

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна технологія управління та аналізу витрат особистого бюджету

Здобувача групи ІТ.м-22 Думанського Тимура Валерійовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Тимур ДУМАНСЬКИЙ
(підпис) (Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доцент Володимир НАГОРНИЙ
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ) _____ (підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Думанському Тимуру Валерійовичу

(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи Інформаційна технологія управління та аналізу витрат особистого бюджету

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «11» грудня 2023 р.

3 Вхідні дані до кваліфікаційної роботи інформація про фінанси, інформація про користувачів та графічне представлення технології, додатковий графічний матеріал.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) вступ, аналіз предметної області, огляд останніх досліджень і публікацій, аналіз сервісів аналогів для ведення особистого бюджету, постановка задачі та методи дослідження, мета та задачі дослідження, розробка візуальної частини інформаційної технології, розробка функціональної частини інформаційної технології, моделювання та проектування, структура інформаційної технології управління та аналізу витрат особистого бюджету, структурно-функціональне моделювання процес, моделювання діаграми варіантів використання інформаційної технології, проектування бази даних, практична реалізація інформаційної технології, програмна реалізація, використання інформаційної технології користувачем, адміністрування додатком інформаційної технології, висновки, список використаних джерел.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, постановка задачі, аналіз додатків, функціональні та не функціональні вимоги, реалізація проекту, структурно-функціональне моделювання, діаграма

варіантів використання, логічна модель бази даних, фізична модель бази даних, демонстрація, демонстрація (користувач), демонстрація (адміністратор), висновки.

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	11.09.23-15.09.23	
2	Створення дизайну	15.09.23-05.10.23	
3	Розробка візуальної частини	06.10.23-02.11.23	
4	Розробка бази даних	03.11.23-20.11.23	
5	Розробка функціональної частини	20.11.23-19.12.23	
6	Тестування	19.12.23-27.12.23	
7	Наповнення контентом	27.12.23-28.12.23	
8	Завантаження на сервер	28.12.23-28.12.23	

Магістрант _____

Думанський Т.В.

Керівник роботи _____

к.т.н., доц. Нагорний В.В.

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Інформаційна технологія управління та аналізу витрат особистого бюджету».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 35 найменувань, 3 додатки. Загальний обсяг роботи – 93 сторінка, у тому числі 65 сторінок основного тексту, 3 сторінки списку використаних джерел, 25 сторінок додатків.

Актуальність розробки інформаційної технології управління та аналізу витрат особистого бюджету є організація та структуризація звітності особистого бюджету, що дозволяє користувачам організувати свої фінансові звітності в одному місці та проводити статистику опираючись на використані данні.

Мета роботи: розробити інформаційну технологію управління та аналізу витрат особистого бюджету.

Також було визначено фінансовий аспект у сфері ІТ, проведено детальний аналіз поточних наукових досліджень у цьому напрямку. Крім цього, було створено план роботи та модель інформаційної системи, виконано представлення процесу через структурно-функціональне моделювання та описано сценарії використання. Розроблена схема бази даних розроблена для оптимальної взаємодії з інформацією.

Було продемонстровано використання інформаційної технології управління та аналізу витрат особистого бюджету з боку користувача та адміністратора. Результатом кваліфікаційної роботи магістра є розроблений web-додаток інформаційної технології управління та аналізу витрат особистого бюджету.

Ключові слова: інформаційна система, інформаційна технологія, web-додаток, фінанси, особистий бюджет, доходи, витрати, заощадження, стаття, економіка, фінансова грамотність, модель.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз сервісів-аналогів ведення особистого бюджету	10
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	24
2.1. Мета та задачі дослідження.....	24
2.2. Обґрунтування вибору інструментів реалізації інформаційної технології у вигляді web-додатку.....	26
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ.....	28
3.1 Структурно-функціональне моделювання процесу.....	28
3.2 Проектування діаграми варіантів інформаційної технології.....	30
3.3 Структура інформаційної технології.....	33
3.4. Проектування бази даних	35
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	40
4.1 Програмна реалізація	40
4.2 Використання додатку інформаційної технології користувачем.....	45
4.3 Адміністрування додатком інформаційної технології	62
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТОК А	70
ДОДАТОК Б.....	78
ДОДАТОК В	89

ВСТУП

Банківська справа, складання бюджету, заощадження, кредит, борг та інвестиції – це стовпи, на яких підтримується більшість фінансових рішень, які ми прийматимемо в нашому житті.

Досягнення фінансової грамотності може допомогти людям уникнути прийняття неправильних фінансових рішень. Це може допомогти їм стати самодостатніми та досягти фінансової стабільності. Ключові кроки до досягнення фінансової грамотності включають навчання, як складати бюджет, відстежувати витрати, виплачувати борги та примножувати накопичення.

Для оптимізації усіх описаних вище процесів можна використовувати інформаційні технології. Логічним висновком є той факт, що це широкий та постійно розвиваючись напрямок. Для кращого аналізу та формування рекомендацій використовують моделювання, аналіз даних та розробки бізнес-моделей на отриманій інформації.

Кожен із процесів збору, обробки та безпосередньо роботи із великими об'ємами даних, полегшується з використанням інструментів інформаційних технологій. Для цього не завжди потрібно бути обізнаним у фінансах, плануванні, управлінні та аналізу доходів та витрат особистого бюджету.

Отже, було прийнято рішення розробити інформаційну технологію для ведення особистого бюджету.

Мета: розробити інформаційну технологію управління та аналізу витрат особистого бюджету.

Об'єкт дослідження: управління та аналіз особистого бюджету користувача, включаючи його доходи, витрати та заощадження.

Предмет дослідження: інформаційна технологія управління та аналізу витрат особистого бюджету.

Для досягнення мети було сформовано ряд задач, які потрібно виконати. Результатом виконаних задач буде готова інформаційна технологія.

Задачі:

- Оцінити впровадження інформаційних технологій у сферу управління, обліку й аналізу особистих витрат;
- Проаналізувати продукти-аналоги та порівняти функціональні можливості, виділити сильні та слабкі сторони;
- Сформувати перелік функціональних та нефункціональних вимог інформаційної технології;
- Спроекувати розробку інформаційної технології;
- Розробити візуальну та функціональну частини додатку;

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

При обговоренні фінансів та фінансової грамотності, люди часто стикаються з такими поняттям як потреби та бажання. Ці поняття, та їх не розуміння, значно впливають на фінансові пріоритети людей [1]. Точніше буде сказати, відсутність переліку категорій чи саме предметів у кожному із напрямків є проблемою на шляху до фінансової грамотності.

На рівень фінансової грамотності серед молодих людей та ставлення до грошей також впливають соціальні фактори, тобто батьки та однолітки. Дослідження на цю тематику можна знайти у статі «Financial Literacy, and Saving Behaviour» [2]. Цікаво було проведено аналіз та оцінено вплив фінансової грамотності на розвиток заощаджувальної поведінки та роль самооцінки жителів Саудівської країни.

Наприклад в іншому дослідженні [3-4], було проведено аналіз інформації де було розглянуто та сформовано прямі та непрямі впливи особистісних рис, поведінки та підтримки сім'ї, щодо інвестиційних намірів з використанням кількісного підходу. Крім того, це дослідження також має на меті додати обмежені емпіричні дані студентів в Індонезії [5].

Щоб компанії продовжували створювати цінності за допомогою цифрових технологій і даних, вони повинні максимально використовувати ІТ-ресурси. З цієї причини управління ІТ-фінансами зараз стає все більш і більш важливим, і все більше ділових і адміністративних відділів хочуть заново вивчити основи управління ІТ-фінансами, які раніше були залишені ІТ-відділу [6].

У інформаційній сфері ведення фінансової звітності вже давно визнано необхідність переходу до структурної взаємодії між людиною та ІТ на різних рівнях. З впровадженням хмарних технологій зростала надійність збереження даних. Не дивлячись на залежність від інтернет-з'єднання, хмарні технології все частіше використовуються для збереження найважливіших даних. Збільшення

інформаційної звітності породжує попит на більші об'єми хмарних технологій із застосуванням найпередовіших розробок для більш оптимізованої та надійної роботи серверів [7].

Прискорення темпів зростання джерел даних та їх об'єм зробили Data Science однією з галузей, що розвиваються найшвидше [8-9]. Все більше підприємств і організацій покладаються на спеціалістів з обробки та інтерпретації даних для надання практичних рекомендацій та покращення результатів прибутковості спираючись на фінансові звіти. Аналітики, в свою чергу, можуть використовувати їх для отримання корисної інформації [10]. Все більше з'являється спеціалістів, створюються наукові групи, моделюється сітка взаємодії між дослідницькими центрами для аналітики масштабніших проєктів та створення висновків для подальшого розвитку тієї чи іншої фінансової галузі [11].

Збір аналітичних даних – це одна із рутинних задач фінансового відділу, яку може бути автоматизовано з використанням цифрових технологій та може підвищити ефективність їх виконання. Обробка транзакцій і різноманітні завдання звітності можуть бути автоматизовані. Використовуючи технології для розширеного аналізу даних на основі яких можна отримати повну фінансову звітність, прибутковість, збитковість та актуальність тих чи інших фінансових процесів [12]. Також покращується структурованість збереження даних з новими алгоритмами кодування, що забезпечує підвищену безпеку не лише фінансової звітності, а й дані учасників різних фінансових договорів [13].

ІТ-технології прогресують з кожним днем та використовуються в банкінгу не лише для покращення безпеки, а й в банківських операціях та транзакціях, що в свою чергу задовольняє широкий список потреб клієнтів. Вже зараз завдяки мобільному банкінгу, банківські послуги доступні кожному в будь-який час та в будь-якому місці без ризику втратити фінансові звіти, бо ті мають електронні копії, які зберігаються на серверах банків в закодованому вигляді [14]. Враховуючи ризики кібератак, то безпека фінансових потоків стоїть на найвищому рівні. Саме тому віртуальні гроші мають специфічні методи кодування та запутану систему

взаємодії. Еволюція інформаційних технологій в грошовій системі породила систему віртуальних валют [15].

Блокчейн використовується у віртуальних валютах, є типом бази даних, яка безпосередньо з'єднує термінали в мережі обробляє та записує записи транзакцій у розподілений спосіб за допомогою криптографічної технології. Блокчейн одразу почав використовуватися завдяки підвищеній безпеці, надійності та універсальності [16]. З появи цих технологій почали розвиватися віртуальні грошові одиниці, на які вже не впливала глобальна банківська система, а рівень кібербезпеки на стільки високий, що обмін криптовалютою являється найбезпечнішим способом передачі фінансів [17].

AI вже трансформують корпоративні фінансові відділи. Пов'язані з фінансами завдання, такі як кредитна оцінка, прийняття рішень і виявлення шахрайства, виконуються спеціалізованими фінансовими установами з використання штучного інтелекту для пришвидшення роботи, а також покращення точності у розрахунках. В свою чергу, інші завдання, такі як оцінка ризиків, складання бюджету та інвестиційне планування, виконуються без використання складних інформаційних технологій [18]. Штучний інтелект проаналізувавши неймовірний об'єм даних за порівняно короткий період часу, може зробити влучні висновки та найточніші прогнози.

Отже, проаналізувавши тематику управління та аналізу витрат особистого бюджету людини, можна зробити висновок, що ця тема є актуальною для значної кількості людей. Одним із підтверджень цьому є значна кількість досліджень у різних країнах.

1.2 Аналіз сервісів-аналогів ведення особистого бюджету

Для формування списку функціональних вимог інформаційної технології необхідно провести аналіз продуктів-аналогів.

Перш за все розглянемо додаток Wallet [19]. Він є одним із найпопулярніших мобільних додатків для управління особистими фінансами. Не дивлячись на це він

має як переваги, так й недоліки. Розглянемо детальніше його функціональні можливості.

При реєстрації користувач повинен ввести поточний баланс (рис.1.1). Це є основою для подальшої роботи в додатку. Не дивлячись на це, не можна ввести від'ємне значення. Дана функціональна можливість вже не передбачає те, що у людини на момент завантаження додатка можуть бути борги чи кредити.

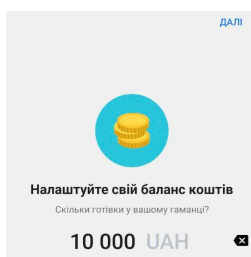


Рисунок 1.1 – Додавання балансу

Джерело: знімок з екрану [19]

При додаванні змін у бюджеті потрібно додати: назву, період, суму, валюту, категорію, тип рахунку та додаткові мітки. Окрім цього, можна було б також додати коментарі для зручнішого використання.

Приклад заповнених параметрів представлений на рисунку 1.2.

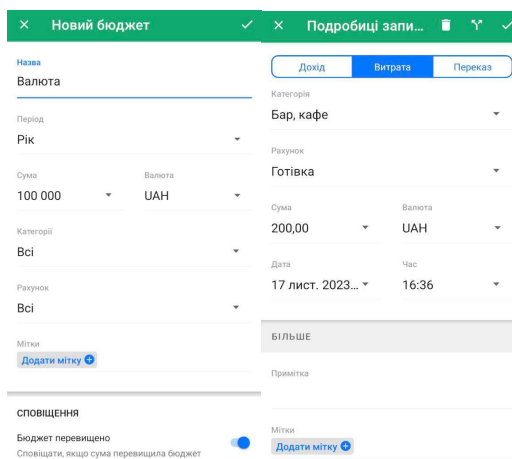


Рисунок 1.2 – Додавання змін у бюджеті

Джерело: знімок з екрану [19]

При виборі змін також можна обрати до чого це відноситься, до доходів, витрат чи переказів, як представлено на рисунку 1.3.

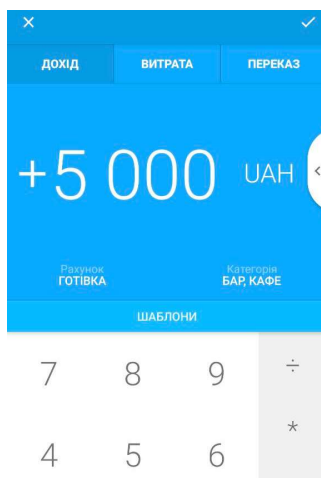


Рисунок 1.3 – Додавання доходів/витрат/переказів

Джерело: знімок з екрану [19]

В додатку категорії вже переналаштовані. Їх також можна змінити, та відредагувати під власні потреби. Також можливе обрання підкатегорії обраної категорії (рис.1.4). Не дивлячись на це, додати більшу кількість категорій неможливо. Це є великим недоліком даного додатку.

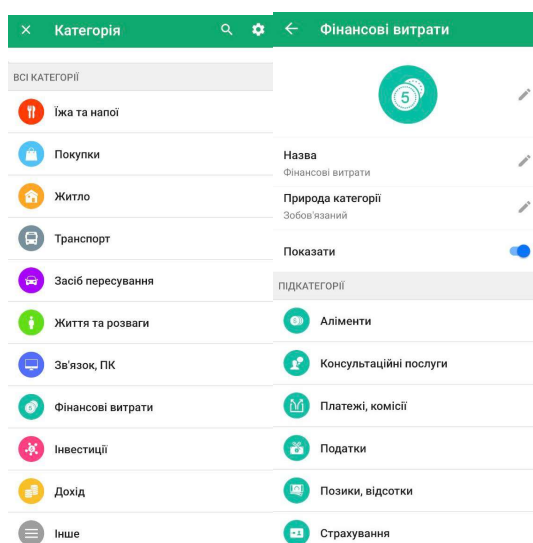


Рисунок 1.4 – Категорії додатку

Джерело: знімок з екрану [19]

Також у додатку можна переглядати статистику. Як зображено на рисунку 1.5 можна переглянути залишки на рахунку, загальні витрати та баланс у валюті.

Не дивлячись на це, при використанні додатка остання з перелічених функцій не працювала. При намаганнях налаштувати її, була надана інформація про розширену платну версію.

Статистика відображається у декількох варіаціях:

- Загальні витрати;
- Загальний баланс;
- Витрати по відношенню сума та місяць.

До недоліків можна віднести обмежені функціональні можливості для перегляду статистичних даних за кожних із напрямків.

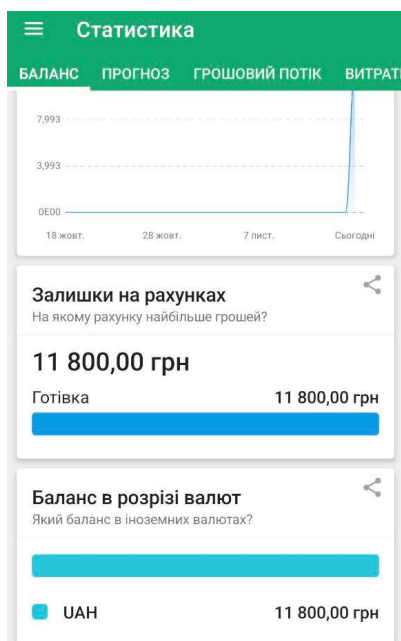


Рисунок 1.5 – Статистика в додатку

Джерело: знімок з екрану [19]

Також в додатку можливий перегляд статистики витрат, доходів. Можна переглядати інформацію за обраною тематикою, проміжком часу (рис.1.6).

Можна відмітити недолік, що відсутня можливість експортування даних за витратами, доходами чи обраними категоріями. Також відсутня можливість створювати додаткові збереження даних.

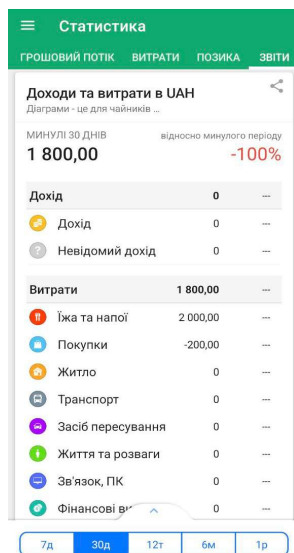


Рисунок 1.6 – Статистика звітів по категоріях

Джерело: знімок з екрану [19]

Окрім цього, в додатку є можливість створювати цілі для формування власного бюджету. Приклад представлено на рисунку 1.7. Усі можливі функціональні можливості Wallet представлено на рисунку 1.8.

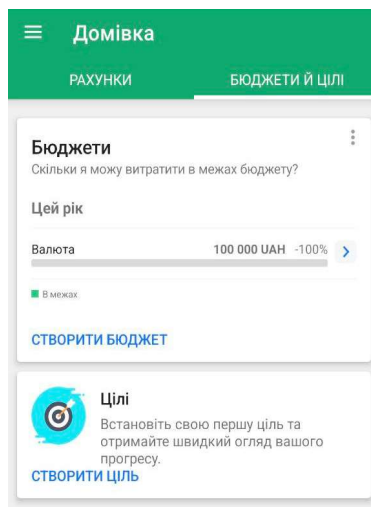


Рисунок 1.7 – Інформація про бюджети та цілі

Джерело: знімок з екрану [19]

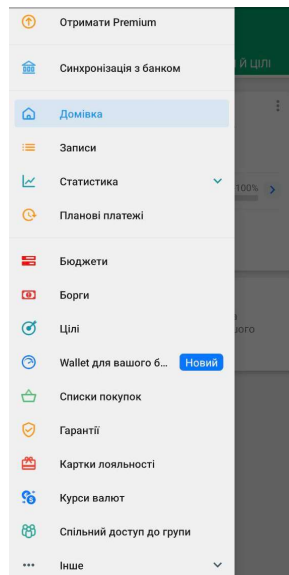


Рисунок 1.8 – Функціональні можливості додатку

Джерело: знімок з екрану [19]

Також у додатку представлена інформації що до курсу валют (рис.1.9), як, наприклад євро та долари. Реалізація саме цього додатку виконана посередньо, адже відсутня інформація, що до того, якого банку чи НБУ представлено курс.

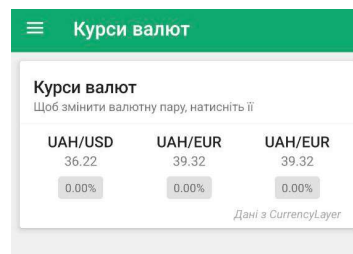


Рисунок 1.9 – Валюти в додатку

Джерело: знімок з екрану [19]

Отже, проаналізувавши додаток, можна зробити висновок, що він має перелік переваг та недоліків. Додаток має розподілення за категоріями та підкатегоріями. Відображення статистики. Хоча, не можна не відмітити, що у Wallet неможливо додавати власні категорії витрат. У додатку також представлено функціональна можливість, яка не реалізований належним чином, як наприклад, синхронізація з

банками, що не працює в Україні. Також лише платна версія Wallet дозволяє користувачеві синхронізувати особисті дані між кількома пристроями.

Наступним розглянемо додаток Money Manager Expense & Budget [20]. Це також доволі популярний додаток. На рисунку 1.10 представлено приклад додавання витрат. Потрібно ввести аналогічну інформацію, як й у минулому додатку, а саме: дату, суму, категорію, коментар.

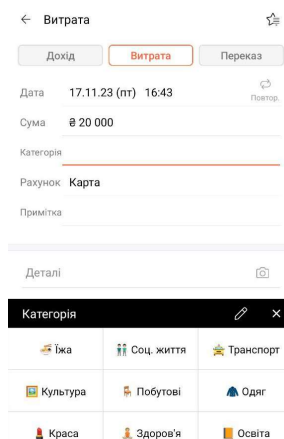


Рисунок 1.10 – Додавання витрат

Джерело: знімок з екрану [20]

Перевагою є можливість відредагувати категорію в момент створення запису. Додавати категорії можна в необмеженій кількості (рис. 1.11).

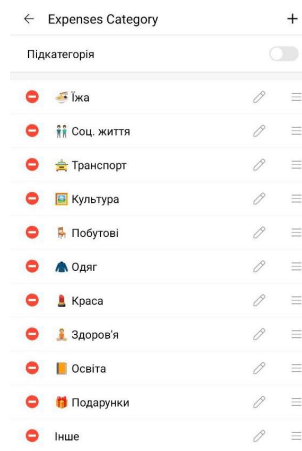


Рисунок 1.11 – Редагування категорій

Джерело: знімок з екрану [20]

Після збереження інформації користувач переходить на головну сторінку, що виконана у вигляді календаря (рис.1.12). У кожній комірці відображається інформація про доходи та витрати.



Рисунок 1.12 – Загальний календар додатку

Джерело: знімок з екрану [20]

Загальну інформацію про записи можна експортувати (рис.1.13). Дані функціональні можливості реалізований не повноцінно, адже дані не можна зберегти саме на пристрій, а лише надіслати через пошту чи інші додатки.

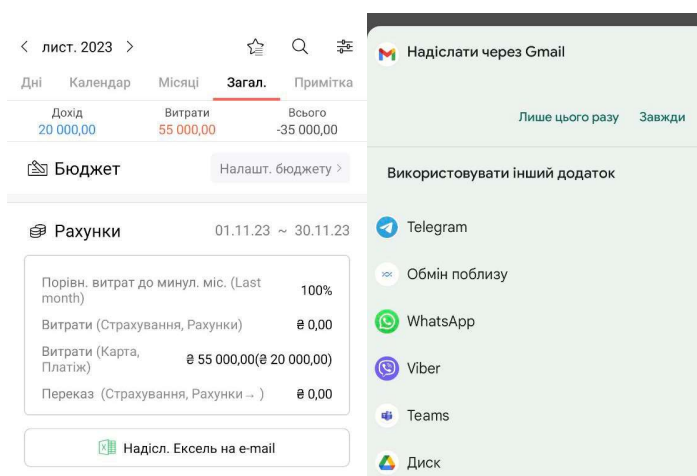


Рисунок 1.13 – Формування та надсилання звітів

Джерело: знімок з екрану [20]

У додатку можна переглянути загальну статистику бюджету у витратах та доходах. Приклад відображення інформації представлено на рисунку 1.14. До недоліків додатку можна віднести те, що відсутня можливість змінювати тип відображення інформації зі статистики.

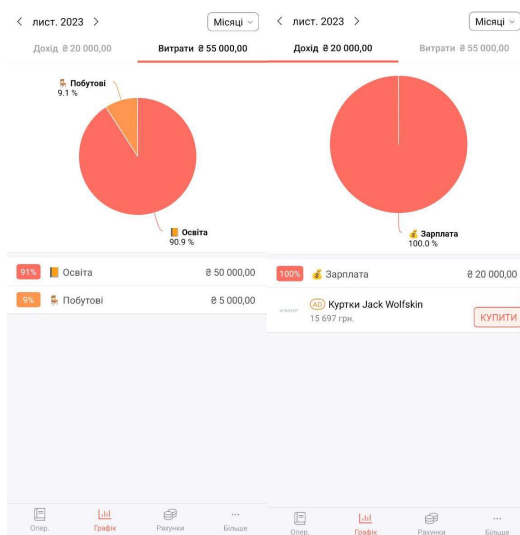


Рисунок 1.14 – Статистика бюджету

Джерело: знімок з екрану [20]

Загальні налаштування додатку представлені на рисунку 1.15. Це включає в себе також створення бекапу даних (рис.1.16). Із недоліків, що бекапи зберігаються не на серверах додатку, а лише надсилаються через пошту чи інші додатки.



Рисунок 1.15 – Налаштування додатку

Джерело: знімок з екрану [20]

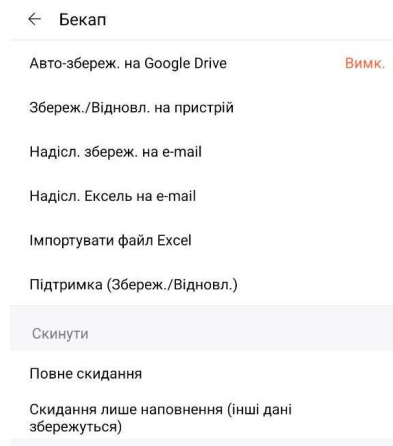


Рисунок 1.16 – Створення бекап збереження

Джерело: знімок з екрану [20]

Окрім цього, в додатку представлена інформація про використання додатку в веб-версії (рис.1.17). Але при переході на сайт, доступу не надається. Виглядає так, ніби він знаходиться в розробці або не використовується взагалі.



Рисунок 1.17 – Інформація про використання додатку в веб-версії

Джерело: знімок з екрану [20]

Отже, проаналізувавши додаток Money Manager Expense & Budget, можна зробити висновок, що він не реалізує необхідні функціональні можливості у повній мірі. Також при використанні було виявлено, що в даному додатку немає функціоналу для синхронізації в режимі реального часу. Навіть при зміні інформації на одному телефоні, зміни не відображаються на інших пристроях. Оновлення відбувається лише раз на годину.

Останнім продуктом-аналогом розглянемо додаток 1Money [21]. На рисунку 1.18 представлена головна сторінка додатку та приклад відображення доданих даних. На момент використання додатка було виявлено, що присутня можливість додавати лише витрати.

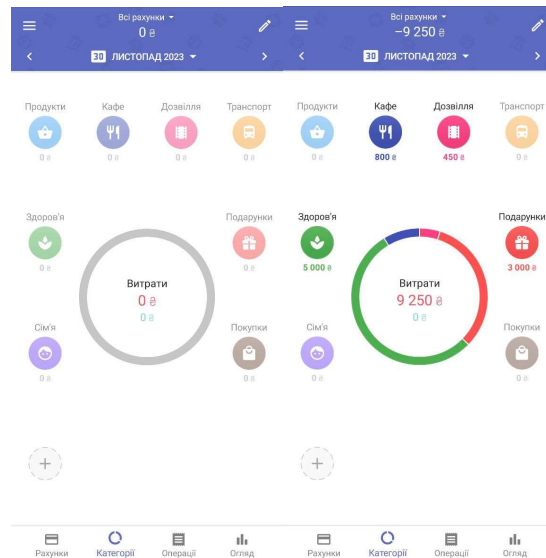


Рисунок 1.18 – Приклад використання додатку

Джерело: знімок з екрану [21]

Як в інших аналогах присутня можливість редагувати категорії для заповнення подальшими даними (рис.1.19). Також гарною функцією додатку є можливість додавати підкатегорії.

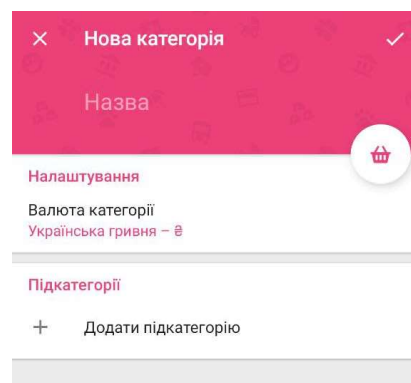


Рисунок 1.19 – Додавання категорії

Джерело: знімок з екрану [21]

Більша частина функціонала відкривається лише при купівлі преміум підписки. Тому оцінити функціональні можливості повною мірою не вдалось. На рисунку 1.20 представлено інформацію про функціональні можливості додатка.

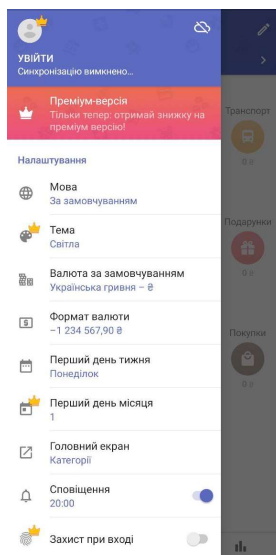


Рисунок 1.20 – Функціональні можливості додатку

Джерело: знімок з екрану [21]

Додаткового відображення статистики окрім як по датам, та загального кругового ввідображення витрат за день, не реалізовано у додатку. Приклад роботи та відображення статистичних даних представлено на рисунку 1.21.

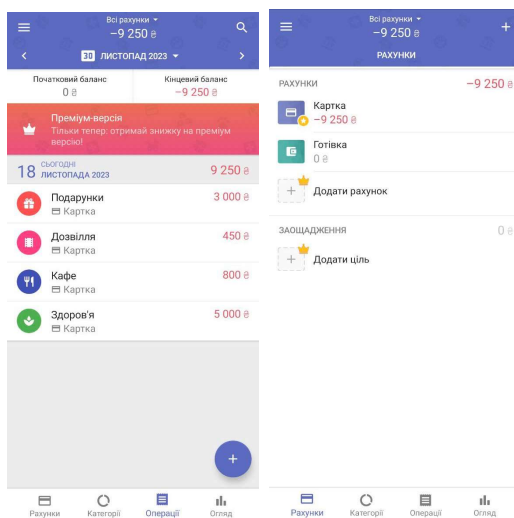


Рисунок 1.21 – Статистика витратів у додатку

Джерело: знімок з екрану [21]

Отже, проаналізувавши додаток можна зробити висновок, більшість інформації та функціоналу присутня лише у платній версії. 1Money дуже рідко оновлюється. Має проблему як й минулий додаток, та у безкоштовній версії відсутня автоматична синхронізація.

У результаті аналізу всіх продуктів аналогів за категоріями, було сформовано перелік із функціоналом. Також сформовані допустимі критерії для розробки інформаційної технології управління та аналізу витрат особистого бюджету.

Детальна інформація у порівнянні додатків Wallet, Money Manager, 1Money та власної розробки представлено в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця

Критерії	Wallet	Money Manager	1Money	Власна розробка
Вибір відображення статистики	+	-	-	+
Відображення витрат/доходів у валюті	+	+	+	+
Відображення курсу обраної валюти	+	-	-	+
Створення власних категорій та редагування створених	-	+	+	+
Синхронізація даних	+	-	-	+
Експорт даних у PDF	-	-	-	+
Експорт даних у Excel	-	+	-	+
Формування цілей для накопичення	+	-	-	+
Використання на різних пристроях	-	+	-	+
Підтримка додатка	-	-	-	+
Надання рекомендацій по запиту користувача	-	-	-	+

Джерело: побудовано автором

Однією із переваг розробленого додатка буде постійна підтримка адміністраторами. Завдяки цьому додаток не буде втрачати актуальності та буде наповнений сучасною інформацією. Постійна підтримка допоможе також в аналізі та вирішенні проблем додатка.

Окрім цього буде розроблено функціональні можливості для надання рекомендацій по запиті користувача. Це дозволить користувачеві швидко розв'язувати питання при роботі з додатком, отримувати відповіді на запитання найближчим часом.

Користувач буде наданий повноцінні функціональні можливості для перегляду статистики у будь-якому форматі. Також буде реалізовано експорт статистичних даних у Excel. Це є зручним функціоналом, для збереження даних для подальшого особистого використання.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Мета та задачі дослідження

Метою роботи є розробка додатку інформаційної технології управління та аналізу витрат особистого бюджету та її реалізація у вигляді зручного та функціонального web-додатку для покращення фінансової грамотності користувачів.

Задачі, які потрібно виконати для досягнення поставленої мети:

1. Оцінити впровадження інформаційних технологій у сферу управління, обліку й аналізу особистих витрат.

Дослідження технологій, застосовуваних у керуванні та аналізі особистих витрат, буде корисним для визначення цілей і напрямків розробки програмного забезпечення з інформаційною технологією [22].

2. Проаналізувати продукти-аналоги та порівняти функціональні можливості, виділити сильні та слабкі сторони.

Під час порівняння аналогів можна чітко визначити вимоги до розробки програмного забезпечення з інформаційною технологією. Також, вирішення цієї задачі допомагає скласти перелік функцій, які можуть бути проблемними та потребувати додаткового часу на розробку й удосконалення [23].

3. Сформування переліку функціональних та не функціональних вимог.

На основі складеного списку формуються умови для створення програмного забезпечення з інформаційною технологією. Це також сприяє створенню критеріїв для вибору технологій, що будуть використані при розробці програми для управління особистими фінансами.

4. Сформувати перелік функціоналу інформаційної технології.

Перелік можливостей обмежує той шлях, по якому розвиватиметься програма з інформаційною технологією, і слугує основою для створення прототипу.

5. Спроектувати розробку інформаційної технології.

Включає в себе графічне представлення діаграм, баз даних та їх опис. Ці кроки дозволяють отримати чітке уявлення про структуру та зв'язків між ними.

6. Розробити візуальну та функціональну частини додатку.

Створення інтерфейсу, дизайну та необхідних функцій для реалізації системи.

Розроблений додаток матиме широку аудиторію. Він охоплюватиме як й студентів, що хочуть налаштувати та ґрунтовно зміцнити фінансову сторону, так й старші люди, що хочуть підготувати фінансову подушку для кризи чи пенсії.

Крім того, розроблена інформаційна технологія повинна містити весь необхідні функціональні можливості для управління та аналізу витрат. Розробимо перелік функціонала користувача:

- Авторизація у системі;
- Реєстрація у системі;
- Зміна особистої інформації;
- Зміна пароля користувача;
- Перегляд загальної інформації у системі;
- Додавання інформації про доходи/витрати/заощадження користувача;
- Редагування інформації про доходи/витрати/заощадження користувача;
- Виведення інформації про кошти у гривневому та доларовому форматі;
- Виведення інформації за обраний період часу;
- Перегляд статистики доходів/витрат/заощаджень користувача;
- Редагування вигляду статистики доходів/витрат/заощаджень користувача;
- Редагування типів даних та діапазон часу для формування статистики доходів/витрат/заощаджень користувача;

- Написання адміністратору питання;
- Перегляд статей щодо фінансової грамотності;
- Експортування даних в Excel;

Розробимо перелік функціонала адміністратора:

- Авторизація у системі;
- Редагування загальних даних;
- Блокування користувачів;
- Функція надання відповіді на запитання користувачів;
- Додавання статей про фінанси;

Для розробки інформаційної технології потрібно також розробити перелік функціональних та нефункціональних вимог. Функціональні вимоги:

- Закритість даних від інших користувачів;
- Шифрування даних, що до фінансів зареєстрованого користувача;
- Мобільна та планшетна адаптація;

Нефункціональні вимоги:

- Вірне збереження даних на сервері;
- Зручне масштабування у разі збільшення кількості користувачів;
- Збереження резервних копій, у разі виникнення проблем на сервері;
- Контроль помилок системи;

Також було проведено планування робіт. Детальна інформація у додатку А

2.2. Обґрунтування вибору інструментів реалізації інформаційної технології у вигляді web-додатку

Було вирішено реалізувати інформаційну технологію у вигляді web-додатку для реалізації усього функціонала. Для реалізації візуальної частини інформаційної технології було обрано Vue.js.

Vue.js є одним із найпередовіших фреймворків. Даний фреймворк часто згадують у порівнянні з Angular і React.js [26]. Не дивлячись на це він відрізняється концепцією використання, такі як директиви та компоненти для керування та відтворення інтерфейсів користувача. Також Vue.js надає можливість керувати HTML, навіть уже на сервері додатка.

Основною причиною використання саме Vue.js – отримання якісного продукту з мінімальними зусиллями у розробці. Завдяки простоті використання та гнучкості, розробники можуть писати різні функціональні можливості.

Також в розробці буде використано Tailwind. Даний підхід використовується для оформлення вебпроект та розширення функціонала для розроблення візуальної частини [27]. Крім того, інструмент має на меті забезпечити гнучкий контроль над процесом проектування.

Також при розробці буде використано додаткові бібліотеки, а саме Flowbite та Chart.js. Адже Flowbite це є екосистема бібліотек з відкритим кодом, що чудово використовуються у поєднанні Tailwind CSS [28].

Бібліотека Chart.js також має відкритий код і є абсолютно безплатний у використанні, що є значною перевагою [29-30]. Крім цього присутня підтримка різних типів візуалізації даних. Chart.js має відносно простий API, але також може бути досить потужним із параметрами налаштування. Також є можливість знайти ресурси про бібліотеку в Інтернеті.

Також буде використано Skeditor5. Це є потужним редактор для розробки програм [31]. Можна використовувати різні плагіни, додатки та бібліотеки, що значно полегшать розробку. Також додатково буде використано Inertia, що полегшує розробку додатків. За допомогою створення контролерів значно легше отримати дані з бази даних та інше.

Для розробки функціональної частини інформаційної технології було обрано Laravel, адже він має багато переваг [32]. Одним з них є легкість у написанні коду. Laravel надає багато функцій та інструментів, які можуть допомогти розробникам писати чистий, ефективний і підтримуваний код. Звичайно можна легко налаштувати будь-яку функціональність у програмі.

Крім того, Laravel має дуже безпечну вбудовану систему контролю доступу [33]. Також він забезпечує надійний механізм обробки винятків і помилок. Laravel дозволяє зберігати паролі в базі даних у зашифрованому вигляді, а не в текстовому вигляді.

Як додаткову бібліотеку для розробки було використано Encryption. Бібліотека надає простий та зручний інтерфейс для шифрування та дешифрування через OpenSSL. Усі зашифровані значення підписуються з використанням коду автентифікації повідомлення. Це означає, що їхнє значення не може бути змінено або підроблено.

Іншою бібліотекою є PhpSpreadsheet. Бібліотека використовується з PHP та дозволяє здійснювати читання та запис даних у форматі таблиць. PhpSpreadsheet дозволяє створювати нові електронні таблиці та редагувати вже існуючі.

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

3.1 Структурно-функціональне моделювання процесу

Щоб побудувати діаграми потрібно: вхідні та вихідні дані, механізми та управління [24].

Вхідними даними: інформація про доходи/витрати, запит користувача, інформація про користувача.

Вихідними даними: статистика доходів/заощаджень/витрат, експортовані файли.

Механізмами: користувач, інформаційна технологія, адаптивне забезпечення.

Управлінням: функції системи, вимоги до виконання.

Розглянемо контекстну діаграму інформаційної технології управління та аналізу витрат особистого бюджету (рис.3.1).



Рисунок 3.1 – Контекстна діаграма системи процесу управління та аналізу витрат особистого бюджету

Джерело: побудовано автором

Розглянемо діаграму декомпозиції першого рівня процесу управління та аналізу витрат особистого бюджету (рис.3.2).

Діаграма декомпозиції першого рівня має такі блоки:

- Реєстрація/авторизація нового користувача;
- Налаштування категорій;
- Додавання інформації про доходи/витрати;
- Формування статистики.

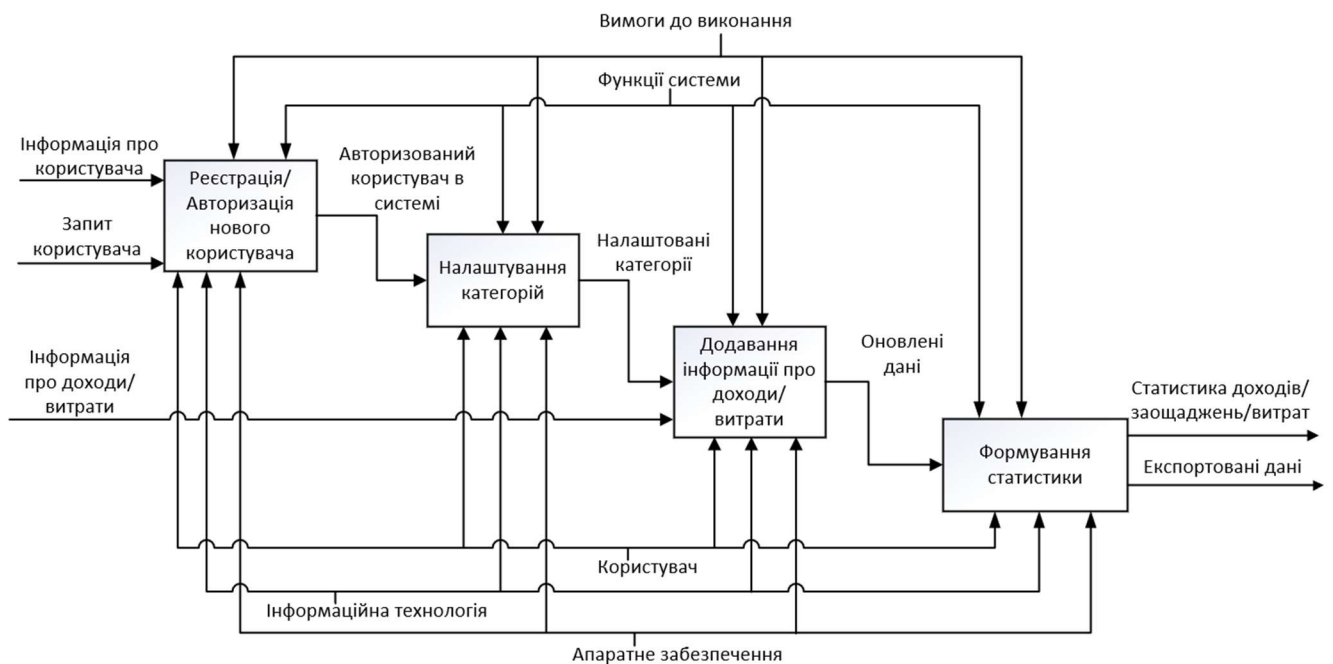


Рисунок 3.2 – Діаграма декомпозиції першого рівня процесу управління та аналізу витрат особистого бюджету

Джерело: побудовано автором

Отже, було розроблено контекстну діаграму та діаграму декомпозиції процесу роботи інформаційної технології управління та аналізу витрат особистого бюджету. Розроблені діаграми чудово демонструють поетапність використання системи.

3.2 Проектування діаграми варіантів інформаційної технології

Для представлення функціоналу інформаційної технології управління та аналізу витрат особистого бюджету було виконано діаграму варіантів використання (рис. 3.3–3.4) та розроблено детальний опис в таблицях 3.1–3.2.

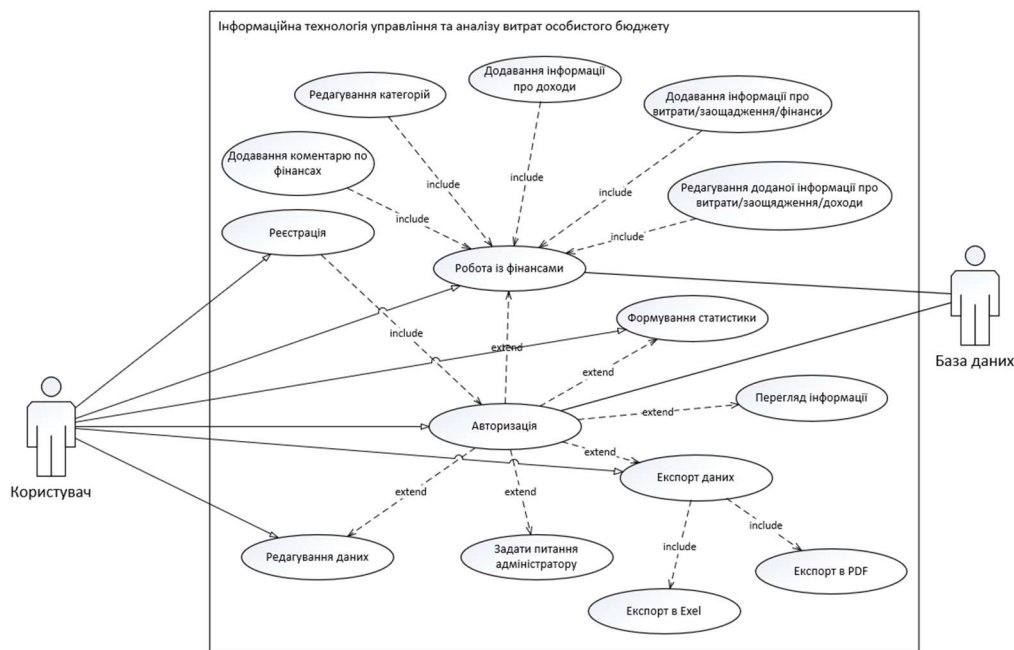


Рисунок 3.3 – Діаграма варіантів використання для користувачів системи

Джерело: побудовано автором

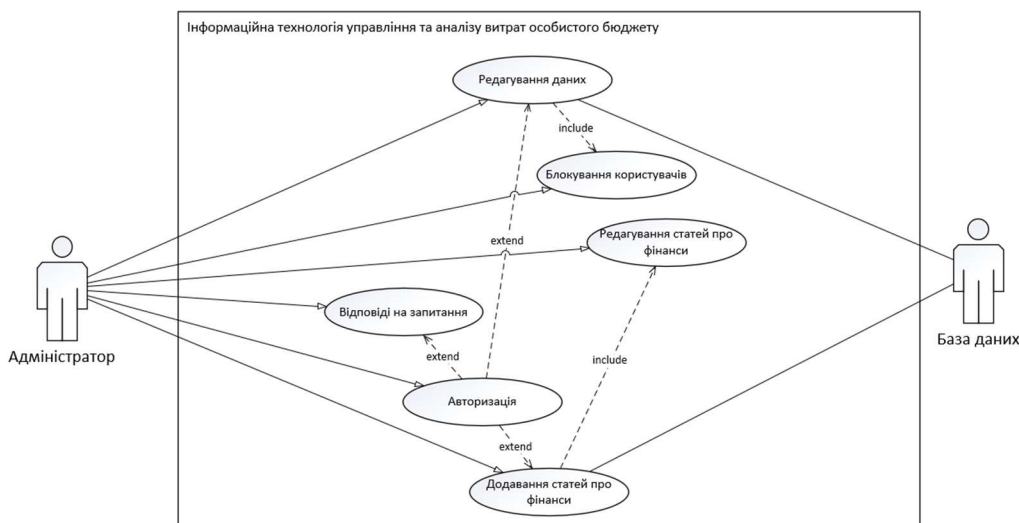


Рисунок 3.4 – Діаграма варіантів використання для адміністратора

Джерело: побудовано автором

Таблиця 3.1. – Опис акторів

№	Іменування актора	Детальний опис
1	Користувач	Зареєстрований користувач може використовувати представлені функціональні можливості інформаційної технології.
2	Адміністратор	Користувач із правами адміністрування, редагування інформації на сайті та іншого.
3	База даних	Користувач, що використовується для збереження та використання інформації.

Джерело: побудовано автором

Таблиця 3.2. – Опис варіантів використання

№	Варіанти використання	Детальний опис
1	Авторизація	Зареєстрований користувач може, використовуючи свої данні, авторизуватися для подальшого користування інформаційною технологією.
2	Реєстрація	Відвідувач може зареєструватися щоб використовувати функціональні можливості інформаційної технології.
3	Редагування даних	Адміністратор має прямий доступ до бази даних для редагування даних
4	Задати запитання адміністратору	Користувач може задати питання стосовно інформаційної технології
5	Робота з фінансами	Авторизований користувач має можливість працювати з фінансами
6	Редагування категорій	Користувач може редагувати категорії при роботі з фінансами
7	Додавання коментарю	Авторизований користувач може додавати коментарі по фінансах

Продовження таблиці 3.2

№	Варіанти використання	Детальний опис
8	Додавання інформації про доходи/ витрати/ заощадження	Користувач може додавати інформацію до бази даних які пов'язана з витратами, заощадженнями та фінансами
9	Редагування інформації про доходи/ витрати/ заощадження	Користувач може редагувати дані які пов'язана з витратами, заощадженнями та фінансами
10	Ведення статистики	Авторизований користувач може самостійно проводити ведення статистики
11	Перегляд інформації	Перегляд інформації доступний на всіх етапах ведення статистики
12	Експорт даних	Користувачі мають можливість загального експорту даних
13	Експорт в PDF	Авторизовані користувачі можуть експортувати дані в документ PDF
14	Експорт в Excel	Авторизовані користувачі можуть експортувати дані в таблиці формату Excel
15	Блокування користувачей	Адміністратор може блокувати користувачів через спеціальну панель з представленими всіма користувачами
16	Відповіді на запитання	Адміністратор може відповідати на задані питання від авторизованих користувачів
17	Додавання статей	Адміністратор може додавати статті в інформаційну технологію
18	Редагування статей	Адміністратор може редагувати статті через спеціальну панель

Джерело: побудовано автором

Отже, було розроблено діаграма варіантів використання з боку адміністратора та користувача. Розроблені діаграми чудово демонструють функціональні можливості інформаційної технології управління та аналізу витрат особистого бюджету.

Окрім цього, спроектовані діаграми використання допомагають краще розуміти структуру інформаційної технології насамперед перед початком розробки. Діаграма допомагає визначити окремі компоненти, їх функціональність та взаємодію, аналізувати вимоги, ідентифікувати можливі проблеми, тощо.

3.3 Структура інформаційної технології

Одним з етапів розробки web-додатка інформаційної технології являється проектування структури та сторінок додатка. Беручи до уваги список функціонала, формується список потрібних сторінок:

- Авторизація. Авторизація користувача в інформаційної технології web-орієнтованого додатка;
- Реєстрація. Реєстрація нового користувача в системі;
- Особиста сторінка. Сторінка для зміни особистих даних, різновид валют, особистого аватару та пароллю.;
- Головна. Сторінка з коротким описом інформаційної технології web-орієнтованого додатка, опис переваг та можливостей до якого має доступ кожен відвідувач;
- Доходи. Сторінка з коротким описом показників доходів витрат та заощаджень з конвертацією в одну з вибраних валют в особистому кабінеті. Сторінка також повинна мати календарний план з можливістю додати та редагувати категорії та додавати суму доходів з можливістю додавання, редагування та видалення коментарів;
- Витрати. Сторінка з показниками інших категорій гаманця з візуальним мінімалістичним представленням календарного плану, в якому легко додавати,

редагувати та видаляти як окремі категорії, так і показники з можливістю вибору потрібного періоду;

- Заощадження. Ця сторінка повинна мати схожий за виглядом та функціоналом зі сторінками про дохідність та витрати;

- Статистика. Сторінка з узагальненою статистикою доходів, витрат та збережень з можливістю візуального відображення різними діаграмами, вибору за категорією, типом та періодом відстежування. Також має бути можливість завантажити статистику в вигляді таблиць та документу;

- Рекомендації. Ця сторінка матиме на меті інформувати користувачів про корисну інформацію за тематикою інформаційної технології з можливістю ставити відмітку та коментувати статтю іншими користувачами;

- Про нас. Детальний опис функціонала інформаційної технології на основі web-орієнтованого додатка та короткий FAQ по найчастішим питанням, стосовно інформаційної технології;

- Обране. Сторінка зі статтями, під якими користувач поставив відмітку;

- Питання. Надати можливість користувачеві через окрему сторінку задавати питання адміністратору.

Також, звертаючи увагу на участь адміністратора в системі, створити потрібні функціональні можливості на окремих сторінках адміністратору для управління інформаційною технологією web-орієнтованого додатка:

- Користувачі. Сторінка з можливістю переглядати та блокувати користувачів;

- Категорії. Редагування категорій користувачів;

- Публікації. Сторінка з функціоналом створення, редагування та видалення публікацій з розширеною можливістю форматування тексту;

- FAQ. Панель керування короткими питаннями з відповідями для відображення на основній сторінці для користувачів;

- Повідомлення. Сторінка для ведення діалогу з користувачами з можливістю видаляти діалоги.

Отже, щоб правильно слідувати вимогам у набору функціонала та можливостей інформаційної технології web-орієнтованого додатка було розглянуто роботу користувача та інформаційної технології.

Більш детальна інформація буде представлена вже при демонстрації роботи інформаційної технології web-орієнтованого додатка. Також, буде докладно показано деякі аспекти використання інформаційної технології не лише з боку користувача, а й адміністратора.

3.4. Проектування бази даних

Важливим компонентом інформаційної технології управління та аналізу витрат особистого бюджету є проектування бази даних. Графічне відображення бази даних сформовано та представлено за всіма правилами розробки логічної моделі бази даних [25]. На рисунку 3.5 представлена розроблена ER-діаграма об'єктів та їх взаємозв'язки.

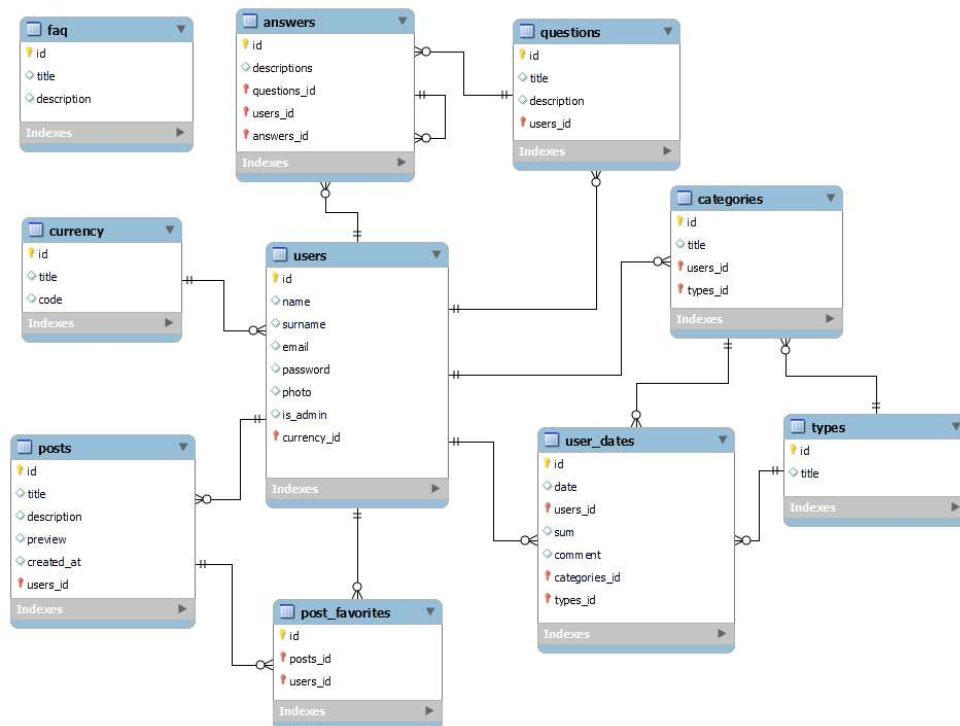


Рисунок 3.5 – Логічна структура бази даних

Джерело: побудовано автором

Уся представлена інформація була спроектована на основі розробленого переліку необхідного функціоналу, функціональних та не функціональних вимог. Було також враховано додаткову інформацію, що буде використана саме у розробленому додатку. Опис кожного об'єкту представлено в таблиці 3.3.

Отже, була розроблена ER-діаграма бази даних для подальшого використання при розробці додатка для управління особистими фінансами. Діаграма була розроблена відповідно правилам нормалізації.

Таблиця 3.3 – Інформація за таблицями діаграми бази даних інформаційної технології управління та аналізу витрат особистого бюджету

№	Таблиця	Поле	Тип	Зміст	Ключі
1	fin_users	fin_id	INTEGER	Ідентифікатор профілю користувача	PK
		fin_name	VARCHAR(255)	Ім'я користувача	
		fin_surname	VARCHAR(255)	Прізвище користувача	
		fin_email	VARCHAR(65)	Електронна пошта зареєстрованого користувача	
		fin_password	VARCHAR(50)	Пароль користувача	
		fin_photo	VARCHAR(200)	Фото завантажено користувачем	
		fin_isadmin	BOOLEAN	Перевірка адміністратора	
		fin_currency_id	INTEGER	Ідентифікатор валюти в залежності від обраної	FK
2	fin_answers	fin_id	INTEGER	Ідентифікатор відповіді користувача чи адміністратора	PK
		fin_description	TEXT	Текст відповіді на запитання	
		fin_question_id	INTEGER	Ідентифікатор запитання задане користувачем адміністратору	FK
		fin_users_id	INTEGER	Ідентифікатор користувача, що задав запитання	FK
		fin_answers_id	INTEGER	Ідентифікатор додаткової відповіді чи запитання	FK
3	fin_question	fin_id	INTEGER	Ідентифікатор запитання задане користувачем адміністратору	PK
		fin_title	VARCHAR(125)	Заголовок створеного запитання користувачем	
		fin_description	TEXT	Текст запитання від зареєстрованого користувача	
		fin_users_id	INTEGER	Ідентифікатор профілю користувача	FK
4	fin_category	fin_id	INTEGER	Ідентифікатор категорії доходів, витрат чи збережень	PK
		fin_title	VARCHAR(65)	Назва створеної категорії доходів, витрат чи збережень	

Продовження таблиці 3.3

№	Таблиця	Поле	Тип	Зміст	Ключі
4	fin_category	fin_users_id	INTEGER	Ідентифікатор користувача, що створив категорію	FK
		fin_types_id	INTEGER	Тип категорії доходів, витрат чи збережень	FK
5	fin_types	fin_id	INTEGER	Ідентифікатор типу збережень	PK
		fin_title	VARCHAR(15)	Назва типу збережень: доходи, витрати чи збереження	
6	fin_users_dates	fin_id	INTEGER	Ідентифікатор доданої інформації користувача: доходи, витрати чи збереження	PK
		fin_date	DATE	Інформація про дату внесення змін інформації	
		fin_users_id	INTEGER	Ідентифікатор користувача, що додав інформацію	FK
		fin_sum	DOUBLE	Інформація про суму у внесення змін	
		fin_comment	TEXT	Коментар користувача, при додаванні змін	FK
		fin_category_id	INTEGER	Ідентифікатор категорії доходів, витрат чи збережень	FK
		fin_types_id	INTEGER	Ідентифікатор типу збережень	FK
7	fin_post_favorites	fin_id	INTEGER	Ідентифікатор проміжних даних	PK
		fin_posts_id	INTEGER	Ідентифікатор статі адміністратора на економічну тематику	FK
		fin_users_id	INTEGER	Ідентифікатор користувача, що додав статтю у збережені	FK
8	fin_posts	fin_id	INTEGER	Ідентифікатор статі адміністратора на економічну тематику	PK
		fin_title	VARCHAR(125)	Заголовок статі доданої адміністратором	
		fin_description	TEXT	Текст статті доданої адміністратором на економічну тематику	
		fin_preview	VARCHAR(155)	Картинка, що доповнює створену статтю	

Продовження таблиці 3.3

№	Таблиця	Поле	Тип	Зміст	Ключі
8	fin_posts	fin_created_at	DATE	Дата створення статті адміністратором	
		fin_users_id	INTEGER	Ідентифікатор адміністратора	FK
9	fin_currency	fin_id	INTEGER	Ідентифікатор додаткової валюти користувача	PK
		fin_title	VARCHAR(25)	Назва додаткової валюти	
		fin_code	VARCHAR(55)	Додатковий код валюти	
10	fin_faq	fin_id	INTEGER	Ідентифікатор питання від користувачів	PK
		fin_title	VARCHAR(125)	Популярне питання від користувачів	
		fin_description	TEXT	Детальна відповідь на запитання користувача	

Джерело: побудовано автором

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

4.1 Програмна реалізація

Програмна реалізація web-додатка для реалізації інформаційної технології для ведення особистого бюджету виконана за допомогою переліку технологій:

- Візуальна частина додатка – Vue.js;
- Функціональна частина додатка – Laravel.

Поєднання Laravel та Vue.js вже є популярним рішенням для розробки багатьох додатків. Структура прозробленого web-додатка інформаційної технології представлена на рисунку 4.1.

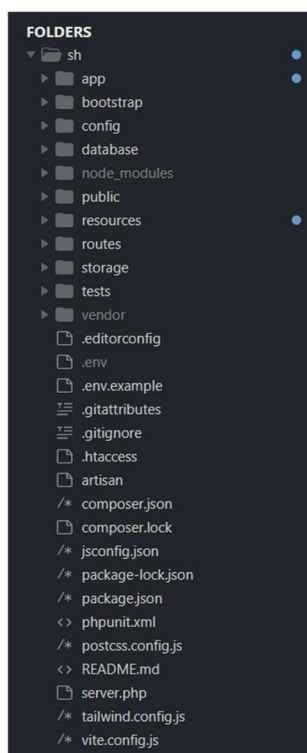


Рисунок 4.1 – Структура додатку інформаційної технології

Джерело: побудовано автором

Також було розроблено базу даних на MySQL в phpMyAdmin [34]. Фізична модель бази даних представлена на рисунку 4.2, а її реалізація в phpMyAdmin представлена на рисунку 4.3.

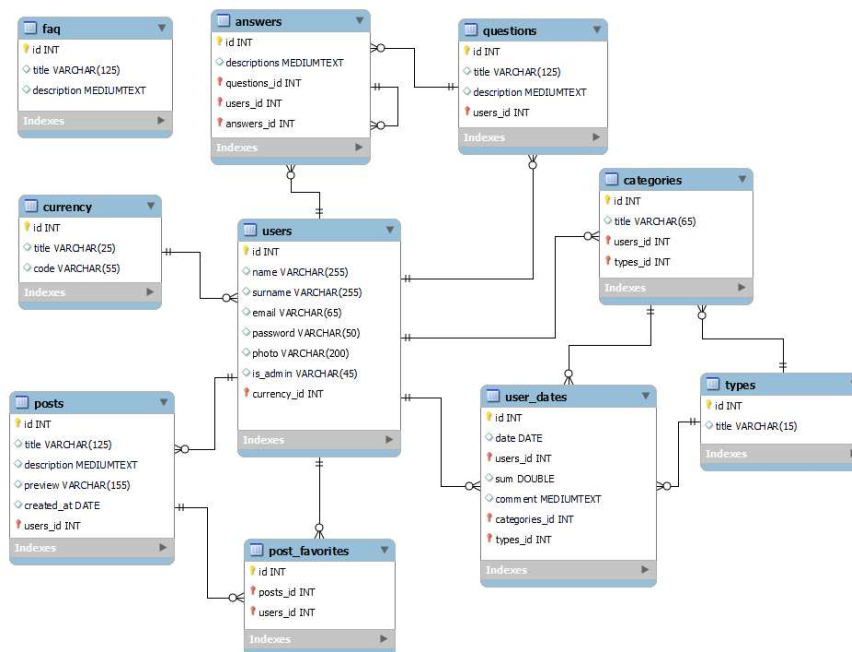


Рисунок 4.2 – Розроблена фізична модель бази даних

Джерело: побудовано автором

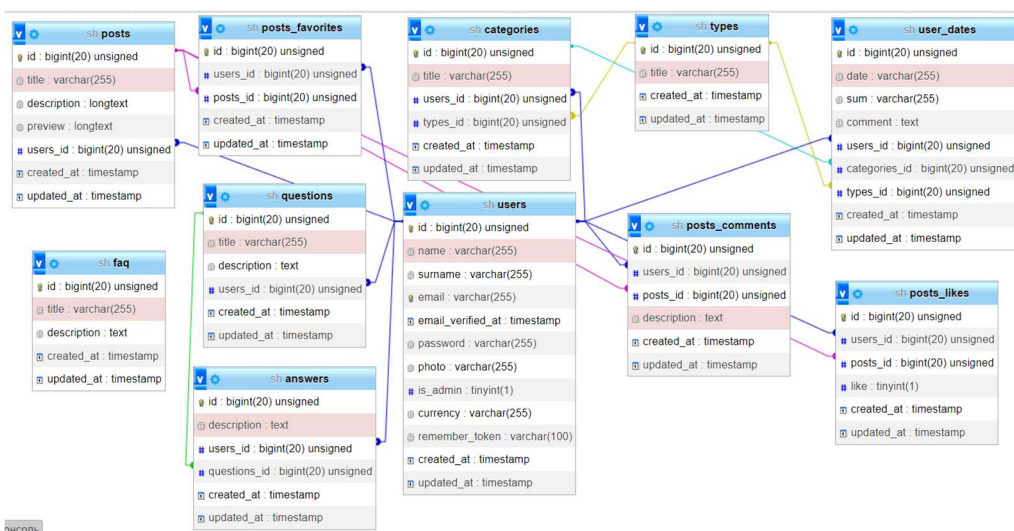


Рисунок 4.3 – Розроблена модель бази даних в phpMyAdmin

Джерело: побудовано автором

Детальну інформацію, що до реалізації кожної таблиці можна переглянути у Додатку В. Крім того, для шифрування використовувалась бібліотека Encryption. Як, наприклад в класі `app/Http/Controllers/UserDatesController.php` та в класі `app/Models/UserDates.php`.

Даний лістинг демонструє приклад шифрування в UserDatesController.php:

```
$response = UserDates::create([
    'date' => $request['date'],
    'sum' => Crypt::encryptString($request['sum']),
    'comment' => $request['comment'],
    'users_id' => Auth::id(),
    'categories_id' => $category['id'],
    'types_id' => $request['types_id']
]);
```

Даний лістинг демонструє приклад дешифрування в UserDates.php:

```
public function getSumAttribute()
{
    return Crypt::decryptString($this->attributes['sum']);
}
```

Після компіляції та використання додатка всі дані, пов'язані з особистими фінансами користувача мають зашифрований вигляд у базі даних (рис.4.4). Детальну інформацію можна переглянути у Додатку Б.



id	date	sum	comment	users_id	categories_id	types_id	created_at	updated_at
13	2023-12-06	eyJpdil6Imh3c3lXbUJ6eUNld1ZrNUdhVIN0Ync9PSlslZhbH...	NULL	1	2	1	2023-12-05 23:23:43	2023-12-05 23:23:43
14	2023-12-06	eyJpdil6ljl3bW9nUTVTRVJpZy9Fd3pOdWI1d3c9PSlslZhbH...	NULL	1	5	1	2023-12-05 23:23:55	2023-12-05 23:23:55
15	2023-12-08	eyJpdil6lNxcy91QVp6dW1GU0hxRUhhWWNpemc9PSlslZhbH...	NULL	1	2	1	2023-12-05 23:24:11	2023-12-05 23:24:11

Рисунок 4.4 – Зашифровані дані в базі даних

Джерело: побудовано автором

Також було виконано вибір валюти та її конвертація. Окрім виведення в гривні користувач обирає додаткове відображення даних. Даний лістинг демонструє приклад конвертації в євро:

```
if (Auth::user()['currency'] == 'EUR') {
    $data['EUR'] = [
        'income' => round($income / $course[0]['buy'], 2),
        'costs' => round($costs / $course[0]['buy'], 2),
        'savings' => round($savings / $course[0]['buy'], 2),
        'remains' => round($remains / $course[0]['buy'], 2)
    ];
}
```

В залежності від вибору користувача можна виконати виведення у іншій валюті, а саме доларах. Даний лістинг демонструє приклад конвертації в долари:

```

if (Auth::user()['currency'] == 'USD') {
    $data['USD'] = [
        'income' => round($income / $course[1]['buy'], 2),
        'costs' => round($costs / $course[1]['buy'], 2),
        'savings' => round($savings / $course[1]['buy'], 2),
        'remains' => round($remains / $course[1]['buy'], 2)
    ];
}

```

Також при розробці додатку інформаційної технології було виконано розробку експорт даних в Exele формат. Як було описано в попередніх пунктах кваліфікаційної роботи, для цього було використано додаткову бібліотеку `phproffice`. Даний лістинг демонструє роботу функції:

```

function exportExcel(Request $request)
{
    $spreadsheet = new Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $data = UserDates::where('users_id', Auth::id())-
>with('category', 'type')->get();

    $header = [
        "A" => "Дата", "B" => "Тип", "C" => "Категорія", "D" => "Сума",
        "E" => "Коментар"
    ];
    foreach ($header as $key => $value) {
        $sheet->setCellValue($key . 1, $value);
    }

    $nextIndex = 2;
    foreach ($data as $item) {
        $sheet->setCellValue('A' . $nextIndex, $item['date']);
        $sheet->setCellValue('B' . $nextIndex, $item['type']['title']);
        $sheet->setCellValue('C' . $nextIndex,
$item['category']['title']);
        $sheet->setCellValue('D' . $nextIndex, $item['sum']);
        $sheet->setCellValue('E' . $nextIndex, $item['comment']);
        $nextIndex += 1;}

    $writer = new Xlsx($spreadsheet);
    return $this->generateFileExcel($writer, 'Data.xlsx');}

```

Наступний лістинг демонструє міграцію даних для бази даних, а саме створення таблиці категорій, за допомогою функціоналу Laravel:

```

public function up(): void
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->foreignId('users_id');
        $table->foreignId('types_id');
        $table->timestamps();
    });
    Schema::table('categories', function (Blueprint $table) {
        $table->index('types_id');
        $table->foreign('types_id')->references('id')->
>on('types')->onDelete('cascade');
        $table->index('users_id');
    });
}

```

Для демонстрації коректної роботи функції виконаємо додавання тестових даних. Наступний лістинг демонструє додання тестових даних:

```

public function run(): void
{
    DB::table('categories')->insert([
        [
            'title' => 'Заклади',
            'users_id' => 1,
            'types_id' => 1
        ],
        [
            'title' => 'Продукти',
            'users_id' => 1,
            'types_id' => 1
        ],
        [
            'title' => 'Комунальні',
            'users_id' => 1,
            'types_id' => 1
        ],
        [
            'title' => 'Аптека й лікарні',
            'users_id' => 1,
            'types_id' => 1
        ]
    ]);
}

```

Успішність додання тестових даних представлена на рисунку 4.5.













	id	title	users_id	types_id	created_at	updated_at
<input type="checkbox"/>   	1	Заклади	1	1	NULL	NULL
<input type="checkbox"/>   	2	Продукти	1	1	NULL	NULL
<input type="checkbox"/>   	3	Комунальні	1	1	NULL	NULL
<input type="checkbox"/>   	4	Аптека й лікарні	1	1	NULL	NULL

Рисунок 4.5 – Приклад успішного додання даних

Джерело: побудовано автором

4.2 Використання додатку інформаційної технології користувачем

Головна сторінка зустрічає користувача мінімалістичним дизайном з коротким описом системи, поєднуючи в собі простоту та привабливість, яка приковує увагу від самого початку користування (рис. 4.6).

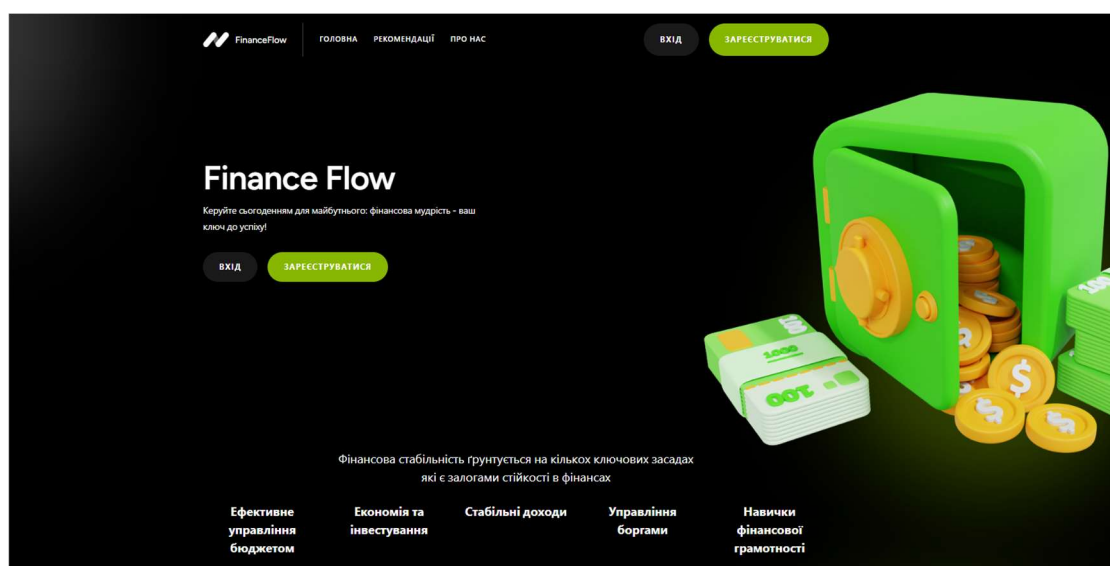


Рисунок 4.6 – Демонстрація головної сторінки

Джерело: знімок з екрану [35]

В наступній частині сторінки, можна побачити головні переваги інформаційної технології на основі додатка, що зображені на рисунку 4.7.

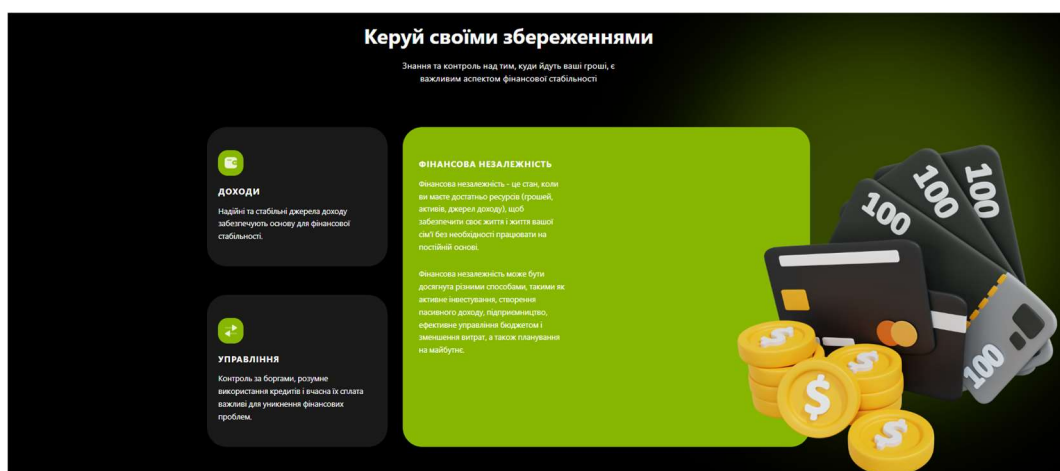


Рисунок 4.7 – Короткий опис інформаційної технології

Джерело: знімок з екрану [35]

В кінці головної сторінки вказані переваги інформаційної технології та ще раз пропонує пройти на сторінку авторизації, щоб почати використання розширеним функціоналом додатком інформаційної технології на рисунку 4.8.



Рисунок 4.8 – Основні переваги додатка інформаційної технології

Джерело: знімок з екрану [35]

Для того, щоб використовувати усі функціональні можливості, потрібно виконати реєстрацію. Форма для реєстрації зображена на рисунку 4.9.

FinanceFlow ГОЛОВНА РЕКОМЕНДАЦІЇ ПРО НАС ВХІД ЗАРЕЄСТРУВАТИСЯ

Реєстрація

ІМ'Я

ПРІЗВИЩЕ

ПОЧТА

ПАРОЛЬ

ПОВТОРІТЬ ПАРОЛЬ

ЗАРЕЄСТРУВАТИСЯ

© 2023 СумДУ. Всі права захищені.

Рисунок 4.9 – Сторінка реєстрації користувача

Джерело: знімок з екрану [35]

Для успішної реєстрації потрібно заповнити поля з ім'ям, прізвищем, електронною поштою та двічі ввести пароль. Приклад заповненої форми представлено на рисунку 4.10.

FinanceFlow ГОЛОВНА РЕКОМЕНДАЦІЇ ПРО НАС ВХІД ЗАРЕЄСТРУВАТИСЯ

Реєстрація

ІМ'Я

ПРІЗВИЩЕ

ПОЧТА

ПАРОЛЬ

ПОВТОРІТЬ ПАРОЛЬ

ЗАРЕЄСТРУВАТИСЯ

© 2023 СумДУ. Всі права захищені.

Рисунок 4.10 – Сторінка реєстрації з заповненими полями

Джерело: знімок з екрану [35]

Всі поля є обов'язковими для заповнення, тому користувач отримує нотифікацію про необхідність заповнення форм даними (рис. 4.11).

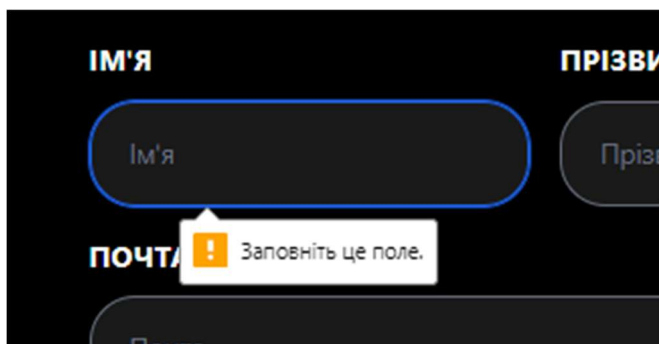


Рисунок 4.11 – Приклад нотифікації у разі незаповнення форми

Джерело: знімок з екрану [35]

Доступ до головного функціоналу відкривається після реєстрації. Для цього, у випадку вже існуючого аккаунту, потрібно виконати авторизацію. При розробці було виконано перенаправлення користувача на сторінку авторизації. Форма для авторизації представлена на рисунку 4.12.

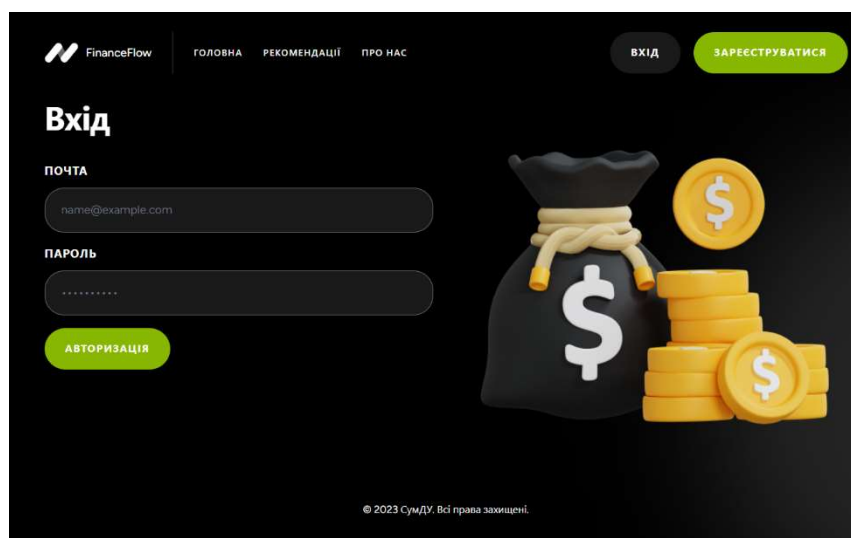


Рисунок 4.12 – Сторінка авторизації

Джерело: знімок з екрану [35]

Після авторизації користувач перенаправляється на особисту сторінку (рис.4.13). На сторінці можна змінити свої дані, а також завантажити свій аватар. Приклад представлено на рисунку 4.14.

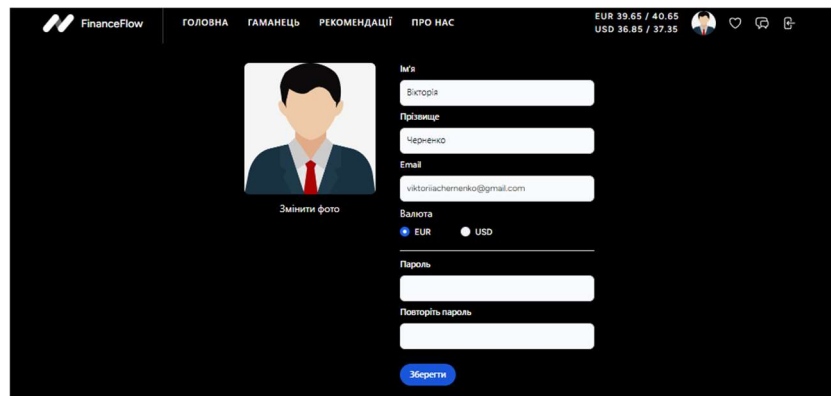


Рисунок 4.13 – Особиста сторінка користувача

Джерело: знімок з екрану [35]

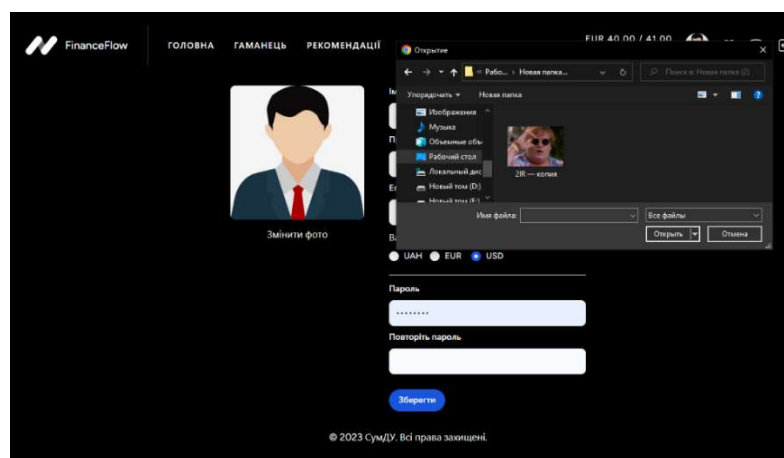


Рисунок 4.15 – Налаштування даних в особистому кабінеті

Джерело: знімок з екрану [35]

Задля підвищення безпеки, змінення пароля на новий потребує від власника знання старого паролю. Також можна змінити валюту для виведення даних загальної інформації про доходи, витрати та заощадження (рис. 4.16).



Рисунок 4.16 – Вибір типу валюти для виведення даних

Джерело: знімок з екрану [35]

Авторизувавшись, користувач отримує повний доступ до функціонала системи. Для зручності користувача, функціональні можливості розділена на спеціальні підрозділи (рис.4.17), а саме:

- Доходи;
- Витрати;
- Збереження;
- Статистика.

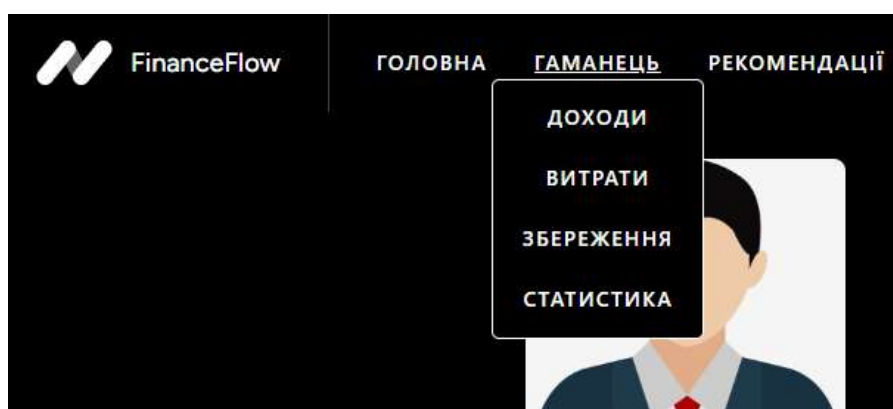


Рисунок 4.17 – Набір функціонала користувача

Джерело: знімок з екрану [35]

Перейшовши на «Доходи», користувач може переглядати детальну інформацію про особисті доходи (рис.4.18).

Категорія	Доходи за місяць																															Сума
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
✓ Заробітна плата	0	0	0	0	0	0	0	0	0	16732	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17105	0	0	0	0	0	33837
✓ Соціальна допомога	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	730	
✓ Інтернет дохід (донат)	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	
✓ Продаж речей	0	0	0	0	0	0	0	0	0	0	0	12650	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13250	
✓ Додатковий підробіток	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	200	200	200	200	0	0	0	0	0	0	0	0	1000	
✓ Продаж послуг	0	2500	0	0	0	0	0	0	0	0	0	0	0	0	0	2500	0	0	0	0	0	0	0	0	0	0	0	0	0	2500	7500	
	0	2500	0	0	0	0	100	0	0	16732	0	0	12650	0	0	2500	0	200	270	800	930	200	0	0	17105	0	0	0	0	2650	0	

Рисунок 4.18 – Доходи користувача

Джерело: знімок з екрану [35]

На цій сторінці користувач може маніпулювати своїми доходами, додавати нові категорії та змінювати місяць для відображення даних (рис.4.19-20). Додані категорії можна як редагувати, так й видаляти.

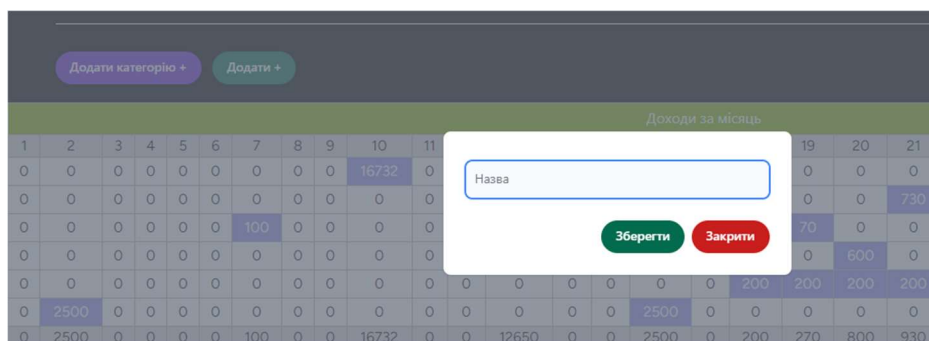


Рисунок 4.19 – Додавання нової категорії

Джерело: знімок з екрану [35]

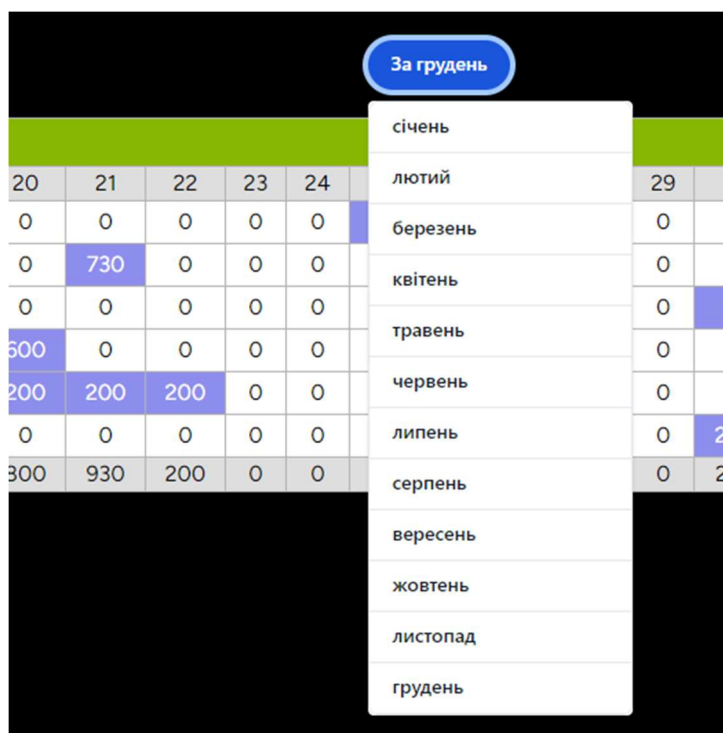


Рисунок 4.20 – Перегляд доходів за окремий період

Джерело: знімок з екрану [35]

Над таблицею також знаходиться інформація про загальні витрати, доходи та заощадження за обраний місяць (рис. 4.21). Також відображається інформація про суму, що ще можна використовувати. Калькуляція залишку включає в себе віднімання від доходів суму витрат та заощаджень.

Під сумою в гривні відображається конвертація сум у обрану раніше валюту на особистій сторінці. У такому випадку це долари. Як можемо бачити на рисунку 4.21, в шапці профілю відображається актуальна інформація по відповідності долара та євро до гривні.



Рисунок 4.21 – Перегляд загальної інформації за окремий період
Джерело: знімок з екрану [35]

У витратах користувача вже є створений шаблон витрат, але нічого не заважає його підлаштувати під себе, змінивши деякі категорії або додавши свої (рис. 4.22).

Категорія	Витрати за місяць																															Сума
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Заклади	0	150	0	0	0	0	0	375	0	0	0	742	0	80	0	0	65	0	0	0	0	405	0	0	0	0	0	200	0	0	0	2017
Продукти	0	1570	0	0	0	0	0	1107	0	0	0	0	0	0	856	0	0	0	0	0	0	1260	0	0	0	0	0	0	0	1630	0	6423
Комунальні	0	0	0	0	0	0	0	0	0	0	2640	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2640	
Аптека й лікарні	0	0	0	0	0	0	0	0	0	0	670	0	0	0	0	0	0	0	0	0	230	0	0	0	0	0	0	0	0	0	900	
Транспорт	16	0	0	16	0	0	0	16	0	0	16	0	16	0	16	0	16	0	16	0	16	16	0	16	0	16	0	0	0	192		
Підписки	0	0	0	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	320	
Подарунки	0	0	0	0	0	2483	0	0	0	0	0	0	0	0	0	0	705	0	0	0	0	0	0	0	0	0	0	0	0	10570		
Догляд	0	0	0	0	0	0	0	0	0	350	0	0	0	0	0	0	0	0	0	0	0	120	0	0	800	0	0	0	0	1270		
Одяг	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5600	0	0	0	0	0	0	0	0	1500	0	0	0	0	0	7100		
Хоббі	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1000	0	0	0	1000		
Борги	0	0	0	1200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1200		
Інше	0	0	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	0	0	0	0	0	0	70	0	0	0	0	0	140		
	16	1720	0	1216	2483	320	0	391	1107	1020	2656	742	86	80	16	6456	0	796	0	16	230	16	1801	1500	86	800	1016	200	7382	1630	0	

Рисунок 4.22 – Сторінка із витратами користувача
Джерело: знімок з екрану [35]

Для додавання змін необхідно натиснути на кнопку «Додати». Після цього відкриється модальне вікно (рис.4.23) для додання інформації про витрати, доходи

чи збереження. Місце, куди буде збережено дані, залежить від сторінки на якій знаходиться користувач.

Крім цього, можна додавати додаткові коментарі, для подальшого перегляду інформації та уточнення, на що саме, наприклад були витрачені кошти. Для того, щоб переглянути інформацію по доданим змінам необхідно натиснути на клітинку. Приклад зображено на рисунку 4.24.

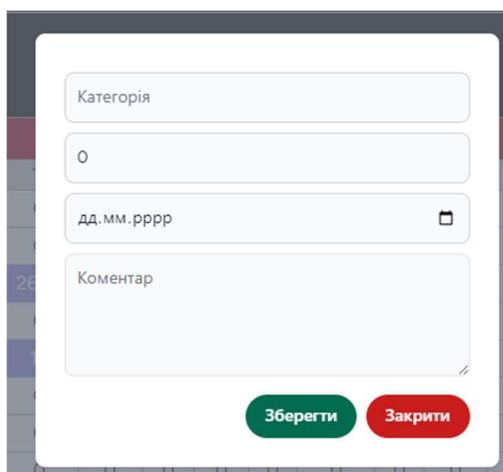


Рисунок 4.23 – Модальне вікно для додання змін у таблицю

Джерело: знімок з екрану [35]

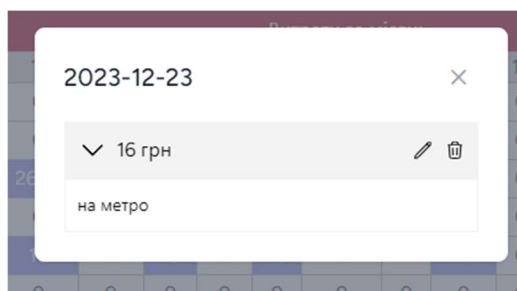


Рисунок 4.24 – Приклад додадих змін

Джерело: знімок з екрану [35]

Також при доданні декількох змін на одну дату та категорію, сума буде складатися. При натисканні на клітинки з такими даними вони будуть відображатися окремо (рис.4.25). Будь які додані дані можна редагувати та видаляти за потреби в незалежності один від одного.

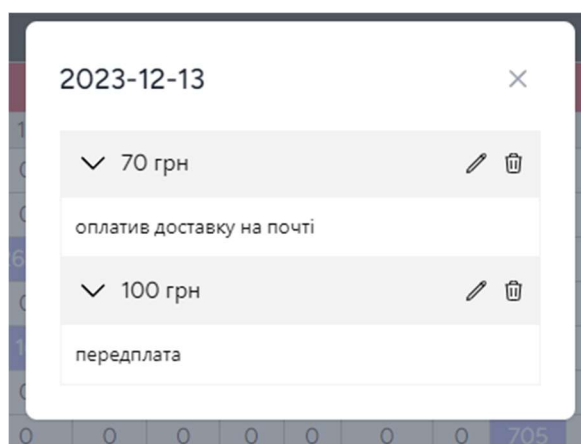


Рисунок 4.25 – Приклад декількох додачих змін

Джерело: знімок з екрану [35]

Сторінка із заощадженнями має аналогічну функціональну можливість, як сторінка із доходами та витратами (рис.4.26).



Рисунок 4.26 – Сторінка із заощадженнями

Джерело: знімок з екрану [35]

Після перегляду усіх сторінок з ведення фінансів, а саме доходів, витрат та збережень можна відмітити особливість, що кожна із цих функціональностей має власний перелік категорій. Кожна з них не впливає на іншу.

Крім того, кожна із таблиць виконана так, щоб сумувати дані по категорії та за день по кожному із напрямлень. Це все дозволяє переглядати статистику. Функціональні можливості статистики був винесений на окрему сторінку.

При відкриванні сторінки зі статистикою відображається перший тип діаграми, а саме круговий (рис.4.27).

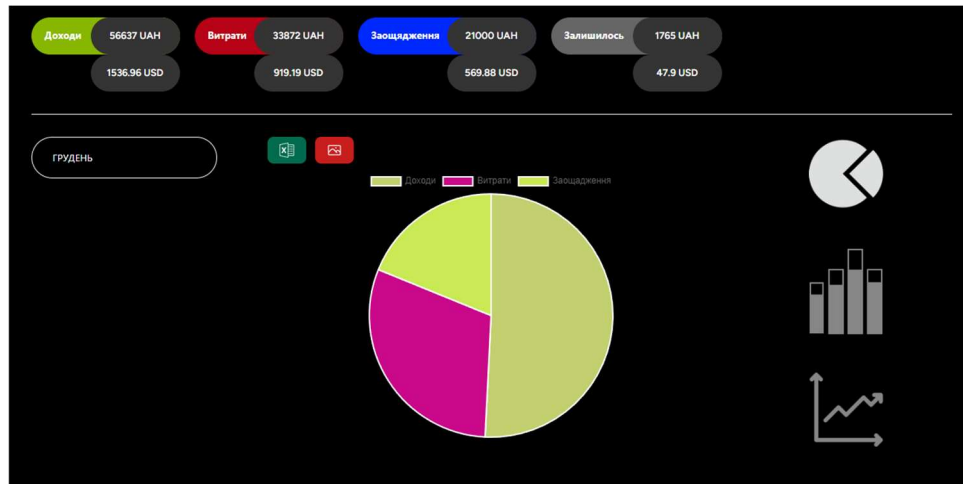


Рисунок 4.27 – Кругова діаграма

Джерело: знімок з екрану [35]

У круговій діаграмі відображається інформація про витрати, доходи та заощадження за обраний місяць. Це дозволяє наочно порівняти розподіл грошей користувача.

Також користувач може обирати тип діаграми справа на сторінці. В залежності від обраної діаграми буде видозмінюватися функціональна можливість. Наприклад, при обранні стовпчикової діаграми на рисунку 4.28.

Зліва на сторінці додається можливість обирати категорії якої групи будуть відображені в діаграмі. На прикладі зображено статистику витрат за грудень за категоріями, де найбільше витрат було здійснено на подарунки.

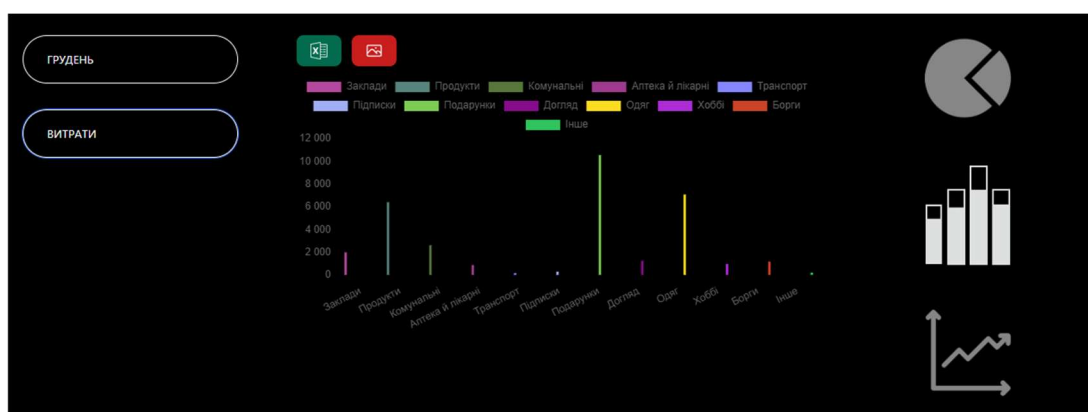


Рисунок 4.28 – Стовбчикова діаграма

Джерело: знімок з екрану [35]

При використанні лінійної діаграми користувач може обирати місяць, тип та категорію для виведення інформації (рис. 4.29).

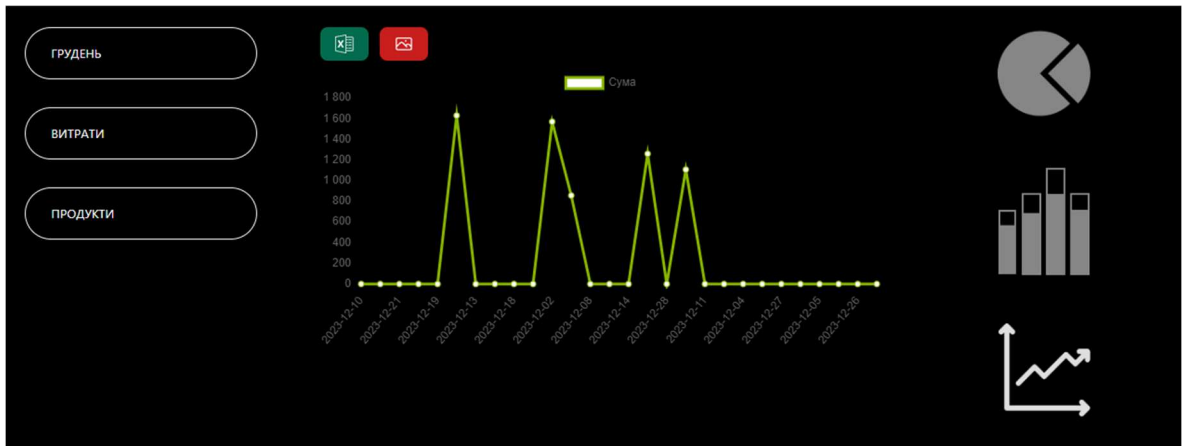


Рисунок 4.29 – Лінійна діаграма

Джерело: знімок з екрану [35]

Функціональні можливості додатка інформаційної технології не закінчуються лише переглядом статистики. Статистику також можна експортувати *xlsx* та зберегти зображення діаграми статистики. Для цього є відповідні функціональні можливості на сторінці (рис.4.30).

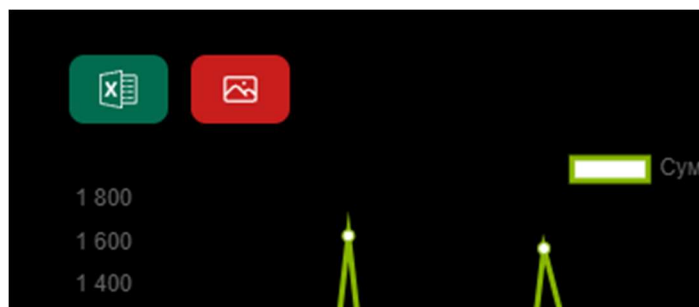


Рисунок 4.30 – Функціональні можливості для зберігання статистики

Джерело: знімок з екрану [35]

Після натискання відповідної клавіші, почнеться завантаження даних статистики в файли відповідного формату (рис. 4.31). Приклади збережених даних представлено на рисунку 4.32-33.

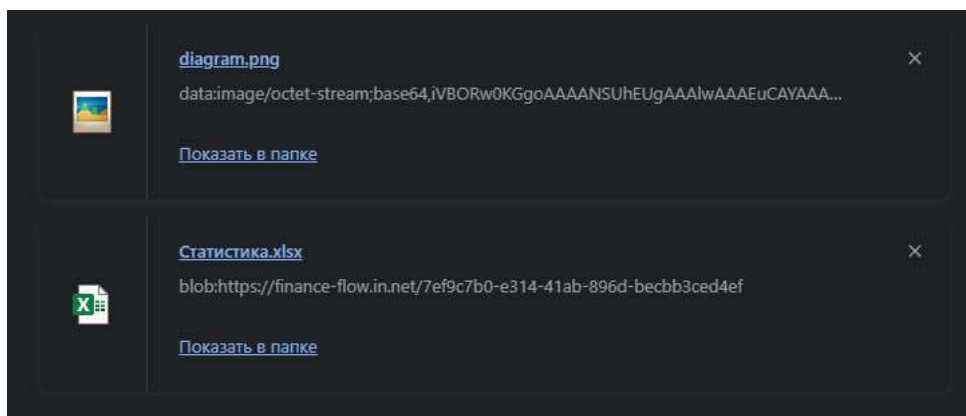


Рисунок 4.31 – Завантаження файлів даних статистики

Джерело: знімок з екрану [35]

	A	B	C	D	E	F
1	Дата	Тип	Категорія	Сума	Коментар	
2	2023-11-28	Витрати	Заклади	100		
3	2023-11-09	Витрати	Неочікувані	500		
4						
5						

Рисунок 4.32 – Інформація із таблиці

Джерело: знімок з екрану [35]

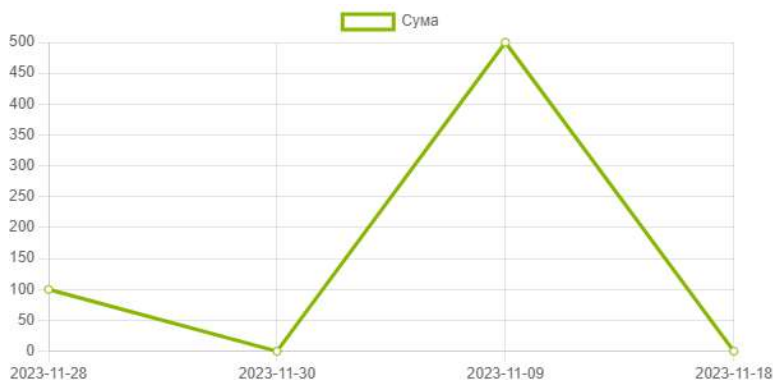


Рисунок 4.33 – Збережений графік

Джерело: знімок з екрану [35]

Також користувач можна прочитати цікаві статті та рекомендації на економічну тематику на сторінці із рекомендаціями (рис.4.34).

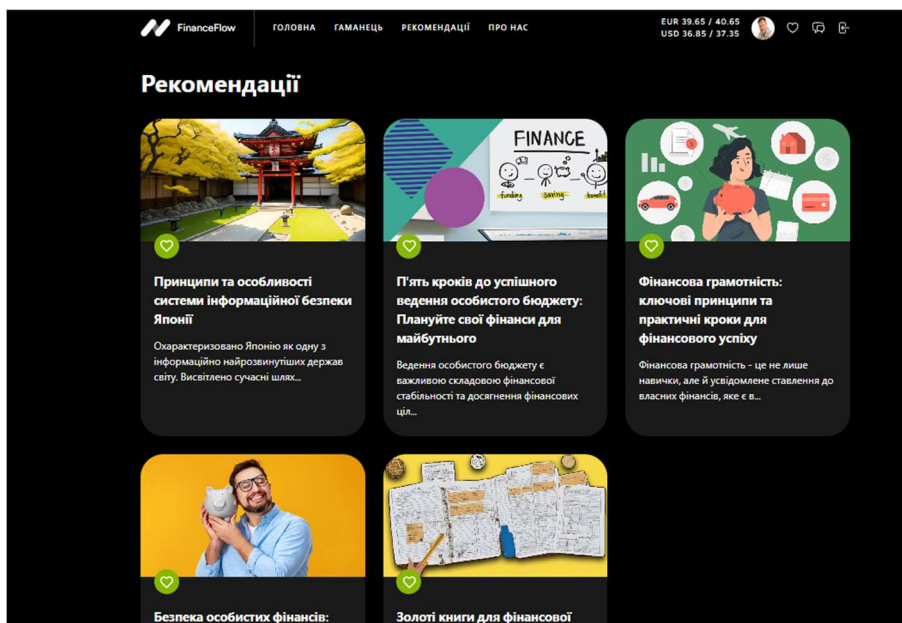


Рисунок 4.34 – Сторінка із рекомендаціями

Джерело: знімок з екрану [35]

Перейшовши на сторінку статі, користувач може залишити коментар та додати статі до обраних за допомогою сердечка у правому верхньому куті (рис.4.35). Приклад доданого коментаря на рисунку 4.36.

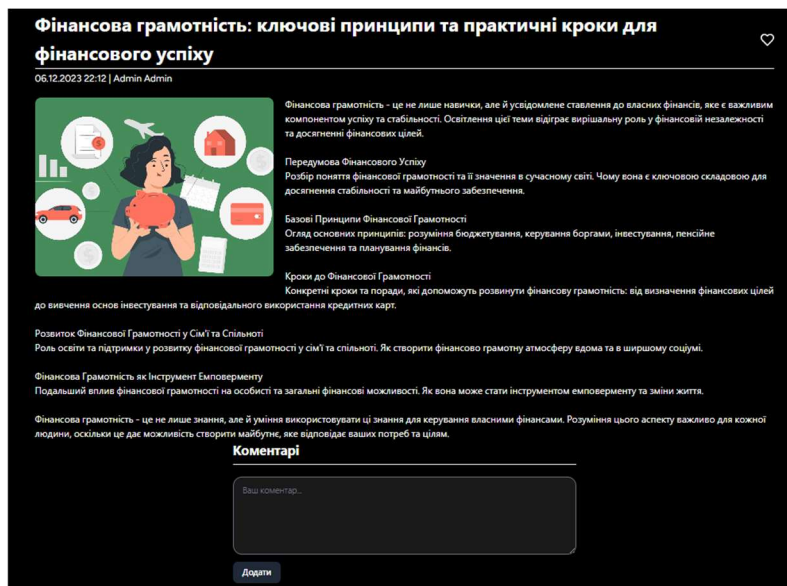


Рисунок 4.35 – Сторінка статі

Джерело: знімок з екрану [35]

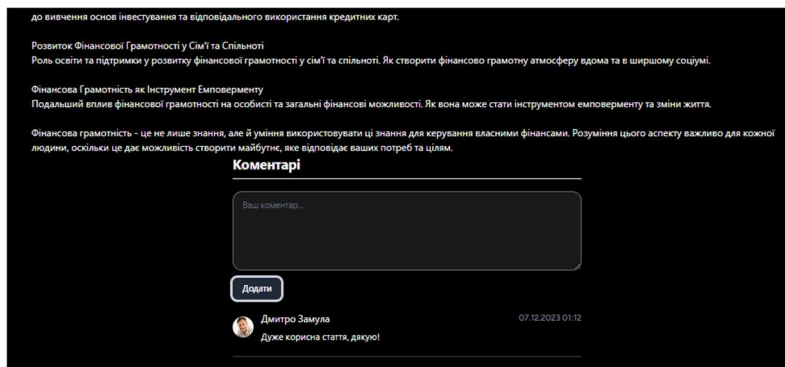


Рисунок 4.36 – Додавання коментарю

Джерело: знімок з екрану [35]

При доданні статі до обраних, користувач отримує нотифікацію про збереження (рис.4.37).

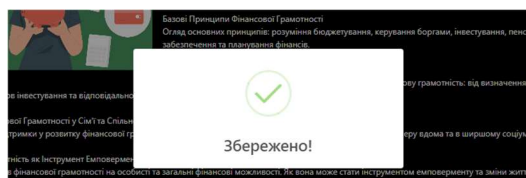


Рисунок 4.37 – Нотифікація про збереження

Джерело: знімок з екрану [35]

На сторінку із обраними статтями можна перейти натиснувши на значок біля профілю користувача. Приклад відображення збережених статей на рисунку 4.38.

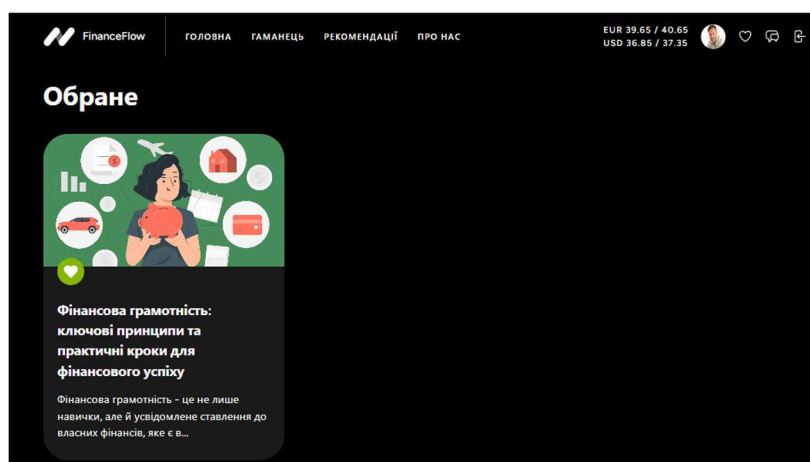


Рисунок 4.38 – Приклад збереженої статі

Джерело: знімок з екрану [35]

На сторінці «Про нас» можна ознайомитися з інформацією про розробників додатка інформаційної технології (рис. 4.39).

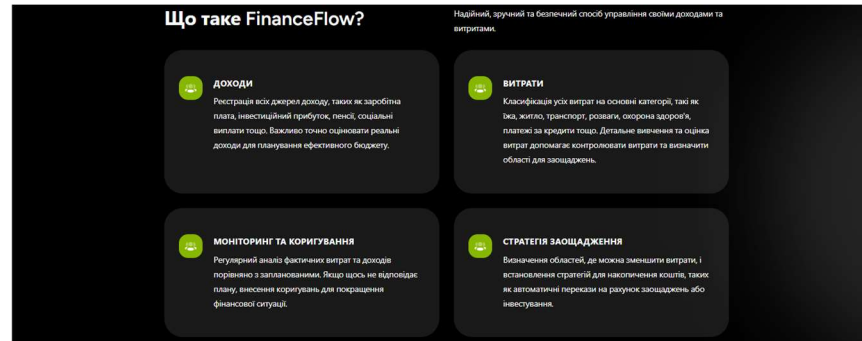


Рисунок 4.39 – Загальна інформація

Джерело: знімок з екрану [35]

Наступна частина сторінки складається з популярних запитань, що можуть надходити від користувачів (рис.4.40).

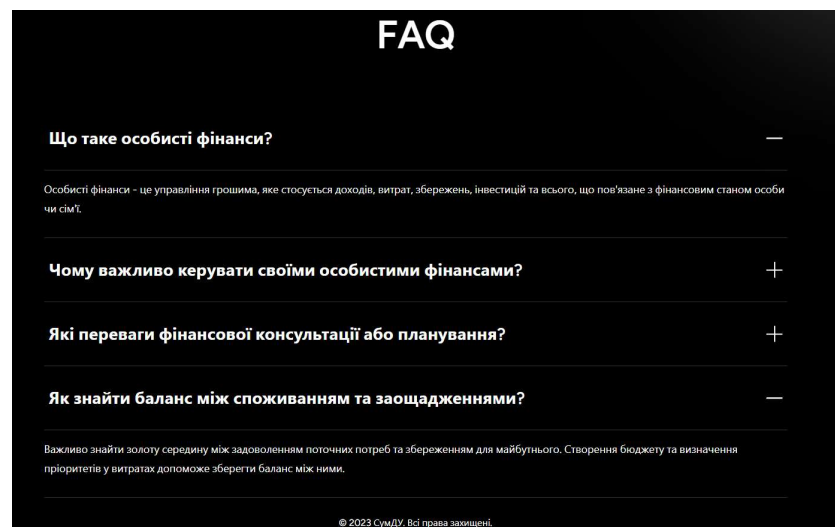


Рисунок 4.40 – Приклад запитання

Джерело: знімок з екрану [35]

Також було реалізовано додаткову функціональну можливість, за допомогою якого користувач має можливість задати питання адміністратору та вести листування відразу у додатку. Список запитань від користувача адміністратору

можна переглянути перейшовши на відповідну сторінку, як зображено на рисунку 4.41.

Для того, щоб надіслати запитання необхідно натиснути «Додати» та заповнити форму в модальному вікні (рис.4.42)



Рисунок 4.41 – Сторінка із запитаннями

Джерело: знімок з екрану [35]

Рисунок 4.42 – Форма для надсилання запитання

Джерело: знімок з екрану [35]

Приклад листування користувача та адміністратора представлено на рисунку 4.43.

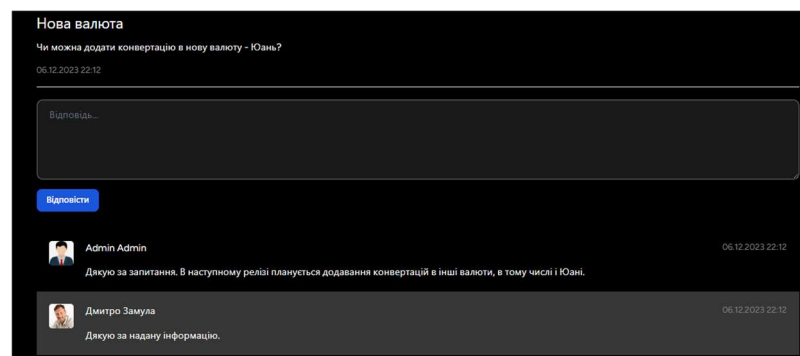


Рисунок 4.43 – Приклад листування

Джерело: знімок з екрану [35]

4.3 Адміністрування додатком інформаційної технології

Вхід на адміністративну панель проходить через особливу сторінку, яка візуально відрізняється від сторінки авторизації користувача, як це показано на рисунку 4.44.

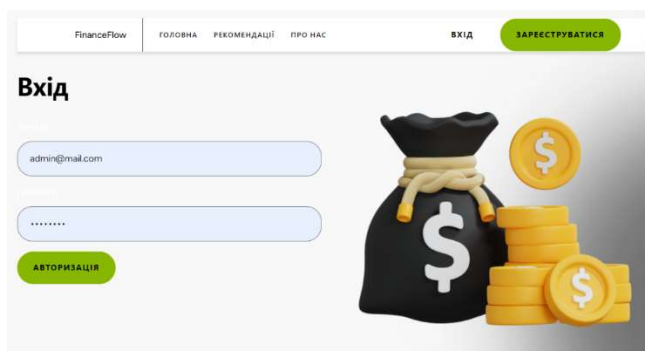


Рисунок 4.44 – Вхід в адміністративну панель

Джерело: знімок з екрану [35]

Після авторизації користувач перенаправляється на сторінку користувачів, яких адміністратор може заблокувати в разі порушення правил використання інформаційної технології на рисунку 4.45.

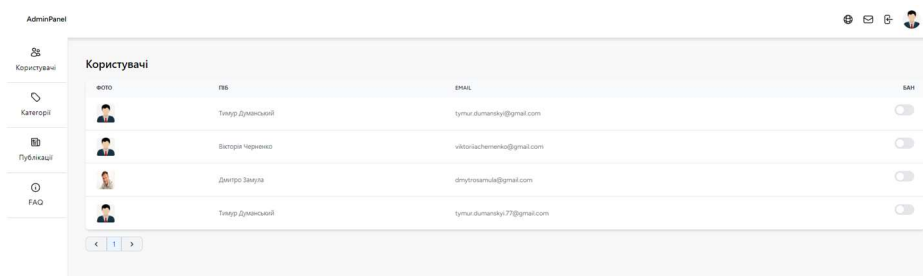
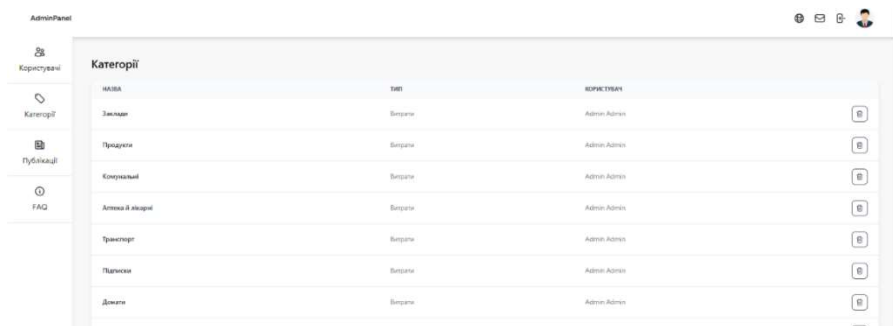


Рисунок 4.45 – Сторінка «Користувачі»

Джерело: знімок з екрану [35]

Адміністратор має можливість редагувати категорії різних підрозділів а також видаляти за необхідності на рисунку 4.46.

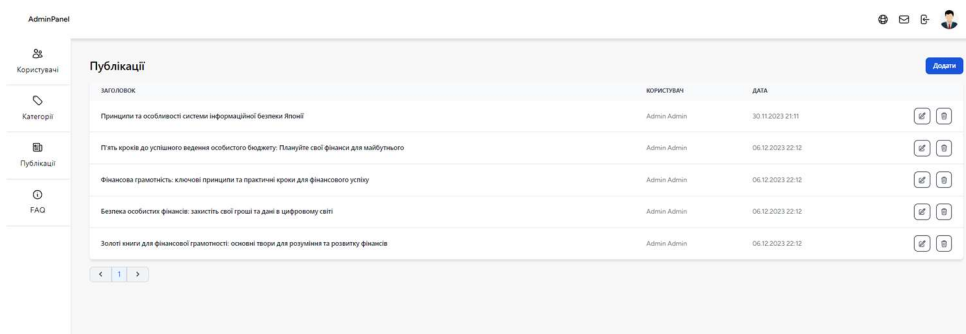


НАЗВА	ТИП	КОРИСТУВАЧ
Заклади	Візитки	Admin-Admin
Принципи	Візитки	Admin-Admin
Концепції	Візитки	Admin-Admin
Апеласія в Україні	Візитки	Admin-Admin
Трансферт	Візитки	Admin-Admin
Підписки	Візитки	Admin-Admin
Доголи	Візитки	Admin-Admin

Рисунок 4.46 – Сторінка «Категорії»

Джерело: знімок з екрану [35]

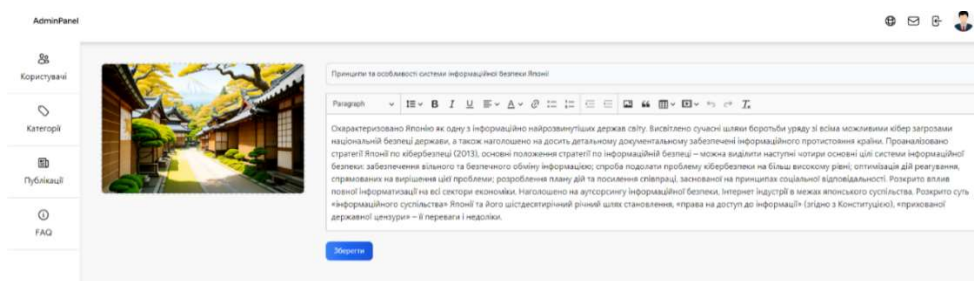
Функціоналу адміністративної панелі достатньо не лише редагувати текст або видаляти публікації, а й налаштовувати розмір та формат шрифту який буде використовуватися в публікації, а також змінити аватар на підходящий (рис. 4.47-48).



ЗАГОЛОВОК	КОРИСТУВАЧ	ДАТА
Принципи та особливості системи інформаційної безпеки Японії	Admin-Admin	30.11.2023 21:11
Пять кроков до успішного ведення особистого бюджету: Плануйте свої фінанси для майбутнього	Admin-Admin	06.12.2023 22:12
Фінансова грамотність: ключові принципи та практичні кроки для фінансового успіху	Admin-Admin	06.12.2023 22:12
Безпека особистих фінансів: заохочуйте свої гроші та дані в цифровому світі	Admin-Admin	06.12.2023 22:12
Золоті книги для фінансової грамотності: основні твори для розуміння та розвитку фінансів	Admin-Admin	06.12.2023 22:12

Рисунок 4.47 – Сторінка «Публікації»

Джерело: знімок з екрану [35]



Принципи та особливості системи інформаційної безпеки Японії

Регулярні

Охарактеризовано Японію як одну з інформаційно найрозвинутіших держав світу. Висвітлено сучасні шляхи боротьби з ризиками кіберзагрози національній безпеці держави, а також наголошено на досвіді державного управління забезпечення інформаційного протективного скрині. Проаналізовано стратегії Японії по кібербезпеці (CSIS), основні положення стратегії по інформаційній безпеці – можна виділити наступні чотири основні цілі системи інформаційної безпеки: забезпечення віального та безпечного обміну інформацією; спроба подолати проблему кібербезпеки на більш високому рівні; оптимізація дій реагування, управління на вирішення цієї проблеми; розроблення плану дій та посилення співпраці, заснованої на принципах соціальної відповідальності. Розкрито вплив нових інформаційних на всі сектори економіки. Наголошено на аутсорсингу інформаційної безпеки, інтернет-індустрії в межах японського суверенітету. Розкрито суть «інформаційного суверенітету» Японії та його віддистингований фічний шлях становлення, «права на доступ до інформації» (згідно з Конституцією), «приватної державної цензури» – її переваги і недоліки.

Рисунок 4.48 – Налаштування окремої публікації

Джерело: знімок з екрану [35]

Адміністратор може додавати, редагувати та видаляти актуальні FAQ які будуть показані в інформаційній системі в спеціально відведеному місці (рис. 4.49-50).

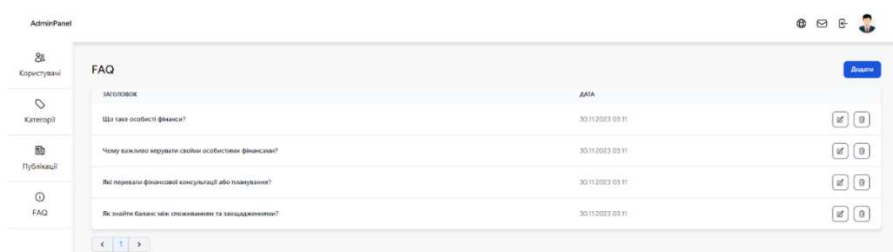


Рисунок 4.49 – Сторінка «FAQ»

Джерело: знімок з екрану [35]

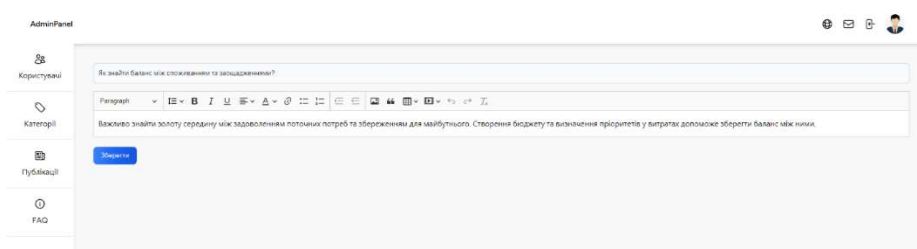


Рисунок 4.50 – Редагування FAQ

Джерело: знімок з екрану [35]

Якщо в користувача виникає питання, або проблема з яким він не може впоратися за допомогою FAQ, він може написати повідомлення адміністратору та пояснити проблему (рис. 4.51 - 2).

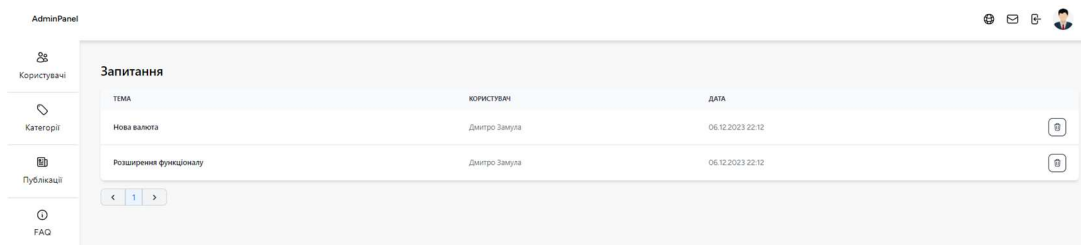


Рисунок 4.51 – Сторінка «Повідомлення»

Джерело: знімок з екрану [35]

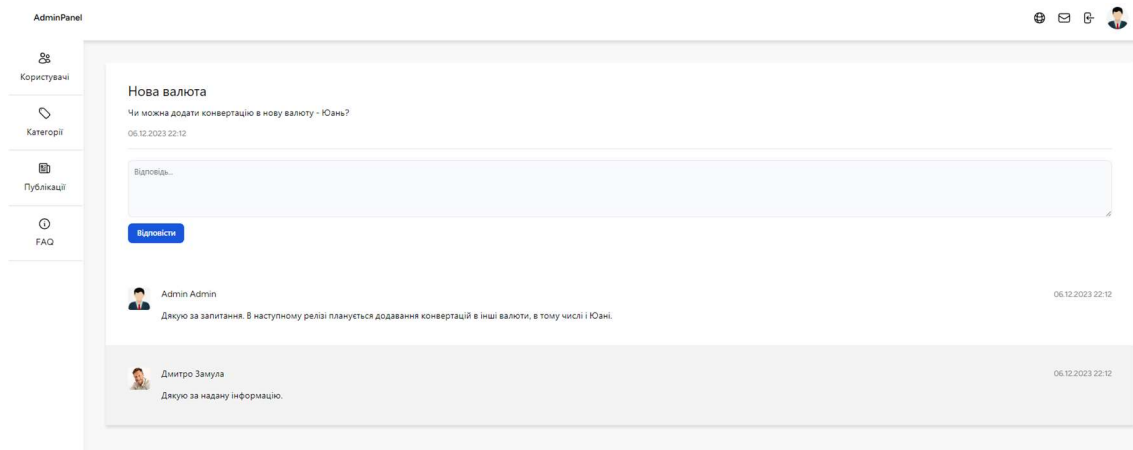


Рисунок 4.51 – Листування

Джерело: знімок з екрану [35]

ВИСНОВКИ

Під час виконання кваліфікаційної роботи на тему "Інформаційна технологія управління та аналізу витрат особистого бюджету" було встановлено мету та сформульовано завдання для її виконання. Також на початкових етапах було проведено планування робіт. Детальний огляд на планування робіт проедставлено у Додатку А.

Були досліджені актуальні джерела інформації, спрямовані на визначення актуальності розробки кваліфікаційної роботи. Результати аналізу підтверджують актуальність та необхідність впровадження інформаційної системи управління та аналізу витрат особистого бюджету.

Після проведення глибокого аналізу аналогів та проведення оцінки, було виконано моделювання та проектування процесу роботи додатка інформаційної технології. Також було розроблено логічну та фізичну модель бази даних. Це дозволило отримати чітке уявлення про функціональні можливості додатка інформаційної технології.

Було виконано розробку додатку за допомогою обраних інструментів та бібліотек. Також було продемонстровано роботу розробленого додатка зі сторони користувача та адміністратора. Всі поставлені задачі були виконані.

Практична цінність розробки проекту полягає у зручності роботи із великими об'ємами даних, а саме даними про доходи, витрати та збереження, полегшує використання інструментів додатка інформаційної технології.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Ultimate Guide to Financial Literacy: вебсайт. URL: <https://www.investopedia.com/guide-to-financial-literacy-4800530> (дата звернення: 02.11.2023);
2. Alshebami A.S., Aldhyani T.H. The Interplay of Social Influence, Financial Literacy, Saving Behaviour among Saudi Youth and the Moderating Effect of Self-Control: Ludenscheid, 2022. 183 с.
3. Widagdo B., Roz K. The role of personality traits, financial literacy and behavior on investment intentions and family support as a moderating variable: Ludenscheid, 2022. 86 с.
4. Harahap S., Thoyib A., Sumiati S. The Impact of Financial Literacy Retirement Planning with Serial Mediation of Financial Risk Tolerance and Saving Behavior: Evidence of Indonesia Entrepreneurs: Ludenscheid, 2022. 206 с.
5. Hastings J.S., Madrian, B.C., Skimmyhorn W.L. Financial literacy, financial education, and economic outcomes: Pearson, 2019. 72 с.
6. Van Rooij, M., Lusardi, A., Alessie R. The financial literacy and stock market participation: Ludenscheid, 2021. 163 с.
7. Mitchell, O.S., Lusardi A. The financial Literacy and Financial Behavior at Older Ages: Pearson, 2022. 73 с.
8. Fernandes D., Lynch Jr J.G. The financial literacy, financial education, and downstream financial behaviors: Pearson, 2014. 67 с.
9. Cole, S., Sampson, T., Zia, B. Prices or knowledge? What drives demand for financial services in emerging markets?: Pearson, 2011. 272 с.
10. Paolo Sironi Fintech Innovation: From Robo-Advisors to Goal Based Investing and Gamification: Wiley, 2018. 202 с.
11. Don Tapscott and Alex Tapscott Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World: Portfolio, 2016. 323 с.

12. László Györfi Machine Learning for Financial Engineering: Springer, 2022. 56 с.
13. Jessica Keyes Financial Technology Handbook: From the Basics to the Latest Applications: CRC Press, 2018. 219 с.
14. Ethan Brown Modeling and design in web development: O'Reilly Media, 2018. 99 с.
15. Marijn Haverbeke Web development as a look into the future: No Starch Press, 2018. 260 с.
16. Walter Isaacson The Innovators: How a Group of Hackers, Geniuses, and Geeks Created the Digital Revolution: Simon & Schuster, 2014. 282 с.
17. Manuel Laguna and Johan Marklund Business Process Modeling, Simulation, and Design: Pearson, 2023. 86 с.
18. Derek Hatley, Peter Hruschka, Imtiaz Pirbhai Process for System Architecture and Requirements Engineering: Dorset House Publishing Co Inc, 2020. 262 с.
19. Budgeting 101: How to Budget Money: вебсайт. URL: <https://www.nerdwallet.com/article/finance/how-to-budget> (дата звернення: 02.10.2022);
20. Making a Personal Budget: вебсайт. URL: <https://openstax.org/books/contemporary-mathematics/pages/6-5-making-a-personal-budget> (дата звернення: 05.11.2022);
21. Wallet: вебсайт. URL: <https://play.google.com/store/apps/details?id=com.droid4you.application.wallet&hl=ru&gl=US&pli=1> (дата звернення: 12.11.2022);
22. Money Manager: вебсайт. URL: <https://play.google.com/store/apps/details?id=com.realbyteapps.moneymanagerfree&hl=ru&gl=US> (дата звернення: 12.11.2022);
23. 1Money: вебсайт. URL: <https://play.google.com/store/apps/details?id=org.pixelrush.moneyiq&hl=ru&gl=US> (дата звернення: 12.11.2022);
24. Kurt Bittner, Ian Spence Use Case Modeling: Addison-Wesley: Professional, 2023. 183 с.

25. Louis Davidson "Practical Database Design for the Web: Morgan Kaufmann, 2019. 152 с.

26. P. Pšenák, M. Tibenský The usage of Vue JS framework for web application creation: Mesterséges intelligencia, 2020. 61 с.

27. Tailwind and bootstrap: вебсайт. URL: <https://ddi-dev.com/programming/the-good-and-the-bad-of-vue-js-framework-programming/https://likims.com/blog/tailwind-vs-bootstrap-pros-and-cons> (дата звернення: 07.12.2023);

28. Flowbite: вебсайт. URL: <https://www.producthunt.com/products/flowbite/reviews> (дата звернення: 07.12.2023);

29. D. Li et al., ECharts A declarative framework for rapid construction of web-based visualization: Visual Informatics, 2018. 136с.

30. Javascript chart libraries: вебсайт. URL: <https://www.luzmo.com/javascript-chart-libraries> (дата звернення: 07.12.2023);

31. What clients love about ckeditor the most: вебсайт. URL: <https://ckeditor.com/what-clients-love-about-ckeditor-the-most/> (дата звернення: 07.12.2023);

32. N. A. Haris, N. Hasim, PHP frameworks usability in web application development: International Journal of Recent Technology and Engineering, 2019. 109 с.

33. C. Vidal-Silva, C. Jiménez, E. Madariaga, L. Urzúa, Applying PHP codeigniter for easy web development: International Journal of Scientific and Technology Research, 2020. 420 с.

34. Реалізований Github проекту «Finance-Flow»: вебсайт. URL: <https://github.com/captim/Finance-Flow> (дата звернення: 07.12.2023);

35. Розроблений додаток інформаційної технології «Finance Flow»: вебсайт. URL: <https://finance-flow.in.net> (дата звернення: 07.12.2023);

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ
для розробки кваліфікаційної роботи магістра
«Інформаційна технологія управління та аналізу
витрат особистого бюджету»

A.1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

Метою проекту є розробка інформаційної технології управління та аналізу витрат особистого бюджету.

Деталізація мети методом SMART:

S – Метою проекту є розробка інформаційної технології управління та аналізу витрат особистого бюджету.

M – Система зможе допомогти в обробці даних, як саме інформаційні технології впливають на процес управління особистим бюджетом та аналізу його витрат.

A – Головною ціллю є розробка інструменту для управління та аналізу витрат особистого бюджету.

R – Розробка та використання програмних засобів для ведення бюджету, аналізу витрат та визначення ефективності витрат за допомогою ведення статистики.

T – на розробку такого проекту потрібно – 15 тижнів. В розрахунку, що в середньому буде використовуватись 8 годин в день.

A.2 ПЛАНУВАННЯ ЗМІСТУ СТРУКТУРИ РОБІТ ІНФОРМАЦІЙНОЇ СИСТЕМИ

WBS (рис. A.1) є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль; наприклад, програма, проект чи договір. Інструмент спрямований на детальне планування, оцінку вартості, визначення та розподіл персональної відповідальності виконавців та інші - тобто, на основні роботи і результати, що визначають зміст проекту.

Після того, як була створена WBS структура проекту, наступним необхідним етапом є розробка OBS (рис.А.2). Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту.

Для максимального скорочення бюджету проекту було вирішено залучити middle Project Manager із навичками бізнес-аналітика, який буде відповідати за бізнес-аспекти процесу та вести розробку. Також було вирішено утриматися від найму devops, оскільки ця позиція є високовартісною на ринку. Намір відмовитися від хмарних рішень означає, що сайт буде розміщено на звичайному сервері, а його міграцію та налаштування здійснить той самий РМ або розробник.



Рисунок А.1 – WBS діаграма

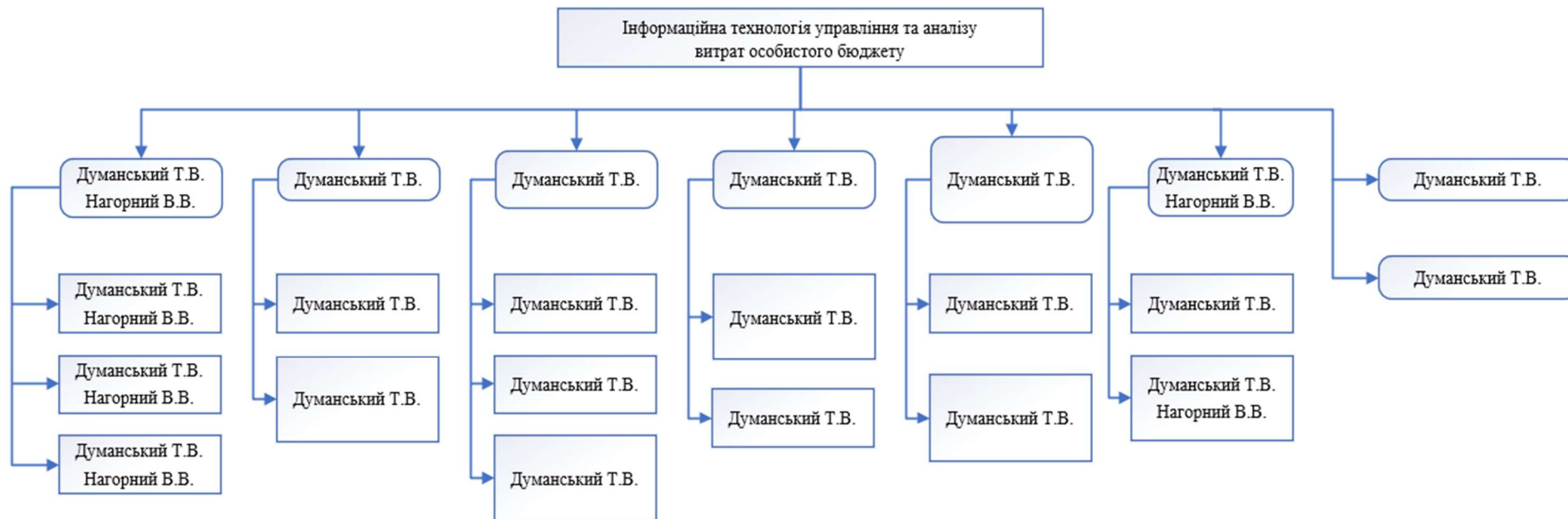


Рисунок А.2 – OBS діаграма

А.3 ПОБУДОВА КАЛЕНДАРНОГО ГРАФІКУ ВИКОНАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Діаграма Ганта – це горизонтальна діаграма з тимчасовою шкалою, яка використовується для ілюстрації плану робіт за проектом з прив'язкою до часу.

За допомогою діаграм Ганта керівники проектів і менеджери по продукту розбивають проекти на робочі завдання для зручності управління, підтримують порядок в роботі і роблять залежності між завданнями наочними.

Діаграми Ганта дозволяють спростити складові проекти. За допомогою цього засобу можна досить наочно і зручно для узагальнення представити велику кількість даних. Завдяки цій гістограмі велика кількість зацікавлених осіб, команд або їх учасників не стане проблемою для запису завдань, як і часті зміни обсягу роботи. Ще одна перевага використання діаграми Ганта полягає в тому, що вона дає загальне уявлення про проект в цілому, в тому числі про всі контрольні точки і терміни виконання. Діаграму Ганта можна уявити як ефективний засіб раннього попередження.

Розглянемо створену діаграму Ганта до заданої інформаційної системи (рис.А.3).

Інформаційна технологія управління та аналізу витрат особистого бюджету

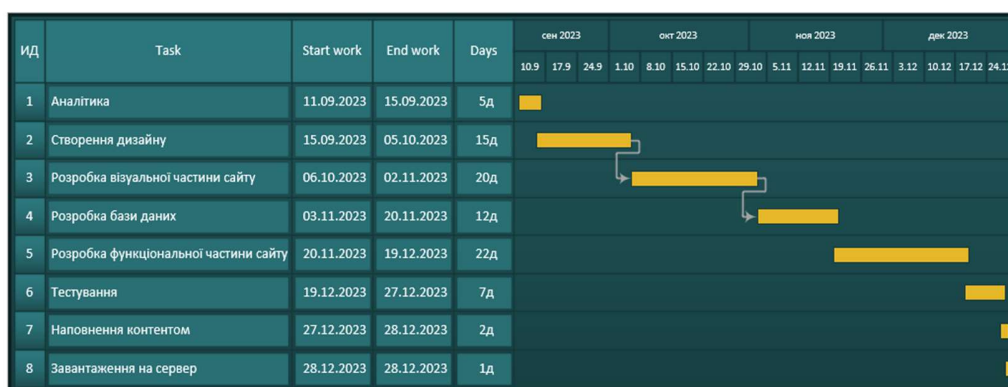


Рисунок А.3 – Діаграма Ганта

Дата початку проекту – 11.09.2023. На виконання проекту відводиться по 8 годин на добу. Робочими днями вважаються також вихідні та святкові дні.

Як видно з графіку виконання проекту загальний час, який відводиться на його виконання – 84 днів робочих днів. Оцінка давалась методом експертної оцінки і аналогової, на основах старих проектів. До кожної оцінки було додано 30% ризику, щоб закласти час та виконати все вчасно.

А.4 ПЛАНУВАННЯ РИЗИКІВ ПРОЕКТУ

Ризиком є ймовірна подія, яка у випадку її виникнення може як негативно, так і позитивно вплинути на конкретний проект. Управління ризиком – це процес зміни ризиків та реагування на події під час виконання проекту. При реалізації проекту важливою частиною є моніторинг ризиків. Отже, в даному випадку можна виділити деякі ризики.

Під час процесу аналізу для визначення числових значень ймовірності появи ступеня впливу, застосовувалася методика експертних оцінок. Виходячи цих оцінок можливо знайти ранг ризиків: $R = P * L$, де R – ранг ризику, P – ймовірність виникнення, L – ступінь впливу.

Шкала оцінки ризику може відповідати емпіричній шкалі оцінки ризику:

- 5 балів - критичний ризик (0,81 - 1);
- 4 бали - максимальний ризик (0,61 - 0,8);
- 3 бали - високий ризик (0,41 - 0,6);
- 2 бали - нормальний ризик (0,31 - 0,4); 1 бал - малий ризик (0 - 0,3).

Таблиця А.1 – Розрахунок ризиків

N	Назва ризику	Опис	P	L	R	План реагування
1	Ризик кібератак	Дані користувачів можуть бути під загрозою	0,7	0,8	0,56	Підвищення безпеки сайту та інформації користувачів
2	Пошкодження даних	Ризик втратити дані користувачів	0,6	0,6	0,36	Резервне зберігання даних у хмарі
3	Серверні проблеми	Контроль стабільності роботи серверу	0,3	0,9	0,27	Моніторинг актуальності сайту та проведення зусиль для оновлення
4	Вихід з ладу обладнання	Вихід з ладу обладнання може уповільнити розробку	0,2	0,5	0,15	Мати можливість продовжити роботу над розробкою сайту на іншому обладнанні

ДОДАТОК Б

КОД ДОДАТКУ
для розробки кваліфікаційної роботи магістра
«Інформаційна технологія управління та аналізу
витрат особистого бюджету»

UserDatesController.php

Контролер для роботи зі збереженнями.

```
<?php

namespace App\Http\Controllers;

use App\Models\UserDates;
use App\Models\Categories;
use App\Models\Types;
use Illuminate\Http\Request;
use Inertia\Inertia;
use Illuminate\Support\Facades\Auth;
use Symfony\Component\HttpFoundation\StreamedResponse;
use Illuminate\Support\Facades\Http;

use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

class UserDatesController extends Controller
{
    protected $result1 = [];
    protected $header1 = [];
    protected $result2 = [];
    protected $header2 = [];
    protected $result3 = [];
    protected $header3 = [];
    protected $general = [];
    protected $backgroundColor = [];

    public function index($id)
    {
        return Inertia::render('Costs', [
            'id' => $id
        ]);
    }

    public function wallet(Request $request, $id)
    {
        $data = UserDates::with('category', 'type')->where('users_id', Auth::id())->where('types_id', $id);
        if (isset($request->month)) {
            $data = $data->whereMonth('date', ($request->month + 1));
        }
        $data = $data->get();
        $categories = Categories::where('types_id', $id)->where('users_id', Auth::id())->get();
        return response()->json([
            'data' => $data,
            'categories' => $categories
        ]);
    }

    function statistic()
    {
        $categories = Categories::where('users_id', Auth::id())->get();
        $types = Types::get();
        return Inertia::render('Statistic', [
            'categories' => $categories,
            'types' => $types
        ]);
    }
}
```

```

}

function exportExcel(Request $request)
{
    $spreadsheet = new Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $model = UserDates::where('users_id', Auth::id())->with('category', 'type');

    if (isset($request->types_id)) {
        $model = $model->where('types_id', $request->types_id);
    }

    if (isset($request->categories_id)) {
        $model = $model->where('categories_id', $request->categories_id);
    }

    if (isset($request->month)) {
        $model = $model->whereMonth('date', ($request->month + 1));
    }

    $data = $model->get();

    $header = [
        "A" => "Дата",
        "B" => "Тип",
        "C" => "Категория",
        "D" => "Сума",
        "E" => "Коментар"
    ];

    foreach ($header as $key => $value) {
        $sheet->setCellValue($key . 1, $value);
    }

    $nextIndex = 2;
    foreach ($data as $item) {
        $sheet->setCellValue('A' . $nextIndex, $item['date']);
        $sheet->setCellValue('B' . $nextIndex, $item['type']['title']);
        $sheet->setCellValue('C' . $nextIndex, $item['category']['title']);
        $sheet->setCellValue('D' . $nextIndex, $item['sum']);
        $sheet->setCellValue('E' . $nextIndex, $item['comment']);
        $nextIndex += 1;
    }

    $writer = new Xlsx($spreadsheet);

    return $this->generateFileExcel($writer, 'Data.xlsx');
}

function apiStatistic(Request $request)
{
    $model = UserDates::where('users_id', Auth::id())->with('category', 'type');

    if (isset($request->types_id)) {
        $model = $model->where('types_id', $request->types_id);
    }

    if (isset($request->categories_id)) {
        $model = $model->where('categories_id', $request->categories_id);
    }

```



```

}

if (isset($request->month)) {
    $model = $model->whereMonth('date', ($request->month + 1));
}

$countPublication = clone $model;
$countPublication = $countPublication->count();

$data = [];

$this->generatePieChart($model);
sort($this->header1);
sort($this->result1);

$data['pie'] = [
    'labels' => $this->header1,
    'datasets' => [[
        'label' => "Всього",
        'backgroundColor' => $this->backgroundColor,
        'data' => $this->result1
    ]]
];

$this->generateColumnBarChart($model);
$result2 = [];
foreach ($this->result2 as $key => $value) {
    array_push($result2, [
        'label' => $value[0],
        'backgroundColor' => '#' . $this->random_color(),
        'data' => array_slice($value, 1)
    ]);
}

$data['bar'] = [
    'labels' => array_values($this->header2),
    'datasets' => $result2
];

$this->generateColumnLineChart($model);

$data['line'] = [
    'labels' => $this->header3,
    'datasets' => [[
        'label' => "Сума",
        'backgroundColor' => '#ffffff',
        'borderColor' => '#88B702',
        'data' => $this->result3
    ]]
];

return response()->json($data);
}

function generalStatistic(Request $request)
{
    $course = Http::timeout(160)->get("https://api.privatbank.ua/p24api/pubinfo?exchange&coursid=5")->json();
    $income = UserDates::where('users_id', Auth::id()->whereMonth('date', ($request->month + 1))->where('types_id',
1);

```

```

$costs = UserDates::where('users_id', Auth::id()->whereMonth('date', ($request->month + 1))->where('types_id', 2);
$savings = UserDates::where('users_id', Auth::id()->whereMonth('date', ($request->month + 1))->where('types_id',
3);

if (isset($request->categories_id)) {
    $income = $income->where('categories_id', $request->categories_id);
    $costs = $costs->where('categories_id', $request->categories_id);
    $savings = $savings->where('categories_id', $request->categories_id);
}

$income = $income->sum('sum');
$costs = $costs->sum('sum');
$savings = $savings->sum('sum');

$remains = $income - ($costs + $savings);

$data = [
    'income' => $income,
    'costs' => $costs,
    'savings' => $savings,
    'remains' => $remains
];

if (Auth::user()['currency'] == 'EUR') {
    $data['EUR'] = [
        'income' => round($income / $course[0]['buy'], 2),
        'costs' => round($costs / $course[0]['buy'], 2),
        'savings' => round($savings / $course[0]['buy'], 2),
        'remains' => round($remains / $course[0]['buy'], 2)
    ];
}

if (Auth::user()['currency'] == 'USD') {
    $data['USD'] = [
        'income' => round($income / $course[1]['buy'], 2),
        'costs' => round($costs / $course[1]['buy'], 2),
        'savings' => round($savings / $course[1]['buy'], 2),
        'remains' => round($remains / $course[1]['buy'], 2)
    ];
}

return response()->json($data);
}

function generatePieChart($model)
{
    $general = Types::get();
    foreach ($general as $key => $value) {
        $query = clone $model;
        $this->backgroundColor[$key] = '#' . $this->random_color();
        $this->header1[$key] = $value->title;
        $this->result1[$key] = $query->where('types_id', $value->id)->sum('sum');
    }
}

function generateColumnBarChart($model)
{
    $models = [];
    $general = [];

```

```

$general = Categories::where('users_id', Auth::id()->get());

foreach ($general as $key => $value) {
    $this->result2[$key] = [$value->title];
    $query = clone $model;
    array_push($models, $query->where('categories_id', $value->id));
}

$indicator = Categories::where('users_id', Auth::id()->get());

foreach ($models as $key => $value) {
    foreach ($indicator as $k => $v) {
        $this->header2[$k + 1] = $v->title;
        $query = clone $value;
        array_push($this->result2[$key], $query->where('categories_id', $v->id)->sum('sum'));
    }
}

function generateColumnLineChart($model)
{
    $general = UserDates::where('users_id', Auth::id()->distinct('date')->pluck('date');
    foreach ($general as $key => $value) {
        $query = clone $model;
        $this->header3[$key] = $value;
        $this->result3[$key] = $query->where('date', $value)->sum('sum');
    }
}

/**
 * Store a newly created resource in storage.
 */
public function store(Request $request)
{
    $category = Categories::firstOrCreate(
        [
            'title' => $request['category'],
            'users_id' => Auth::id(),
            'types_id' => $request['types_id']
        ],
        [
            'types_id' => $request['types_id'],
            'title' => $request['category']
        ]
    );
    $response = UserDates::create([
        'date' => $request['date'],
        'sum' => $request['sum'],
        'comment' => $request['comment'],
        'users_id' => Auth::id(),
        'categories_id' => $category['id'],
        'types_id' => $request['types_id']
    ]);
    return response()->json($response);
}

/**
 * Update the specified resource in storage.
 */

```

```

*/
public function update(Request $request, $id)
{
    $data = $request->all();
    $category = Categories::firstOrCreate(
        [
            'title' => $data['category'],
            'users_id' => Auth::id()
        ],
        [
            'types_id' => $data['types_id'],
            'title' => $data['category']
        ]
    );
    UserDates::find($id)->update([
        'date' => $request['date'],
        'sum' => $request['sum'],
        'comment' => $request['comment'],
        'categories_id' => $category['id']
    ]);
}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    UserDates::find($id)->delete();
}

function random_color_part()
{
    return str_pad(dechex(mt_rand(0, 255)), 2, '0', STR_PAD_LEFT);
}

function random_color()
{
    return $this->random_color_part() . $this->random_color_part() . $this->random_color_part();
}

function generateFileExcel($writer, $fileName): StreamedResponse
{
    $response = new StreamedResponse(
        function () use ($writer) {
            $writer->save('php://output');
        }
    );

    $response->headers->set('Content-Type', 'application/vnd.ms-excel');
    $response->headers->set('Content-Disposition', 'attachment;filename="' . $fileName . '"');
    $response->headers->set('Cache-Control', 'max-age=0');

    return $response;
}
}

```

Statistic.vue

КОМПОНЕНТ СТАТИСТИКИ.

<script setup>

```

import { Head } from '@inertiajs/vue3';
import Header from '../Components/Header.vue'
import Footer from '../Components/Footer.vue'
import Bar from '../Components/diagram/Bar.vue'
import Pie from '../Components/diagram/Pie.vue'
import Line from '../Components/diagram/Line.vue'
import Statistic from '../Components/Statistic.vue'
import { FwbButton } from 'flowbite-vue'
defineProps({
  categories: {
    type: Array,
  },
  types: {
    type: Array,
  }
});
</script>
<template>
  <Head title="Витрати" />
  <div>
    <Header />

    <Statistic :filter="filter" />

    <div class="container md:px-0 px-4">
      <div class="grid grid-cols-12 gap-8">
        <div class="md:col-span-3 col-span-12 md:pr-9">
          <select class="select" v-model="filter.month" @change="getData()">
            <option disabled :value="null">Місяць</option>
            <option v-for="(item, index) in month" :key="index" :value="index">{{ item }}</option>
          </select>
          <select class="select" v-model="filter.types_id" @change="getData()">
            <option disabled :value="null">Тип</option>
            <option v-for="(item, index) in types" :key="index" :value="item.id">{{ item.title }}</option>
          </select>
          <select class="select" v-model="filter.categories_id" @change="getData()">
            <option disabled :value="null">Категорія</option>
            <option v-for="(item, index) in filteredCategories" :key="index" :value="item.id">{{ item.title }}</option>
          </select>
        </div>
        <div class="md:col-span-6 col-span-12">
          <div class="flex gap-3 mb-2">
            <fwb-button color="green" @click="exportExcel()">
              <svg xmlns="http://www.w3.org/2000/svg" x="0px" y="0px" width="20" height="20"
                viewBox="0,0,256,256">
                <g fill="#ffffff" fill-rule="nonzero" stroke="none" stroke-width="1" stroke-linecap="butt"
                  stroke-linejoin="miter" stroke-miterlimit="10" stroke-dasharray="" stroke-dashoffset="0"
                  font-family="none" font-weight="none" font-size="none" text-anchor="none"
                  style="mix-blend-mode: normal">
                  <g transform="scale(5.12,5.12)">
                    <path
                      d="M28.875,0c-0.01953,0.00781 -0.04297,0.01953 -0.0625,0.031251-28,5.3125c-
0.47656,0.08984 -0.82031,0.51172 -0.8125,1v37.3125c-0.00781,0.48828 0.33594,0.91016
0.8125,1128,5.3125c0.28906,0.05469 0.58984,-0.01953 0.82031,-0.20703c0.22656,-0.1875 0.36328,-0.46484 0.36719,-
0.76172v-5h17c1.09375,0 2,-0.90625 2,-2v-34c0,-1.09375 -0.90625,-2 -2,-2h-17v-5c0.00391,-0.28906 -0.12109,-0.5625 -
0.33594,-0.75391c-0.21484,-0.19141 -0.50391,-0.28125 -0.78906,-0.24609zM28,2.1875v4.34375c-0.13281,0.27734 -
0.13281,0.59766 0,0.875v35.40625c-0.02734,0.13281 -0.02734,0.27344 0,0.40625v4.593751-26,-4.96875v-
35.6875zM30,8h17v34h-17v-5h4v-2h-4v-6h4v-2h-4v-5h4v-2h-4v-5h4v-2h-4zM36,13v2h8v-
2zM6.6875,15.687515.46875,9.343751-5.96875,9.34375h513.25,-6.03125c0.22656,-0.58203 0.375,-1.02734 0.4375,-

```



```

    }
  },
  filter: {
    month: null,
    types_id: 1,
    categories_id: null
  },
  month: [
    'Січень',
    'Лютий',
    'Березень',
    'Квітень',
    'Травень',
    'Червень',
    'Липень',
    'Серпень',
    'Вересень',
    'Жовтень',
    'Листопад',
    'Грудень'
  ]
}
},
computed: {
  filteredCategories() {
    return this.categories.filter(item => item.types_id === this.filter.types_id)
  }
},
created() {
  const d = new Date();
  this.filter.month = d.getMonth();
  this.getData();
},
methods: {
  generateReport() {
    this.$refs.vue3SimpleHtml2pdf.download()
  },
  getData() {
    axios.get('/statistics', {
      params: this.filter
    })
      .then((response) => {
        this.data = response.data
      })
  },
  exportExcel() {
    axios.get('/api/export-excel', { params: this.filter, responseType: "blob" }).then(response => {
      const url = window.URL.createObjectURL(new Blob([response.data]));
      const link = document.createElement('a');
      link.href = url;
      link.setAttribute('download', 'Статистика.xlsx');
      document.body.appendChild(link);
      link.click();
    });
  },
  exportPdf() {
    html2canvas(document.querySelector("#divToPrint")).then(canvas => {
      const link = document.createElement("a");
      link.setAttribute("download", "diagram.png");
      link.setAttribute(

```

```

        "href",
        canvas
        .toDataURL("image/png")
        .replace("image/png", "image/octet-stream")
    );
    link.click();
    })
}
}
}
</script>
<style scoped>
.select {
    border: 1px solid #fff;
    background: none;
    border-radius: 40px;
    padding: 15px 28px;
    font-size: 13.219px;
    font-weight: 400;
    width: 100%;
    text-transform: uppercase;
    text-align: left;
    margin-bottom: 30px;
}

.select option {
    color: #000;
}

.btn-diagram {
    width: 100px;
    margin: auto;
    margin-bottom: 50px;
    cursor: pointer;
    opacity: 0.6;
    transition: 0.2s;
}

.btn-diagram:hover,
.btn-diagram.active {
    opacity: 1;
}

@media screen and (max-width: 600px) {
    .btn-diagram {
        display: inline-block;
        margin: 20px;
        width: 60px;
    }</style>
}

```


ДОДАТОК В

**РЕАЛІЗАЦІЯ ТАБЛИЦЬ БАЗИ ДАНИХ
для розробки кваліфікаційної роботи магістра
«Інформаційна технологія управління та аналізу
витрат особистого бюджету»**

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|-------------------------|------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 description | text | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 questions_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 5 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 6 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.1 – Таблица «answers»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 title | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 types_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 5 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 6 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.2 – Таблица «categories»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 title | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 description | text | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 5 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.3 – Таблица «faq»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 title | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 description | longtext | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 preview | longtext | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 5 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 6 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 7 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.4 – Таблица «posts»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------------|------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 posts_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 description | text | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 5 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 6 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.5 – Таблица «posts_comments»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---------------------|------------|-----------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 posts_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 5 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.6 – Таблица «posts_favourites»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 title | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 description | text | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 5 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 6 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.7 – Таблица «questions»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 title | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 4 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.8 – Таблица «types»

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|--------------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | Ещё |
| <input type="checkbox"/> | 2 date | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 3 sum | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 4 comment | text | utf8mb4_unicode_ci | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 5 users_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 6 categories_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 7 types_id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | | Ещё |
| <input type="checkbox"/> | 8 created_at | timestamp | | | Да | NULL | | | Ещё |
| <input type="checkbox"/> | 9 updated_at | timestamp | | | Да | NULL | | | Ещё |

Рисунок В.9 – Таблица «uset_datas»

























| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|----------------------------|--------------|--------------------|----------|------|-----------------|-------------|----------------|---|
| <input type="checkbox"/> | 1 id 🔑 | bigint(20) | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT |   Ещё |
| <input type="checkbox"/> | 2 name | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | |   Ещё |
| <input type="checkbox"/> | 3 surname | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | |   Ещё |
| <input type="checkbox"/> | 4 email 📧 | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | |   Ещё |
| <input type="checkbox"/> | 5 email_verified_at | timestamp | | | Да | NULL | | |   Ещё |
| <input type="checkbox"/> | 6 password | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | |   Ещё |
| <input type="checkbox"/> | 7 photo | varchar(255) | utf8mb4_unicode_ci | | Нет | /img/avatar.jpg | | |   Ещё |
| <input type="checkbox"/> | 8 is_admin | tinyint(1) | | | Нет | 0 | | |   Ещё |
| <input type="checkbox"/> | 9 currency | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | |   Ещё |
| <input type="checkbox"/> | 10 remember_token | varchar(100) | utf8mb4_unicode_ci | | Да | NULL | | |   Ещё |
| <input type="checkbox"/> | 11 created_at | timestamp | | | Да | NULL | | |   Ещё |
| <input type="checkbox"/> | 12 updated_at | timestamp | | | Да | NULL | | |   Ещё |

Рисунок В.10 – Таблица «users»