICS, CHARITY, DONATIONS, DATABASE.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. МОДЕЛЬ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ.....	11
1.1. Сучасні інформаційно-аналітичні системи управління та аналізу благодійних внесків.....	11
1.2 Інформаційні процеси, обґрунтування вибору засобів моделювання та програмного середовища.....	18
1.3. Постановка технічного завдання.....	20
Висновки до розділу 1.....	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ.....	25
2.1. Вибір технологій для розробки системи.....	25
2.2. Створення інтерфейсу користувача.....	26
2.3. Побудова аналітичної системи.....	34
2.4. Проєктування модулів системи.....	42
Висновки до розділу 2.....	46
РОЗДІЛ 3. ВПРОВАДЖЕННЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ.....	47
3.1. Використання аналітичної системи для управління внесками.....	47
3.2. Порівняння показників на основі аналітичних даних.....	49
3.3.Оцінка ефективності інформаційно-аналітичної системи.....	
Висновки до розділу 3.....	62
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	– база даних.
ІС	– інформаційна система
СУБД	– система управління базами даних
CRUD	– create, read, update, delete. створити, прочитати, оновити, видалити
ЗСУ	– збройні сили України
JS	Javascript (мова програмування)
ІІІ	Штучний інтелект

ВСТУП

У сучасних умовах, коли національна безпека є однією з найважливіших складових стабільності країни, підтримка Збройних Сил України (ЗСУ) набуває особливого значення. Фінансові внески у благодійність ЗСУ стають важливим джерелом забезпечення армії необхідними ресурсами, необхідними для виконання стратегічних завдань у сфері оборони та безпеки.

Програмне забезпечення для благодійності – це цифрові інструменти, які дозволяють некомерційним організаціям оптимізувати пожертвування, управляти кампаніями та покращувати стосунки з донорами. Це програмне забезпечення є єдиною платформою для збору коштів, відстеження надходжень та забезпечення прозорості, вирішуючи проблеми, з якими стикаються традиційні методи.

У сучасному світі технологій, програмне забезпечення для збору благодійних внесків вийшло на новий рівень, забезпечуючи автоматизацію, прозорість і ефективність управління фінансовими ресурсами. Такі програмні забезпечення роблять пожертвування простішими, що дозволяє людям вносити пожертвування по всьому світу в кілька кліків.

Сучасне програмне забезпечення для благодійності пропонує звіти та оновлення у режимі реального часу, дозволяючи волонтерам та благодійним організаціям відстежувати ефективність своїх пожертвувань, а автоматизація таких процесів, як управління даними та зв'язок, знижує накладні витрати, дозволяючи некомерційним організаціям ефективніше розподіляти ресурси.

Програмне забезпечення для благодійних організацій дає змогу некомерційним організаціям розширювати свою діяльність за кордоном. Завдяки багатомовній підтримці, спеціалізованим кампаніям та регіональним платіжним системам ці інструменти полегшують зв'язок із донорами по всьому світу. У регіонах, де традиційні банківські системи обмежені, програмне забезпечення для благодійності заповнює цю прогалину, дозволяючи людям робити

пожертвування через мобільні гаманці та цифрові платіжні системи. Така інклюзивність сприяла значному внеску країн, що розвиваються. Довіра має життєво важливе значення у благодійній діяльності, а програмне забезпечення для благодійності сприяє прозорості, надаючи докладну інформацію про те, як використовуються кошти.

У зв'язку з цим, розробка ефективної web-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність ЗСУ стає важливою складовою стратегії забезпечення армії необхідними ресурсами. Ця система має на меті оптимізувати процеси збору, обробки та аналізу фінансової інформації, пов'язаної з пожертвами на підтримку Збройних Сил України.

Мета дослідження полягає в розробці та впровадженні web-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність Збройних Сил України (ЗСУ). Ця система спрямована на оптимізацію процесів збору, обробки та аналізу фінансових даних, пов'язаних з жертвами на підтримку української армії.

Завданням дослідження є розробка ефективної web-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків на підтримку Збройних Сил України. У рамках дослідження необхідно провести аналіз існуючих інформаційних систем та технологій автоматизації процесів управління благодійними жертвами, визначити вимоги до розроблюваної системи та розробити її архітектуру. Важливими етапами є створення бази даних для структурованого зберігання інформації, реалізація модулів збору, збереження, аналізу та візуалізації фінансових даних, а також інтеграція системи з платіжними платформами. Особливу увагу приділяється тестуванню системи, оцінці її ефективності та розробці рекомендацій щодо впровадження в роботу благодійних фондів. Завдання також включає створення пропозицій для подальшого вдосконалення, таких як розробка модуля прогнозування та інтеграція з іншими аналітичними інструментами, що дозволить оптимізувати

процеси управління фінансовими ресурсами і забезпечити прозорість та результативність благодійної діяльності.

Об'єкт дослідження цієї роботи - це web-орієнтована інформаційно-аналітична система для управління та аналізу фінансових внесків у благодійність Збройних Сил України (ЗСУ). Основна увага приділяється розробці, впровадженню та функціонуванню системи, що забезпечує збір, зберігання, обробку та аналіз фінансових даних, пов'язаних з пожертвами на підтримку української армії.

Предмет дослідження полягає в процесі розробки та використанні цієї інформаційно-аналітичної системи. Основні аспекти, які досліджуються, включають розробку інтерфейсу користувача, реалізацію функціональності для збору та обробки фінансових даних, аналіз ефективності використання фінансових ресурсів, а також забезпечення безпеки та конфіденційності даних.

Практичне значення роботи полягає у створенні інструменту, який може бути використаний благодійними фондами та волонтерськими організаціями для прозорого та ефективного управління пожертвами на підтримку Збройних Сил України. Система надає можливість автоматизувати процеси збору та обробки даних, забезпечити структуроване зберігання інформації, створювати візуалізовані аналітичні звіти та прогнозувати майбутні надходження. Її впровадження сприятиме підвищенню довіри донорів, зниженню ризиків помилок та ефективнішому плануванню благодійних кампаній.

В контексті нинішніх викликів та потреб суспільства, ефективна система управління фінансовими внесками у благодійність ЗСУ є важливим інструментом забезпечення армії необхідними ресурсами для успішного виконання своїх завдань. Розробка та впровадження цієї системи відображає загальне прагнення до зміцнення обороноздатності та безпеки країни.

РОЗДІЛ 1. МОДЕЛЬ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ WEB-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

1.1. Сучасні інформаційно-аналітичні системи управління та аналізу благодійних внесків.

Інформаційно-аналітичні системи (ІАС) в сучасному світі стали невід'ємною частиною управління благодійними внесками. Вони дозволяють автоматизувати процеси збору, аналізу та управління фінансовими ресурсами, що надаються на благодійні цілі. Такі системи не лише спрощують роботу організацій, але й підвищують рівень довіри серед донорів, завдяки прозорості та доступності інформації.

У контексті благодійної діяльності інформаційно-аналітичні системи (ІАС) виконують ряд важливих функцій, що сприяють оптимізації управління фінансовими ресурсами та підвищенню ефективності організацій. Передусім, вони забезпечують обробку та збереження даних про благодійні внески, включаючи суми, джерела та напрями використання, що дозволяє організувати фінансові потоки, відстежувати рух коштів і забезпечувати їх правильний розподіл. Однією з ключових переваг ІАС є здатність здійснювати глибокий аналіз даних, що дозволяє виявляти тренди, оцінювати ефективність кампаній, аналізувати активність донорів та їх географічне розташування, а також прогнозувати майбутні результати.

Прозорість і автоматизація звітності є ще однією важливою складовою таких систем. Вони дозволяють волонтерам і громадськості отримувати детальну інформацію про використання коштів, що підвищує рівень довіри до організації та сприяє залученню додаткових ресурсів. Крім того, сучасні ІАС інтегруються з платіжними системами, соціальними мережами та іншими платформами, що значно спрощує процес збору коштів.

Використання сучасних технологій і платформ для збору благодійних внесків дозволяє організаціям оперативно обробляти дані в реальному часі,

зберігати інформацію про волонтерів, моніторити платежі та управляти різними видами кампаній. Такі системи також забезпечують накопичення аналітичних даних, які можуть бути використані для прийняття стратегічних управлінських рішень, підвищуючи ефективність і результативність роботи благодійних організацій.

Багато благодійних платформ також забезпечують інтеграцію з популярними платіжними системами, такими як **PayPal, Stripe, Visa**, що дозволяє волонтерам та благодійним організаціям з усього світу легко робити внески. Система забезпечує автоматичне генерування квитанцій та підтверджень для кожного платежу, що також підвищує рівень прозорості [10].

Ще одним важливим фактором є те, що ІАС дозволяють використовувати різноманітні аналітичні інструменти, включаючи **BI-системи (Business Intelligence)** для візуалізації даних, що дає змогу керівникам організацій швидко оцінювати ефективність кампаній, зрозуміти, які види допомоги найбільше цікавлять волонтерів, та побудувати найбільш вдалі стратегії, щоб досягти найкращих результатів у залученні коштів.

Теорія і практика управління породили низку різноманітних підходів, методів і моделей, націлених підвищенням ефективності. Ці методи та моделі, у свою чергу, зумовили появу та розвиток різноманітних аналітичних інформаційних систем. Такі системи займають певне становище в інформаційній інфраструктурі та відіграють важливу роль у процесах управління [4].

Ці обставини змушують використовувати наявні нині розвинені програмно-технічні засоби. Широке та ефективне застосування цих засобів стало одним із факторів виживання та успіху благодійної діяльності в сучасних умовах. Набули широкого поширення автоматизовані інформаційні системи – так останніми роками частіше називають інформаційні системи, маючи на увазі, що без автоматизації їх просто неможливо уявити.

Проблема аналізу вихідної інформації для прийняття рішень виявилася настільки серйозною, що з'явився окремий напрямок чи вид інформаційних

систем — інформаційно-аналітичні системи, під якими розуміють комплекс апаратних, програмних засобів, інформаційних ресурсів, методик, які використовуються для забезпечення автоматизації аналітичних робіт з метою обґрунтування прийняття управлінських рішень та інших можливих застосувань. перетворення великих обсягів сильно деталізованих даних в керовані та вивіряються оглядові звіти, які можуть бути використані для прийняття обґрунтованих рішень [3].

Під час розробки модельно-програмного комплексу вирішено такі задачі:

- розроблено форми введення та виведення даних, схем, діаграм, графіків та таблиць, необхідних для аналізу;
- перевірено та апробовано алгоритми розрахунків та аналізу;
- створено інформаційні бази, що містять вихідні дані щодо аналізу
- розроблено технологію та алгоритми вирішення завдань;
- забезпечені зручні форми виведення інформації для користувачів;
- розроблено методичні рекомендації для користувача щодо роботи з програмою.

Таким чином, можна резюмувати, що в країні з'явилася об'єктивна необхідність у впровадженні інформаційно-аналітичних систем у різні сфери діяльності, зокрема при зборі та розподіленні благодійних внесків, де потрібна підтримка стратегічного планування та досягнення високоефективних управлінських рішень.

На даний момент на ринку благодійності існує багато інформаційних систем, що збирають благодійні внески за різними категоріями. Наведемо приклади інформаційно-аналітичних систем та розглянемо три сервіси, які пропонують українські банки, які можна використовувати для збору коштів:

1. "Банка" від monobank
2. "Конверт" від Приватбанку
3. «Скринька» від О. Bank

Сервіс «Банку», що з'явився три роки тому, запропонований командою

Олега Гороховського, став першим і унікальним у своєму роді, т.к. вона, крім традиційних способів накопичення (шляхом списання зі своїх карт чи то відсотка від купівлі, чи округлення залишку) запропонувала всім охочим можливість збору донатів.

Після створення «Банки» можна згенерувати посилання для охочих її поповнити та розмістити її у соціальних мережах чи месенджерах типу Viber, Telegram тощо [6].

Перейшовши за цим посиланням, кожному бажаючому пропонується меню, де можна вказати суму поповнення, ім'я та прізвище (за бажанням), а також вибрати спосіб поповнення: GooglePay, monoPay (якщо у того, хто поповнює Банку, є картка monobank) або просто оплата карткою (у разі знадобиться ввести основні реквізити карти — номер, термін дії і CVV/CVC-код).

monobank

Віталій С. збирає
ТЕРМІНОВО ЗСУ
IBAN UA 40 322001 00000 2620 8346 1594 01

Збір на:
- 30 пікапів та 3 вантажівки 4x4
- 40 дронів
- 3 ОШБ, 15 БрОП, 23 ОШБ, 31 ОШБ, 33 ОШБ
Наша ціль: 28 мільйонів гривень до 31.07

👉 Закриваємо збір - рандомно розіграємо котедж у Львові серед тих, хто задонатить 250 грн. Кожен донат = 1 шанс!

Накопичено 232 125.97 ₴ | Ціль 28 000 000 ₴

Сума поповнення 🇺🇦

0 ₴

+100 ₴ +500 ₴ +1 000 ₴

Ім'я або коментар (необов'язково)

mono Pay

G Pay

📱 Оплатити карткою

Рис.1.1. Спосіб оплати

Після початку повномасштабного вторгнення росії в Україну саме «Банки» від monobank стали головним (і спочатку єдиним) інструментом для збору грошей на ЗСУ волонтерами, військовими та просто небайдужими українцями.

Саме її використання для збору донатів зробило її знаменитою, без перебільшення у всьому світі, т.к. поповнити її можна було з будь-якої точки земної кулі, з будь-якої карти будь-якого банку світу.

Після закінчення збору Банку «розбивається», а кошти з неї переміщуються на карту того, хто створив Банку, і надалі використовуються для цієї мети.

Звичайно ж, банки не могли довго «терпіти» наявність у конкурента такого потужного та унікального сервісу, тому вже у 2023 році Приватбанк розробив дуже схожий сервіс для збору коштів та назвав його «Конвертом».

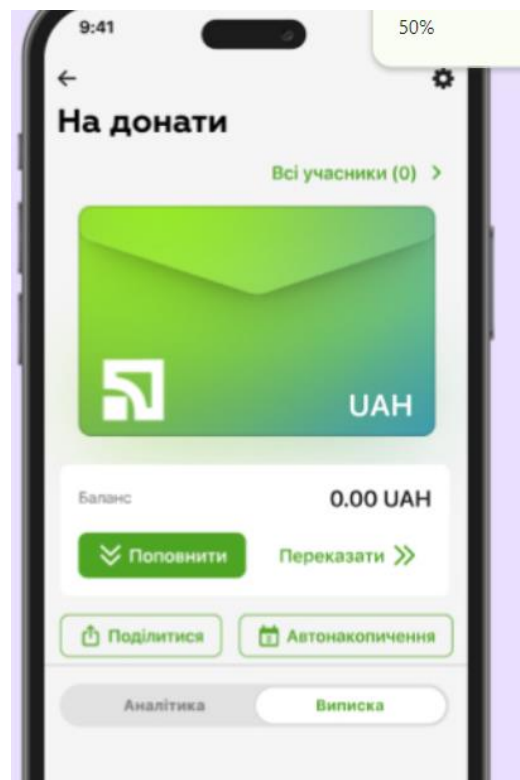


Рис.1.2. Конверт

Як бачимо, розробники не сильно морочилися і скопіювали зовнішній вигляд у Банки від monobank, та це й не важливо.

Тут же бачимо наявність QR-коду, за допомогою якого можна, відсканувавши його, потрапити до додатку Приват24 (якщо воно є) і продовжити надсилання донату там. Також, як бачимо, можна використовувати для внесення коштів систему Google Pay.

Втім, Приватбанк додав у сервіс кілька своїх новинок, що вигідно

відрізняють Конверт від монобанківської Банки.

Насамперед, Конвертом можна розраховуватися, тобто. е. після закінчення збору не треба переміщати з нього гроші на карту. Платежі, перекази, розрахунки в магазинах за допомогою Конверту — все, як і будь-яка інша банківська картка.

Крім того, Конверт можна "заклеїти" до певної дати чи суми, тобто закрити можливість витратити з нього. Ця функція важливіша для управління власними фінансами, але й у разі збирання коштів також може бути використана.

Третій сервіс під назвою «Скринька» («Шкатулка») з'явився зовсім недавно — лише кілька місяців тому. Його розробив та впровадив у мобільному додатку О. Bank Ідея Банк.

Сервіс дозволяє поповнювати та знімати гроші у будь-який час і без комісії.

Для того, щоб використовувати Скриньку для спільного збору, потрібно натиснути «Поділитися» та вибрати один із двох доступних варіантів:

- скопіювати номер рахунку;
- відправити реквізити Скриньки.

Поповнити Скриньку можна або звичайним Р2Р-перекладом з картки на карту (у Скриньки є свій 16-значний номер, як у звичайної картки), або перекладом по реквізитах.

Спеціальне посилання для поповнення згенерувати не можна. Як бачимо, тут поки що про зручність донатерів ще не подбали. Сподіваємось, у перспективі це буде реалізовано.

Важливий нюанс, що відрізняє скриньку від двох попередніх: банк нараховує на залишок 5% річних. Т. е. якщо гроші збиратимуться довго, банк трохи додасть у вигляді відсотка на залишок.

Також необхідно відмітити онлайн платформи та сайти для відомих Українських благодійних фондів для збору донатів на ЗСУ.

1. Повернись живим — один із найбільших та авторитетних фондів допомоги українським військовим, який розпочав свою роботу ще у 2014 році. Тому не дивно, що він першим спав на думку десяткам тисяч небайдужих

співвітчизників в очікуванні російського вторгнення — за добу до вторгнення 23 лютого на рахунки організації надійшло рекордне 48 млн грн. З того часу масштаби допомоги відчутно зросли і станом на кінець липня Повернись живим прозвітували про 4 млрд грн. донатів від 866 тис. благодійників. За ці гроші фонд закуповує обладнання, що допомагає рятувати життя військових: тепловізори, засоби захисту, квадрокоптери та авто.

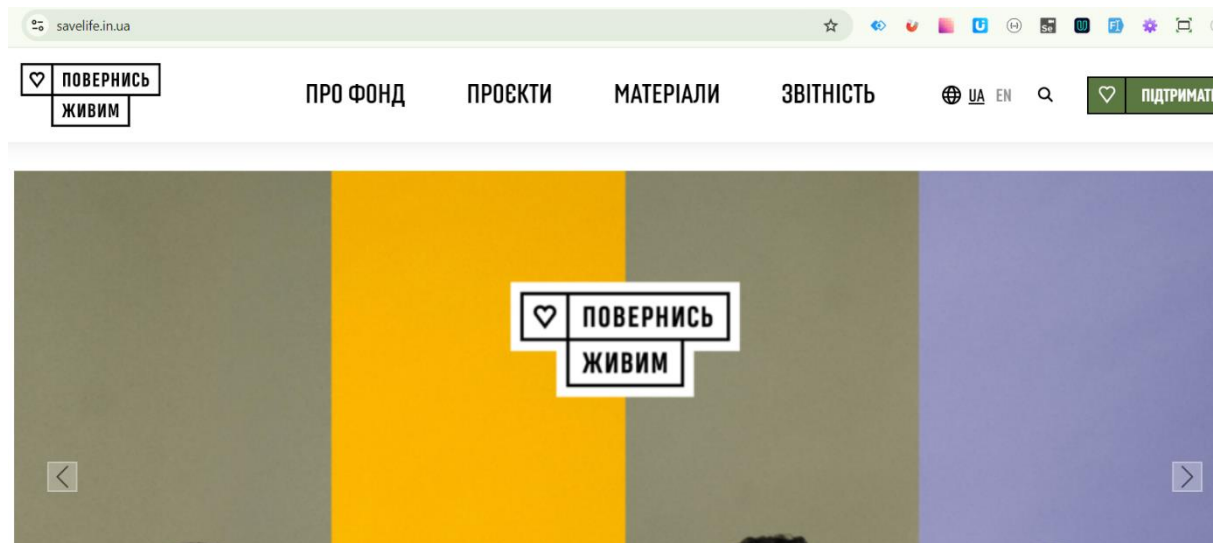


Рис.1.3. Онлайн платформа «Повернись живим»

2. Благодійний фонд Сергія Притули - мілітарна допомога армії, гуманітарна допомога цивільним. Загалом, починаючи з лютого, фонду Притули вдалося акумулювати понад 1 млрд грн., за рахунок яких українську армію додатково забезпечують засобами зв'язку, оптичними пристроями, дронами та БПЛА, транспортом та засобами тактичної медицини. цивільним на окупованих територіях та в радіусі 70 км від лінії фронту. ефективно перевозити та зберігати допомогу з усього світу фонд відкрив п'ять логістичних центрів в Україні та Польщі.

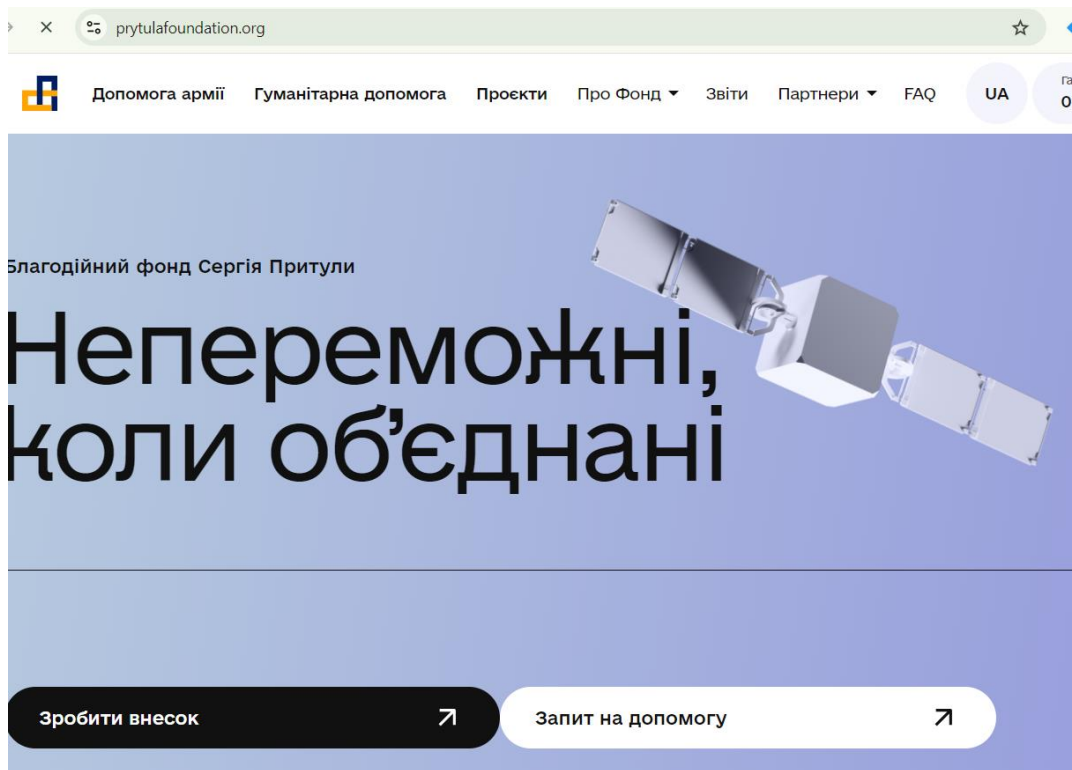


Рис.1.4 Благодійний фонд

Всі сайти мають схожу функціональність длябору грошових коштів, але не мають розгорнутої аналітики та прогнозування подальших надходжень.

Даний аналіз ліг в основу формування основного функціоналу програмного забезпечення, що розробляється.

1.2 Інформаційні процеси, обґрунтування вибору засобів моделювання та програмного середовища

При розробці інформаційно-аналітичних систем (ІАС), зокрема для управління благодійними внесками, важливим етапом є правильний вибір засобів моделювання та програмного середовища. Вибір технологій безпосередньо впливає на ефективність системи, її здатність обробляти великі обсяги даних, а також на зручність і швидкість роботи користувачів. У цьому контексті важливо детально розглянути інформаційні процеси в системі, а також обґрунтувати вибір необхідних інструментів для моделювання та розробки [9].

Для реалізації програмного комплексу використовувався патерн проектування HMVC.

Для зменшення початкових вкладень зазвичай приймають рішення про те, що програма має бути спроектовано у вигляді цілісної програми, що містить всі необхідні функції. Якщо сайт швидко знайде популярність, це стане проблемою. Рефакторинг погано масштабованих кодових баз справив погане враження. До того ж, це може зажадати великої кількості ресурсів і грошей. В ідеалі додатки повинні рости в міру необхідності і не вимагати в процесі цього великих фінансових витрат. Структура порівняння патернів MVC та HMVC рис.1.5

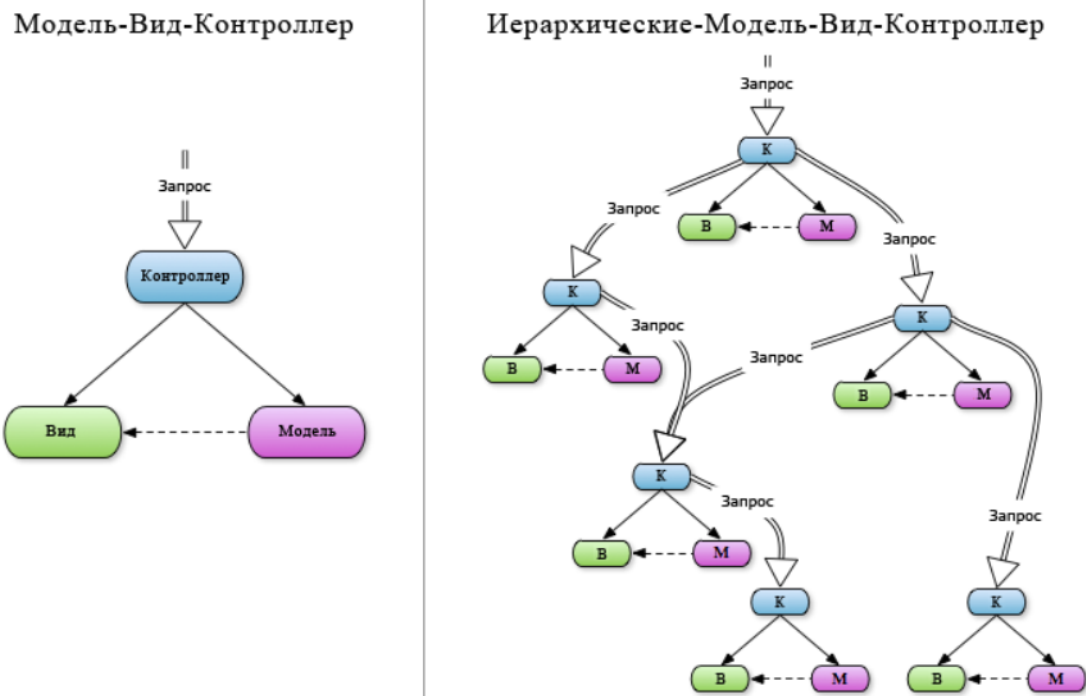


Рис. 1.5. Порівняння патернів

Патерн Ієрархічний-Модель-Вид-Контролер (HMVC) є розширенням MVC, яке дозволяє вирішити безліч вже згаданих проблем масштабованості. Вперше він був описаний в липні 2000 року в статті блогу JavaWorld під назвою «HMVC: багат шаровий патерн для розробки клієнтських рівнів [15].

HMVC - це набір традиційних тріад MVC, що працюють як один додаток. Кожна тріада абсолютно незалежна і може виконуватися при відсутності будь-якої іншої. Всі запити до тріадах повинні використовувати інтерфейс контролера, ніколи не підключайте моделі або бібліотеки поза ними домену. Фізичне місцезнаходження тріади на сервері не має значення, головне, щоб вона була доступна з усіх інших частин системи. Характерними особливостями HMVC є заохочення повторного використання коду, спрощення тестування окремих частин системи і гарантія того, що додаток може бути ускладнено і розширено без особливих складнощів.

Успішне проектування додатку на основі патерну HMVC неможливо без розбивки його функцій на окремі системи. Кожна така система повинна являти собою одну тріаду MVC, незалежно керуючу методами зберігання і поданням всередині більшого HMVC додатки. В даний час існує декілька фреймворків, що підтримують HMVC без додаткових модулів. Один з них - Kohana PHP третьої версії, який спочатку створювався з орієнтацією на HMVC.

1.3. Постановка технічного завдання

Розробка веб-орієнтованої інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність передбачає створення багатофункціонального сервісу, що забезпечує зручну взаємодію з користувачами та прозорість фінансових операцій. Основна мета проєкту полягає у створенні вебресурсу, який дозволяє обліковувати пожертви, аналізувати їх ефективність, а також презентувати дані в зрозумілій формі для залучення нових донорів і забезпечення довіри до роботи фонду.

Функціональні вимоги системи включають можливість реєстрації та авторизації користувачів для персоналізованого досвіду роботи. Користувачі повинні мати доступ до перегляду активних і завершених проєктів із деталізованою інформацією про кожен із них, включаючи опис, фінансові цілі,

статус виконання та прогрес збору коштів. Для забезпечення зручності внесення пожертв передбачено інтерактивну форму з вибором суми та валюти, а також можливістю додати примітки чи вказати особисті дані для звітності [2].

Інтеграція інформаційно-аналітичних функцій дозволяє автоматично збирати й обробляти дані про внески, що надходять. Система повинна забезпечувати збереження інформації про донорів, суми внесків, географію пожертв і категорії проєктів. На основі цих даних створюються аналітичні звіти, які можна візуалізувати у вигляді графіків, таблиць і діаграм. Аналітика включає відображення динаміки надходжень у часі, розподілу коштів за категоріями проєктів, а також звіти про найбільш активних донорів чи країни, звідки надходять пожертви.

Технічна частина реалізації системи включає створення бази даних для обліку користувачів, проєктів, пожертв і звітів.

Система має бути адаптивною, забезпечувати швидку роботу та доступ із будь-яких пристроїв. Інтерфейс розробляється з акцентом на зручність використання та інтуїтивність, щоб користувачі могли легко взаємодіяти із сервісом. Крім цього, система повинна мати можливість інтеграції з зовнішніми аналітичними інструментами або іншими сервісами, що дозволить розширити функціонал у майбутньому [1].

Технічне завдання охоплює також реалізацію звітності, яка дозволяє фонду демонструвати досягнуті результати, кількість зібраних коштів, завершених проєктів і кількість людей, яким була надана допомога. Інструменти візуалізації допомагають користувачам краще розуміти вплив їхніх внесків та стимулюють подальшу взаємодію з платформою.

Таким чином, запропонована система об'єднує функції збору, обліку та аналізу даних із гнучкими можливостями візуалізації та прозорим підходом до звітності, що дозволить фонду ефективно виконувати свої завдання та залучати нових донорів.

Проведений аналіз дозволив виявити і систематизувати основні вимоги та

функціональність програмного комплексу.

Можна перерахувати коло задач, які необхідно вирішити на теоретичному й експериментальному рівні в рамках дослідження проблеми:

- 1) розробити програмний комплекс інформаційно-аналітичної системи для управління та аналізу фінансових внесків у благодійність;
- 2) розробити модуль реєстрації користувачів системи, авторизацію та відновлення;
- 3) розробити модуль реєстрації запитів на допомогу відфільтрованих за категоріями;
- 4) розробити модуль збереження та надання адресної грошової допомоги;
- 5) розробити модуль пошуку за основними параметрами;
- 6) розробити модуль звітності надходжень та потреб;
- 8) розробити модуль статистичного оброблення інформації;
- 9) розробити модулі побудови аналітичної та графічної інформації на основі даних.

Реалізація програмного комплексу складає в основі використання web технологій.

- Інструментарій розробки PHP + MySQL + JS + HTML
- Мережевий протокол https

Підводячи підсумки, необхідно відзначити, що на сьогоднішній день процес автоматизації та аналізу збору грошових коштів перейшов вже на принципово новий рівень.

Висновки до розділу 1

У першому розділі роботи виконано аналіз сучасних інформаційно-аналітичних систем (ІАС) управління та аналізу благодійних внесків, а також обґрунтовано вибір засобів моделювання та програмного середовища для реалізації програмного комплексу. Дослідження показали, що сучасні ІАС у

сфері благодійності значно спрощують процеси управління благодійними внесками, забезпечуючи ефективність, прозорість та автоматизацію. До ключових функцій таких систем належать управління внесками, аналіз даних, автоматизація звітності та інтеграція з платіжними платформами. Завдяки використанню сучасних аналітичних інструментів, таких як Business Intelligence (BI) системи, стає можливим візуалізувати дані, моніторити ефективність кампаній і вдосконалювати стратегії залучення коштів.

Аналіз існуючих рішень показав, що на ринку благодійності широко використовуються сервіси, наприклад, "Банка" від monobank, "Конверт" від Приватбанку та "Скринька" від O.Bank, які забезпечують ефективний збір коштів, але мають обмежені можливості аналітики та прогнозування. Відомі благодійні фонди України, зокрема "Повернись живим" і Фонд Сергія Притули, активно впроваджують сучасні інформаційні системи для забезпечення прозорості та звітності.

Для розробки веб-орієнтованої інформаційно-аналітичної системи обрано патерн HMVC, який гарантує масштабованість, модульність і простоту тестування. Використання таких технологій, як PHP, MySQL, JavaScript та HTML, забезпечує швидкодію системи, її інтеграцію з платіжними платформами та адаптивність до вимог користувачів. Було визначено основні функціональні модулі системи, зокрема реєстрацію користувачів, обробку запитів на допомогу, аналітику, побудову звітів та інтерактивну візуалізацію даних. Розроблена схема бази даних містить основні елементи, необхідні для зберігання та обробки інформації про внески, донорів і проєкти.

Результати аналізу свідчать, що впровадження сучасних ІАС у сферу благодійності є важливим для підвищення ефективності та прозорості діяльності фондів. Розробка програмного комплексу, що включає інтегровані аналітичні інструменти та зручний інтерфейс користувача, сприятиме автоматизації процесів управління внесками, підвищенню довіри донорів та збільшенню фінансової підтримки благодійних ініціатив.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

2.1. Вибір технологій для розробки системи

Для створення програмного комплексу, було застосовано: мову розмітки гіпертекстових документів HTML5, каскадні таблиці стилів CSS3, браузерну мову програмування JavaScript, мову програмування PHP, веб-сервер Apache з підтримкою PHP та MySQL-баз даних, графічний редактор Adobe Photoshop CS6 та середовище розробки PhpStorm 21.0.1.

HTML це мова, яка дозволяє представляти інформацію в Internet. Те, що ви бачите при перегляді сторінки в Internet, це інтерпретація вашим браузером HTML-тексту. Щоб побачити HTML-коди сторінки в Internet, клацніть "View" в лінійці меню вашого браузера і виберіть "Source".

JavaScript – об'єктно-орієнтована, скрипкова мова програмування, розроблена компанією Netscape. JavaScript, зазвичай використовується, як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке вживання знаходить в браузерах, як мова сценаріїв для додання інтерактивності веб-сторінкам.

PHP – це скриптова мова (scripting language), що вбудовується в код сторінки, інтерпретується і виконується на сервері. Основна відмінність від CGI-скриптів, написаних на інших мовах, типу Perl або C – це те, що в CGI-програмах програміст сам пише HTML-код, що виводиться, а використовуючи PHP - вбудовує свою програму у готову HTML-сторінку. Відмінність PHP від JavaScript, полягає в тому, що PHP-скрипт виконується на сервері, а клієнтові передається результат роботи, тоді як в JavaScript-код повністю передається на клієнтську машину і лише там виконується [5].

JetBrains PhpStorm - комерційне кроссплатформене інтегроване середовище розробки для PHP. Розробляється компанією JetBrains на основі платформи IntelliJ IDEA. PhpStorm є інтелектуальним редактором для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок в коді і

автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3, 5.4, 5.5 і 5.6 (сучасні та традиційні проекти), включаючи генератори, спів програми, простори імен, замикання, типажі і синтаксис коротких масивів, повноцінний SQL -редактор з можливістю редагування отриманих результатів запитів, PHP UML.

База даних MySQL - це найпопулярніша в світі база даних з відкритим кодом. Завдяки своїй перевіреної продуктивності, надійності і простоті використання база даних MySQL найбільш часто використовується для веб-додатків на таких ресурсах, як Facebook, Twitter, YouTube і всі п'ять з п'яти кращих веб-сайтів *. Крім того, її особливо часто вибирають в якості вбудованої бази даних, яка розповсюджується тисячами постачальників програмного забезпечення і виробників обладнання.

MySQL є системою керування базами даних (СКБД). На сьогоднішній день це одна з найпопулярніших систем управління базами даних.

Початковим розробником даної СУБД була шведська компанія MySQL AB. У 1995 році вона випустила перший реліз MySQL. У 2008 році компанія MySQL AB була куплена компанією Sun Microsystems, а в 2010 році вже компанія Oracle поглинула Sun і тим самим придбала права на торговельну марку MySQL. Тому MySQL на сьогоднішній день розвивається під егідою Oracle [2].

2.2. Створення інтерфейсу користувача

Створення інтерфейсу користувача є важливим етапом розробки веб-орієнтованої інформаційно-аналітичної системи, оскільки саме він забезпечує зручність та ефективність взаємодії кінцевого користувача із системою. Основна увага була зосереджена на досягненні інтуїтивності, простоти використання, адаптивності та сучасного дизайну.

Інтерфейс користувача розроблявся відповідно до сучасних стандартів веб-дизайну з урахуванням принципів зручності використання. Основною метою було забезпечити доступ до всіх функціональних можливостей системи, включаючи подання аналітичної інформації, взаємодію з базою даних,

можливість формування та експорту звітів, а також внесення інформації. Особлива увага приділялася створенню логічної та зрозумілої навігації, яка дозволяє користувачу швидко знаходити необхідну інформацію.

На головній сторінці розміщено верхнє навігаційне меню, яке містить ключові розділи: "Фонд", "Про нас", "Звітність", "Проекти", "Сувеніри", "Події" та "Контакти".



Рис.2.1. Головна сторінка

Це дозволяє користувачу швидко перейти до потрібного розділу сайту. У центральній частині сторінки розташовано блок із заголовком "Як отримати допомогу від нашого фонду?", який пояснює порядок взаємодії з фондом.

Тут же користувачеві пропонується кнопка "Отримати допомогу", що переводить до детальної інструкції.

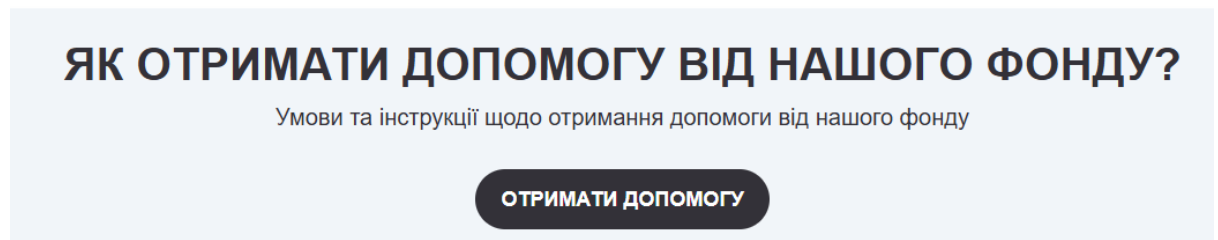


Рис.2.2. Як отримати допомогу

У розділі "Як отримати допомогу?" реалізовано детальну покрокову інструкцію для підрозділів, які потребують допомоги. Користувачам пропонується завантажити бланк заявки, заповнити його та надіслати на вказану електронну пошту або через форму на сайті. Додатково в цьому розділі передбачено інформацію про ключові умови отримання допомоги, що підвищує прозорість взаємодії.

ЯК ОТРИМАТИ ДОПОМОГУ ВІД НАШОГО ФОНДУ?

Фонд заснований під час відкритого вторгнення російської армії та створений для допомоги українським збройним силам.

Якщо Ваш підрозділ потребує допомоги нашого фонду вам потрібно:

1. Завантажити бланк заявки з нашого сайту.
2. Заповнити бланк заявки.
3. Відправити бланк на нашу електронну пошту **office@ukr.ua** або через форму нижче.

СКАЧАТИ БЛАНК ЗАЯВКИ

Рис.2.3. Скачати бланк

Ваші П.І.Б: *

Вкажіть Ваші П.І.Б

Email: *

Вкажіть Ваш email

Телефон: *

Вкажіть Ваш телефон

Область: *

Область

Організація: *

Вкажіть організацію

Сумма: *

вкажіть суму в грн

Рис.2.4. Заповнити форму

Однією з основних функціональних секцій є розділ "Наші проєкти". Цей блок інтерактивно відображає всі активні проєкти фонду за категоріями. У верхній частині блоку користувачу надається можливість обирати проєкти за тематиками: "Оборона та розмінування", "Відбудована Україна", "Живлення", "Дрони", "Автівки" та "Зв'язок".

Наші проєкти

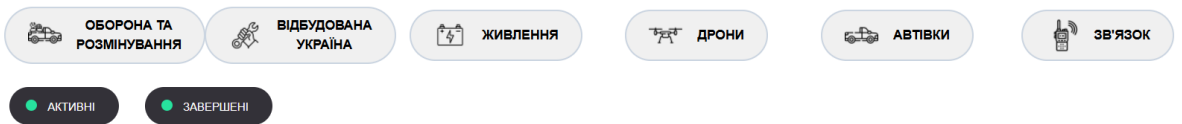


Рис.2.5. Наші проєкти

Під категоріями розміщені інтерактивні картки проєктів. Кожна картка містить назву, опис, статус (активний або завершений), а також фінансову інформацію про хід збору коштів: сума зібраних коштів і залишок до завершення збору. Користувачі мають можливість переглядати деталі проєкту та натискати кнопку "Підтримати проєкт", яка спрямовує до форми здійснення пожертв. Це спрощує участь користувачів у підтримці фонду.

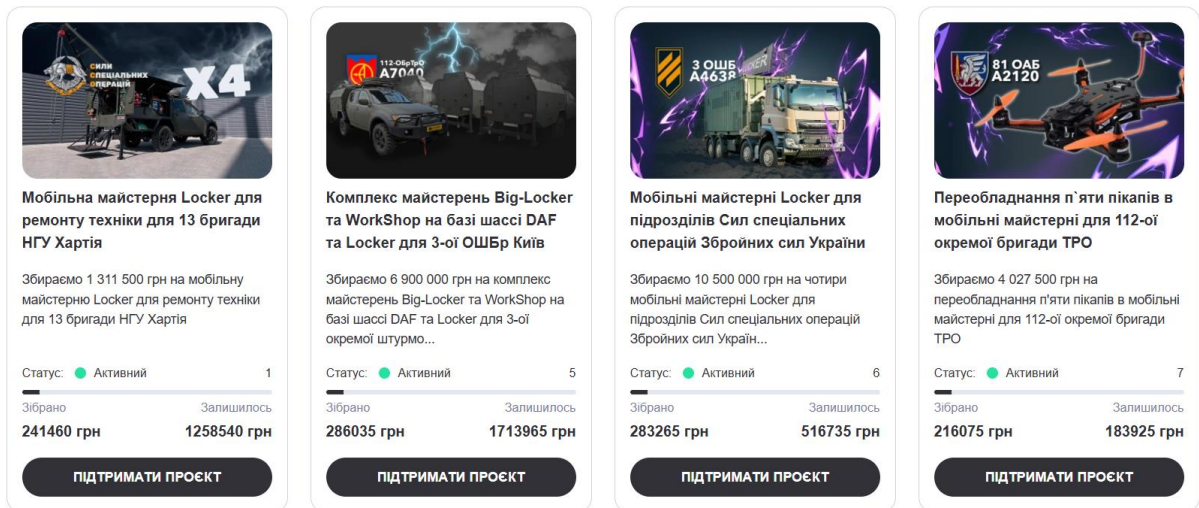


Рис.2.6. Підтримка проєкту

Для кожного проєкту передбачено кнопку "Підтримати проєкт", яка дозволяє користувачеві здійснити внесок. Ця структура дозволяє швидко орієнтуватися у поточній діяльності фонду та приєднуватися до важливих ініціатив.

Статистичні дані розташовані у спеціальному інформаційному блоці, який демонструє ефективність роботи фонду. Тут вказано суму переданої допомоги, кількість завершених проєктів, кількість людей, які отримали допомогу, та відсоток прозорості операцій.

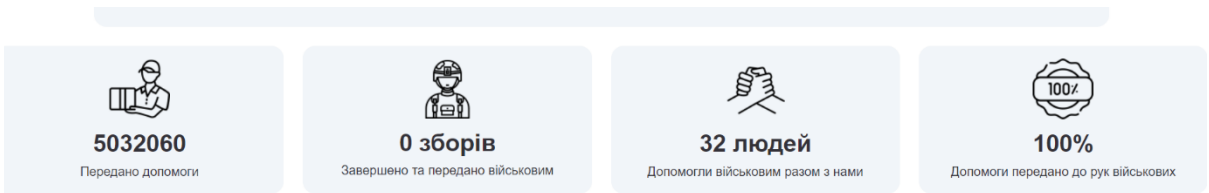


Рис.2.8. Статистичні дані

Натиснувши на кнопку «Підтримати проєкт», користувач переходить до іншої сторінки, де детально описано благодійний проєкт, а також присутня форма для внесків.

Зарядні станції для 23-ої окремої механізованої бригади

Опубліковано: 2024-11-10

Переглядів: 1162

16

Статус: Активний

ОПИС

НАДХОДЖЕННЯ

Зарядні станції для 23-ої окремої механізованої бригади 23-тя окрема механізована бригада (23 ОМБр) — з'єднання механізованих військ Сухопутних військ Зброй

23-тя окрема механізована бригада (23 ОМБр) — з'єднання механізованих військ Сухопутних військ Збройних сил України чисельністю у бригаду. У 2023 році у складі військ ОСУВ «Таврія» вела бої на запорізькому напрямку. Військовослужбовці 23 окремої механізованої бригади Збройних Сил України беруть безпосередню активну участь в обороні, забезпеченні суверенітету, територіальної цілісності та недоторканності України.

Зроби свій крок до ПЕРЕМОГИ!

Рис.2.9. Опис проєкту

На закладці «Надходження» можна переглянути всі донати на обраний проєкт.

Зарядні станції для 23-ої окремої механізованої бригади

Опубліковано: 2024-11-10 ☐ Переглядів: 1162 16 Статус: ● Активний



ОПИС			НАДХОДЖЕННЯ
Дата	Email	Сума	
2024-08-15	bwnv5o39lx@icloud.com	21600 UAH	
2024-08-15	pitkevych.andrii.1991@gmail.com	21600 UAH	
2024-08-15	bwnv5o39lx@icloud.com	41040 UAH	
2024-08-15	globina999@gmail.com	21600 UAH	
2024-08-15	bwnv5o39lx@icloud.com	21600 UAH	

Інші проєкти

Рис.2.10. Усі донати

У верхній частині блоку знаходиться інформація про статус збору коштів, яка включає загальну суму, вже зібрану, та залишок, необхідний для досягнення мети. Ці дані візуалізовані за допомогою горизонтального індикатора прогресу, що дає користувачу візуальне уявлення про хід кампанії.

Нижче розміщено блок вибору валюти внеску, який представлений трьома кнопками: UAH (українська гривня), EUR (євро) та USD (долар США). Активна валюта виділена темним кольором для інтуїтивної взаємодії, тоді як решта кнопок залишаються світлими. Сума внеску може бути обрана з попередньо встановлених варіантів, або ж користувач може натиснути кнопку "Інша сума" для введення власного значення.

Також у формі передбачено текстове поле для введення електронної адреси, яке містить підказку "Вкажіть Ваш email". Нижче розташоване випадаюче меню для вибору країни з підписом "Країна". Крім цього, користувачу пропонується визначити свою категорію, вибравши один із трьох варіантів: "Фонд", "Волонтер" або "Бізнес".

Підтримати

Зібрано

170430 грн

Залишилось

469570 грн

Валюта внеску

UAH

EUR

USD

Сума внеску

250

500

950

ІНША СУМА

Email

Вкажіть Ваш email

Країна

Країна

Ви...? *

☐ Фонд

☒ Волонтер

☐ Бізнес

ПІДТРИМАТИ ПРОЄКТ

Рис.2.11. Підтримка проєкту

На сторінці «Контакти» реалізована можливість відіслати приватне повідомлення із запитанням.

Контакти

Контактна інформація

Контактний телефон:
(096) 676-13-91

Email:
office@krok.com.ua

Адреса:
м. Київ

Ми в соціальних мережах:

f

ig

td

yt

Напишіть нам

Ваше ім'я: *

Вкажіть Ваше ім'я

Email: *

Вкажіть Ваш email

Телефон: *

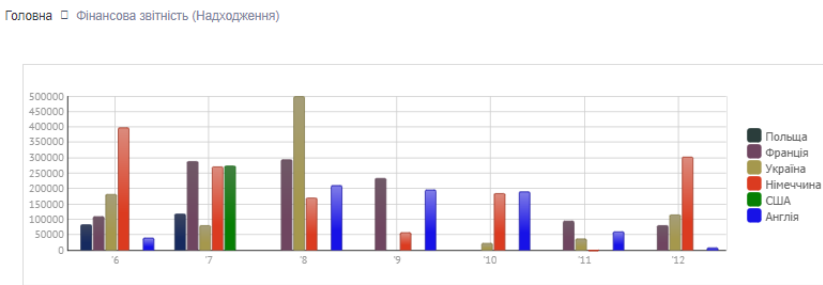
Вкажіть Ваш телефон

Ваше звернення: *

НАДІСЛАТИ ПОВІДОМЛЕННЯ

Рис.2.12. Заповнення форми для благодійників

Для прозорості звітності реалізована сторінка «Звітність». На сторінці виведені всі донати на всі проекти та динаміка грошових надходжень за країнами.



Фінансова звітність (Надходження)

Інформація про цільове використання благодійних внесків надається в порядку, визначеному чинним законодавством України

НАДХОДЖЕННЯ

Дата	Сума	Назва проєкту	Email
2024-12-10	21600 UAH	Комплекс майстерень Big-Locker та WorkShop на базі шасі DAF та Locker для 3-ої ОШБр Київ	bysuq@mailinator.com
2024-12-10	250 UAH	Комплекс майстерень Big-Locker та WorkShop на базі шасі DAF та Locker для 3-ої ОШБр Київ	zetojy@mailinator.com
2024-11-15	250 UAH	Інструменти та обладнання для для 57-ої окремої мотопіхотної бригади	ududiak@gmail.com
2024-11-15	21600 UAH	Переобладнання автомобіля ГАЗ-66 в майстерню Locker для 103-ої окремої бригади територіальної оборони	bwv5639lx@icloud.com
2024-11-15	20050 UAH	Переобладнання автомобіля ГАЗ-66 в майстерню Locker для 103-ої окремої бригади територіальної оборони	mhl83o2x@gmail.com
2024-11-15	38095 UAH	Переобладнання автомобіля ГАЗ-66 в майстерню Locker для 103-ої окремої бригади територіальної оборони	regx21@gmail.com

Рис.2.13. Сторінка «Звітність»

Також слід зазначити, що сайт було виконано з урахування адаптивного дизайну. Що забезпечує коректне відображення на різних пристроях, включаючи мобільні телефони й планшети. Окремі елементи, такі як категорії проєктів та статистичні показники, було оформлено з акцентом на простоту та естетичність.

Таким чином, створений інтерфейс поєднує у собі функціональність, естетику та простоту, відповідаючи потребам користувачів і підвищуючи ефективність роботи фондів та волонтерів.

2.3. Побудова аналітичної системи

Побудова аналітичної системи для управління та аналізу благодійних внесків є важливим етапом реалізації проєкту. Вона включає створення функціоналу для збору, обробки, зберігання та аналізу даних, а також їх візуалізації у зрозумілому форматі. Аналітична система базується на розробленій базі даних, SQL-запитах і вбудованих інструментах для роботи з великими обсягами даних [13].

Структура бази даних

Основою аналітичної системи є база даних MySQL, яка включає 10 таблиць, призначених для зберігання всіх необхідних даних про донорів, проєкти, надходження та категорії допомоги. Нижче наведено конфігурацію системи для розробки платформи:

The image shows a configuration window with four sections, each with a dropdown menu and a checkbox:

- HTTP:** The dropdown menu is set to "Apache-PHP-7-x64+N". The checkbox "Вести лог запитів" (Keep request log) is checked.
- PHP:** The dropdown menu is set to "PHP-7.1-x64". The checkbox "Вести лог запитів" is unchecked.
- MySQL / MariaDB:** The dropdown menu is set to "MySQL-5.7-x64". The checkbox "Вести лог запитів" is unchecked.
- PostgreSQL:** The dropdown menu is set to "Не використовувати" (Do not use). The checkbox "Вести лог запитів" is unchecked.

Рис.2.14. - Конфігурація системи розробки.

Для створення бази даних було вирішено використовувати MySQL 5.7 завдяки її функціональності і простоті у використанні. MySQL 8.0 ще дуже сира і по ній накопичено мало статистики. Її слід використовувати тільки тоді, коли необхідні нові функції цієї версії. MySQL - це популярна система керування базами даних (СКБД), дуже часто застосовується в поєднанні з PHP. MySQL - це система керування базами даних. У реляційній базі даних дані зберігаються не всі скопом, а в окремих таблицях, завдяки чому досягається виграш в швидкості і гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мову структурованих запитів плюс найбільш поширений стандартний мова, яка використовується для доступу до баз даних [15].

Реалізована база даних MySQL складається з 10 таблиць, які містять у собі всі дані для роботи програми. База даних має назву donaty.

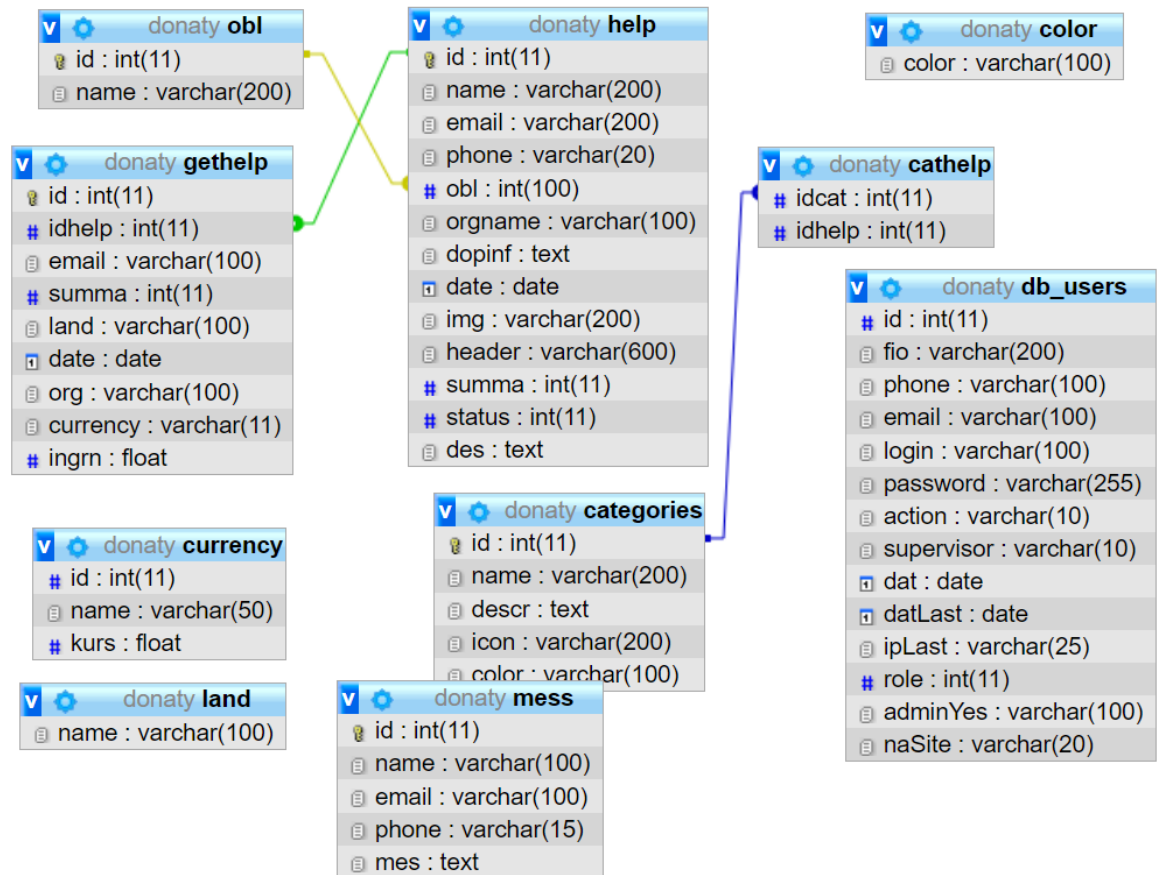


Рис.2.15. Діаграма бази даних.

Таблиця 2.1

Інформація про таблиці бази даних donaty

Назва таблиці	Призначення
<u>obl</u>	Області, що потребують допомоги
gethelp	Таблиця містить інформацію про надання допомоги
currency	Валюти надання допомоги
land	Довідник країн, з яких надається допомога
mess	Приватні запити та запитання
categories	Категорії допомоги
Db_user	Користувачі програми

cathelp	Таблиця, що поєднує категорії допомоги та запити на допомогу (реалізує зв'язок багато до багатьох)
help	Запити на допомогу (проекти)
color	Список кольорів (використовується для відображення діаграм)

Далі опишемо детально основні таблиці бази даних donatu.

Таблиця help призначена для зберігання інформації про запити на допомогу. Структура таблиці наведена нижче:

Таблиця 2.2

Таблиця help

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор запиту про допомогу
Name	Varchar	-	-	ПІБ відповідального за запит
img	Varchar	-	-	Рекламне зображення
Email	Varchar	-	-	Електронна пошта для контакту
phone	varchar	-	-	Телефон
Obl	Int	-	+	Код області України
Orgname	Varchar	-	-	Назва організації
Dopinf	Text	-	-	Додаткова інформація
Date	Date	-	-	Дата запиту
Header	Varchar	-	-	Заголовок запиту

Summa	Int	-	-	Грошова сума
Status	Int	-	-	Статус запиту
Des	text	-	-	Повний опис проблеми

Таблиця gethelp призначена для зберігання інформації про надання грошової допомоги. Структура таблиці наведена нижче:

Таблиця 2.3

Таблиця gethelp

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор
Idhelp	int	-	+	Ідентифікатор проекту, якому надається допомога
summa	float	-	-	Сума допомоги у обраній валюті
email	varchar	-	-	Електронна пошта
date	date	-	-	Дата надання допомоги
Land	varchar	-	-	Країна з якої надійшла допомога
org	varchar	-	-	Хто надав допомогу (волонтери, організації, фонди)
currency	varchar	-	-	Валюта допомоги
Ingrn	float	-	-	Сума допомоги в гривнях

Таблиця categories містить список категорій проектів для подальшого аналізу та аналітики.

Таблиця 2.4.

Таблиця categories

Назва	Тип даних	ПК	ЗК	Опис поля
Id	Int	+	-	Ідентифікатор категорії
name	varchar	-	-	Назва категорії
descr	text	-	-	Опис категорії
icon	varchar	-	-	Іконка
color	varchar	-	-	Колір для діаграм

Таблиця cathelp призначена для реалізації зв'язку багато до багатьох між таблицями categories та help. Так як запити на допомогу можуть бути віднесені до декількох категорій одночасно. Структура таблиці наведена нижче:

Таблиця 2.5

Таблиця cathelp

Назва	Тип даних	ПК	ЗК	Опис поля
idcat	Int	-	+	Ідентифікатор категорії
idhelp	int	-	+	Ідентифікатор проекту

SQL-запити для аналітики

Для аналізу даних було створено низку SQL-запитів: для виведення динаміки надходжень за місяцями та країнами; аналізу загальної суми допомоги за категоріями, щоб визначити найбільш затребувані сфери; обчислення мінімальних і максимальних сум допомоги за місяць; виведення інформації про проекти, які потребують додаткового фінансування; а також аналізу потреб за областями України та категоріями допомоги. Ці запити забезпечують глибокий аналіз даних і є основою для формування звітів та графіків.

Інструменти для аналітики та візуалізації

Для побудови інтерактивних графіків і таблиць використовуються наступні інструменти: **Chart.js** — для створення діаграм (лінійних, кругових, стовпчикових); **WebDataRocks** — для побудови зведених таблиць із можливістю інтерактивної взаємодії з даними; **Google Charts** — для візуалізації трендів та аналізу великих обсягів інформації. Для побудови діаграм використовується бібліотека **Webix**.

Бібліотека **Webix** пропонує величезну кількість UI віджетів, які можуть задовольнити побажання навіть найвибагливіших веб-розробників на JavaScript. Ці легко кастомізовані UI компоненти можна без особливих зусиль вбудувати в існуючі веб-додатки.

Однією з головних переваг **Webix** є здатність працювати на різних платформах. Бібліотека включає не тільки віджети для розробки швидких десктопних веб-додатків, але й дозволяє розробникам використовувати ті ж самі UI компоненти для створення легких у використанні мобільних додатків. Інтерфейс користувача мобільних програм розроблений з **Webix** повністю оптимізований під тач-екрани: події обробляються ідеально навіть на невеликих пристроях [7].

Що стосується інтеграції з сервером, то і тут розроблена компанією ІксБі Софтвар JavaScript бібліотека UI компонентів надає кілька простих рішень. Інтеграція може бути здійснена з використанням серверних конекторів для Java, PHP та ASP.NET. Більш того, компоненти **Webix** можуть працювати з такими фреймворками, як RoR, ASP.NET MVC, Spring, Struts та Grails. Бібліотека дозволяє розробникам використовувати REST API з усіма компонентами **Webix**.

Ще одна важлива характеристика **Webix** – це можливість інтегрування бібліотеки з такими популярними MVC фреймворками як Backbone.js та, починаючи з версії 1.2, з AngularJS. Це дозволяє розробникам будувати потужні веб-програми, використовуючи всі бонуси MVC фреймворків.

Крім того, бібліотека легко інтегрується з jQuery. Одним з головних факторів є те, що розробники можуть додавати UI компоненти Webix на сторінку, використовуючи jQuery, і бути впевненими, що компоненти не конфліктуватимуть з фреймворком jQuery. Варто також зазначити і інші характеристики Webix: послідовна та легка в освоєнні документація, інтеграція із редакторами Mercury, NicEdit, TinyMCE, інтеграція з картами Google та Nokia, швидке завантаження даних, налаштування зовнішнього вигляду віджетів повністю через CSS-код та крос-доменне завантаження даних.

Зважаючи на всі вище викладені переваги бібліотеки, можна зробити висновок, що Webix – це передовий JavaScript інструмент, який допомагає розробникам будувати прості у використанні мобільні та десктопні програми, використовуючи HTML5 та CSS3 технології.

Для реалізації зведених таблиць та аналізу даних використовується WebDataRocks. Це безкоштовний інструмент веб-звітів для аналізу та візуалізації даних, який написаний на JavaScript і не обмежений жодною зовнішньою структурою. WebDataRocks легко вбудовується у веб-сторінку та є потужним аналітичним інструментом із добре розробленим готовим до використання сучасним інтерфейсом користувача. WebDataRocks працює з наборами даних JSON і CSV розміром не більше 1 МБ, що має задовольнити більшість випадків використання

WebDataRocks має певні технічні характеристики: віртуальна сітка, яка забезпечує плавне відтворення та прокручування рядків, сортувати за значеннями, розгортання і згортання ієрархії в сітці, підсумки та проміжні підсумки, фільтрування звітів, додавання стовпців та комірок, а також побудова зведених таблиць.

2.4. Проектування модулів системи

При реалізації програмного комплексу було розроблено ряд модулів, які відповідають за виконання ключових функцій системи. Ці модулі забезпечують

зручність користування, ефективність роботи з даними та можливість аналізу інформації в різних розрізах. Усі модулі були спроектовані з урахуванням принципів модульності, що дозволяє легко розширювати та модернізувати систему в майбутньому.

Таблиця 2.6.

Модулі програми

№	Назва модулю	Функціональність, що реалізується
1	gethelp	Модуль роботи із наданням допомоги
2	Main	Головний модуль програми
3	needhelp	Модуль роботи із запитами про допомогу
4	analit	Модуль аналізу даних
5	prognoz	Модуль зведених таблиць
6	auth	Модуль авторизації в програмі
7	zv	Модуль формування звітів
8	admin	Модуль адміністративної частини

Важливо зазначити, що система складається з кількох функціональних модулів, кожен з яких виконує специфічні завдання для забезпечення ефективної роботи інформаційно-аналітичної системи. Модуль **gethelp** відповідає за збереження даних про транзакції, джерела допомоги та їх призначення, забезпечуючи обробку й відображення інформації про надану допомогу, а також виведення статистики за категоріями та датами. Головний модуль (**Main**) реалізує базову логіку завантаження сторінок, обробки запитів користувачів і

відображення основного інтерфейсу, координуючи роботу інших модулів. Модуль **needhelp** займається обробкою запитів про допомогу, включаючи валідацію даних, їх збереження до бази даних та представлення у зручному форматі для адміністраторів.

Модуль **analit** виконує функцію збору й обробки даних для формування аналітичних звітів, підтримуючи побудову графіків, діаграм і статистичних таблиць з використанням бібліотек візуалізації, таких як Chart.js. Модуль **prognoz** застосовує методи аналізу часових рядів для прогнозування динаміки надходжень і потреб благодійних фондів, забезпечуючи створення зведених таблиць і функціональність прогнозування. Для забезпечення безпеки використовується модуль **auth**, що включає механізми хешування паролів, валідацію даних і управління сесіями. Модуль **zv** підтримує формування та експорт звітів у різні формати, такі як PDF і Excel, з автоматичним оновленням даних. Нарешті, модуль **admin** забезпечує доступ до адміністративної частини системи, що дозволяє управляти всіма аспектами її роботи.

Діаграма класів.

На основі спроектованих модулів було побудовано діаграму класів, яка ілюструє взаємодію компонентів системи. Вона включає такий ключовий клас, як клас `Url`, який аналізує адресу сторінки та повертає назву контролера для завантаження відповідного класу контролера. Також клас `Autoloader`, який перевіряє на відповідність регулярному виразу шлях, що вводить користувач. Клас `AdminController` працює з даними адміністративної частини, взаємодіючи з моделями `AdminModel` та `AdminIModel`, а клас `Router` парсить адресу сторінки та передає необхідні параметри до контролерів.

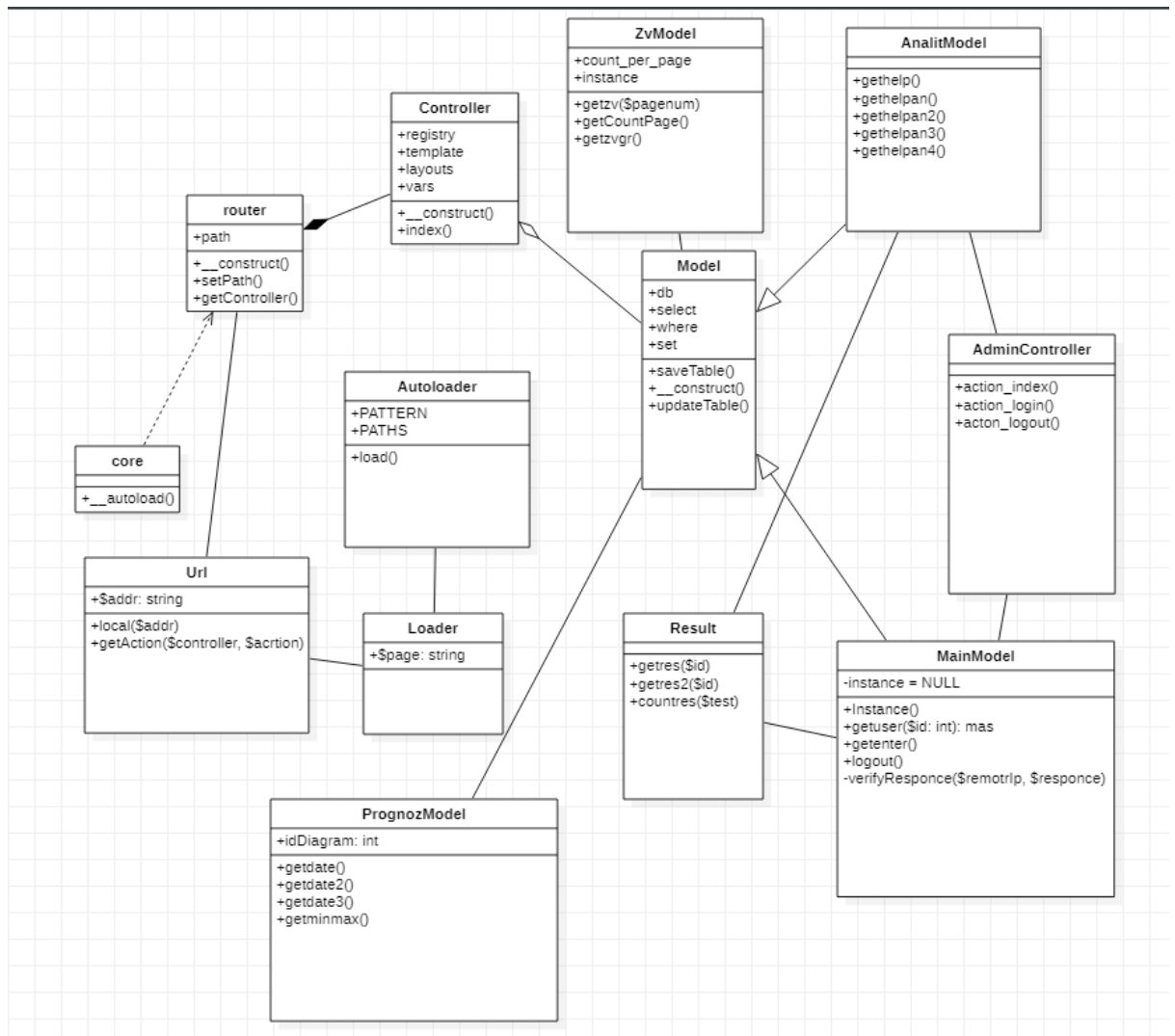


Рис. 2.16 Діаграма класів

Спроектовані модулі забезпечують гнучкість, масштабованість та ефективність роботи системи. Їх чітка структура дозволяє легко інтегрувати нові функції, а також оптимізувати існуючі процеси. Код модулів та детальна реалізація представлена у додатках до цієї роботи.

Код для реалізації авторизації та реєстрації наведено в Додатку Б.

Висновки до розділу 2

У другому розділі роботи було реалізовано ключові етапи розробки інформаційно-аналітичної системи для управління благодійними внесками. У ході виконання роботи створено архітектуру системи, визначено основні модулі

та їх функціональність, що забезпечило ефективну та масштабовану структуру програмного комплексу. Для розробки системи обрано оптимальні технології, зокрема PHP для бекенд-розробки, MySQL для управління даними та бібліотеки для візуалізації, такі як Chart.js. Спроектовано модулі, які відповідають за управління користувачами, роботу із запитом на допомогу, аналіз і прогнозування даних, формування звітів та адміністрування.

Реалізована база даних складається з десяти таблиць, що забезпечує централізоване зберігання та обробку інформації про пожертви, проекти та користувачів. Побудована діаграма класів демонструє взаємодію між компонентами системи та їх структуру. Запропоновано механізми для розширення функціональності, зокрема впровадження прогнозування на основі часових рядів і розширення модулів інтеграції з іншими сервісами.

Модульний підхід до розробки забезпечує гнучкість системи, що дозволяє легко додавати нові функції та адаптуватися до змінюваних потреб користувачів. Результати цього розділу закладають основу для подальшої реалізації аналітичного модуля та інтеграції додаткових функцій у наступних етапах розробки.

РОЗДІЛ 3. ВПРОВАДЖЕННЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

3.1. Використання аналітичної системи для управління внесками

Аналітична система є важливим інструментом, який забезпечує автоматизацію процесів збору, обробки та аналізу даних, а також візуалізації отриманих результатів, що значно підвищує ефективність роботи благодійних організацій і прозорість фінансових операцій. Завдяки інтеграції аналітичних інструментів організації можуть здійснювати глибокий аналіз даних і приймати обґрунтовані рішення, що сприяє оптимізації використання наявних ресурсів.

Існує кілька форм бізнес-аналітики, які можуть бути корисними для благодійних організацій. **Описова аналітика** надає знімок поточного стану справ, що дозволяє оцінити динаміку пожертвувань і демографічні характеристики донорів. **Діагностична аналітика** допомагає детально аналізувати поведінку користувачів, виявляти неефективні області та покращувати їх. **Прогностична аналітика** використовується для аналізу тенденцій і моделей, прогнозування внесків за демографічними групами й оптимального часу для залучення донорів. **Наказова аналітика** визначає найкращі стратегії дій, дозволяючи організаціям зосередити зусилля на досягненні цілей.

На основі даних аналітики донорів, що охоплює висновки з отриманих даних і способи їх практичного застосування, були реалізовані SQL-запити для аналізу різних аспектів роботи системи. Серед них запити для визначення динаміки надходжень за країнами, загальної суми допомоги за категоріями, мінімальних і максимальних сум пожертвувань за місяць, а також для оцінки потреб за проєктами, категоріями й областями. Запити також використовуються для побудови зведених таблиць, які допомагають візуалізувати динаміку потреб і внесків.

Ця система надає благодійним організаціям потужні інструменти для моніторингу й управління фінансовими потоками, а також для визначення

пріоритетних напрямів допомоги. Уся структура запитів детально описана в Додатку А, що дає змогу ефективно використовувати систему для аналізу й управління даними.

На першому етапі система автоматизує збір даних про фінансові внески, включаючи інформацію про волонтерів та благодійні організації, суми внесків, дати транзакцій та способи оплати.

Після збору даних система виконує їх обробку, що включає перевірку коректності введених даних, фільтрацію транзакцій за категоріями, географією чи періодом, а також збереження даних у централізованій базі. Це забезпечує структурованість, актуальність інформації та зручний доступ до історії транзакцій.

Аналітична система також забезпечує глибокий аналіз зібраної інформації. Вона дозволяє оцінити динаміку надходжень фінансів за певний період, активність донорів, визначити найбільш активних благодійників, аналізувати популярність категорій допомоги та визначати регіони з найбільшим обсягом внесків або недостатнім фінансуванням. Це дозволяє фонду ефективно розподіляти ресурси та концентрувати зусилля на найбільш затребуваних напрямках.

Однією з важливих функцій системи є можливість впровадження алгоритмів аналізу часових рядів для прогнозування надходжень. Це дозволяє передбачати майбутні фінансові потоки, оптимізувати розподіл коштів і планувати благодійні кампанії з урахуванням потенційних ризиків і можливостей.

Технології розробленої аналітичної системи автоматизують всі етапи роботи з внесками — від збору до аналізу та звітності, що значно підвищує результативність роботи фондів та довіру донорів.

3.2. Порівняння показників на основі аналітичних даних

Розвиток благодійного сектора супроводжується підвищеною увагою суспільства до діяльності благодійних фондів. Одним із показників відкритості фондів є фінансова звітність з якісним аудиторським висновком. Якість аудиту у свою чергу безпосередньо залежить від планування аудиту, яке буде ефективнішим із застосуванням аналізу. У цій статті розглянуто маркетинговий, стратегічний та інвестиційний аналіз – традиційні види аналізу для комерційних організацій [8].

Головною метою цього дослідження є вивчення та аналіз благодійних внесків в умовах війни, зокрема вивчення донатів на різні категорії допомог від волонтерських та гуманітарних організацій різних країн після повномасштабного вторгнення. Для досягнення цілей дослідницького проєкту, крім опитування благодійних фондів та аналізу динаміки донатів за останні пів року було також проведено кабінетний аналіз (desk research) вторинних даних щодо діяльності благодійних фондів. Аналізу підлягали всі суспільно-політичні матеріали, відібрані за ключовими словами «благодійний фонд», «благодійна організація», «громадська організація», «волонтерська організація».

Для аналізу були відібрані організації, які мають наступні КВЕД: 88.99 (КВЕД 2010) / 85.32.5 (КВЕД 2005) (благодійна діяльність).

На третьому році повномасштабної війни збори на підтримку армії вже стали звичною частиною нашого життя.

Минулого літа понад 80% українців регулярно донатили в середньому 10% своїх доходів.

У червні українці перевели на допомогу армії півмільярда гривень. У липні вже 1.9 млрд грн.

Сума пожертвувань у червні знизилася втричі за місяць — і у 12 разів у порівнянні з першими днями після вторгнення РФ у лютому. Так, за останній тиждень грудня на рахунок ЗСУ було переведено 6 мільярдів гривень.

У березні українці задонатили армії рекордну суму - 8,1 мільярда гривень. З того часу сума надходжень на рахунки ЗСУ лише зменшувалася. Загалом за понад 4 місяці з початку повномасштабної війни українці перевели на підтримку армії 19,6 млрд гривень.

Зменшення сум донатів можна пояснити як втомою українців від війни та загальним зменшенням доходів, так і появою нових масштабних волонтерських чи державних ініціатив.

У публікаціях зазначалося, що на початку повномасштабного вторгнення відбувся сплеск активності від громадян щодо донатів на ЗСУ та ТРО, порівняно з періодом до початку вторгнення. Весною кількість новин про активність громадян щодо донатів на ЗСУ трималась на високому рівні. ЗМІ приділяли значну увагу висвітленню новин про те, скільки грошей на армію вдалось зібрати. Таке звітування показувало, що підтримка армії від українців та країн Європи лишається на високому рівні і зростає. Влітку медійне висвітлення активності щодо допомоги ЗСУ зберігалось на високому рівні, однак зазначалось, що поступово кількість та суми донатів зменшуються. Восени збереглася тенденція щодо висвітлення активності донатити на військові потреби, зазначалось, що масовані обстріли критичної інфраструктури заохотили людей донатити більше. Окрім звітування про великі суми зібраних коштів, в інфопросторі з'явилося більше історій персональних невеликих вкладів.

Щодо бізнесу, то весною він активно вкладався в гуманітарну допомогу. Якоюсь мірою, для великих українських бізнес-гравців участь в гуманітарній підтримці постраждалих від війни стала засобом не тільки допомогти, а й укріпити власну репутацію. Влітку тенденція щодо активної участі бізнесу в питаннях гуманітарної допомоги закріпилась та посилилась. Висвітлення в ЗМІ участі бізнесу в питаннях гуманітарної допомоги залежало від масштабу компаній: якщо великі бізнес-гравці звітували про кількість сплачених податків та писали про участь в соціальних проєктах, то невеликі бізнеси згадувалися в

контексті закриття точкових потреб пожертвами в місцеві благодійні організації. В ЗМІ та соціальних мережах висвітлювалась як норма ситуація, коли частину прибутку бізнеси передавали на гуманітарні потреби. Значну увагу приділяли допомозі дітям, постраждалим від війни. Восени ЗМІ продовжували активно писати про участь бізнесу у гуманітарних питаннях. Найбільш актуальними були тенденції щодо тем допомоги деокупованим регіонам, грошової підтримки та акценті на відновлені. Допомогу деокупованим регіонам та її висвітлення в ЗМІ і соціальних мережах великі бізнес-гравці використовували як частину маркетингової стратегії. Продовжували також допомагати в медичній сфері.

Аналітичний модуль системи реалізовано в адміністративній частині проекту. Зайти в адміністративну частину можна лише за логіном та паролем зареєстрованому адміністратору.

№	Проект	Пошта	Сума	Країна	Дата	Організація	Валюта	В грн
2	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	vasul@ukr.net	950	Франція	2024-06-21	found	USD	38095
4	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	peschko5@uke.net	950	Німеччина	2024-06-21	volunteer	EUR	41040
5	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	peschko5@uke.net	950	Англія	2024-06-21	volunteer	EUR	41040
6	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	peschko5@uke.net	950	Німеччина	2024-06-21	volunteer	EUR	41040
23	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	tzp196o3wn@zoho.com	500	Україна	2024-06-21	volunteer	UAH	500
24	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	tzp196o3wn@zoho.com	500	Німеччина	2024-06-21	volunteer	EUR	21600
25	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	mhl83o2x@gmail.com	950	Німеччина	2024-06-21	business	USD	38095
26	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	globins999@gmail.com	500	Німеччина	2024-06-21	volunteer	USD	20050
1	Комплекс майстерень Big-Locker та WorkShop на базі шасі DAF та Locker для 3-ої ОШБр Київ	petrov@ukr.net	500	Німеччина	2024-06-21	business	EUR	21600
8	Комплекс майстерень Big-Locker та WorkShop на базі шасі DAF та Locker для 3-ої ОШБр Київ		500	Англія	2024-06-21	volunteer	UAH	500

Рис.3.1. Аналітичний модуль

В адміністративній частині реалізована можливість пошуку, сортування та виведення інформації про надходження та запити на допомогу.

Search

Entries per Page: 10

Лп	Проект	Пошта	Сума	Телефон	Дата	Статус	Допомога	Залишок
1	Мобільна майстерня Locker для ремонту техніки для 13 бригади НГУ Харків	Ivanenko	1500000	(435) 453-54-35	2023-08-09	1	241460	1258540
5	Комплекс майстерень Big-Locker та WorkShop на базі шасі DAF та Locker для 3-ої ОШБр Київ	andr@ukr.net	2000000	+38096 2323432	2023-07-13	1	483780	1516220
6	Мобільні майстерні Locker для підрозділів Сил спеціальних операцій Збройних сил України	fedko@ukr.net	800000	+380967896767	2024-07-02	1	283265	516735
7	Переобладнання п'яти пікапів в мобільні майстерні для 112-ої окремої бригади ТРО	frtd12@ukr.net	400000	+380974567887	2023-10-18	1	291675	108325
9	Два пікапа для 34-го окремого моторизованого батальйону ЗСУ	mhl83o2s@gmail.com	640000	+380936006118	2023-09-05	1	170430	469570
10	Інструменти та обладнання для для 57-ої окремої моторизованої бригади	vo29lpwxt@icloud.com	259200	+380932406118	2024-11-05	1	104790	154410
11	Дрон DJI Mavic 3 Pro для підрозділу РДК	wn8lp36vo@protonmail.com	76830	+380636033118	2024-09-04	1	143935	-67105
12	Два комплексу майстерень WorkShop-Locker на шасі 4*4 та Locker для 92 ОШБр	trpl96o3wn@zoho.com	6375600	+380736333118	2024-11-12	1	152290	6223310
13	Мобільна майстерня Locker для ремонту техніки для 30 ОМБр ім. Костянтина Острозького	vl2o93xtp@yahoo.com	655000	+3806865333118	2024-06-10	1	45600	609400
14	Комплекти інструментів та обладнання на 5 сервісів для 12-ої бригади спеціального призначення "Азов" НГУ	xplvo92tn@outlook.com	3721880	+380936006118	2024-10-14	1	246120	3475760

» 1 2 3 «

Рис.3.2. Адміністративна частина

Динаміка благодійних внесків за країнами за останні пів року

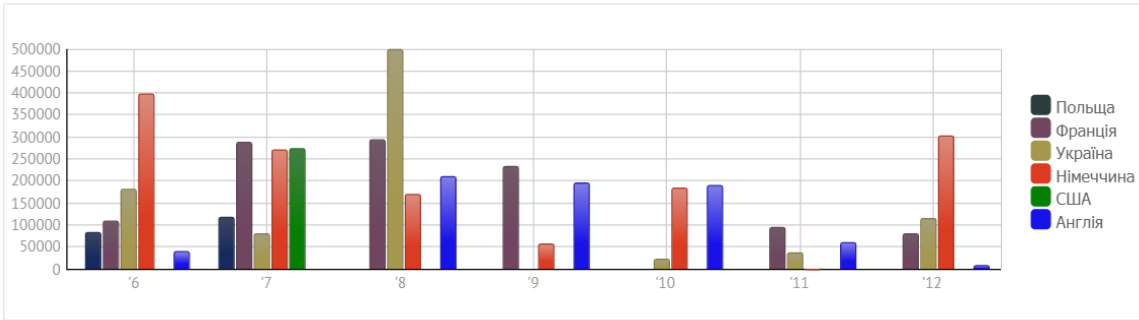


Рис.3.3. Динаміка благодійних внесків

За кількістю загальних грошових надходжень бачимо, що Німеччина (за останні пів року) посідає перше місце – 1500000 грн (завдяки тому, що на даний момент Німеччина приймає велику кількість Українських біженців).

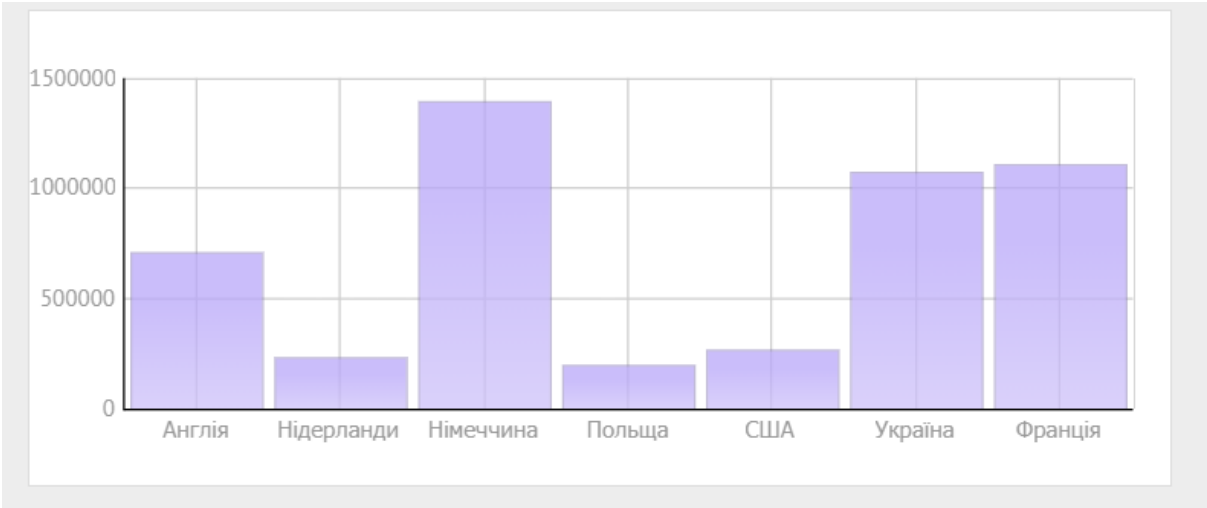


Рис.3.4. Динаміка благодійних внесків

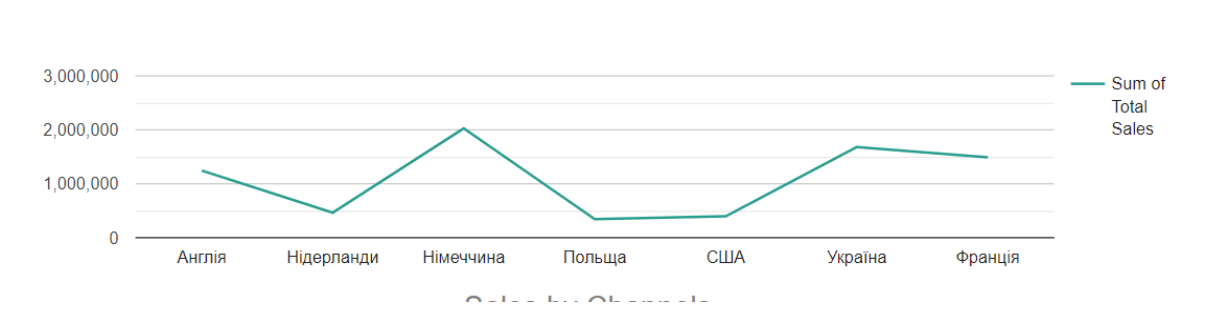


Рис.3.4. Динаміка благодійних внесків

Кількість надходжень у відсотковому відношенні

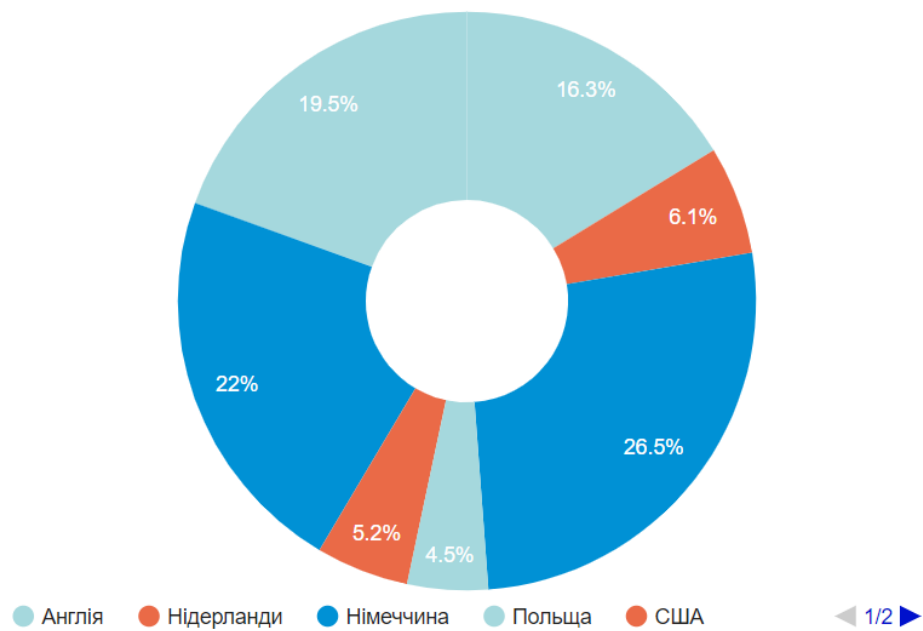


Рис.3.5. Кількість надходжень у відсотковому відношенні

Аналізуємо кількість грошових коштів та надходжень від різних організацій (волонтерів, бізнесу та фондів) з різних країн світу. Робимо висновок, що приватні особи та волонтери допомагають майже в 4 рази більше ніж фонди. Фонди таких країн як Нідерланди та Польща взагалі приймають мінімальну участь у допомозі Україні.

TOTAL NUMBER ⚙ NAME ⚙					
LAND ⚙	> business	> found	> volunteer	Total Sales	
Англія	587 500	74 050	1 628 995		
Нідерланди	381 340		82 080		
Німеччина	420 740	179 530	340 320		
Польща	129 850		215 020		
США	182 980	250	251 530		
Україна	744 330	261 435	676 110		
Франція	584 370	385 060	521 270		
Grand Total	3 031 110	900 325	3 715 325	7 646 760	

Рис.3.6. Аналіз кількості грошових надходжень

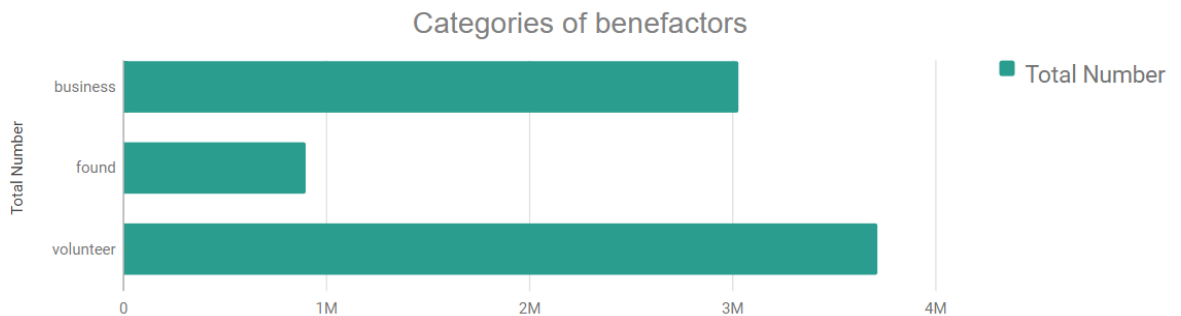


Рис.3.7. Аналіз кількості грошових надходжень

TOTAL NUMBER ⚙ NAME ⚙							
LAND ⚙	Автівки	Відбудована Україна	Дрони	Живлення	Зв'язок	Оборона та розмінування	Total Donats
> business	737 040	377 565	187 920	881 655	62 890	784 040	3 031 110
> found	263 760	65 050	31 625	199 835	82 940	257 115	900 325
> volunteer	644 825	736 320	59 695	1 166 710	273 165	834 610	3 715 325
Grand Total	1 645 625	1 178 935	279 240	2 248 200	418 995	1 875 765	7 646 760

Рис.3.8. Аналіз кількості грошових надходжень

TOTAL NUMBER	NAME							
LAND	Автівки	Відбудована Україна	Дрони	Живлення	Зв'язок	Оборона та розмінування	Total Donats	
business	737 040	377 565	187 920	881 655	62 890	784 040	3 031 110	
Англія	83 640	86 400		165 535		132 535		
Нідерланди	97 790			201 470		82 080		
Німеччина	209 845	64 800	43 200	162 590		59 695		
Польща	64 800				250	64 800		
США		59 695	41 040	22 550		59 695		
Україна	63 590	62 040	103 680	269 565	41 040	204 415		
Франція	217 375	104 630		59 945	21 600	180 820		
found	263 760	65 050	31 625	199 835	82 940	257 115	900 325	
Англія		10 800		10 800		52 450		
Німеччина	41 040	10 800	21 600	43 200	41 290	21 600		
США		250						
Україна	83 890		10 025	85 190		82 330		
Франція	138 830	43 200		60 645	41 650	100 735		

Рис.3.9. Аналіз кількості грошових надходжень

При аналізі категорій на які донатять різні країни ми бачимо, що в Україні майже не було донатів на категорію «Відбудована Україна» за останні пів року, тоді як країни Євросоюзу перерахували на цю категорію значні кошти.

LAND	TOTAL NUMBER			
NAME	business	found	volunteer	Total Sales
▼ Англія	587 500	74 050	1 628 995	2 290 545
Автівки	143 335		157 680	
Відбудована Україна	86 400	10 800	358 180	
Живлення	165 535	10 800	369 480	
Зв'язок			263 140	
Оборона та розмінування	192 230	52 450	480 515	
> Нідерланди	381 340		82 080	463 420
▼ Німеччина	420 740	179 530	340 320	940 590
Автівки	150 150	41 040	84 240	
Відбудована Україна	64 800	10 800	163 985	
Дрони	43 200	21 600	38 095	
Живлення	162 590	43 200	32 400	
Зв'язок		41 290		
Оборона та розмінування		21 600	21 600	

Рис.3.10. Аналіз категорії «Відбудована Україна»

▼ Україна	744 330	261 435	676 110	1 681 875
Автівки	63 590	83 890	203 880	
Відбудована Україна	62 040		81 295	
Дрони	103 680	10 025	21 600	
Живлення	269 565	85 190	347 735	
Зв'язок	41 040			
Оборона та розмінування	204 415	82 330	21 600	

Рис.3.11. Аналіз категорії «Відбудована Україна»

Якщо аналізувати потреби та запити на допомогу, бачимо що потреби зростають за останні півроку зросли майже в 7 разів.

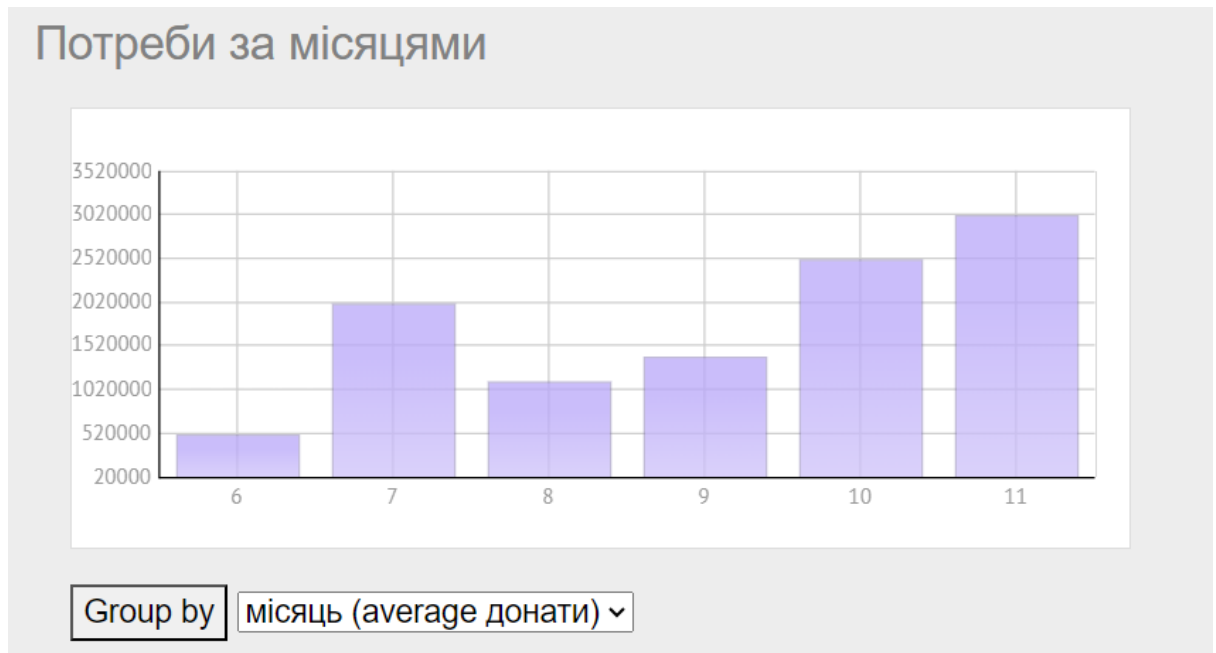


Рис.3.12. Аналіз потреб та запитів

Потреби за категоріями та областями України. Бачимо, що більше всього запитів на допомогу надійшло з Донецької області та Одеської. В Одеській області передусє категорія «Відбудована Україна» тоді як в Харківській на першому місці категорія «Живлення»

TOTAL NUMBER							
LAND	зaporізька область	Миколаївська область	Одеська область	Полтавська область	Сумська область	Харківська область	Total Donats
Автівки				384 920	800 000	405 925	
Відбудована Україна			5 587 930			632 700	
Дрони	2 485 415					76 830	
Живлення		2 000 000	3 215 450			7 008 300	
Зв'язок						632 700	
Оборона та розминування			5 269 680		800 000	1 038 625	
Grand Total	2 485 415	2 000 000	14 073 060	384 920	1 600 000	9 795 080	31 2

Рис.3.13. Аналіз потреби за категоріями та областями України

З категорій максимальні потреби має категорія «Живлення».

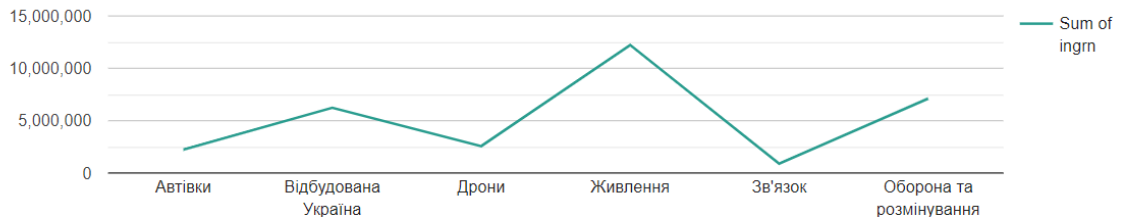


Рис.3.14. Аналіз потреби за категоріями та областями України

Відсоткові відношення категорій за потребами відображено на наступній діаграмі

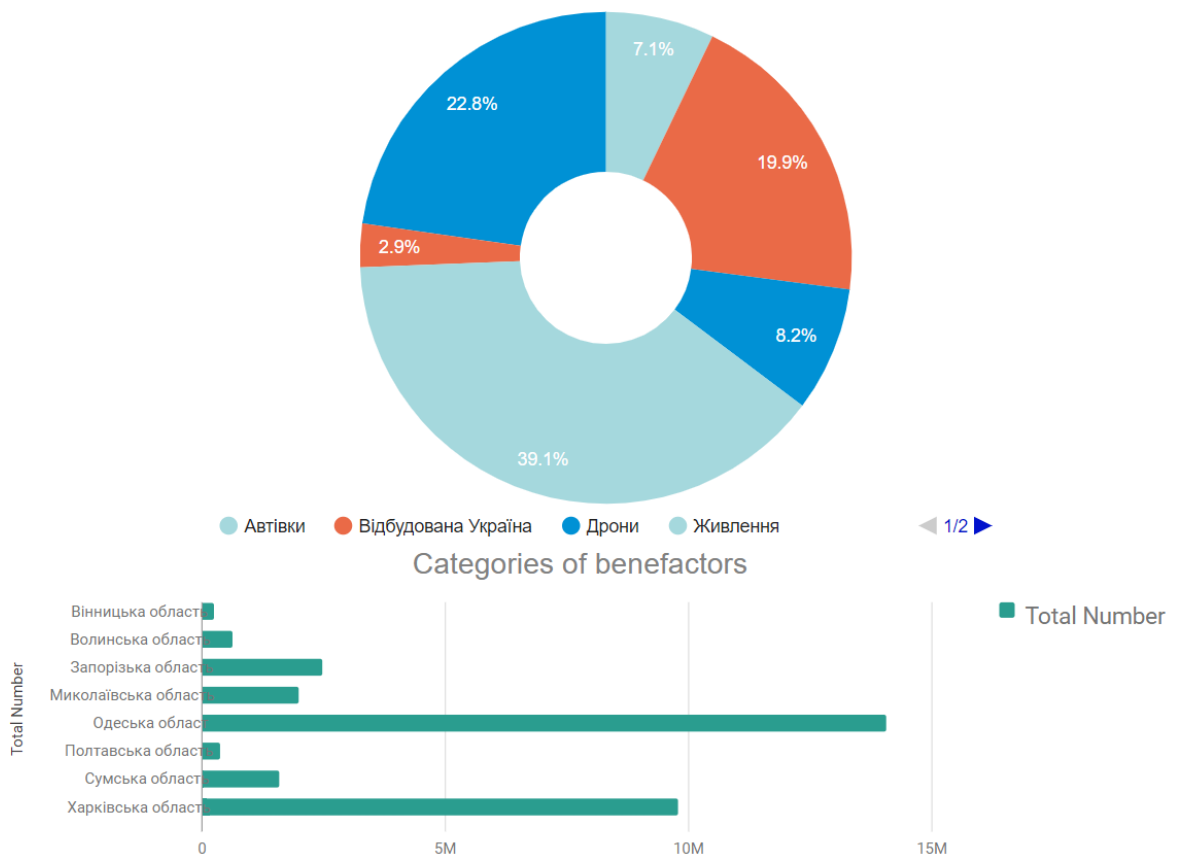


Рис.3.15. Відсоткові відношення категорій

Код зведеної таблиці та діаграм наведено у Додатку В

3.3. Оцінка ефективності та перспективи використання інформаційно-аналітичної системи

Оцінка ефективності інформаційно-аналітичної системи також базується на даних, отриманих із пункту 3.2. Результати аналітики свідчать про значний вплив впровадження системи на оптимізацію процесів управління благодійними внесками. Запроваджена система забезпечує прозорий та зручний механізм збору, аналізу та обробки даних про надходження коштів та інформацію щодо волонтерів та благодійних організацій, дозволяючи підвищити рівень довіри серед донорів та ефективно розподіляти ресурси.

Оцінка ефективності системи включала аналіз таких ключових показників: швидкість обробки транзакцій і генерації звітів; якість візуалізації великих обсягів даних; рівень точності прогнозування потреб і надходжень; зручність використання системи для кінцевих користувачів.

Опираючись на ці ключові показники, можна зробити наступний аналіз та порівняти, чи відповідає розроблена система ефективності та перспективи інформаційно-аналітичної системи. Тому треба взяти до уваги наступні дані:

1. Система дозволила зменшити час обробки даних на 75%, що підвищило продуктивність роботи.
2. Візуалізація трендів за допомогою інтегрованих інструментів, таких як Chart.js та WebDataRocks, забезпечила інтуїтивність і точність аналізу даних.
3. Система допомогла визначити, що найбільший обсяг пожертв надходить із країн Європи, а найпоширеніша категорія допомоги — "Живлення", яка покриває 40% загальних потреб.
4. Виявлено, що найбільші запити на допомогу надходять із Донецької області, причому основними категоріями є "Живлення" та "Медична допомога".

За результатами тестування система показала високий рівень продуктивності, здатність обробляти великі масиви даних у режимі реального часу та забезпечувати надійний захист інформації про донорів, що повністю відповідає критеріям ключових показників.

Але все ж таки існують тенденції вдосконалення системи. Перспективи подальшого розвитку включають впровадження нових модулів для прогнозування майбутніх фінансових потоків із використанням методів аналізу часових рядів, що дозволить завчасно планувати кампанії збору коштів. Крім того, передбачається інтеграція з додатковими платформами для обробки даних, такими як аналітичні сервіси Big Data, що розширить можливості системи у виявленні тенденцій і формуванні стратегічних планів.

Таким чином, розроблена система демонструє значний потенціал для масштабування та адаптації до потреб різних благодійних організацій, сприяючи зростанню їх ефективності та підвищенню впливу на соціальну сферу. А також надає змогу волонтерам та благодійним організаціям легко та швидко робити свої внески.

Висновки до розділу 3

Впровадження інформаційно-аналітичної системи для управління благодійними внесками стало важливим кроком для підвищення ефективності, прозорості та прогнозованості роботи благодійних організацій. Використання сучасних технологій забезпечило автоматизацію збору, обробки та аналізу даних про донорів, транзакції та запити на допомогу. Це дозволило ефективно управляти фінансовими ресурсами, аналізувати динаміку надходжень і потреб, а також створювати звітність, яка підвищує довіру донорів.

Система надає інструменти для аналізу великих обсягів даних, зокрема динаміки надходжень, популярності категорій допомоги та географічного розподілу фінансових потоків. Візуалізація даних за допомогою графіків,

діаграм і зведених таблиць спрощує прийняття управлінських рішень. Інтеграція з популярними платіжними системами та банківськими сервісами, такими як "Банка" та "Конверт", дозволяє автоматично обробляти транзакції та зменшує ризики втрати інформації.

Однією з ключових функцій системи є прогнозування надходжень на основі аналізу часових рядів. Це дозволяє передбачати майбутні фінансові потоки, оптимізувати розподіл коштів і планувати благодійні кампанії з урахуванням потенційних ризиків. Аналіз даних також допоміг виявити важливі тенденції, такі як зростання потреб за категоріями та областями України, що дозволяє більш ефективно розподіляти ресурси.

Завдяки цій системі благодійні організації отримали можливість більш точно оцінювати свою діяльність і адаптувати стратегії роботи відповідно до змін потреб і умов. Перспективи вдосконалення системи пов'язані з розробкою додаткових модулів, зокрема прогнозування, що дозволить ще більше оптимізувати процеси управління та підвищити ефективність благодійної діяльності.

ВИСНОВКИ

У кваліфікаційній роботі запропоновано нове рішення для статистичного спостереження та аналізу надходжень благодійних коштів, а також оцінки покриття потреб Збройних Сил України. Основний акцент зроблено на створенні сучасної інформаційно-аналітичної системи, яка дозволяє автоматизувати процеси збору, обробки та аналізу даних, а також забезпечити прозорість і зручність роботи з фінансовими потоками.

В результаті виконання роботи проведено аналіз літературних джерел, аналогічних рішень та технологій для автоматизації процесів, що дозволило сформулювати чіткі вимоги до системи. Обрано оптимальний підхід до реалізації автоматизації процесів та описано архітектуру системи, яка враховує потреби користувачів і забезпечує високу продуктивність. Розроблений програмний комплекс включає базу даних, наповнену реальними даними для проведення статистичного аналізу за різними параметрами. Ця база забезпечує можливість зберігання, швидкого доступу та обробки великих обсягів інформації.

Запропонована система рекомендована до використання благодійними фондами та волонтерськими організаціями, оскільки вона дозволяє ефективніше управляти благодійними внесками, здійснювати аналіз надходжень і потреб за різними категоріями, а також планувати майбутні дії. Впровадження системи дозволяє організаціям автоматизувати більшість рутинних процесів, знизити ризик помилок, підвищити прозорість і довіру з боку донорів.

Система надає широкі можливості для аналізу, включаючи оцінку динаміки надходжень, виявлення найбільш активних донорів, аналіз потреб у різних регіонах та категоріях допомоги. Це дозволяє приймати обґрунтовані управлінські рішення та концентрувати ресурси на найпріоритетніших напрямках. Особливо важливим є інструментарій прогнозування, який забезпечує можливість передбачати майбутні фінансові надходження та потреби.

Перспективи вдосконалення системи передбачають реалізацію додаткових модулів, зокрема модуля прогнозування, що базується на аналізі часових рядів. Цей модуль дозволить не лише оцінювати поточний стан надходжень і потреб, але й робити прогнози для оптимізації фінансового планування. Також можливе впровадження інтеграції з іншими аналітичними інструментами та системами для забезпечення комплексного аналізу.

Крім того, система сприятиме підвищенню рівня звітності, що є ключовим фактором для залучення нових донорів. Автоматизовані звіти та візуалізація даних допомагають організаціям не лише показати свою ефективність, але й створити прозорі та зрозумілі для донорів матеріали. Це робить систему потужним інструментом для розвитку благодійної діяльності в сучасних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бернштейн П.Л. Проти богів: видатна історія ризику. – К.: Дух і Літера, 2019. – 432 с.
2. Білейчук В.В. Аналітичні системи управління: підручник. – Київ: КНЕУ, 2020. – 320 с.
3. Дашко Н.М. Основи програмування Python. – Львів: ЛНУ ім. І. Франка, 2020. – 320 с.
4. Ентоні Р. Управлінський облік. – К.: Видавничий дім «Києво-Могилянська академія», 2018. – 280 с.
5. Гаврилук О.В. Розробка інформаційних систем: сучасні підходи. – Київ: КНТЕУ, 2021. – 256 с.
6. Kaner S., Bach J., Pettichord B. Lessons Learned in Software Testing. – New York: John Wiley & Sons, 2002. – 580 p.
7. Лемпель О.М. Системи управління базами даних: сучасні рішення. – Одеса: ОНУ ім. І.І. Мечникова, 2020. – 350 с.
8. Лутц М. Програмування на Python. Т. 1. – Київ: Символ, 2016. – 992 с.
9. Лутц М. Програмування на Python. Т. 2. – Київ: Символ, 2016. – 992 с.
10. Marr B. Data Strategy: How to Profit from a World of Big Data, Analytics, and the Internet of Things. – Kogan Page, 2017. – 240 p.
11. Марченко І.О. Автоматизація збору даних для бізнес-аналітики. – Дніпро: ДНУ, 2019. – 310 с.
12. Матвієнко Ю.В. Інструменти аналізу даних: навчальний посібник. – Київ: КНУ ім. Тараса Шевченка, 2020. – 270 с.
13. Назаренко О.В. Методи прогнозування на основі часових рядів. – Харків: НТУ "ХПІ", 2020. – 240 с.
14. Новіков А.В. Інформаційні системи в управлінні. – Львів: ЛНУ ім. І. Франка, 2020. – 320 с.

15. Пономаренко В.С., Луценко Є.С. Сучасні підходи до розробки програмного забезпечення. – Харків: ХНЕУ, 2010. – 342 с.
16. Provost F., Fawcett T. Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. – O'Reilly Media, 2013. – 414 p.
17. Wickham H., Grolemund G. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. – O'Reilly Media, 2017. – 520 p.

ДОДАТКИ

Запити до таблиць БД

ДОДАТОК А

Запит, що виводить динаміку надходження допомоги з різних країн за останній рік.

```
SELECT month(date) as year, sum(gethelp.ingrn) as ingrn, gethelp.land as
cantry

from gethelp where year(date) = year(CURRENT_DATE) group by
month(date), gethelp.land
```

Запит, що виводить загальну суму про допомогу за кожною категорією. Використовується для аналізу та знаходження більш потребуючих категорій.

```
select sum(help.summa) as s, categories.name, categories.color from help
inner join cathelp on cathelp.idhelp=help.id inner JOIN categories

on categories.id=cathelp.idcat

GROUP BY categories.name, categories.color
```

Запит, що виводить максимальну та мінімальну суму грошових надходжень за місяць в поточному році.

```
select min(s) as min, max(s) as max from (SELECT month(date) as m,
sum(gethelp.ingrn) as s from gethelp where year(date) = year(CURRENT_DATE) group
by month(date)) t
```

Запит, що виводить загальну суму допомоги, що надійшла за основними країнами, що допомагають.

```
select month(t.date) as mn,

sum(IF(land = 'Польща', t.ingrn, null)) as 'Польща',

sum(IF(land = 'Англія', t.ingrn, null)) as 'Англія',

sum(IF(land = 'США', t.ingrn, null)) as 'США',

sum(IF(land = 'Франція', t.ingrn, null)) as 'Франція',

sum(IF(land = 'Нідерланди', t.ingrn, null)) as 'Нідерланди',

sum(IF(land = 'Україна', t.ingrn, null)) as 'Україна',

sum(IF(land = 'Німеччина', t.ingrn, null)) as 'Німеччина'

from gethelp t

group by month(t.date)
```

Запит, що виводить інформацію про проекти та суми, що надійшли та ще залишились для закриття потреб. Дані із запиту використовуються для побудови зведених таблиць.

```
SELECT help.id,
help.name,help.email,help.phone,help.obl,help.orgname,help.dopinf,help.date,help
.img,help.header,

help.summa,help.status,help.des, sum(gethelp.ingrn) as a,help.summa-
sum(gethelp.ingrn) as b FROM help left join gethelp on gethelp.idhelp=help.id

group by help.id,
help.name,help.email,help.phone,help.obl,help.orgname,help.dopinf,help.date,help
.img,help.header,

help.summa,help.status,help.des
```

Запит, що виводить інформацію про надходження грошових коштів за категоріями запитів про допомогу та країнами надходжень. Дані із запиту використовуються для побудови зведених таблиць.

```
SELECT gethelp.ingrn as a,categories.name,help.header, gethelp.land,
help.obl,help.orgname,help.date,

gethelp.summa as b FROM help left join gethelp on gethelp.idhelp=help.id
inner join cathelp on cathelp.idhelp=help.id

inner join categories on categories.id=cathelp.idcat
```

Запит, що виводить інформацію про надходження грошових коштів за категоріями запитів про допомогу та областями України. Для аналізу більш потребуючих областей.

```
SELECT categories.name as land,summa as 'ingrn',obl.name FROM help

inner join cathelp on cathelp.idhelp=help.id inner join categories on
categories.id=cathelp.idcat

inner join obl on obl.id=help.obl
```

Запит, що виводить інформацію про необхідну суму збору для кожної категорії за місяць. Аналізується динаміка потреб за категоріями.

```
SELECT categories.name, sum(summa) as sales, month(date) as year FROM help
inner join cathelp on cathelp.idhelp=help.id inner join categories on
categories.id=cathelp.idcat
group by month(date), categories.name
```

Код для реалізації авторизації та реєстрації

ДОДАТОК Б

```

class AuthModule
{
private function __construct(){ }
private static $instance=NULL;
public static function instance(){
return self::$instance===NULL
? self::$instance=new self()
: self::$instance;
}
private $user=NULL;
private function GUID(){
return (function_exists('com_create_guid')===true)?
str_replace("-", "",trim(com_create_guid(),'{}'))
: sprintf("%04X%04X-%04X-%04X-%04X-%04X%04X%04X", mt_rand(0, 65535), mt_rand(0, 65535),
mt_rand(0, 65535), mt_rand(16384, 20479), mt_rand(32768, 49151), mt_rand(0, 65535), mt_rand(0, 65535),
mt_rand(0, 65535));
}
const SALT_LEN=10;
const SALT_POS = 12;
private function generateSalt(){
$s1=md5(rand(1000000,9999999));
$s2=sha1(time().rand());
return substr(md5($s1.$s2),0,self::SALT_LEN);
}
private function passhash($pass,$salt){

$h1=hash('sha256',$salt.$pass.$salt);

$h2 = hash('sha256',$pass.$salt.$pass);

$res=hash('sha256',$h1.$h2);
return substr_replace($res,self::SALT_POS,self::SALT_LEN);
}
private function repairSalt($hash){
return substr($hash,self::SALT_POS,self::SALT_LEN);
}
private function testPass($pass,$hash){
$salt = $this->repairSalt($hash);
$hash2=$this->passhash($pass,$salt);

return $hash=== $hash2;
}
const SESSION_LIVE_TIME=3600;
const SESSION_LIVE_TIME_BIG = 3600*24*10;
const SESSION_COOKIE_NAME = "SESID";
public function createSession($user_id,$save=false){
$ip=$_SERVER["REMOTE_ADDR"];
$agent = md5($_SERVER["HTTP_USER_AGENT"]);
$token = $this->GUID();
$exp = time()+$save?self::SESSION_LIVE_TIME_BIG:self::SESSION_LIVE_TIME;
DatabaseModule::instance()->tokens->insert([
"user_id"=>$user_id,
"user_ip"=>$ip,
"user_agent"=>$agent,

```

```

        "expires"=>$exp,
        "logintime"=>time(),
        "token"=>$token
    });
    setcookie(self::SESSION_COOKIE_NAME, $token,$exp,URLROOT."/");
}
private function sessionTimeTest($token){
    $exp = $token["expires"];
    if($exp<time()){
        DatabaseModule::instance()->tokens->delete($token["id"]);
        return false;
    }
    $lt=$token["expires"]-$token["logintime"];
    if(($token["logintime"]+$lt/2)<time()){
        $exp=time()+$lt;
        setcookie(self::SESSION_COOKIE_NAME, $token["token"],$exp,URLROOT."/");
        DatabaseModule::instance()->tokens->update($token["token"],[
            "expires"=>$exp,
        ]);
    }
}

}

public function register($login, $pass,$email, array $userdata=[]){
    $c=DatabaseModule::instance()->users->getCount("`login`=?",[$login]);
    if($c>0) throw new AuthException("User exists");
    $userdata["login"]=$login;
    $userdata["email"]=$email;
    $userdata["pass"] = password_hash($pass, PASSWORD_DEFAULT);
    $path=$_FILES['img']['tmp_name'];
    $img=addslashes(file_get_contents($path));
    $userdata["img"] =addslashes(file_get_contents($path));
    // $st="Insert into img values ('$img')";

    // $id=DatabaseModule::instance()->img->query($st);
    $id=DatabaseModule::instance()->users->insert($userdata);
    return $id;
}

public function login($login,$pass){
    $user = DatabaseModule::instance()->users->selectOne("`login`=?",[$login]);
    if(!$user) throw new AuthException("user not exist");

    if(!$this->testPass($pass,$user["pass"])) throw new AuthException("invalid password");

    $this->createSession((int)$user["id"]);
}

public function isAuth(){
    echo "dsdfsd";
    if(empty($_COOKIE[self::SESSION_COOKIE_NAME])) return false;
    $ip=$_SERVER["REMOTE_ADDR"];
    $agent = md5($_SERVER["HTTP_USER_AGENT"]);

    $token = DatabaseModule::instance()->tokens->selectOne("`token`=? AND `user_id`=? AND
`user_agent`=?",[$_COOKIE[self::SESSION_COOKIE_NAME],$ip,$agent]);
    if(!$token) return false;
    if(!$this->sessionTimeTest($token)) return false;
    $this->user = DatabaseModule::instance()->getById($token["user_id"]);
    if(!$this->user) return false;

```

```

return true;
}
public function getUser(){
    return $this->user;
}
public function logout($all=false){
    if(!$this->isAuth())return;
    if($all){
        DatabaseModule::instance()->tokens->deleteWhere("`user_id`=?",[(int)$this->user["id"]]);
    }else{
        DatabaseModule::instance()->tokens-
>deleteWhere("`token`=?",[$_COOKIE[self::SESSION_COOKIE_NAME]]);
        setcookie(self::SESSION_COOKIE_NAME, "",time()-1,URLROOT."/");
    }
}
}
}

```

Код двох контролерів.

1. NeedHelpController

```

<?php
session_start();
class NeedhelpController extends Controller
{
    public function action_index(){
        $v=new View("needhelp");
        $v->title="Потребую допомоги";
        $v->cat=NeedhelpModel::Instance()->getCat();
        $v->obl=NeedhelpModel::Instance()->getObl();
        $v->viewTemplate();
        $v->dop=MainModel::Instance()->getDop();
        $this->responce($v);
    }
    public function action_add(){
        $uploaddir = 'D:\OSPanel\domains\Donaty\media\img/';
        $uploadfile = $uploaddir . basename($_FILES['file']['name']);

        if (!is_uploaded_file($_FILES['file']['tmp_name'])) {
            echo "Файл не завантажено";
            die(); //or throw exception...
        }
        //Проверка что это картинка
        if (!getimagesize($_FILES["file"]["tmp_name"])) {
            echo "Це не зображення";
            die(); //or throw exception...
        }
        if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)) {
            echo "Ок.\n";
        } else {
            echo "Возможная атака с помощью файловой загрузки!\n";
        }
        if(!$this->post("name")) throw new RoutException("404 param not valid",404);
        $id=NeedhelpModel::Instance()->add($this->apostr($this->post("name")), $this->post("email"),
            $this->post("phone"), $this->apostr($this->post("obl")), $this->post("summa"), $_FILES['file']['name'],
            $this->post("orgname"), $this->post("dopin"), $this->post("cat"), $this->apostr($this->post("header")))
    );
}

```

```

    $v=new View("needhelp");
    $v->title="Потребую помощи";
    $v->id=$id;
    $v->dop=MainModel::Instance()->getDop();
    $v->viewTemplate();
    $this->responce($v);
  }
}

```

```

    2. <main>
    <h2 style="text-align:center; margin-bottom:2;">Analytics Dashboard</h2>
    <div id="wdr-component"></div>

    <h3 style="text-align:center; margin-bottom:20px;">Max Price by Country</h3>
    <div id="line-chart-container" style="width:900px;height:450; margin-left: 35px; margin-bottom:0;"></div>

    <h3 style="text-align:center;">Total donations in percentage by country</h3>
    <div id="pie-chart-container" style="width:900px;height:450; margin-left: 50px;"></div>

    <h3 style="text-align:center; margin-bottom:0;">Categories of benefactors</h3>
    <div id="bar-chart-container" style="width:900px;height:450; margin-left: 35px; margin-top:15px;"></div>
  </main>
  <script>
    var pivot = new WebDataRocks({
      container: "#wdr-component",
      toolbar: true,
      height: 550,
      width: 1000,
      customizeCell: customizeCellFunction,
      report: {
        dataSource: {
          data: getJSONData()
        },
        slice: {
          rows: [
            {
              uniqueName: "land"
            }
          ],
          columns: [
            {
              uniqueName: "Measures"
            },
            {
              uniqueName: "org"
            },
            {
              uniqueName: "name"
            }
          ],
          measures: [
            {
              uniqueName: "ingrn",
              aggregation: "sum"
            }
          ]
        },

```

```

    formats: [
      {
        name: "currency",
        thousandsSeparator: " ",
        decimalSeparator: ".",
        currencySymbol: "$",
        currencySymbolAlign: "left",
        nullValue: "",
        textAlign: "right",
        isPercent: false
      }
    ],
    options: {
      grid: {
        showHeaders: false
      }
    }
  },
  reportcomplete: function () {
    pivot.off("reportcomplete");
    pivotTableReportComplete = true;
    createLineChart();
    createPieChart();
    createBarChart();
  }
});
var pivotTableReportComplete = false;
var googleChartsLoaded = false;

google.charts.load("current", { packages: ["corechart", "bar"] });

google.charts.setOnLoadCallback(onGoogleChartsLoaded);

function onGoogleChartsLoaded() {
  googleChartsLoaded = true;
  if (pivotTableReportComplete) {
    createLineChart();
    createPieChart();
    createBarChart();
  }
}

function createLineChart() {
  if (googleChartsLoaded) {
    pivot.googlecharts.getData(
      {
        type: "line",
        slice: {
          rows: [
            {
              uniqueName: "land"
            }
          ],
          columns: [
            {
              uniqueName: "Measures"
            }
          ],

```

```

    ],
    measures: [
      {
        uniqueName: "ingrn",
        aggregation: "avg"
      },
    ]
  }
},
drawLineChart,
drawLineChart
);
}
}

```

```

function drawLineChart(_data) {
  var data = google.visualization.arrayToDataTable(_data.data);
  var formatter = new google.visualization.NumberFormat({
    fractionDigits: 2,
    prefix: "$"
  });
  formatter.format(data, 1);
  var options = {
    colors: ["#2A9D8F", "#E9C46A", "#E76F51"],
    bars: "horizontal"
  };
  var chart = new google.visualization.LineChart(
    document.getElementById("line-chart-container")
  );
  chart.draw(data, options);
}

```

```

function createPieChart() {
  if (googleChartsLoaded) {
    pivot.googlecharts.getData(
      {
        type: "pie",
        slice: {
          rows: [
            {
              uniqueName: "land"
            }
          ],
          columns: [
            {
              uniqueName: "Measures"
            }
          ],
          measures: [
            {
              uniqueName: "ingrn",
              formula: 'sum("ingrn")',
              individual: true,
              caption: "Total Donats"
            }
          ]
        }
      },
      drawPieChart,
    );
  }
}

```

```

        drawPieChart
    );
}
}

function drawPieChart(_data) {
    var data = google.visualization.arrayToDataTable(_data.data);
    var formatter = new google.visualization.NumberFormat({
        fractionDigits: 2,
        prefix: "$"
    });
    formatter.format(data, 1);
    var options = {
        legend: {
            position: "bottom"
        },
        pieHole: 0.35,
        chartArea: { height: "85%" },
        pieSliceBorderColor: "none",
        colors: ["#A5D8DD", "#EA6A47", "#0091D5"]
    };
    var chart = new google.visualization.PieChart(
        document.getElementById("pie-chart-container")
    );
    chart.draw(data, options);
}

function createBarChart() {
    if (googleChartsLoaded) {
        pivot.googlecharts.getData(
            {
                type: "bar",
                slice: {
                    rows: [
                        {
                            uniqueName: "org"
                        }
                    ],
                    columns: [
                        {
                            uniqueName: "Measures"
                        }
                    ],
                    measures: [
                        {
                            uniqueName: "org",
                            formula: 'sum("ingrn")',
                            caption: "Total Number"
                        }
                    ]
                }
            }
        ),
        drawBarChart,
        drawBarChart
    );
}
}

function drawBarChart(_data) {

```

```

var data = google.visualization.arrayToDataTable(_data.data);
var formatter = new google.visualization.NumberFormat({
  fractionDigits: 2,
  prefix: "$"
});
formatter.format(data, 1);
var options = {
  colors: ["#2A9D8F"],
  bars: "horizontal"
};
var chart = new google.charts.Bar(
  document.getElementById("bar-chart-container")
);
chart.draw(data, options);
}

/* Insert icons to the cells */
/*Icons made by Smashicons from www.flaticon.com */
function customizeCellFunction(cell, data) {
  if (
    data.type == "value" &&
    !data.isDrillThrough &&
    data.isGrandTotalColumn &&
    !data.isTotalRow
  ) {
    if (data.value < 15000) {
      cell.text =
        "<img src='https://static.webdatarocks.com/uploads/2019/02/21213347/icons8-decline-64-1.png' class='centered'>";
    } else if (data.value >= 15000) {
      cell.text =
        "<img src='https://static.webdatarocks.com/uploads/2019/02/21213348/icons8-account-64.png' class='centered'>";
    }
  }
}

function getJSONData() {
let head= {
  Price: {
    type: "number"
  },
  Quantity: {
    type: "number"
  },
  ProductGroup: {
    type: "string"
  },
  Product: {
    type: "string"
  },
  Country: {
    type: "string"
  },
  Channel: {
    type: "string"
  }
}

```

```

    }
};

}

</script>

```

3. GetHelpController

```

<?php
session_start();
class GethelpController extends Controller
{
    public function action_index(){
        $v=new View("gethelp");
        $v->title="Надаю допомогу";
        $v->id=$this->post("id");
        $v->help=GethelpModel::Instance()->gethelp($this->post("id"));
        $v->helpemail=GethelpModel::Instance()->gethelpemail($this->post("id"));
        $v->cat=GethelpModel::Instance()->getCat();
        $v->help3=GethelpModel::Instance()->getHelp3();
        $v->dop=MainModel::Instance()->getDop();
        $v->viewTemplate();

        $this->responce($v);
    }
    public function action_add(){
        $idhelp=$this->post("id");
        if(!$this->post("amount")) throw new RountExeption("404 param not valid",404);
        $id=GethelpModel::Instance()->add($this->apostr($this->post("land")), $this->apostr($this->post("user-
occupation")),
            $this->post("amount"), $this->apostr($this->post("email")), $this->post("id"), $this->post("currency")
        );

        $v=new View("gethelp");
        $v->title="Надаю допомогу";
        $v->id=$id;
        $v->viewTemplate();
        $v->dop=MainModel::Instance()->getDop();
        $v->help=GethelpModel::Instance()->gethelp($idhelp);
        $v->helpemail=GethelpModel::Instance()->gethelpemail($this->post("id"));
        $v->cat=GethelpModel::Instance()->getCat();
        $v->help3=GethelpModel::Instance()->getHelp3();
        $this->responce($v);
    }
}

```

```

1      <main>
2          <h2 style="text-align:center; margin-
3 bottom:2;">Analytics Dashboard</h2>
4          <div id="wdr-component"></div>
5          <h3 style="text-align:center; margin-bottom:20px;">Max
6 Price by Category</h3>
7          <div id="line-chart-container"
8 style="width:900px;height:450; margin-left: 35px; margin-
9 bottom:0;"></div>
10         <h3 style="text-align:center;">Total donations in
11 percentage by category</h3>
12         <div id="pie-chart-container"
13 style="width:900px;height:450; margin-left: 50px;"></div>
14         <h3 style="text-align:center; margin-
15 bottom:0;">Categories of benefactors</h3>
16         <div id="bar-chart-container"
17 style="width:900px;height:450; margin-left: 35px; margin-
18 top:15px;"></div>
19     </main>
20     <script>
21         var pivot = new WebDataRocks({
22             container: "#wdr-component",
23             toolbar: true,
24             height: 550,
25             width: 1000,
26             customizeCell: customizeCellFunction,
27             report: {
28                 dataSource: {
29                     data: getJSONData()
30                 },
31                 slice: {
32                     rows: [
33                         {
34                             uniqueName: "land"
35                         }
36                     ],
37                     columns: [
38                         {
39                             uniqueName: "Measures"
40                         },
41                         {
42                             uniqueName: "org"
43                         },
44                         {
45                             uniqueName: "name"
46                         }
47                     ],
48                     measures: [
49                         {
50                             uniqueName: "ingrn",
51                             aggregation: "sum"
52                         }
53                     ]
54                 },
55                 formats: [
56                     {
57                         name: "currency",
58                         thousandsSeparator: " ",

```

```

59         decimalSeparator: ".",
60         currencySymbol: "$",
61         currencySymbolAlign: "left",
62         nullValue: "",
63         textAlign: "right",
64         isPercent: false
65     }
66     ],
67     options: {
68         grid: {
69             showHeaders: false
70         }
71     }
72 },
73     reportcomplete: function () {
74         pivot.off("reportcomplete");
75         pivotTableReportComplete = true;
76         createLineChart();
77         createPieChart();
78         createBarChart();
79     }
80 });
81 var pivotTableReportComplete = false;
82 var googleChartsLoaded = false;
83
84 google.charts.load("current", { packages: ["corechart",
85 "bar"] });
86
87 google.charts.setOnLoadCallback(onGoogleChartsLoaded);
88
89 function onGoogleChartsLoaded() {
90     googleChartsLoaded = true;
91     if (pivotTableReportComplete) {
92         createLineChart();
93         createPieChart();
94         createBarChart();
95     }
96 }
97
98 function createLineChart() {
99     if (googleChartsLoaded) {
100         pivot.googlecharts.getData(
101             {
102                 type: "line",
103                 slice: {
104                     rows: [
105                         {
106                             uniqueName: "land"
107                         }
108                     ],
109                     columns: [
110                         {
111                             uniqueName: "Measures"
112                         }
113                     ],
114                 },
115             ],
116             measures: [
117                 {
118                     uniqueName: "ingrn",
119                     aggregation: "avg"

```

```

120         },
121     ]
122     }
123     },
124     drawLineChart,
125     drawLineChart
126 );
127 }
128 }
129
130     function drawLineChart(_data) {
131         var data =
132 google.visualization.arrayToDataTable(_data.data);
133         var formatter = new
134 google.visualization.NumberFormat({
135             fractionDigits: 2,
136             prefix: "$"
137         });
138         formatter.format(data, 1);
139         var options = {
140             colors: ["#2A9D8F", "#E9C46A", "#E76F51"],
141             bars: "horizontal"
142         };
143         var chart = new google.visualization.LineChart(
144             document.getElementById("line-chart-container")
145         );
146         chart.draw(data, options);
147     }
148
149     function createPieChart() {
150         if (googleChartsLoaded) {
151             pivot.googlecharts.getData(
152                 {
153                     type: "pie",
154                     slice: {
155                         rows: [
156                             {
157                                 uniqueName: "land"
158                             }
159                         ],
160                         columns: [
161                             {
162                                 uniqueName: "Measures"
163                             }
164                         ],
165                         measures: [
166                             {
167                                 uniqueName: "ingrn",
168                                 formula: 'sum("ingrn")',
169                                 individual: true,
170                                 caption: "Total Donats"
171                             }
172                         ]
173                     }
174                 },
175                 drawPieChart,
176                 drawPieChart
177             );
178         }
179     }
180

```

```

181         function drawPieChart(_data) {
182             var data =
183 google.visualization.arrayToDataTable(_data.data);
184             var formatter = new
185 google.visualization.NumberFormat({
186                 fractionDigits: 2,
187                 prefix: "$"
188             });
189             formatter.format(data, 1);
190             var options = {
191                 legend: {
192                     position: "bottom"
193                 },
194                 pieHole: 0.35,
195                 chartArea: { height: "85%" },
196                 pieSliceBorderColor: "none",
197                 colors: ["#A5D8DD", "#EA6A47", "#0091D5"]
198             };
199             var chart = new google.visualization.PieChart(
200                 document.getElementById("pie-chart-container")
201             );
202             chart.draw(data, options);
203         }
204
205         function createBarChart() {
206             if (googleChartsLoaded) {
207                 pivot.googlecharts.getData(
208                     {
209                         type: "bar",
210                         slice: {
211                             rows: [
212                                 {
213                                     uniqueName: "name"
214                                 }
215                             ],
216                             columns: [
217                                 {
218                                     uniqueName: "Measures"
219                                 }
220                             ],
221                             measures: [
222                                 {
223                                     uniqueName: "name",
224                                     formula: 'sum("ingrn")',
225                                     caption: "Total Number"
226                                 }
227                             ]
228                         }
229                     },
230                     drawBarChart,
231                     drawBarChart
232                 );
233             }
234         }
235
236         function drawBarChart(_data) {
237             var data =
238 google.visualization.arrayToDataTable(_data.data);
239             var formatter = new
240 google.visualization.NumberFormat({
241                 fractionDigits: 2,

```

```

242         prefix: "$"
243     });
244     formatter.format(data, 1);
245     var options = {
246         colors: ["#2A9D8F"],
247         bars: "horizontal"
248     };
249     var chart = new google.charts.Bar(
250         document.getElementById("bar-chart-container")
251     );
252     chart.draw(data, options);
253 }
254
255 /* Insert icons to the cells */
256 /*Icons made by Smashicons from www.flaticon.com */
257 function customizeCellFunction(cell, data) {
258     if (
259         data.type == "value" &&
260         !data.isDrillThrough &&
261         data.isGrandTotalColumn &&
262         !data.isTotalRow
263     ) {
264         if (data.value < 15000) {
265             cell.text =
266                 "<img
267 src='https://static.webdatarocks.com/uploads/2019/02/21213347/icons8-
268 decline-64-1.png' class='centered'>";
269         } else if (data.value >= 15000) {
270             cell.text =
271                 "<img
272 src='https://static.webdatarocks.com/uploads/2019/02/21213348/icons8-
273 account-64.png' class='centered'>";
274         }
275     }
276 }
277
278 function getJSONData() {
279     let head= {
280         Price: {
281             type: "number"
282         },
283         Quantity: {
284             type: "number"
285         }
286     }
287 }
288 </script>

```