

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

ВЕРЕДІЮК Тетяна Петрівна

Програмний модуль для відстеження та аналізу фінансів/
Software module for tracking and analyzing finances

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
Т. П. Вередюк

Науковий керівник:
Викл. В. І. Дорош

Кваліфікаційну роботу допущено до
захисту
«___» _____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Н.М. Васильків
« ____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
ВЕРЕДЮК Тетяна Петрівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програмний модуль для відстеження та аналізу фінансів / Software Module for Tracking and Analyzing Finances

керівник роботи В.І. Дорош

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– проаналізувати існуючі підходи до автоматизованого відстеження фінансових операцій, включно з методами, що базуються на машинному навчанні, глибокому навчанні та фінансовій аналітиці; визначити їх переваги та обмеження в умовах обробки великих обсягів транзакційних даних;

– сформулювати концепцію побудови інтелектуального програмного модуля для моніторингу фінансів, який би забезпечував автоматичне виявлення аномалій, підозрілих транзакцій або фінансових ризиків у режимі реального часу, використовуючи інтеграцію глибокого навчання та аналітики великих даних;

– розробити архітектуру програмного модуля для обробки фінансових потоків, яка забезпечує масштабованість, інтеграцію з різними джерелами даних;

– провести експериментальні дослідження ефективності розробленого модуля, зокрема порівняння його точності та продуктивності з традиційними методами фінансового моніторингу.

5. Перелік графічного матеріалу в роботі:

- загальна архітектура сайту;
- макет головної сторінки;
- сторінка додавання та перегляду транзакцій;

– графік фінансової активності.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студентка _____ Т. П. Вередюк
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ В.І. Дорош
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль для відстеження та аналізу фінансів» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» написана обсягом 68 сторінок та містить 17 ілюстрацій, 1 таблицю, 5 додатків та 29 використаних джерел.

Метою дослідження є розробка та впровадження програмного модуля, який дозволяє ефективно відстежувати, аналізувати та керувати особистими або бізнес-фінансами.

У цьому дослідженні для досягнення своєї мети було використано кілька методів. Теоретичний аналіз літератури та існуючих рішень у галузі фінансового менеджменту та інформаційних систем дозволив сформувати базу знань та визначити ключові напрямки роботи.

Для розробки структурно-функціональної моделі було використано методи структурного та об'єктного моделювання, зокрема IDEF0. Алгоритм робочого процесу системи було розроблено з використанням методів алгоритмізації та моделювання процесів. Для проектування бази даних було використано методи реляційного моделювання даних. Користувацький інтерфейс було створено з урахуванням принципів UX/UI дизайну для забезпечення зручності використання.

Запропонований модуль значно покращить організацію та управління фінансами користувачів, забезпечуючи автоматичну категоризацію транзакцій, генерування фінансових звітів, планування бюджету та моніторинг цілей заощаджень. Система також дозволить аналізувати тенденції витрат і доходів, що допоможе приймати обґрунтовані фінансові рішення.

Ключові слова: ФІНАНСОВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, БАЗА ДАНИХ, ФІНАНСОВИЙ МЕНЕДЖМЕНТ, ВІДСЛІДКУВАННЯ ВИТРАТ, БЮДЖЕТУВАННЯ, АНАЛІЗ ТРАНЗАКЦІЙ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

ANNOTATION

Qualification thesis on the topic «Software Module for Tracking and Analyzing Finances» for obtaining the bachelor's degree in the specialty 122 «Computer Science» consists of 68 pages and includes 17 illustrations, 1 table, 5 appendices, and 29 references.

The purpose of the study is to develop and implement a software module that allows you to effectively track, analyze and manage personal or business finances.

In this study, several methods were used to achieve its goal. Theoretical analysis of the literature and existing solutions in the field of financial management and information systems allowed to form a knowledge base and identify key areas of work.

To develop a structural-functional model, structural and object modeling methods were used, in particular IDEF0. The algorithm of the system's workflow was developed using algorithmization and process modeling methods. Relational data modeling methods were used to design the database. The user interface was created taking into account the principles of UX/UI design to ensure ease of use.

The proposed module will significantly improve the organization and management of users' finances by providing automatic transaction categorization, financial report generation, budget planning, and savings goal monitoring. The system will also allow analyzing trends in expenses and income, which will help make informed financial decisions.

Keywords: FINANCIAL SOFTWARE, DATABASE, FINANCIAL MANAGEMENT, EXPENDITURE TRACK, BUDGETING, TRANSACTION ANALYSIS, SOFTWARE DEVELOPMENT.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка задачі	10
1.1 Теоретичні основи фінансового планування та контролю	10
1.2 Огляд існуючих рішень	12
1.3 Постановка задачі проектування	16
2 Моделювання та проектування програмного модуля	18
2.1 Структурно-функціональне моделювання	18
2.2 Алгоритм аналізу фінансового стану та управління бюджетом	19
2.3 Проектування моделі бази даних	23
2.4 Архітектура програмного модуля	25
3 Реалізація програмного модуля	27
3.1 Програмна реалізація.....	27
3.2 Інтерфейс користувача	30
Висновки	35
Список використаних джерел	38
Додаток А Копія публікації.....	41
Додаток Б Опис стрілок з IDEF0-діаграми.....	46
Додаток В Вміст файлу MAIN.HTML	49
Додаток Г Вміст файлу STYLES.CSS	55
Додаток Д Вміст файлу INDEX.JS	63

ВСТУП

У сучасному світі управління фінансами стало набагато складнішим, ніж будь-коли раніше. Динамічно мінливі економічні умови, зростання інфляції та зростаюча доступність різних фінансових інструментів означають, що ефективний контроль над бюджетом домогосподарства вимагає не лише фінансової обізнаності, але й відповідних технологічних інструментів. Традиційні методи відстеження витрат, такі як ведення паперових нотаток або простих електронних таблиць, є неадекватними з огляду на складність сучасних фінансових операцій. Люди використовують багато різних форм оплати – дебетові картки, кредитні картки, мобільні платежі, онлайн-перекази – що робить ручний моніторинг усіх транзакцій практично неможливим.

Розвиток цифрових технологій відкрив нові можливості в сфері управління особистими фінансами. Сучасні мобільні додатки та веб-системи дозволяють автоматично класифікувати витрати, створювати бюджети та генерувати детальні фінансові звіти. Особливо важливо те, що такі рішення можуть інтегруватися з банківськими системами, що дозволяє автоматично завантажувати дані про транзакції без необхідності ручного введення кожної операції. Дослідження показують, що люди, які використовують автоматичні інструменти відстеження фінансів, демонструють кращий контроль над своїми витратами та мають більше шансів досягти своїх цілей заощадження. Це пояснюється тим, що регулярний моніторинг фінансового стану підвищує обізнаність щодо витрат і допомагає визначити сфери, де можна заощадити [1].

Неналежний контроль особистих фінансів може призвести до серйозних наслідків. Неконтрольовані витрати часто призводять до боргу, який може наростати у вигляді відсотків та штрафів. Це особливо проблематично для молодих людей, які тільки починають своє самостійне фінансове життя та ще не розвинули звичок управління грошима. Крім того, відсутність прозорості щодо ваших фінансів ускладнює планування майбутнього. Без точних знань

про ваші доходи та витрати важко визначити, скільки грошей можна виділити на заощадження, інвестиції чи більші покупки. Це, у свою чергу, може призвести до ситуацій, коли люди не готові до непередбачених витрат або не можуть досягти своїх довгострокових фінансових цілей [2].

Сучасний стан технологій дозволяє створювати передові системи управління фінансами, які можуть значно полегшити життя користувачів. Однак, вкрай важливо, щоб такі рішення були не лише функціональними, але й безпечними та простими у використанні. Особливу увагу слід приділяти питанням конфіденційності та захисту фінансових даних, які є особливо чутливими. Сучасні рішення також повинні пропонувати можливості персоналізації, оскільки фінансові потреби різних користувачів можуть суттєво відрізнятися. Студент матиме інші вимоги, ніж сім'я з дітьми, а підприємцю потрібні інші функції, ніж пенсіонеру.

Метою цієї роботи є розробка та впровадження програмного модуля для відстеження та аналізу особистих фінансів, який поєднує функціональність, безпеку та простоту використання. Розроблена система має на меті дозволити користувачам автоматично контролювати свої фінансові операції, класифікувати витрати, створювати бюджети та генерувати прозорі аналітичні звіти.

Для досягнення цієї мети необхідно буде вирішити кілька ключових технічних та операційних задач:

- проаналізувати існуючі підходи до автоматизованого відстеження фінансових операцій, включно з методами, що базуються на машинному навчанні, глибокому навчанні та фінансовій аналітиці; визначити їх переваги та обмеження в умовах обробки великих обсягів транзакційних даних;
- сформулювати концепцію побудови інтелектуального програмного модуля для моніторингу фінансів, який би забезпечував автоматичне виявлення аномалій, підозрілих транзакцій або фінансових ризиків у режимі реального часу, використовуючи інтеграцію глибокого навчання та аналітики великих даних;

- розробити архітектуру програмного модуля для обробки фінансових потоків, яка забезпечує масштабованість, інтеграцію з різними джерелами даних;
- провести експериментальні дослідження ефективності розробленого модуля, зокрема порівняння його точності та продуктивності з традиційними методами фінансового моніторингу.

Об’єкт дослідження – процес автоматизованого відстеження, аналізу та управління особистими з використанням інформаційних технологій.

Предмет дослідження – методи, моделі та програмні засоби реалізації інтелектуального програмного модуля для фінансового моніторингу, включаючи архітектурні рішення, алгоритми обробки транзакцій, механізми бюджетування, візуалізації та забезпечення безпеки фінансових даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Теоретичні основи фінансового планування та контролю

У наш час ефективне управління особистими фінансами стало не просто навичкою, а необхідністю для підтримки економічної стабільності. Фінансове планування – це процес систематичного прийняття рішень щодо розподілу грошових ресурсів для досягнення конкретних життєвих цілей. Він включає аналіз поточного фінансового стану, прогнозування майбутніх доходів і витрат, а також розробку стратегій мінімізації фінансового ризику.

Фінансовий контроль є невід'ємною частиною цього процесу, що дозволяє відстежувати фактичні грошові потоки відповідно до бюджетних припущень. Це безперервний процес порівняння планових та фактичних фінансових результатів, який дозволяє виявляти відхилення та вживати коригувальних заходів. Без належного контролю навіть найкраще розроблений фінансовий план може виявитися неефективним. Ключові компоненти фінансового планування охоплюють кілька взаємопов'язаних областей. Домашнє бюджетування передбачає систематичний розподіл доходів на окремі категорії витрат з урахуванням поточних потреб, цілей заощаджень та резервів на непередбачені ситуації. Відстеження ваших витрат дозволяє визначити моделі споживання та сфери потенційної економії, що є основою для прийняття обґрунтованих фінансових рішень. Сучасні дослідження показують, що люди, які ведуть детальний облік витрат, економлять в середньому 12-18% від загального бюджету. Цей ефект пояснюється "ефектом Хоторна" – зміною поведінки під впливом усвідомлення того, що вона відстежується. Категоризація витрат відіграє ключову роль у фінансовому аналізі. Основні категорії включають: фіксовані витрати (оренда, страхування), змінні витрати (продукти, транспорт), дискреційні витрати (розваги, хобі) та інвестиційні витрати (освіта, покращення навичок). Правильна категоризація дозволяє ідентифікувати пріоритети та оптимізувати розподіл ресурсів.

Управління прибутками виходить за рамки простого обліку доходів – воно також включає диверсифікацію джерел доходу та оптимізацію оподаткування. Сучасні підходи до особистих фінансів також включають інвестування як спосіб захисту від інфляції та створення довгострокового багатства. Водночас, відповідальне управління боргом допомагає уникнути боргової пастки та мінімізує витрати на обслуговування фінансових зобов'язань.

Фінансове прогнозування на основі історичних даних дозволяє вам бути краще підготовленими до майбутніх економічних викликів. Він використовує аналіз тенденцій витрат, сезонності доходів та зовнішніх факторів, що впливають на фінансове становище домогосподарства.

Відсутність системного підходу до фінансів може призвести до серйозних наслідків. Дослідження показують, що домогосподарства без бюджету втричі частіше стикаються з проблемами ліквідності. Неконтрольовані витрати можуть призвести до зростання боргу, що, своєю чергою, негативно впливає на кредитоспроможність та фінансові можливості в майбутньому.

В умовах сучасної цифрової економіки традиційні методи фінансового менеджменту, такі як ведення бюджету в блокноті або електронній таблиці, виявляються недостатніми. Зростаюча складність фінансових операцій, кількість банківських рахунків та платіжних інструментів вимагає використання більш досконалих інструментів. Цифрова трансформація фінансових послуг створила нові можливості та виклики. Cashless payments становлять вже понад 60% всіх транзакцій в розвинених країнах, що ускладнює відстеження витрат без спеціалізованих інструментів.

Штучний інтелект та машинне навчання революціонізують персональне фінансове планування. Алгоритми можуть аналізувати тисячі транзакцій для виявлення паттернів, прогнозування майбутніх витрат з точністю до 85% та надання персоналізованих рекомендацій. Robo-advisors вже управляють

активами на суму понад \$1 трильйон глобально, демонструючи ефективність автоматизованого фінансового планування.

Поведінкова економіка також впливає на дизайн фінансових додатків. Принципи м'якого підштовхування використовуються для заохочення позитивної фінансової поведінки: автоматичні заощадження, нагадування про бюджетні ліміти, візуалізація прогресу до цілей. Дослідження показують, що правильно розроблені "nudges" можуть збільшити рівень заощаджень на 3-5% річних.

1.2 Огляд існуючих рішень

Ринок додатків для управління особистими фінансами характеризується широким розмаїттям рішень, від простих інструментів відстеження витрат до передових аналітичних платформ. Кожне з цих рішень відповідає різним потребам користувачів та пропонує різні функціональні можливості. Впровадження директиви PSD2 у Європі та аналогічних нормативних актів на інших ринках дозволило створити екосистему відкритого банкінгу. Споживачі тепер можуть авторизувати зовнішні програми для доступу до своїх банківських даних, що революціонізує спосіб управління кількома фінансовими рахунками. Мобільні та веб-додатки для управління особистими фінансами складають найбільшу категорію доступних рішень.

На рисунках 1.1-1.3 представлені головні сторінки популярних додатків, таких як Mint, YNAB та Goodbudget, пропонують автоматизоване відстеження транзакцій шляхом підключення до банківських рахунків користувачів. Ці платформи використовують технологію відкритого банкінгу, яка забезпечує безпечний доступ до фінансових даних за згодою користувача.

У таблиці 1.1 відображено порівняльну характеристику додатків-аналогів. Розширені функції категоризації автоматично призначають транзакції відповідним групам витрат, використовуючи алгоритми машинного навчання для розпізнавання закономірностей. Візуалізація фінансових даних

за допомогою діаграм та інформаційних панелей полегшує розуміння ваших звичок витрачання коштів та визначення областей для оптимізації.

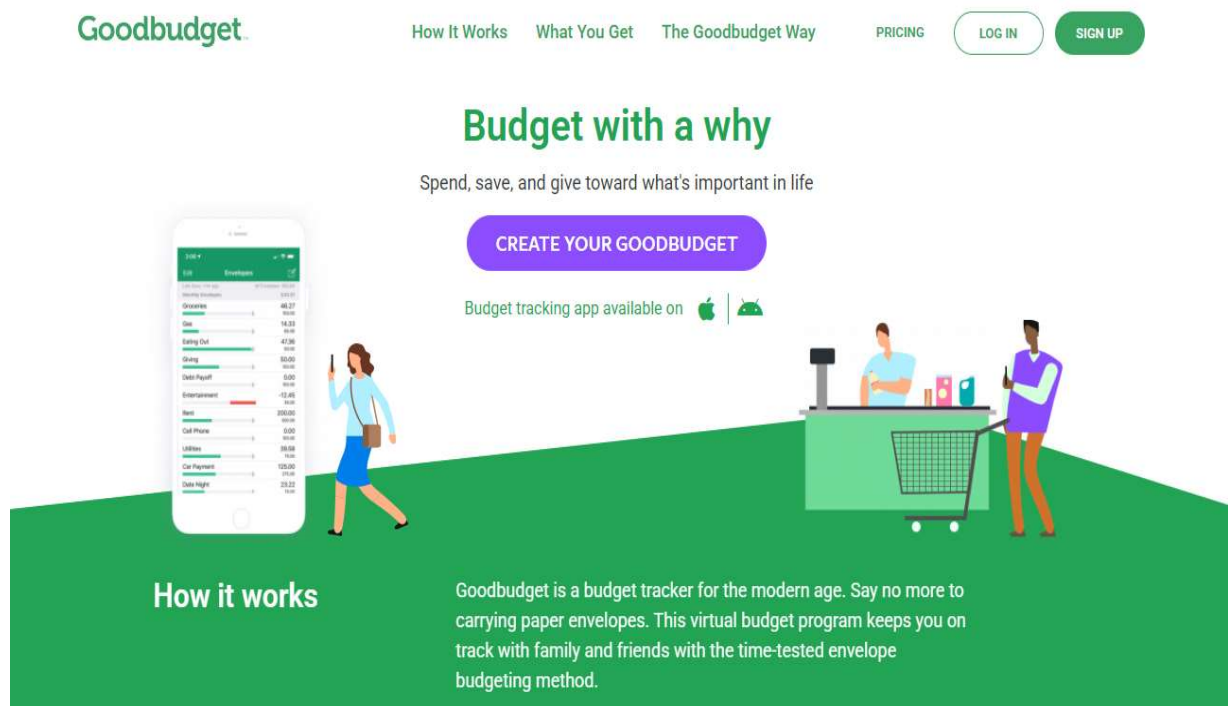


Рисунок 1.1 – Головна сторінка Goodbudget

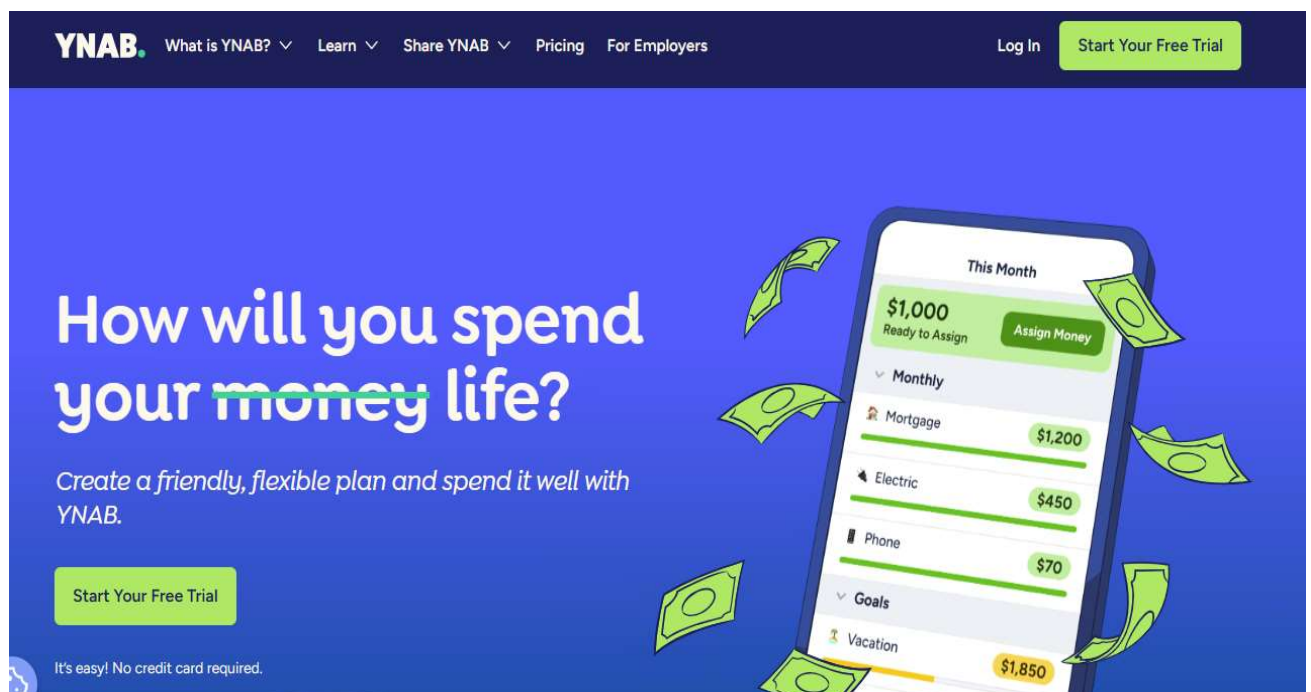


Рисунок 1.2 – Головна сторінка YNAB

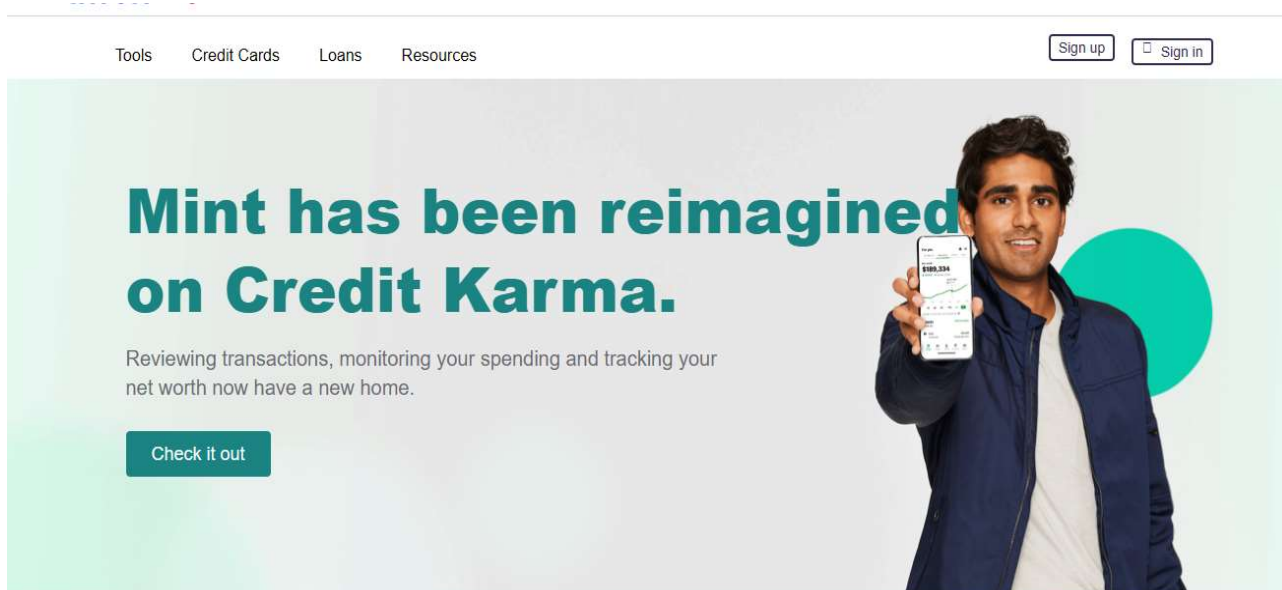


Рисунок 1.3 – Головна сторінка Mint

Рішення, призначені для малих та середніх підприємств, такі як QuickBooks або Xero, зосереджені на розширених функціях бухгалтерського обліку. До них належать управління виставленням рахунків, розрахунки з підрядниками, розрахунки податків та прогнози руху грошових коштів. Ці системи часто інтегруються із зовнішніми сервісами, такими як електронний банкінг або платіжні системи.

Інструменти на основі електронних таблиць залишаються популярними серед користувачів, які цінують повний контроль над своїми даними. Excel та Google Таблиці пропонують гнучкість у створенні власних шаблонів бюджету, але вимагають більше часу та технічних знань.

Зростаючий сегмент додатків на основі блокчейну реагує на зростання інтересу до криптовалют. Такі додатки, як CoinTracking та Blockfolio, дозволяють відстежувати ваш портфель криптовалют поряд із традиційними фінансовими активами.

Порівняльна характеристика веб-додатків–аналогів понада в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика веб-додатків–аналогів

Характеристика/ Онлайн ресурс	Mi nt	YN AB	PocketG uard	GoodBu dget	Exc el	PersonalCA pital
Зручний інтерфейс	+	+	+	-	-	+
Сучасний дизайн	+	+	+	-	-	+
Функціональність	+	+	+	+	+	+
Синхронізація з банкми	+	+	+	+	-	+
Реєстрація користувачів	+	+	+	+	+	+
Інформаційна сторінка	+	+	+	+	+	+
Звіти та аналіз	+	+	-	+	+	+
Наявність відгуків	+	+	+	+	-	+
Планування фінансових цілей	+	+	-	-	-	+

Після дослідження аналогічних проєктів, формуються певні висновки. Більшість із наведених вище веб-сайтів насичені інформацією. Водночас, для створення програмного продукту, орієнтованого на користувача, проєкт вимагає ретельної розробки. Інформаційну базу, структуру, палітру кольорів та візуальні елементи можна запозичити з аналогів, збагативши власну систему новими функціями. Це надасть розробленому продукту унікальність та допоможе виправити згадані раніше недоліки. Незважаючи на широкий спектр пропозицій, сучасні рішення мають певні обмеження. Відсутність повноцінних можливостей персоналізації часто унеможлиблює адаптацію застосунку до конкретних потреб користувача [20].

Побоювання щодо конфіденційності фінансових даних становлять значну перешкоду для багатьох потенційних користувачів, особливо в контексті вимог GDPR.

Висока вартість підписки на розширені функції може відштовхнути деяких користувачів, тоді як надмірна складність деяких рішень може відлякати тих, хто шукає прості інструменти для щоденного використання.

Крім того, більшість застосунків орієнтовані на англomовні ринки, що обмежує їх доступність для користувачів в інших регіонах.

1.3 Постановка задачі проектування

Сучасне суспільство стикається зі зростаючими викликами в управлінні особистими фінансами, що є результатом складності сучасної фінансової системи та недостатньої економічної освіти. Більшість людей мають труднощі із систематичним відстеженням своїх витрат, плануванням бюджету та досягненням довгострокових фінансових цілей.

Сучасні технологічні рішення, хоча й численні, не повністю відповідають потребам польських користувачів. Більшість популярних додатків були розроблені з урахуванням західних ринків, що призводить до відсутності підтримки місцевої банківської специфіки, відмінностей у структурі витрат та різних моделей споживання.

Доступність та безпека також є ключовими питаннями. Багато передових рішень вимагають підключення до банківських рахунків, що викликає обґрунтовані побоювання у користувачів щодо безпеки даних. Водночас, локальні рішення часто характеризуються обмеженою функціональністю або застарілим інтерфейсом користувача. Дослідження показують, що 78% потенційних користувачів відмовляються від використання фінансових додатків через побоювання щодо безпеки персональних даних та банківської інформації. Водночас, 45% користувачів вважають існуючі рішення занадто складними для щоденного використання.

У відповідь на ці виклики, ця робота спрямована на розробку програмного модуля для фінансового відстеження, який характеризуватиметься високим рівнем безпеки, інтуїтивно зрозумілим керуванням та адаптацією до специфіки польського фінансового ринку. Система пропонуватиме комплексні інструменти для управління домашнім бюджетом, зберігаючи при цьому простоту використання.

Отже, метою цієї роботи є розробка та впровадження програмного модуля для відстеження та аналізу особистих фінансів, який поєднує функціональність, безпеку та простоту використання.

Для досягнення поставленої мети необхідно виконати ряд завдань:

- проаналізувати існуючі підходи до автоматизованого відстеження фінансових операцій, включно з методами, що базуються на машинному навчанні, глибокому навчанні та фінансовій аналітиці; визначити їх переваги та обмеження в умовах обробки великих обсягів транзакційних даних;
- сформулювати концепцію побудови інтелектуального програмного модуля для моніторингу фінансів, який би забезпечував автоматичне виявлення аномалій, підозрілих транзакцій або фінансових ризиків у режимі реального часу, використовуючи інтеграцію глибокого навчання та аналітики великих даних;
- розробити архітектуру програмного модуля для обробки фінансових потоків, яка забезпечує масштабованість, інтеграцію з різними джерелами даних;
- провести експериментальні дослідження ефективності розробленого модуля, зокрема порівняння його точності та продуктивності з традиційними методами фінансового моніторингу.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

2.1 Структурно-функціональне моделювання

Для моделювання функцій системи та організації бізнес-процесів використовуються структурні та об'єктно-орієнтовані методи. Одним з найбільших аналітичних інструментів для цієї мети є IDEF0. Цей метод дозволяє описати систему як набір взаємопов'язаних функцій на графічному рівні, що дає змогу аналізувати процеси без прив'язки до конкретних місць їх реалізації.

У стандарті IDEF0 управління відображає потоки інформації та ресурсів, які перетворюються на бізнес-процеси.

На рисунку 2.1 зображено функціональну схему IDEF0, яка відображає процеси введення, обробки та аналізу фінансових даних у системі.



Рисунок 2.1 – Діаграма IDEF0 процесу фінансового управління

Таким чином, використання підходу IDEF0 дозволяє ефективно організувати процеси фінансового обліку, забезпечуючи структурне управління та співпрацю між основними компонентами системи.

Більш детально про стрілки та їх опис написано в таблиці А.1.

2.2 Алгоритм аналізу фінансового стану та управління бюджетом

На рисунку 2.2 показано схему алгоритму аналізу фінансового стану та управління бюджетом, який включає основні процеси введення, обробки та візуалізації фінансових даних.



Рисунок 2.2 – Схема алгоритму аналізу фінансового стану та управління бюджетом

Загальний алгоритм аналізу фінансового стану та управління бюджетом включає такі етапи(рисунок 2.3):

- дані збираються з бази даних транзакцій за вибраний період часу;
- визначаються ключові фінансові показники (загальні витрати, доходи, баланс);
- проводиться аналіз відповідності витрат встановленому бюджету;
- користувач має можливість внести корективи до бюджету або змінити категорію витрат.;
- якщо бюджет перевищено, система надсилає сповіщення;
- діаграми та звіти створюються для візуалізації даних у доступному форматі.

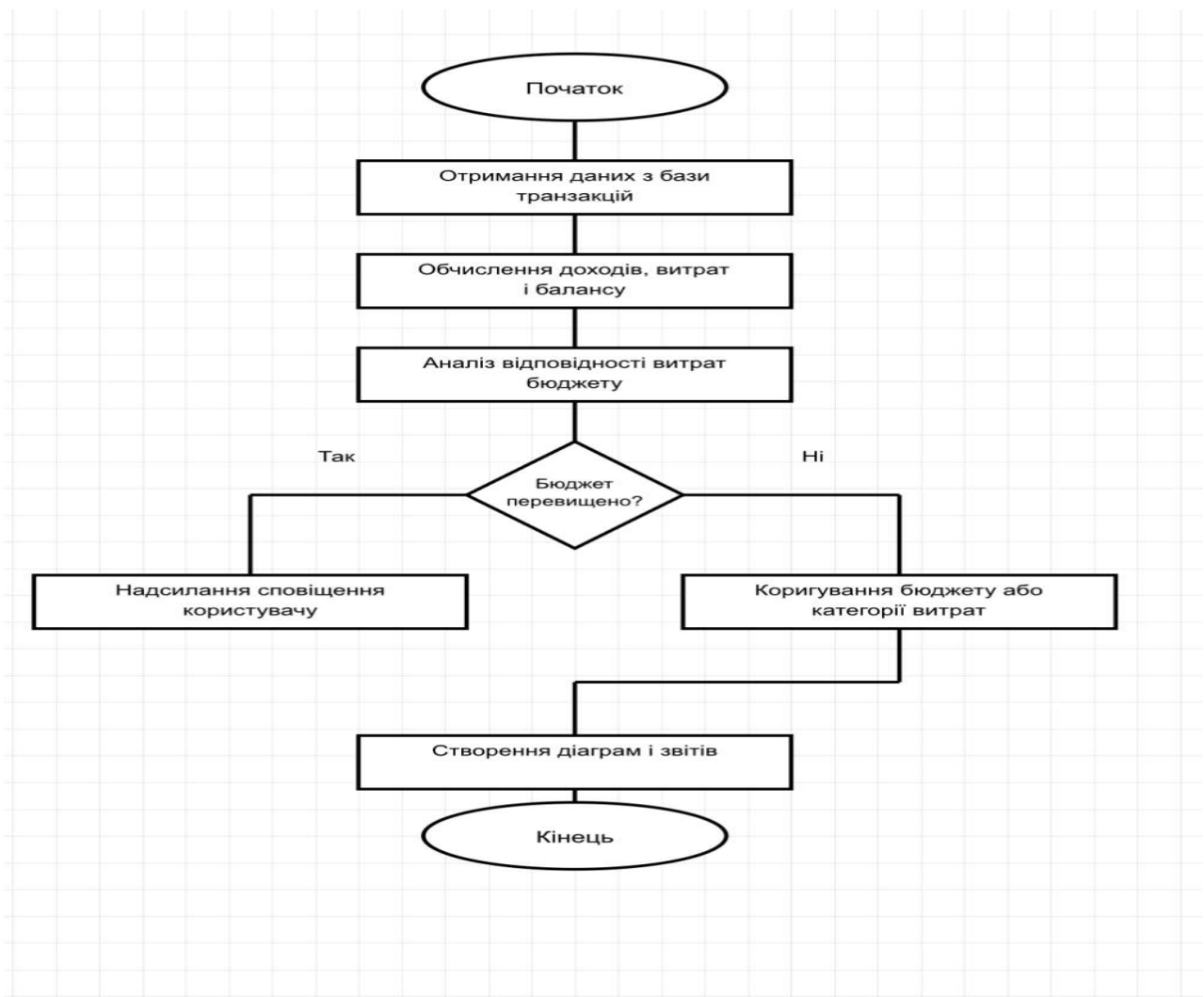


Рисунок 2.3 – Схема алгоритму аналізу та управління бюджетом

Загальний алгоритм процесу створення та перевірки фінансових звітів включає такі кроки (рисунок 2.4):

- отримання даних про фінансові операції користувача;
- аналіз витрат, визначення категорій з найбільшими витратами;
- генерація звітів у вигляді таблиць, діаграм та PDF-файлів;
- надсилання фінансових звітів на адресу електронної пошти користувача (за запитом);
- збереження історії звітів у базі даних для подальшого доступу.

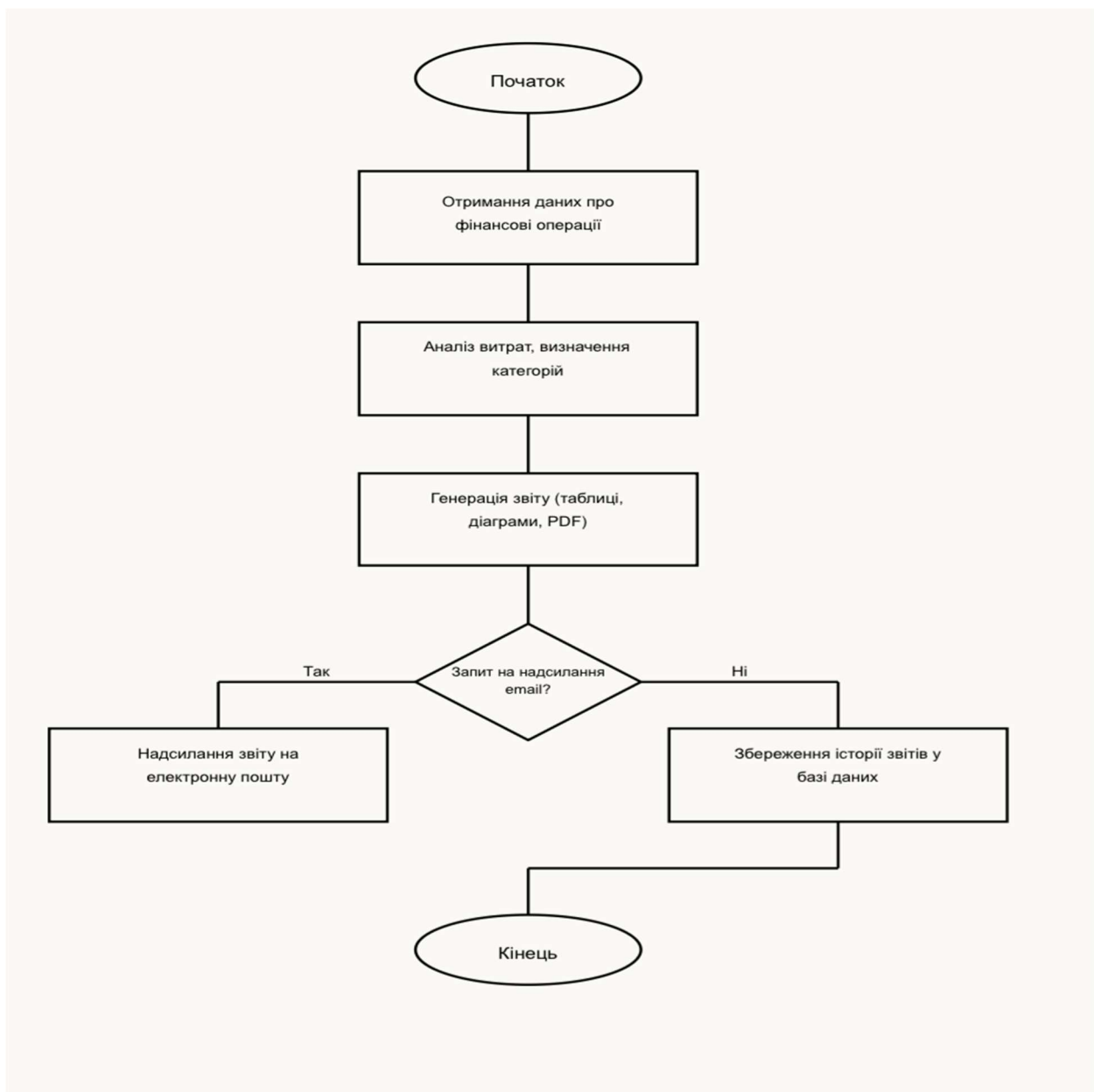


Рисунок 2.4 – Схема алгоритму фінансової звітності

Сучасний ринок пропонує широкий вибір програмних рішень для управління особистими фінансами. Кожен з доступних інструментів характеризується специфічним підходом до організації фінансових даних та різними функціональними можливостями.

Більшість доступних програм зосереджені на автоматизації процесів введення даних та створенні інтуїтивно зрозумілих користувацьких інтерфейсів. Американські рішення, такі як Mint або Personal Capital, пропонують розширену інтеграцію з банківськими системами, що дозволяє автоматично завантажувати інформацію про транзакції.

Європейські заявки часто зосереджені на універсальності та дотриманні місцевих фінансових правил. Такі додатки, як Spendee або MoneyLover, зосереджені на міжнародній функціональності з підтримкою різних валют та форматів даних.

Мобільний тренд домінує в сегменті фінансових додатків, оскільки користувачі очікують можливості керувати своїми бюджетами в режимі реального часу. Мобільні рішення дозволяють негайно фіксувати витрати та отримувати сповіщення про стан фінансів.

Ключовим завданням у проектуванні фінансових систем є забезпечення належного рівня безпеки даних користувачів. Впровадження протоколів шифрування та дотримання міжнародних стандартів, таких як PCI DSS, є основою довіри клієнтів.

Європейські регламенти, зокрема GDPR та директива PSD2, вимагають від постачальників фінансових додатків впроваджувати механізми захисту конфіденційності та надавати користувачам контроль над їхніми персональними даними.

Таким чином, запропонований алгоритм забезпечує ефективний нагляд за фінансами, їх аналіз та створення звітів, що дозволяє користувачеві контролювати свої фінансові потоки комплексно та безпечно.

2.3 Проектування моделі бази даних

Для роботи веб-орієнтованої системи відстеження фінансів була розроблена база даних, яка зберігає інформацію про користувачів, їхні фінансові операції, категорії витрат і доходів, бюджетні обмеження та звіти.

На рисунку 2.5 представлена структура бази даних, яка складається з п'яти основних таблиць: users, transactions, categories, budgets і reports.

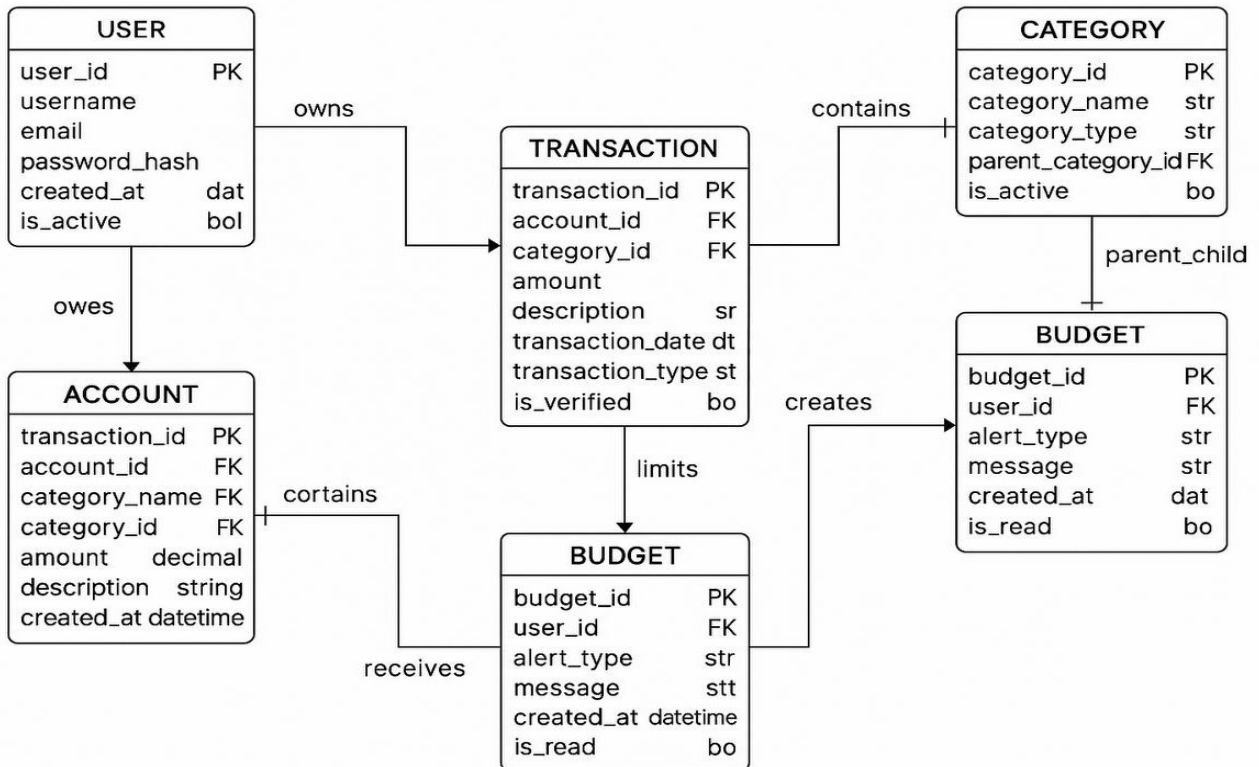


Рисунок 2.5 - Структура бази даних.

Таблиця users:

- id – унікальний ідентифікатор користувача.
- name – ім'я користувача.
- email – електронна пошта.
- password_hash – хеш пароля.
- date_registered – дата реєстрації.

Таблиця categories:

- id – унікальний ідентифікатор категорії.

- name – назва категорії (наприклад, "Продукти", "Транспорт").
- user_id – ідентифікатор користувача, який створив категорію.

Таблиця transactions:

- id – унікальний ідентифікатор транзакції.
- user_id – ідентифікатор користувача, якому належить транзакція.
- category_id – зв'язок з таблицею categories.
- amount – сума транзакції.
- type – тип операції (income або expense).
- date_created – дата створення.
- description – опис транзакції.

Таблиця budgets:

- id – унікальний ідентифікатор бюджету.
- user_id – ідентифікатор користувача.
- category_id – зв'язок з таблицею categories.
- amount_limit – ліміт бюджету для категорії.
- start_date – початок періоду бюджету.
- end_date – кінець періоду бюджету.

Таблиця reports:

- id – унікальний ідентифікатор звіту.
- user_id – ідентифікатор користувача.
- date_generated – дата створення звіту.
- report_data – збережена інформація про фінансовий звіт у форматі

JSON.

Зв'язки між таблицями:

- transactions -> users: Кожна транзакція належить конкретному користувачеві через user_id.
- transactions -> categories: Транзакції відносяться до певних категорій через category_id.
- budgets -> users: Кожен бюджет прив'язаний до користувача.

- budgets -> categories: Бюджетні обмеження встановлюються для певних категорій витрат.
- reports -> users: Кожен фінансовий звіт прив'язаний до конкретного користувача.

Проектування цієї моделі бази даних забезпечує зручне та ефективне управління фінансовими даними користувачів. Структура бази даних забезпечує масштабованість, безпеку та ефективне управління даними. Ця модель забезпечує міцну основу для розвитку функціональності фінансового трекера, дозволяючи легко додавати нові функції та адаптувати їх до змінних вимог користувачів.

2.4 Архітектура програмного модуля

Сучасні фінансові програми вимагають надійної архітектури, яка забезпечує масштабованість, безпеку та легкість обслуговування. Розроблений модуль використовує шаблон Model-View-Controller (MVC), який є перевіреним рішенням у проектуванні вебзастосунків.

Архітектура MVC поділяє систему на три основні рівні, кожен з яких відповідає за певні функціональні можливості:

Модель (рівень даних) – відповідає за управління фінансовими даними користувачів, зв'язок з базою даних та реалізацію бізнес-логіки. У контексті фінансового застосунку це включає операції з транзакціями, категоріями витрат, бюджетами та звітами. Цей рівень забезпечує цілісність даних, впроваджуючи відповідні перевірки та обмеження.

Представлення (рівень презентації) – відповідає за представлення даних користувачеві в доступній графічній формі. Він включає інтерфейси для перегляду транзакцій, створення фінансових звітів, візуалізації даних у вигляді діаграм та форм для введення нових фінансових операцій.

Контролер (рівень керування) – виступає посередником між рівнями даних та представлення, обробляючи запити користувачів та координуючи

потік інформації в системі. Контролери обробляють логіку, пов'язану з автентифікацією, авторизацією та обробкою HTTP-запитів.

Процес обробки запитів у системі виглядає наступним чином:

- користувач ініціює запит, взаємодіючи з веб-інтерфейсом, наприклад, натискаючи кнопку для додавання нової транзакції або запитуючи створення фінансового звіту;
- система маршрутизації аналізує запит і направляє його до відповідного контролера на основі URL-адреси та методу HTTP;
- контролер обробляє запит, виконуючи необхідні операції перевірки вхідних даних та прав доступу користувача;
- зв'язок відбувається з шаром моделі, який отримує або змінює дані в базі даних, виконує фінансові розрахунки або генерує необхідні звіти;
- контролер отримує оброблені дані з шару моделі та передає їх до відповідного представлення разом із додатковими параметрами форматування;
- рівень презентації генерує відповідь у вигляді HTML-сторінки з фінансовими даними, діаграмами або повідомленнями про стан операцій, яку користувач отримує у браузері.

Використання архітектури MVC у фінансовому модулі має такі переваги: вона дозволяє незалежну розробку окремих системних рівнів, полегшує тестування шляхом ізоляції бізнес-логіки, збільшує можливість повторного використання коду та спрощує процес впровадження модифікацій та розширень функціональності.

Крім того, архітектура підтримує реалізацію шаблонів проектування, таких як шаблон репозиторію для доступу до даних або шаблон рівня сервісу для бізнес-логіки, що ще більше підвищує якість та зручність підтримки коду.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Програмна реалізація

Перед початком розробки програмного модуля для відстеження фінансів було обрано сучасні веб-технології, що забезпечують високу продуктивність, масштабованість, безпеку та зручність користувача. Основною метою проєкту стало створення функціонального веб-застосунку, який дозволяє користувачам зручно керувати своїми фінансами, аналізувати витрати і доходи, формувати бюджети, будувати графіки та отримувати детальні звіти. Правильний вибір технологій має вирішальне значення для фінансових систем, особливо в контексті безпеки даних, швидкості обробки транзакцій і зручності масштабування. У процесі розробки було використано такі інструменти:

- HTML5 і CSS3 — для побудови структури веб-сторінок та стилізації інтерфейсу користувача;
- JavaScript — для реалізації динамічних та інтерактивних елементів на стороні клієнта;
- Bootstrap — CSS-фреймворк, який забезпечує адаптивність інтерфейсу на різних пристроях;
- Laravel (PHP) — серверний фреймворк, який пропонує елегантний синтаксис, потужні інструменти для безпечної авторизації, обробки запитів та роботи з базами даних;
- MySQL — реляційна СУБД, що гарантує збереження фінансових даних із підтримкою ACID-транзакцій;
- Laravel Sanctum — для безпечної авторизації користувачів через API;
- Laravel Queue — для асинхронної обробки задач, зокрема створення звітів та розсилки сповіщень;
- Chart.js — бібліотека для побудови графіків витрат, доходів та бюджетів;
- JetBrains PhpStorm / IntelliJ IDEA — професійні середовища розробки, що використовувалися під час написання коду.

На рисунку 3.1 зображено автоматичне оновлення змін через hot reload у dev-середовищі (вміст файлу webpack.mix.js).

```
const mix = require('laravel-mix');

mix.js('resources/js/app.js', 'public/js')
  .sass('resources/sass/app.scss', 'public/css')
  .version(); // Для кешування в production
```

Рисунок 3.1 – Вміст файлу webpack.mix.js

Для забезпечення безпеки конфіденційної інформації (паролі, токени доступу, дані бази) використовувалися конфігураційні файли .env (рисунок 3.2). Це дозволяє ізолювати чутливу інформацію від коду, який зберігається в репозиторіях.

```
APP_NAME=FinanceTracker
APP_ENV=local
APP_KEY=base64:somebase64keyhere==
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=finance_db
DB_USERNAME=root
DB_PASSWORD=your_secure_password

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=database
SESSION_DRIVER=file
SESSION_LIFETIME=120

SANCTUM_STATEFUL_DOMAINS=localhost:3000
API_URL=http://localhost/api

MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=your_mailtrap_user
MAIL_PASSWORD=your_mailtrap_pass
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=finance@local.com
MAIL_FROM_NAME="Finance Tracker"

AWS_COGNITO_POOL_ID=your_pool_id
AWS_COGNITO_CLIENT_ID=your_client_id
AWS_COGNITO_REGION=your_region
|
```

Рисунок 3.2 – Вміст файлу .env

Авторизація реалізована через Laravel Sanctum для API-запитів, а також сесіями для звичайного веб-інтерфейсу. Було реалізовано рольову модель

доступу, яка дозволяє розмежувати права між адміністраторами та звичайними користувачами.

У базі даних MySQL створено основні таблиці:

- users — зберігає дані користувачів;
- transactions — реєструє всі фінансові операції;
- categories — категорії витрат та доходів;
- budgets — інформація про задані бюджети користувачів;
- reports — зберігання згенерованих звітів.

Таблиці містять відповідні індекси, що дозволяє пришвидшити пошук і фільтрацію великих обсягів фінансових даних.

Рівень бізнес-логіки реалізовано за шаблоном Service + Repository, що дозволяє гнучко розділяти функціональні частини програми:

- сервісні класи відповідають за основну логіку — розрахунки бюджетів, генерацію звітів, перевірку транзакцій;
- репозиторії абстрагують доступ до бази даних, полегшуючи тестування та зміну структури даних у майбутньому.

Також використано систему черг завдань Laravel Queue, яка дозволяє відкладати тривалі операції (наприклад, формування звіту на основі декількох тисяч транзакцій), не блокуючи роботу інтерфейсу.

Валідація вхідних даних реалізована на двох рівнях:

- клієнтський рівень — перевірка через HTML5 та JavaScript;
- серверний рівень — перевірка за допомогою Laravel Form Requests.

Це забезпечує захист від SQL-ін'єкцій, XSS та інших векторів атак, що критично важливо у фінансових веб-застосунках.

3.2 Інтерфейс користувача

При завантаженні web-додатку для відстеження фінансів, користувач попадає на вітальна сторінку (рисунок 3.3).

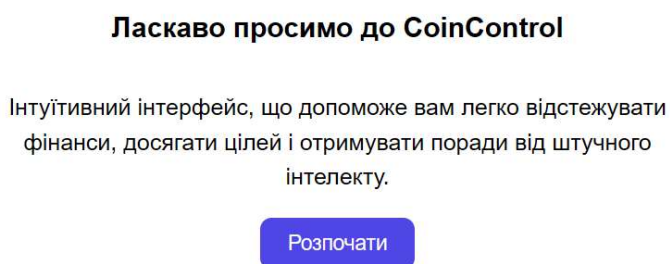


Рисунок 3.3– Вітальна сторінка веб-орієнтованої інформаційної системи

Користувач повинен зареєструватися, щоб отримати доступ до всіх наявних функцій, або здійснити вхід у вже існуючий акаунт. Рисунки 3.4–3.5 показують форми для реєстрації та авторизації.

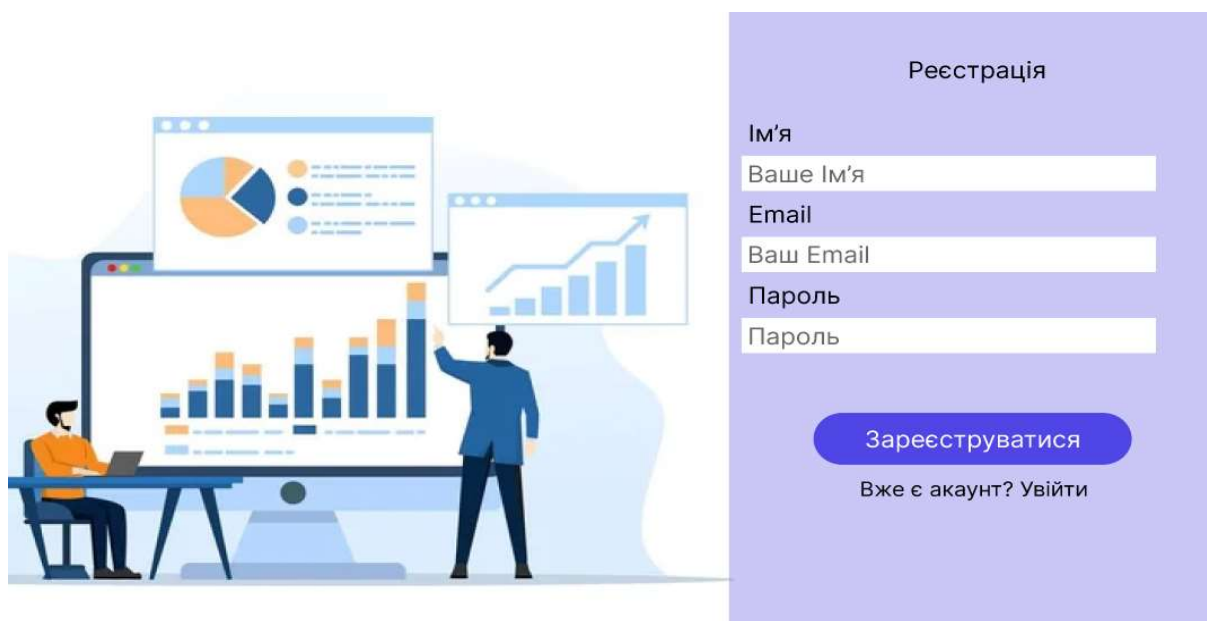


Рисунок 3.4 - Форма реєстрації через поштову адресу

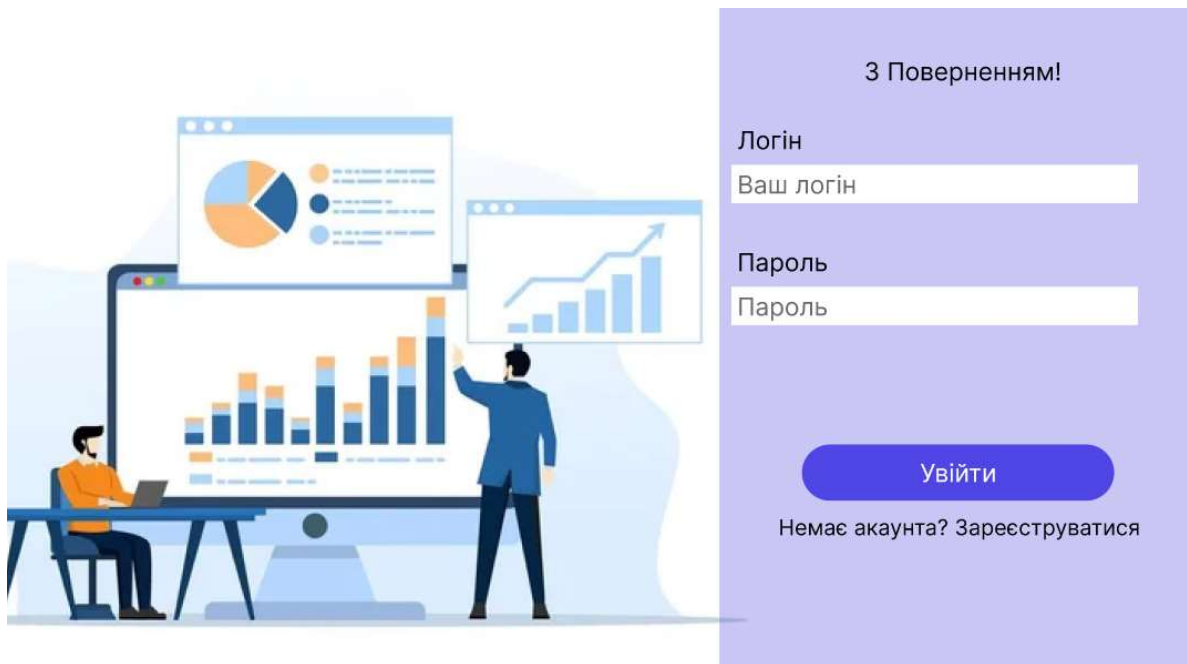


Рисунок 3.5 – Форма авторизації

Наступною, після вітальної сторінки ми попадаємо на головну сторінку (рисунок 3.6) на якій можна побачити таку інформацію як: загальний баланс, доходи цього місяця, витрати, та бюджет на поточний місяць.

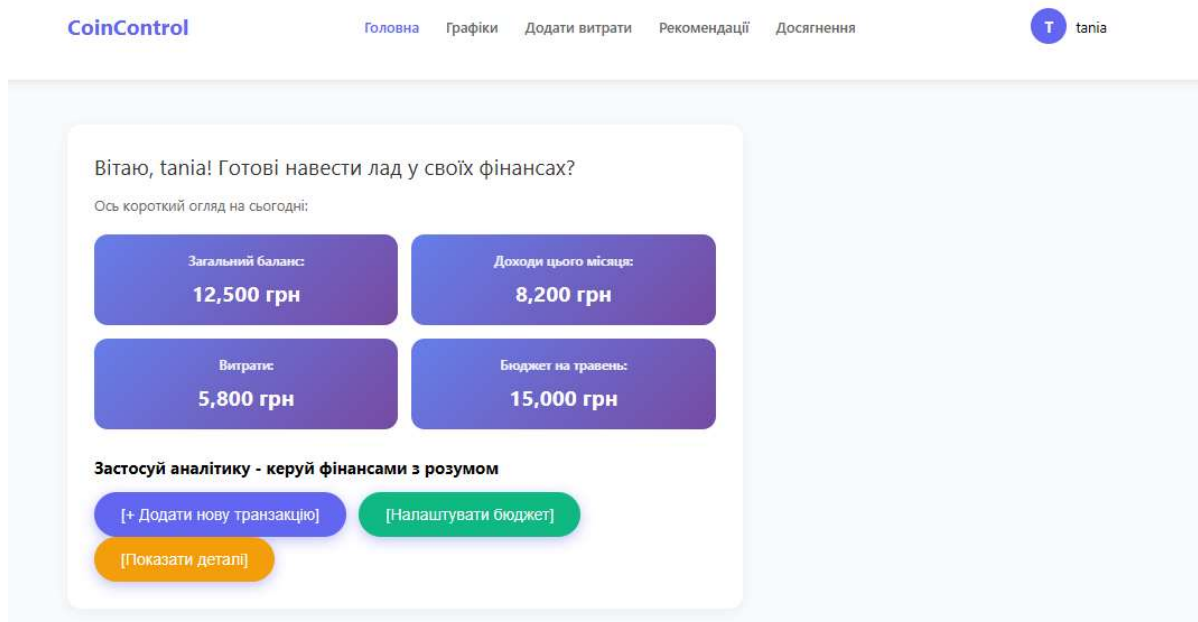


Рисунок 3.6– Головна сторінка

“Графіки” є наступним пунктом меню, як показано на рисунку 3.7, 3.8. Це сторінка на якій відображаються графіки та діаграми ось їх назви “Зведена діаграма витрат”, “Динаміка витрат”, “Доходи і витрати”, “Прогрес фінансової цілі”.

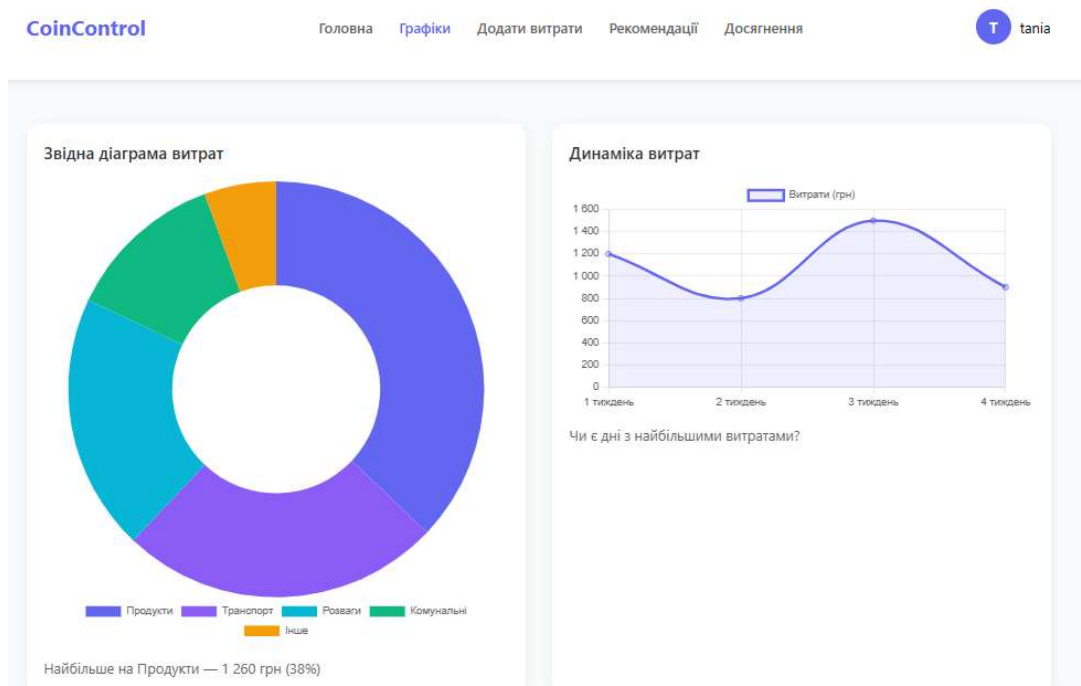


Рисунок 3.7 - Зведена діаграма витрат

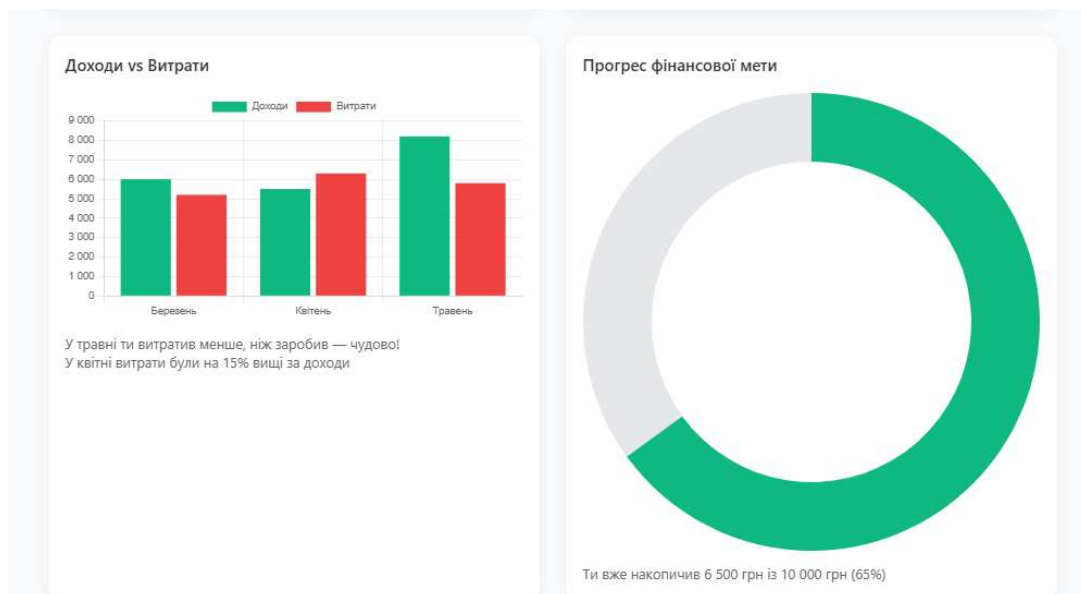


Рисунок 3.8 – Графік доходів і витрат

Також є сторінки “Рекомендації” та “Досягнення” (рисунки 3.9–3.10 відповідно). Сторінка рекомендації відображає рекомендації щодо ваших витрат та контролюю фінансів в поточному та наступному місяці або році. “Досягнення” – це твої успіхи в заощадженні бюджету, прогрес та відкриті бейджі за накопичення та витрати.

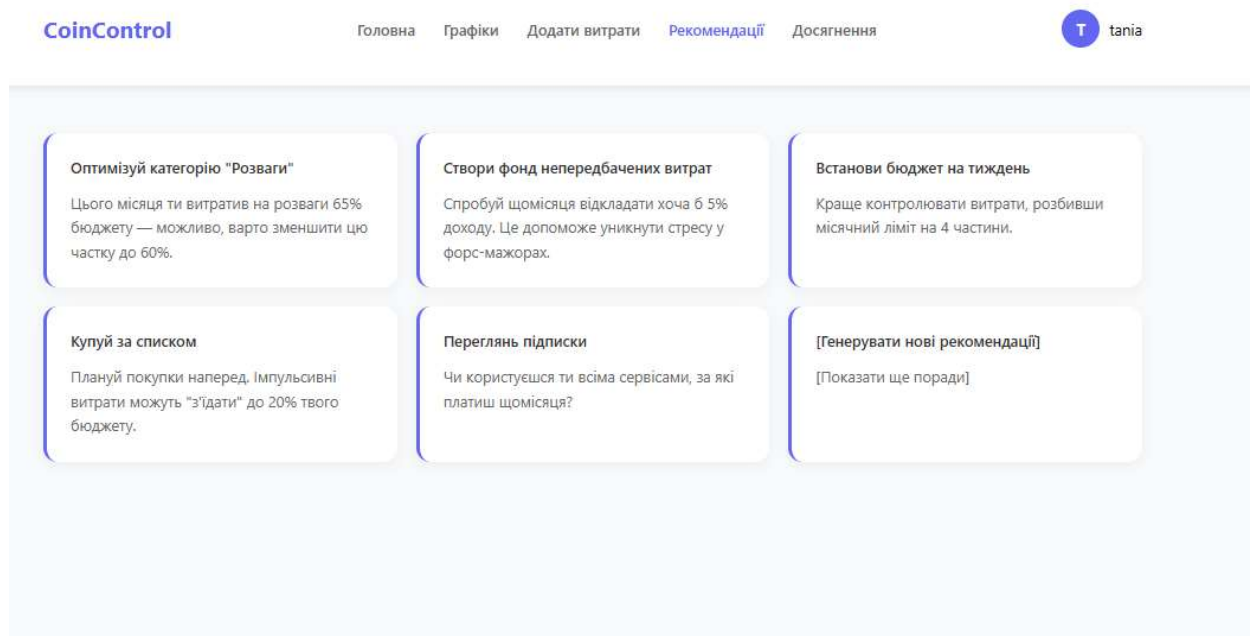


Рисунок 3.9– Сторінка Рекомендації

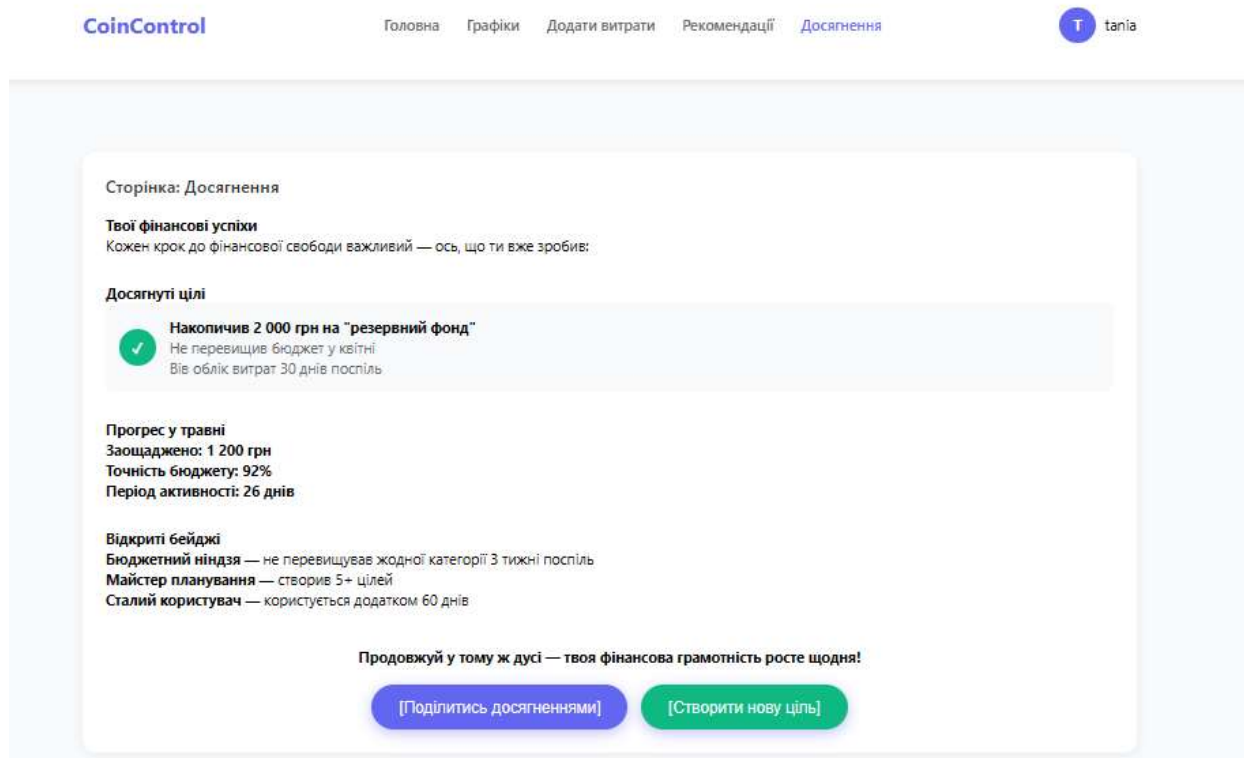


Рисунок 3.10– Сторінка Досягнення

Інтерфейс користувача веб-орієнтованої інформаційної системи відстеження фінансів забезпечує зручну навігацію та доступ до основних функцій системи.

На головній сторінці представлені основні дані щодо витрат та надходження коштів, а меню дозволяє користувачам переходити до інших розділів, таких як графіки, рекомендації та досягнення. Користувачі повинні зареєструватися або увійти до свого облікового запису, щоб мати повний доступ до функцій, включаючи додавання витрат та надходження, рекомендації від ШІ, та отримання досягнень. Інтерфейс також включає форми для реєстрації та авторизації.

ВИСНОВКИ

1. Робота з розробки програмного модуля для відстеження та аналізу фінансових операцій продемонструвала високу ефективність створеного рішення, що забезпечує повноцінну функціональність для керування домашнім бюджетом користувачів.

2. Початковий етап дослідження включав комплексне вивчення галузі персонального фінансового менеджменту та стратегічного планування коштів. Виявлено ключові недоліки наявних сервісів щодо всебічного моніторингу сімейного бюджету та проблематичність мануального фіксування грошових потоків. Такий підхід сприяв формуванню чіткого бачення основних цілей створюваної веб-платформи. Проаналізовано функціонал існуючих фінансових додатків, серед яких Mint, YNAB, PocketGuard та аналогічні продукти. Здійснено детальний розбір їхніх сильних сторін та вразливих місць, що стало підґрунтям для формулювання технічних специфікацій майбутньої системи. Даний аналітичний огляд забезпечив встановлення функціональних та експлуатаційних критеріїв до розроблюваного продукту, а також дозволив уникнути стандартних прорахунків, притаманних аналогічним платформам.

3. Другий розділ роботи представив структурно-архітектурне планування системи через застосування IDEF0 методології. Створено концептуальну модель, що ілюструє базові функціональні модулі платформи та механізми їх інтеграції. Документовано користувацькі ролі, центральні бізнес-сценарії фінансового контролю та їх кореляції. Розроблено комплексний алгоритмічний опис функціонування платформи, що охоплює процедури збирання, опрацювання та інтерпретації монетарних даних у логічній послідовності. Контекстна IDEF0 діаграма відображає синергію головних елементів фінансового управлінського комплексу. Додатково створено алгоритмічні схеми для різноманітних операцій, включно з бюджетним аналізом, формуванням фінансової звітності, та архітектурою інформаційного сховища. Вони становлять фундамент для технічної імплементації системи.

4. Інформаційне сховище спроектовано та втілено засобами клієнтських веб-технологій з використанням локального збереження даних. Створено п'ять центральних структур: користувачі, операції, класифікації, бюджетні плани та аналітичні матеріали, які гарантують надійне архівування персональних фінансових відомостей. Для результативного управління значними масивами грошових транзакцій застосовано передові технологічні підходи. Розроблена архітектура даних задовольняє усі проектні вимоги та забезпечує захищене збереження приватних фінансових відомостей через механізми валідації та контролю цілісності.

5. HTML, CSS та JavaScript - це технологічний набір, обраний для створення основного функціоналу веб-додатку. Обслуговування користувацьких запитів, маніпулювання фінансовими масивами, автоматизована класифікація операцій та координація між компонентами системи реалізовані через клієнтську частину програми. Для гарантування високої швидкодії та стабільності застосовано сучасні JavaScript бібліотеки та фреймворки, включаючи Local Storage API для збереження даних та Web Workers для асинхронного виконання задач. Особливий акцент зроблено на створенні інтуїтивного користувацького досвіду. Платформа характеризується зрозумілим інтерфейсом з наочною презентацією фінансових показників через діаграми та графічні елементи, адаптованим для широкого спектра користувачів - від студентської аудиторії до багатодітних родин. Проведено тестування інтерфейсних рішень для підтвердження відповідності ергономічним стандартам та принципам usability.

6. Розроблений модуль містить наступні основні функціональні компоненти: автоматизовану класифікацію фінансових операцій, формування та адміністрування бюджетних планів, генерування комплексних фінансових аналітичних матеріалів, візуалізацію інформації через різноманітні графічні представлення, систему персональних рекомендацій щодо оптимізації витратної частини, а також гейміфіковані елементи для стимулювання

користувачів. Клієнт-серверна архітектура забезпечує горизонтальну масштабованість системи та спрощує процес майбутніх удосконалень.

7. Значну увагу приділено аспектам інформаційної безпеки фінансових даних. Реалізовано багаторівневу перевірку вхідних даних, захищені протоколи ідентифікації користувачів та авторизації доступу, а також захисні механізми проти поширених типів веб-загроз. Система відповідає актуальним безпековим регламентам для фінансових веб-сервісів.

8. Узагальнюючи, реалізація даної роботи дозволила розробити функціональний програмний модуль для трекінгу та аналітики персональних фінансів, яка повністю задовольняє визначені технічні вимоги та гарантує високі показники якості і надійності функціонування. Система демонструє потужний потенціал для подальшого розвитку та модернізації, включаючи перспективи інтеграції з банківськими API, розширення аналітичного інструментарію та розробки мобільних клієнтів, що відкриває додаткові можливості для підвищення рівня фінансової грамотності в сучасному цифровому суспільстві.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kapoor J. R., Dlabay L. R., Hughes R. J., Hart M. Personal Finance. New York: McGraw-Hill Education, 2019.
2. Bodie Z., Kane A., Marcus A. J. Investments. New York: McGraw-Hill Education, 2018.
3. Tyson E. Personal Finance For Dummies. Hoboken: John Wiley & Sons, 2020.
4. Bach D. The Automatic Millionaire: A Powerful One-Step Plan to Live and Finish Rich. New York: Broadway Books, 2018.
5. Malkiel B. G. A Random Walk Down Wall Street. New York: W. W. Norton & Company, 2019.
6. Ramsey D. The Total Money Makeover: Classic Edition. Nashville: Thomas Nelson, 2013.
7. Siegel J. J. Stocks for the Long Run. New York: McGraw-Hill Education, 2014.
8. Consumer Financial Protection Bureau. Financial Well-Being in America. Washington, D.C.: CFPB, 2019.
9. Federal Reserve Bank of St. Louis. Household Debt Service Payments as a Percent of Disposable Personal Income. St. Louis: FRED Economic Data, 2020.
10. McKinsey & Company. The Future of Work in Technology. McKinsey Global Institute, 2021.
11. Statista. Personal Finance Software Market Size Worldwide. Market Research Report, 2022.
12. Plaid Inc. The Rise of Fintech Applications. Industry Analysis Report, 2021.
13. Open Banking Implementation Entity. Open Banking Standards. Technical Documentation, 2020.
14. Accenture. AI in Financial Services. Technology Trends Report, 2020.

15. Nielsen Norman Group. Data Visualization Guidelines. UX Research Report, 2019.
16. Intuit Inc. Small Business Financial Management Trends. Market Research, 2021.
17. Xero Limited. Cloud Accounting Adoption Report. Industry Survey, 2020.
18. Microsoft Corporation. Excel for Financial Planning. Product Documentation, 2021.
19. CoinMarketCap. Cryptocurrency Portfolio Management. Market Analysis, 2021.
20. Deloitte. Personalization in Financial Services. Industry Report, 2020.
21. European Banking Authority. GDPR Compliance in Financial Applications. Regulatory Guidelines, 2019.
22. PwC. Fintech Pricing Strategies. Market Analysis Report, 2021.
23. Capgemini. World Fintech Report. Global Industry Analysis, 2020.
24. OECD. Financial Literacy Around the World. Economic Survey, 2020
25. National Endowment for Financial Education. Personal Financial Management Study. Research Report, 2019.
26. Urząd Ochrony Danych Osobowych. Безпека даних у фінансових додатках. Регуляторні рекомендації. Варшава: UODO, 2020.
27. Związek Banków Polskich. Стан польського банківського сектору. Щорічний звіт. Варшава: ZBP, 2021.
28. Дорош В.І., Вередюк Т.П. Автоматизація процесів планування та моніторингу проєктів з використанням штучного інтелекту. Збірник тез доповідей студентської науково-практичної конференції “Інтелектуальні інформаційні технології в прикладних дослідженнях” (ІТАР-2025). Тернопіль: ЗУНУ, 2025 330-333 с.
29. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп’яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки»

спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

30. Островерхов В. М., Біловус Л. І., Возьний К. З., Луцишин О. О., Монастирський Г. Л., Надвиничний С. А., Питель С. В., Шандрук С. К. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів / Укладачі: Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А
Копія публікації

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління



ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції
ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ
ДОСЛІДЖЕННЯХ
(ІПТАР-2025)

27-29 травня 2025 року

Тернопіль
2025

Свірський Максим, Кочан Володимир	315
АРХІТЕКТУРА СИСТЕМИ РОЗУМНОЇ БІБЛІОТЕКИ НА БАЗІ ІoT І SDN	315
Трач Мар'ян, Осолінський Олександр	319
КОНЦЕПЦІЯ ПРОГРАМНО КЕРОВАНОЇ МОДУЛЬНОЇ АРХІТЕКТУРИ ІОТ	319
Цанько Андрій, Осолінський Олександр	322
ІНТЕЛЕКТУАЛЬНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ ТЕПЛИЦІ НА БАЗІ ІОТ	322
Циб Андрій	325
ІНТЕЛЕКТУАЛЬНА СИСТЕМА ВИЯВЛЕННЯ АТАК У СЕРЕДОВИЩІ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ ТУМАННИХ ОБЧИСЛЕНЬ	325
СЕКЦІЯ 4. ШІ В УПРАВЛІННІ ПРОЄКТАМИ	327
Бубнюк Надія, Черній Іван	327
ОГЛЯД ІНТЕЛЕКТУАЛЬНИХ МЕТОДІВ В УПРАВЛІННІ ПРОЄКТАМИ	327
Вередюк Тетяна, Дорош Віталій	330
АВТОМАТИЗАЦІЯ ПРОЦЕСІВ ПЛАНУВАННЯ ТА МОНІТОРИНГУ ПРОЄКТІВ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	330
Грицюк Іванна, Гладій Григорій	333
ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ УПРАВЛІННЯ ПРОЦЕСАМИ ГУМАНІТАРНИХ МІСІЙ НА ОСНОВІ РСА ТА КЛАСТЕРИЗАЦІЇ	333
Демчук Максим, Лендюк Тарас	336
МОДУЛЬ КЛАСИФІКАЦІЇ КОМЕНТАРІВ REDDIT З ВИКОРИСТАННЯМ МОДЕЛІ BERT	336
Кузьмич Богдан, Биковий Павло	339
ПРОГРАМНИЙ МОДУЛЬ ПРОГНОЗУВАННЯ ПЕРЕШКОД У ЛАБІРИНТІ НА ОСНОВІ МАШИННОГО НАВЧАННЯ	339
Лимар Вікторія, Турченко Ірина	342
ІНФОРМАЦІЙНА ПАНЕЛЬ ЗГАДОК ДЛЯ ВІДСТЕЖЕННЯ ТА КЕРУВАННЯ КОМУНІКАЦІЯМИ ПРОГРАМНОГО СЕРЕДОВИЩА JIRA	342
Лось Марія, Ліп'яніна-Гончаренко Христина	346
ВИКОРИСТАННЯ МОДЕЛІ CodeBERT ДЛЯ АНАЛІЗУ ТА ГЕНЕРАЦІЇ КОДУ В СУЧАСНОМУ ПРОГРАМУВАННІ	346
Магильницький Дмитро, Коваль Василь	349
МЕТОД ПОБУДОВИ ОПТИМАЛЬНИХ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ПРЯМОГО ПОШИРЕННЯ	349

Вередюк Тетяна
студентка групи КН-41
t.verediuk@st.wunu.edu.ua

Дорош Віталій

викладач

vdo@wunu.edu.ua

Західноукраїнський національний університет
Тернопіль, Україна

АВТОМАТИЗАЦІЯ ПРОЦЕСІВ ПЛАНУВАННЯ ТА МОНІТОРИНГУ ПРОЄКТІВ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Штучний інтелект - це проривна технологія, яка трансформує управління проектами шляхом автоматизації процесів планування та моніторингу, що призводить до значного підвищення ефективності, точності прогнозування та зниження ризику невдачі проекту [1]. Основа полягає в тому, що алгоритми машинного навчання не лише здатні замінити традиційні, ручні методи планування та моніторингу проектів, але, перш за все, вони можуть забезпечити значно кращі результати завдяки здатності обробляти величезні набори історичних даних та виявляти залежності, невидимі для людини [2]. Сучасні проекти характеризуються зростаючою складністю, збільшенням кількості зацікавлених сторін та динамічно змінюваними вимогами, а це означає, що традиційні інструменти управління проектами часто виявляються недостатніми [3]. Дослідження показують, що понад 70% проектів перевищують свій бюджет, причому значна частина невдач є наслідком неточних оцінок та затримки з виявленням ризиків. Саме в цих сферах штучний інтелект демонструє свій найбільший трансформаційний потенціал. На відміну від традиційних методів, рішення на основі штучного інтелекту здатні постійно навчатися на нових даних, адаптуючи рекомендації до змінних умов проектування в режимі реального часу. Це особливо цінно у високоінноваційних проектах, де історичні дані можуть мати обмежену прогностичну цінність(таблиця 1).

Таблиця 1 – Ключові переваги систем штучного інтелекту в управлінні проектами.

	Традиційні методи	Рішення на основі AI	Переваги
Оцінка	На основі досвіду експертів	Аналіз історичних даних і контекстних параметрів	Зменшення похибки оцінки на 50-70%[4]
Планування	Вручну, часто не оптимальний результат	Алгоритми оптимізації з урахуванням сотень змінних	Скорочення термінів реалізації проекту на 15-25%[3,5]
Моніторинг	Періодичні, звітні	Безперервний, автоматичний, з прогнозуванням тренду	Виявлення проблем в середньому на 10 днів раніше[8]
Управління ресурсами	Статичний розподіл на основі доступності	Динамічна оптимізація з урахуванням компетенцій і навантаження	Підвищення продуктивності команди на 20-30%[9]
Спілкування	Регулярні зустрічі та звіти	Автоматична ідентифікація ключової інформації та її розподіл	Скорочення часу, витраченого на спілкування, на 40%[9,10]

Впровадження систем штучного інтелекту в організаціях стикається з низкою специфічних перешкод, які необхідно систематично вирішувати. Дослідження, проведене серед 120 організацій, що впроваджують рішення на основі штучного інтелекту в управлінні проектами, вказує на п'ять основних перешкод: недостатня якість історичних даних (78%), відсутність відповідних компетенцій у командах (65%), культурний опір (61%), проблеми з інтерпретацією результатів (57%) та труднощі з інтеграцією з існуючими системами (52%) [4, 5]. Подолання цих перешкод вимагає комплексного підходу, який включає:

1. впровадження процесів, що забезпечують високу якість зібраних проектних даних;
2. розвиток компетенцій з аналізу даних серед керівників проектів;
3. поступове впровадження змін за активної участі команди;
4. застосування методів пояснюваного штучного інтелекту (XAI);
5. розробка рішень на основі штучного інтелекту як розширення існуючих систем, а не як заміни; [6-8]

Підсумовуючи, автоматизація процесів планування та моніторингу проектів за допомогою штучного інтелекту є не лише природним продовженням існуючих практик управління проектами, але й фундаментальною зміною парадигми, що призводить до значного підвищення рівня успішності проектів у бізнес-середовищі [8,9]. Прогнози показують, що до 2027 року понад 80% організацій, які реалізують складні проекти, використовуватимуть рішення на основі штучного інтелекту для підтримки процесів прийняття рішень в управлінні проектами [9]. Подальший розвиток автоматизації управління проектами, ймовірно, буде спрямований на глибшу інтеграцію з іншими цифровими технологіями, такими як Інтернет речей (IoT), доповнена реальність

(AR) або блокчейн, що дозволить ще точніше контролювати та прогнозувати параметри проекту. Однак ключовим завданням залишається пошук балансу між автоматизацією та збереженням людського нагляду та відповідальності, особливо в проектах з високим ризиком або сильним суспільним впливом.

Список використаних джерел

1. Сміт Дж., Браун Т. Штучний інтелект у сучасному управлінні проектами: комплексний аналіз // Журнал управління проектами. – 2023. – Т. 45, № 2. – С. 112–128.
2. Чен Й., Вілсон Д. Підходи машинного навчання до ідентифікації ризиків у складних проектах // Міжнародний журнал управління ризиками. – 2023. – Т. 18, № 4. – С. 345–361.
3. Джонсон К., Мартінес А. Еволюція практик управління ризиками в цифрову епоху // Щоквартальник з управління проектами. – 2022. – Т. 37, № 3. – С. 78–95.
4. Патель Р., Андерсон С. Обробка природної мови для автоматизованого виявлення ризиків у проектній документації // IEEE Transactions on Engineering Management. – 2024. – Т. 71, № 1. – С. 45–62.
5. Лі Х., Ван Дж. Порівняльний аналіз методів ідентифікації ризиків на основі штучного інтелекту в IT-проектах // Інформаційні технології та люди. – 2023. – Т. 36, № 2. – С. 178–194.
6. Вільямс П., Томпсон Г. Моделювання методом Монте-Карло, покращене глибоким навчанням, для оцінки ризиків проекту // Міжнародний журнал організації та управління проектами. – 2023. – Т. 15, № 3. – С. 267–285.
7. Ерікссон Т., Лі М. Прийняття користувачами інструментів управління ризиками на основі штучного інтелекту: емпіричне дослідження // Міжнародний журнал управління проектами. – 2024. – Т. 42, № 3. – С. 289–305.
8. Купер Л., Девіс Н. Майбутнє управління ризиками проектів: співпраця людини та штучного інтелекту // Гарвардський бізнес-огляд. – 2024. – Т. 102, № 2. – С. 112–124.
9. Ернандес В., Ямамото К. Пояснювальний ШІ в управлінні ризиками проектів: поточний стан та майбутні напрямки // Технологічне прогнозування та соціальні зміни. – 2023. – Т. 188.

Додаток Б

Опис стрілок з IDEF0-діаграми

Таблиця А.1 - Опис стрілок з IDEF0-діаграми

Назва стрілки		Опис
Вхід	Фінансові транзакції	- Дані, які містять інформацію про всі грошові операції: надходження, витрати, перекази. - Є основою для розрахунків, обліку та аналітики фінансової системи. - Можуть надходити з банківських систем, платіжних сервісів або бути введені вручну. - Містять важливі реквізити: дата, сума, тип операції, контрагент. - Трансформуються у категорії або піддаються фільтрації для аналізу.
	Запити користувачів	- Ініціюють конкретні дії в системі, наприклад: “створити звіт”, “показати баланс”, “аналіз витрат”. - Формуються користувачем через інтерфейс у вигляді команд, натискань, фільтрів. - Можуть бути як одноразовими, так і регулярними (періодичні звіти). - Визначають обсяг, глибину та формат виведених даних. - Враховують права доступу, які визначають, хто і які дані може запитувати.
	Параметри користувача	- Встановлюють індивідуальні налаштування для кожного користувача (валюта, мова, часовий пояс). - Визначають роль користувача (бухгалтер, керівник, аналітик), яка впливає на доступ до функцій. - Зберігаються в профілі користувача і застосовуються автоматично при роботі в системі. - Забезпечують персоналізоване відображення інтерфейсу та звітів. - Можуть змінюватися вручну або відповідно до політик підприємства.
Керування	Правила обліку фінансів	- Встановлюють методики ведення бухгалтерського обліку (наприклад, згідно з МСФЗ або національними стандартами). - Визначають, як формуються документи, проводки, облік амортизації тощо. - Регламентують періодичність обліку, закриття періодів, валютну переоцінку. - Можуть бути внутрішні (для компанії) або зовнішні (законодавчі вимоги). - Автоматично впливають на обробку транзакцій у системі.

Назва стрілки		Опис
	Алгоритми категоризації транзакцій	- Автоматично визначають тип транзакції за її параметрами (наприклад, “зарплата”, “оренда”, “податки”). - Використовують ключові слова, шаблони або машинне навчання для класифікації. - Полегшують побудову аналітики за категоріями витрат/доходів. - Зменшують потребу в ручному втручанні користувача. - Можуть бути адаптовані під конкретні бізнес-процеси.
	Параметри безпеки даних	- Визначають політику збереження та захисту фінансових даних від несанкціонованого доступу. - Включають багаторівневу автентифікацію, шифрування, контроль сесій. - Передбачають резервне копіювання і план відновлення у разі втрати даних. - Визначають, які дії користувача фіксуються у логах для аудиту. - Є частиною ІТ-комплаєнсу та кібербезпеки компанії.
Механізм	Система звітності	- Програмна підсистема або модуль, який генерує фінансову звітність на основі оброблених даних. - Містить шаблони звітів, розрахункові формули, графічні інтерфейси. - Підтримує збереження, друк, експорт або автоматичне надсилання звітів. - Дозволяє гнучко формувати звіти за періодами, підрозділами, проектами. - Інтегрується з іншими підсистемами, такими як CRM або ERP.
	Алгоритми аналізу фінансів	- Використовують фінансові показники для побудови аналітичних моделей: прибутковість, рентабельність, ліквідність. - Працюють з історичними даними для виявлення трендів та закономірностей. - Можуть включати прогнозування (predictive analytics) і моделі оптимізації бюджету. - Реалізовані як аналітичні панелі або блоки в системі. - Підтримують прийняття управлінських рішень.

Назва стрілки		Опис
	Інтерфейс введення даних	- Графічний або текстовий інтерфейс, через який користувач вводить первинні дані (транзакції, параметри). - Забезпечує валідацію, підказки, автозаповнення для зручності користувача. - Може мати форму веб-додатку, мобільного застосунку або десктопного клієнта. - Взаємодіє з базою даних та іншими модулями в режимі реального часу. - Підтримує масове або поодинокі введення інформації.
Вихід	Звіти про фінансовий стан	- Формуються як результат обробки вхідних фінансових даних і користувацьких запитів. - Містять ключову інформацію: баланси, прибутки, витрати, прогнози. - Призначені для управлінських, аналітичних або податкових цілей. - Можуть експортуватися в різних форматах: PDF, Excel, графіки, діаграми. - Підлягають регулярному створенню (місячні, квартальні звіти) або за запитом.

Додаток В

Вміст файлу MAIN.HTML

```
<!DOCTYPE html>
<html lang="uk">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CoinControl - Фінансовий контроль</title>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.9.1/chart.min.js"></scri
pt>
  <link rel="stylesheet" href="styles.css">
</head>

<body>
  <!-- Welcome Screen -->
  <div id="welcomeScreen" class="welcome-screen">
    <h1>Ласкаво просимо до CoinControl</h1>
    <p>Інтуїтивний інтерфейс, що допоможе вам легко відстежувати фінанси,
досягати цілей і отримувати поради від
штучного інтелекту.</p>
    <button class="btn" onclick="showAuth('login')">Розпочати</button>
  </div>

  <!-- Login Form -->
  <div id="loginScreen" class="auth-container">
    <div class="auth-card">
      <div class="auth-illustration"></div>
      <h2>З Поверненням!</h2>
      <form id="loginForm">
        <div class="form-group">
          <label>Логін</label>
          <input type="text" id="loginUsername" placeholder="Ваш логін"
required>
        </div>
        <div class="form-group">
          <label>Пароль</label>
          <input type="password" id="loginPassword" placeholder="Пароль"
required>
        </div>
        <button type="submit" class="btn" style="width: 100%;">Увійти</button>
      </form>
      <div class="auth-link">
        Немає акаунта? <a href="#"
onclick="showAuth('register')">Зареєструватися</a>
      </div>
    </div>
  </div>

  <!-- Register Form -->
  <div id="registerScreen" class="auth-container">
    <div class="auth-card">
      <div class="auth-illustration"></div>
      <h2>Реєстрація</h2>
```

```

    <form id="registerForm">
      <div class="form-group">
        <label>Ім'я</label>
        <input type="text" id="registerName" placeholder="Ваше Ім'я"
required>
      </div>
      <div class="form-group">
        <label>Email</label>
        <input type="email" id="registerEmail" placeholder="Ваш Email"
required>
      </div>
      <div class="form-group">
        <label>Пароль</label>
        <input type="password" id="registerPassword" placeholder="Пароль"
required>
      </div>
      <button type="submit" class="btn" style="width:
100%;">Зареєструватися</button>
    </form>
    <div class="auth-link">
      Вже є акаунт? <a href="#" onclick="showAuth('login')">Увійти</a>
    </div>
  </div>
</div>

<!-- Dashboard -->
<div id="dashboard" class="dashboard">
  <div class="header">
    <div class="container">
      <div class="nav">
        <div class="logo">CoinControl</div>
        <div class="nav-links">
          <a href="#" class="nav-link active" data-page="home">Головна</a>
          <a href="#" class="nav-link" data-page="charts">Графіки</a>
          <a href="#" class="nav-link" data-page="expenses">Додати
витрати</a>
          <a href="#" class="nav-link" data-
page="recommendations">Рекомендації</a>
          <a href="#" class="nav-link" data-
page="achievements">Досягнення</a>
        </div>
      <div class="user-info">
        <div class="user-avatar" id="userAvatar">I</div>
        <span id="userName">Ім'я</span>
      </div>
    </div>
  </div>
</div>

  <div class="container">
    <!-- Home Page -->
    <div id="homePage" class="page-section active">
      <div class="main-content">
        <div class="content-card">
          <div class="welcome-message">
            <h2>Вітаю, <span id="welcomeName">Ім'я</span>! Готові навести лад
у своїх фінансах?</h2>

```

```

    <p>Ось короткий огляд на сьогодні:</p>
  </div>

  <div class="balance-cards">
    <div class="balance-card">
      <h3>Загальний баланс:</h3>
      <div class="amount">12,500 грн</div>
    </div>
    <div class="balance-card">
      <h3>Доходи цього місяця:</h3>
      <div class="amount">8,200 грн</div>
    </div>
    <div class="balance-card">
      <h3>Витрати:</h3>
      <div class="amount">5,800 грн</div>
    </div>
    <div class="balance-card">
      <h3>Бюджет на травень:</h3>
      <div class="amount">15,000 грн</div>
    </div>
  </div>

  <div style="margin-top: 30px;">
    <h3>Застосуй аналітику - керуй фінансами з розумом</h3>
    <div style="margin-top: 15px;">
      <button class="btn" style="margin-right: 10px;">[+ Додати нову
транзакцію]</button>
      <button class="btn" style="background: #10b981;">[Налаштувати
бюджет]</button>
      <button class="btn" style="background: #f59e0b;">[Показати
деталі]</button>
    </div>
  </div>
</div>
</div>
</div>
</div>

<!-- Charts Page -->
<div id="chartsPage" class="page-section">
  <div class="charts-grid">
    <div class="chart-container">
      <h3>Звідна діаграма витрат</h3>
      <canvas id="expensesChart"></canvas>
      <p style="margin-top: 15px; color: #666;">Найбільше на Продукти – 1
260 грн (38%)</p>
    </div>

    <div class="chart-container">
      <h3>Динаміка витрат</h3>
      <canvas id="dynamicsChart"></canvas>
      <p style="margin-top: 15px; color: #666;">Чи є дні з найбільшими
витратами?</p>
    </div>

    <div class="chart-container">
      <h3>Доходи vs Витрати</h3>
      <canvas id="incomeVsExpenses"></canvas>
    </div>
  </div>
</div>

```

```

        <p style="margin-top: 15px; color: #666;">У травні ти витратив
менше, ніж заробив – чудово!</p>
        <p style="color: #666;">У квітні витрати були на 15% вищі за
доходи</p>
    </div>

    <div class="chart-container">
        <h3>Прогрес фінансової мети</h3>
        <canvas id="progressChart"></canvas>
        <p style="margin-top: 15px; color: #666;">Ти вже накопичив 6 500
грн із 10 000 грн (65%)</p>
    </div>
</div>
</div>

<!-- Add Expenses Page -->
<div id="expensesPage" class="page-section">
    <div class="expense-form">
        <h3>Додати нову витрату</h3>
        <form id="expenseForm">
            <div class="form-row">
                <div class="form-group">
                    <label>Сума</label>
                    <input type="number" id="expenseAmount" placeholder="0.00"
required>
                </div>
                <div class="form-group">
                    <label>Категорія</label>
                    <select id="expenseCategory" required>
                        <option value="">Оберить категорію</option>
                        <option value="food">Продукти</option>
                        <option value="transport">Транспорт</option>
                        <option value="entertainment">Розваги</option>
                        <option value="utilities">Комунальні</option>
                        <option value="other">Інше</option>
                    </select>
                </div>
            </div>
            <div class="form-group">
                <label>Опис</label>
                <input type="text" id="expenseDescription" placeholder="Опис
витрати">
            </div>
            <button type="submit" class="btn">Додати витрату</button>
        </form>
    </div>

    <div class="content-card">
        <h3>Останні витрати</h3>
        <div id="expensesList">
            <div class="achievement-item">
                <div class="achievement-icon">🛒</div>
                <div>
                    <strong>Продукти - 450 грн</strong>
                    <p style="color: #666; font-size: 0.9em;">Сьогодні, 14:30</p>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="achievement-item">
            <div class="achievement-icon"><img alt="Car icon" data-bbox="568 68 588 82"/></div>
            <div>
                <strong>Паливо - 800 грн</strong>
                <p style="color: #666; font-size: 0.9em;">Вчора, 18:45</p>
            </div>
        </div>
    </div>
</div>
<!-- Recommendations Page -->
<div id="recommendationsPage" class="page-section">
    <div class="recommendations">
        <div class="recommendation-card">
            <h4>Оптимізуй категорію "Розваги"</h4>
            <p>Цього місяця ти витратив на розваги 65% бюджету – можливо, варто зменшити цю частку до 60%.</p>
        </div>
        <div class="recommendation-card">
            <h4>Створи фонд непередбачених витрат</h4>
            <p>Спробуй щомісяця відкладати хоча б 5% доходу. Це допоможе уникнути стресу у форс-мажорах.</p>
        </div>
        <div class="recommendation-card">
            <h4>Встанови бюджет на тиждень</h4>
            <p>Краще контролювати витрати, розбивши місячний ліміт на 4 частини.</p>
        </div>
        <div class="recommendation-card">
            <h4>Купуй за списком</h4>
            <p>Плануй покупки наперед. Імпульсивні витрати можуть "з'їдати" до 20% твого бюджету.</p>
        </div>
        <div class="recommendation-card">
            <h4>Переглянь підписки</h4>
            <p>Чи користуєшся ти всіма сервісами, за які платиш щомісяця?</p>
        </div>
        <div class="recommendation-card">
            <h4>[Генерувати нові рекомендації]</h4>
            <p>[Показати ще поради]</p>
        </div>
    </div>
</div>
<!-- Achievements Page -->
<div id="achievementsPage" class="page-section">
    <div class="achievements">
        <h3>Сторінка: Досягнення</h3>
        <p><strong>Твої фінансові успіхи</strong></p>
        <p>Кожен крок до фінансової свободи важливий – ось, що ти вже зробив:</p>
        <div style="margin: 30px 0;">
            <h4>Досягнуті цілі</h4>
            <div class="achievement-item">

```

```

        <div class="achievement-icon">✓</div>
        <div>
            <strong>Накопичив 2 000 грн на "резервний фонд"</strong>
            <p style="color: #666;">Не перевищив бюджет у квітні</p>
            <p style="color: #666;">Вів облік витрат 30 днів поспіль</p>
        </div>
    </div>
</div>
<div style="margin: 30px 0;">
    <h4>Прогрес у травні</h4>
    <p><strong>Заощаджено: 1 200 грн</strong></p>
    <p><strong>Точність бюджету: 92%</strong></p>
    <p><strong>Період активності: 26 днів</strong></p>
</div>
<div style="margin: 30px 0;">
    <h4>Відкриті бейджі</h4>
    <p><strong>Бюджетний ніндзя</strong> – не перевищував жодної
категорії 3 тижні поспіль</p>
    <p><strong>Майстер планування</strong> – створив 5+ цілей</p>
    <p><strong>Сталий користувач</strong> – користується додатком 60
днів</p>
</div>

    <div style="text-align: center; margin-top: 40px;">
        <p><strong>Продовжуй у тому ж дусі – твоя фінансова грамотність
росте щодня!</strong></p>
        <div style="margin-top: 20px;">
            <button class="btn">[Поділитись досягненнями]</button>
            <button class="btn" style="background: #10b981; margin-left:
10px;">[Створити нову ціль]</button>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

    <script src="index.js"></script>
</body>
</html>

```

Додаток Г

Вміст файлу STYLES.CSS

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Segoe UI', sans-serif;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  min-height: 100vh;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}

/* Welcome Screen */
.welcome-screen {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  text-align: center;
  color: white;
}

.welcome-screen h1 {
  font-size: 2.5em;
  margin-bottom: 20px;
  font-weight: 300;
}

.welcome-screen p {
  font-size: 1.2em;
  margin-bottom: 40px;
  max-width: 500px;
  line-height: 1.6;
}

.btn {
  background: #6366f1;
  color: white;
  border: none;
  padding: 15px 30px;
  border-radius: 25px;
  font-size: 1.1em;
  cursor: pointer;
  transition: all 0.3s ease;
```

```

    box-shadow: 0 4px 15px rgba(99, 102, 241, 0.4);
}

.btn:hover {
    background: #5855eb;
    transform: translateY(-2px);
    box-shadow: 0 6px 20px rgba(99, 102, 241, 0.6);
}

/* Auth Forms */
.auth-container {
    display: none;
    min-height: 100vh;
    align-items: center;
    justify-content: center;
}

.auth-card {
    background: rgba(255, 255, 255, 0.95);
    backdrop-filter: blur(10px);
    border-radius: 20px;
    padding: 40px;
    box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);
    width: 400px;
    margin: 0 20px;
}

.auth-illustration {
    width: 300px;
    height: 200px;
    background: linear-gradient(45deg, #667eea, #764ba2);
    border-radius: 15px;
    margin-bottom: 30px;
    position: relative;
    overflow: hidden;
}

.auth-illustration::before {
    content: '';
    position: absolute;
    top: 20px;
    left: 20px;
    width: 60px;
    height: 60px;
    background: rgba(255, 255, 255, 0.2);
    border-radius: 50%;
}

.auth-illustration::after {
    content: '';
    position: absolute;
    bottom: 20px;
    right: 20px;
    width: 80px;
    height: 40px;
    background: rgba(255, 255, 255, 0.15);
    border-radius: 20px;
}

```



```

}

.auth-card h2 {
  color: #333;
  margin-bottom: 30px;
  font-weight: 300;
}

.form-group {
  margin-bottom: 20px;
}

.form-group label {
  display: block;
  margin-bottom: 8px;
  color: #666;
  font-weight: 500;
}

.form-group input,
.form-group select {
  width: 100%;
  padding: 12px 15px;
  border: 2px solid #e1e5e9;
  border-radius: 10px;
  font-size: 1em;
  transition: border-color 0.3s ease;
}

.form-group input:focus,
.form-group select:focus {
  outline: none;
  border-color: #6366f1;
}

.auth-link {
  text-align: center;
  margin-top: 20px;
  color: #666;
}

.auth-link a {
  color: #6366f1;
  text-decoration: none;
  font-weight: 500;
}

/* Dashboard */
.dashboard {
  display: none;
  background: #f8fafc;
  min-height: 100vh;
}

.header {
  background: white;
  padding: 20px 0;
}

```

```

    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
    margin-bottom: 30px;
}

.nav {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.logo {
    font-size: 1.5em;
    font-weight: bold;
    color: #6366f1;
}

.nav-links {
    display: flex;
    gap: 30px;
}

.nav-links a {
    text-decoration: none;
    color: #666;
    font-weight: 500;
    transition: color 0.3s ease;
}

.nav-links a:hover,
.nav-links a.active {
    color: #6366f1;
}

.user-info {
    display: flex;
    align-items: center;
    gap: 10px;
}

.user-avatar {
    width: 40px;
    height: 40px;
    background: #6366f1;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-weight: bold;
}

.main-content {
    display: grid;
    grid-template-columns: 2fr 1fr;
    gap: 30px;
    margin-bottom: 30px;
}

```

```

.content-card {
  background: white;
  border-radius: 15px;
  padding: 30px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.05);
}

.welcome-message h2 {
  color: #333;
  margin-bottom: 15px;
  font-weight: 400;
}

.welcome-message p {
  color: #666;
  line-height: 1.6;
  margin-bottom: 20px;
}

.balance-cards {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 15px;
  margin-bottom: 30px;
}

.balance-card {
  background: linear-gradient(135deg, #667eea, #764ba2);
  color: white;
  padding: 20px;
  border-radius: 15px;
  text-align: center;
}

.balance-card h3 {
  font-size: 0.9em;
  opacity: 0.9;
  margin-bottom: 10px;
}

.balance-card .amount {
  font-size: 1.5em;
  font-weight: bold;
}

.charts-grid {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 30px;
  margin-bottom: 30px;
}

.chart-container {
  background: white;
  border-radius: 15px;
  padding: 20px;
}

```

```

    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.05);
}

.chart-container h3 {
    color: #333;
    margin-bottom: 20px;
    font-weight: 500;
}

.recommendations {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
}

.recommendation-card {
    background: white;
    border-radius: 15px;
    padding: 25px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.05);
    border-left: 4px solid #6366f1;
}

.recommendation-card h4 {
    color: #333;
    margin-bottom: 15px;
    font-weight: 500;
}

.recommendation-card p {
    color: #666;
    line-height: 1.6;
}

.achievements {
    background: white;
    border-radius: 15px;
    padding: 25px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.05);
    margin-top: 30px;
}

.achievements h3 {
    color: #333;
    margin-bottom: 20px;
    font-weight: 500;
}

.achievement-item {
    display: flex;
    align-items: center;
    gap: 15px;
    margin-bottom: 15px;
    padding: 15px;
    background: #f8fafc;
    border-radius: 10px;
}

```

```

.achievement-icon {
  width: 40px;
  height: 40px;
  background: #10b981;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  color: white;
  font-weight: bold;
}

.hidden {
  display: none !important;
}

.page-section {
  display: none;
}

.page-section.active {
  display: block;
}

.expense-form {
  background: white;
  border-radius: 15px;
  padding: 25px;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.05);
  margin-bottom: 20px;
}

.form-row {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 15px;
  margin-bottom: 15px;
}

@media (max-width: 768px) {
  .auth-container {
    padding: 20px;
  }

  .auth-card {
    width: 100%;
    max-width: 400px;
  }

  .main-content {
    grid-template-columns: 1fr;
  }

  .charts-grid {
    grid-template-columns: 1fr;
  }
}

```

```
.nav-links {  
  display: none;  
}  
}
```

Додаток Д
Вміст файлу INDEX.JS

```
// Current user data
let currentUser = null;
let expenses = [
  { id: 1, amount: 450, category: 'food', description: 'Продукти', date: new
Date() },
  { id: 2, amount: 800, category: 'transport', description: 'Паливо', date: new
Date(Date.now() - 86400000) }
];

// Navigation
function showAuth(type) {
  document.getElementById('welcomeScreen').style.display = 'none';
  document.getElementById('loginScreen').style.display = type === 'login' ?
'flex' : 'none';
  document.getElementById('registerScreen').style.display = type === 'register'
? 'flex' : 'none';
}

function showDashboard(user) {
  document.getElementById('welcomeScreen').style.display = 'none';
  document.getElementById('loginScreen').style.display = 'none';
  document.getElementById('registerScreen').style.display = 'none';
  document.getElementById('dashboard').style.display = 'block';

  // Update user info
  document.getElementById('userName').textContent = user.name;
  document.getElementById('welcomeName').textContent = user.name;
  document.getElementById('userAvatar').textContent =
user.name.charAt(0).toUpperCase();

  // Initialize charts
  setTimeout(initCharts, 100);
}

// Page navigation
document.addEventListener('DOMContentLoaded', function () {
  const navLinks = document.querySelectorAll('.nav-link');
  const pageSection = document.querySelectorAll('.page-section');

  navLinks.forEach(link => {
    link.addEventListener('click', function (e) {
      e.preventDefault();

      // Remove active class from all links
      navLinks.forEach(l => l.classList.remove('active'));
      // Add active class to clicked link
      this.classList.add('active');

      // Hide all pages
      pageSection.forEach(page => page.classList.remove('active'));
      // Show selected page
      const targetPage = this.getAttribute('data-page') + 'Page';
      document.getElementById(targetPage).classList.add('active');
    });
  });
});
```

```

        // Reinit charts if charts page is selected
        if (this.getAttribute('data-page') === 'charts') {
            setTimeout(initDetailedCharts, 100);
        }
    });
});
});

// Form handlers
document.getElementById('loginForm').addEventListener('submit', function (e) {
    e.preventDefault();
    const username = document.getElementById('loginUsername').value;
    const password = document.getElementById('loginPassword').value;

    // Simple validation
    if (username && password) {
        currentUser = { name: username, email: username + '@example.com' };
        showDashboard(currentUser);
    }
});

document.getElementById('registerForm').addEventListener('submit', function (e)
{
    e.preventDefault();
    const name = document.getElementById('registerName').value;
    const email = document.getElementById('registerEmail').value;
    const password = document.getElementById('registerPassword').value;

    if (name && email && password) {
        alert('Реєстрацію завершено успішно!');
        // Clear form
        this.reset();
        // Stay on registration page
    }
});

document.getElementById('expenseForm').addEventListener('submit', function (e)
{
    e.preventDefault();
    const amount = parseFloat(document.getElementById('expenseAmount').value);
    const category = document.getElementById('expenseCategory').value;
    const description = document.getElementById('expenseDescription').value;

    if (amount && category) {
        const newExpense = {
            id: expenses.length + 1,
            amount: amount,
            category: category,
            description: description || 'Витрата',
            date: new Date()
        };
        expenses.push(newExpense);

        // Reset form
        this.reset();
    }
});

```



```

    // Update expenses list
    updateExpensesList();

    alert('Витрату додано успішно!');
  }
});

function updateExpensesList() {
  const expensesList = document.getElementById('expensesList');
  expensesList.innerHTML = '';

  expenses.slice(-5).reverse().forEach(expense => {
    const categoryIcons = {
      food: '🛒',
      transport: '🚗',
      entertainment: '🎪',
      utilities: '💡',
      other: '💰'
    };

    const expenseItem = document.createElement('div');
    expenseItem.className = 'achievement-item';
    expenseItem.innerHTML = `
      <div class="achievement-icon">${categoryIcons[expense.category]} ||
'💰'</div>
      <div>
        <strong>${expense.description} - ${expense.amount} грн</strong>
        <p style="color: #666; font-size:
0.9em;">${expense.date.toLocaleDateString('uk-UA')}</p>
      </div>
    `;
    expensesList.appendChild(expenseItem);
  });
}

// Chart initialization
function initCharts() {
  // Home page charts
  const ctx1 = document.getElementById('expensesPieChart');
  const ctx2 = document.getElementById('incomeChart');

  if (ctx1) {
    new Chart(ctx1, {
      type: 'pie',
      data: {
        labels: ['Продукти', 'Транспорт', 'Розваги', 'Комунальні', 'Інше'],
        datasets: [{
          data: [38, 25, 20, 12, 5],
          backgroundColor: [
            '#6366f1',
            '#8b5cf6',
            '#06b6d4',
            '#10b981',
            '#f59e0b'
          ]
        }]
      }
    });
  }
}

```

```

    },
    options: {
      responsive: true,
      plugins: {
        legend: {
          position: 'bottom'
        }
      }
    }
  });
}

if (ctx2) {
  new Chart(ctx2, {
    type: 'bar',
    data: {
      labels: ['Березень', 'Квітень', 'Травень'],
      datasets: [{
        label: 'Доходи',
        data: [6000, 5500, 8200],
        backgroundColor: '#10b981'
      }, {
        label: 'Витрати',
        data: [5200, 6300, 5800],
        backgroundColor: '#ef4444'
      }]
    },
    options: {
      responsive: true,
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
}

function initDetailedCharts() {
  // Expenses pie chart
  const ctx1 = document.getElementById('expensesChart');
  if (ctx1) {
    new Chart(ctx1, {
      type: 'doughnut',
      data: {
        labels: ['Продукти', 'Транспорт', 'Розваги', 'Комунальні', 'Інше'],
        datasets: [{
          data: [1260, 850, 680, 420, 190],
          backgroundColor: [
            '#6366f1',
            '#8b5cf6',
            '#06b6d4',
            '#10b981',
            '#f59e0b'
          ],
          borderWidth: 0
        }]
      }
    });
  }
}

```

```

    ]]
  },
  options: {
    responsive: true,
    cutout: '50%',
    plugins: {
      legend: {
        position: 'bottom'
      }
    }
  }
});
}

// Dynamics chart
const ctx2 = document.getElementById('dynamicsChart');
if (ctx2) {
  new Chart(ctx2, {
    type: 'line',
    data: {
      labels: ['1 тиждень', '2 тиждень', '3 тиждень', '4 тиждень'],
      datasets: [{
        label: 'Витрати (грн)',
        data: [1200, 800, 1500, 900],
        borderColor: '#6366f1',
        backgroundColor: 'rgba(99, 102, 241, 0.1)',
        fill: true,
        tension: 0.4
      }]
    },
    options: {
      responsive: true,
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
}

// Income vs Expenses
const ctx3 = document.getElementById('incomeVsExpenses');
if (ctx3) {
  new Chart(ctx3, {
    type: 'bar',
    data: {
      labels: ['Березень', 'Квітень', 'Травень'],
      datasets: [{
        label: 'Доходи',
        data: [6000, 5500, 8200],
        backgroundColor: '#10b981'
      }, {
        label: 'Витрати',
        data: [5200, 6300, 5800],
        backgroundColor: '#ef4444'
      }]
    }
  });
}

```

```

    },
    options: {
      responsive: true,
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
}

// Progress chart
const ctx4 = document.getElementById('progressChart');
if (ctx4) {
  new Chart(ctx4, {
    type: 'doughnut',
    data: {
      labels: ['Накопичено', 'Залишилось'],
      datasets: [{
        data: [6500, 3500],
        backgroundColor: ['#10b981', '#e5e7eb'],
        borderWidth: 0
      }]
    },
    options: {
      responsive: true,
      cutout: '70%',
      plugins: {
        legend: {
          display: false
        }
      }
    }
  });
}

// Initialize expenses list on load
updateExpensesList();

```