

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет/(ННІ) інформаційних технологій

ПОГОДЖЕНО

Декан факультету (Директор ННІ)
інформаційних технологій
(назва факультету (ННІ))

Ігор БОЛБОТ
(ім'я ПРІЗВИЩЕ)

“ ” 2025 р.

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри
комп'ютерних наук
(назва кафедри)

Белла ГОЛУБ
(ім'я ПРІЗВИЩЕ)

“ ” 2025 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Аналітичний ВЕБ-додаток управління особистими фінансами

Спеціальність 121 «Інженерія програмного забезпечення»
(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем
(назва)

Орієнтація освітньої програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

к.ф.-м.н. доцент
(науковий ступінь та вчене звання)

Віктор КИРИЧЕНКО
(підпис)

Віктор КИРИЧЕНКО
(ім'я ПРІЗВИЩЕ)

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент
(науковий ступінь та вчене звання)

Ірина БОРОДКІНА
(підпис)

Ірина БОРОДКІНА
(ім'я ПРІЗВИЩЕ)

Виконав

Микита МОШНІН
(підпис)

Микита МОШНІН
(ім'я ПРІЗВИЩЕ здобувача)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерних наук

к.т.н., доцент

Белла ГОЛУБ

(науковий ступінь, вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

“ 01 ” листопада 2024 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Мошнін Микита Андрійович

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія програмного забезпечення»

(код і найменування)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи Аналітичний ВЕБ-додаток управління особистими фінансами

затверджена наказом від “ 01 ” листопада 2024 р. № 1963 «С»

Термін подання завершеної роботи на кафедру 2024.11.29

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи фінансові транзакції користувачів (доходи, витрати, заощадження), довідкові дані про категорії витрат.

Перелік питань, що підлягають дослідженню:

1. Аналіз існуючих рішень для управління особистими фінансами та визначення їхніх переваг і недоліків.
2. Дослідження методів збору, обробки та зберігання фінансових даних із забезпеченням їхньої конфіденційності та безпеки.
3. Вибір і обґрунтування методів фінансової аналітики та прогнозування для підтримки прийняття рішень користувачами.
4. Розробка принципів побудови інтерфейсу та інструментів візуалізації, що забезпечують зручність і ефективність використання веб-додатком.

Перелік графічного матеріалу (за потреби) _____

Дата видачі завдання “ 01 ” листопада 2024 р.

Керівник магістерської кваліфікаційної роботи _____

(підпис)

Ірина БОРОДКІНА

(ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання _____

Микита МОШНІН

(ім'я ПРІЗВИЩЕ)

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Аналіз сутності процесів управління особистими фінансами.....	9
1.2. Особливості збору, структурування та зберігання фінансових даних користувачів.....	12
1.3. Огляд і класифікація існуючих веб-додатків для управління особистими фінансами.....	15
1.4. Порівняльний аналіз функціональних і аналітичних можливостей сучасних веб-систем.....	19
1.5. Визначення недоліків наявних підходів і формулювання проблеми дослідження.....	23
1.6. Постановка завдання на створення аналітичного веб-додатку управління фінансами.....	26
Висновки до розділу 1.....	29
РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ АНАЛІТИЧНОГО УПРАВЛІННЯ ОСОБИСТИМИ ФІНАНСАМИ.....	31
2.1. Аналіз предметної області та постановка наукової задачі.....	31
2.2. Функціональне моделювання процесів системи.....	33
2.3. Об'єктно-орієнтоване моделювання системи.....	37
2.4. Алгоритми фінансового аналізу та прогнозування.....	43
2.5. Вимоги до інформаційної безпеки та захисту фінансових даних.....	45
Висновки до розділу 2.....	47
РОЗДІЛ 3. РОЗРОБКА АНАЛІТИЧНОГО ВЕБ-ДОДАТКУ.....	49
3.1. Загальна архітектура програмної системи.....	49
3.2. Вибір технологічного стеку та середовища розробки.....	52
3.3. Проектування структури бази даних фінансових транзакцій.....	55
3.4. Реалізація серверної частини: логіка обробки запитів, API, модуль аналітики.....	59
3.5. Реалізація клієнтської частини: інтерфейс користувача, динамічна візуалізація даних.....	62
3.6. Інтеграція аналітичних і прогнозних моделей у систему.....	65
3.7. Забезпечення надійності, безпеки та масштабованості веб-додатку.....	67
Висновки до розділу 3.....	69
РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ.....	71
4.1. Методика експериментальних досліджень.....	71
4.2. Тестування працездатності програмних модулів.....	73
4.3. Перевірка точності прогнозування та якості аналітичних звітів.....	76

4.4. Практичні результати впровадження та приклади використання системи.....	78
4.5. Перспективи розвитку, оптимізації та масштабування розробленого рішення.....	79
Висновки до розділу 4.....	81
ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	86
ДОДАТКИ.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – програмний інтерфейс прикладного програмування;

DB – база даних;

DFD – діаграма потоків даних;

GDPR – загальний регламент захисту персональних даних;

HTTP/HTTPS – протокол передавання гіпертексту;

JSON – формат обміну структурованими даними;

KPI – ключові показники ефективності;

ML – машинне навчання;

REST API – архітектурний стиль взаємодії клієнт–сервер;

SSL/TLS – протоколи захищеного з'єднання;

UML – уніфікована мова моделювання;

UI/UX – інтерфейс користувача / досвід користувача.

ВСТУП

Актуальність теми

У сучасному цифровому суспільстві ефективне управління особистими фінансами є важливою складовою економічної стабільності людини. Зростання кількості електронних транзакцій, поява різних джерел доходів і витрат створюють потребу в інструментах, що дозволяють контролювати фінансові потоки та приймати обґрунтовані рішення.

Більшість існуючих застосунків для обліку фінансів забезпечують лише фіксацію операцій, але не надають глибокого аналізу чи прогнозування. Тому актуальним є створення аналітичного веб-додатку, який поєднує автоматизований облік, інтелектуальну аналітику та зручну візуалізацію даних. Така система допоможе користувачам підвищити фінансову грамотність і ефективніше керувати власним бюджетом.

Об'єкт дослідження

Процес управління особистими фінансами користувачів у межах інформаційних систем.

Предмет дослідження

Моделі, методи та засоби побудови аналітичного веб-додатку для обліку, аналізу, кластеризації та прогнозування фінансових даних користувачів.

Мета дослідження

Розробити аналітичний веб-додаток управління особистими фінансами, який забезпечує повний цикл роботи з фінансовими даними користувача — від збору і візуалізації до аналітики та прогнозування — з урахуванням вимог безпеки, масштабованості та зручності використання.

Завдання дослідження

1. Аналіз існуючих рішень для управління особистими фінансами та визначення їхніх переваг і недоліків.
2. Дослідження методів збору, обробки та зберігання фінансових даних із забезпеченням їхньої конфіденційності та безпеки.

3. Вибір і обґрунтування методів фінансової аналітики та прогнозування для підтримки прийняття рішень користувачами.
4. Розробка принципів побудови інтерфейсу та інструментів візуалізації, що забезпечують зручність і ефективність користування веб-додатком.

Методи дослідження

У роботі використано методи системного аналізу для формалізації процесів управління фінансами, методи функціонального та об'єктно-орієнтованого моделювання (DFD, UML) для побудови структури системи.

У межах аналітичної частини застосовано алгоритми кластеризації (K-Means, DBSCAN) для групування користувачів за фінансовими звичками та моделі прогнозування на основі часових рядів для оцінювання майбутніх витрат.

Програмну реалізацію виконано з використанням технологій React, Node.js, PostgreSQL, REST API і бібліотек Chart.js для інтерактивної візуалізації даних.

Наукова новизна

- вперше запропоновано комплексний підхід до управління особистими фінансами, що об'єднує кластеризацію користувачів і прогнозування фінансових показників у межах веб-додатку;
- удосконалено архітектурну модель аналітичних систем за рахунок інтеграції інтелектуальних алгоритмів обробки даних;
- розроблено методику підвищення точності прогнозів за рахунок попередньої кластеризації фінансових даних;
- реалізовано архітектуру веб-додатку з підвищеним рівнем безпеки та гнучкою масштабованістю.

Апробація результатів

Результати дослідження доповідалися на студентських науково-практичних конференціях факультету інформаційних технологій НУБіП України та опубліковані у збірниках тез доповідей.

Структура магістерської роботи

Робота складається зі вступу, чотирьох розділів, висновків та списку джерел.

У першому розділі здійснено системний аналіз предметної області.

У другому — розроблено функціональні, об'єктно-орієнтовані та математичні моделі системи.

Третій розділ присвячено реалізації аналітичного веб-додатку.

Четвертий розділ містить результати дослідження, тестування та оцінку точності прогнозних моделей. Загальний обсяг роботи становить 98 сторінок, містить 22 рисунків, 2 таблиць і 25 найменувань у списку використаних джерел, а також оформлені додатки А та Б.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз сутності процесів управління особистими фінансами

Управління особистими фінансами є невід’ємною складовою життєдіяльності сучасної людини, оскільки дозволяє забезпечити стабільність, передбачуваність та ефективність використання наявних грошових ресурсів. Під цим поняттям розуміють процес планування, контролю та аналізу доходів і витрат з метою досягнення фінансових цілей — збереження, накопичення або примноження капіталу.[5]

Сутність управління особистими фінансами полягає в раціональному розподілі грошових потоків, що включає:

- формування та ведення бюджету (планування доходів і витрат на певний період);
- контроль виконання бюджету та коригування витратної частини;
- аналіз фінансового стану користувача та оцінку ефективності рішень;
- розробку стратегії накопичення та інвестування;
- мінімізацію фінансових ризиків.

У контексті інформаційних технологій управління особистими фінансами розглядається як процес обробки фінансових даних, що генерує знання, необхідні для прийняття обґрунтованих рішень.[4] Тобто, користувач, маючи систематизовану інформацію про свої доходи, витрати та заощадження, може оптимізувати фінансову поведінку, визначити пріоритети та прогнозувати майбутні потреби. Детальніше розглянуто на рисунку 1.1 умовний процес управління фінансами можна поділити на кілька етапів.

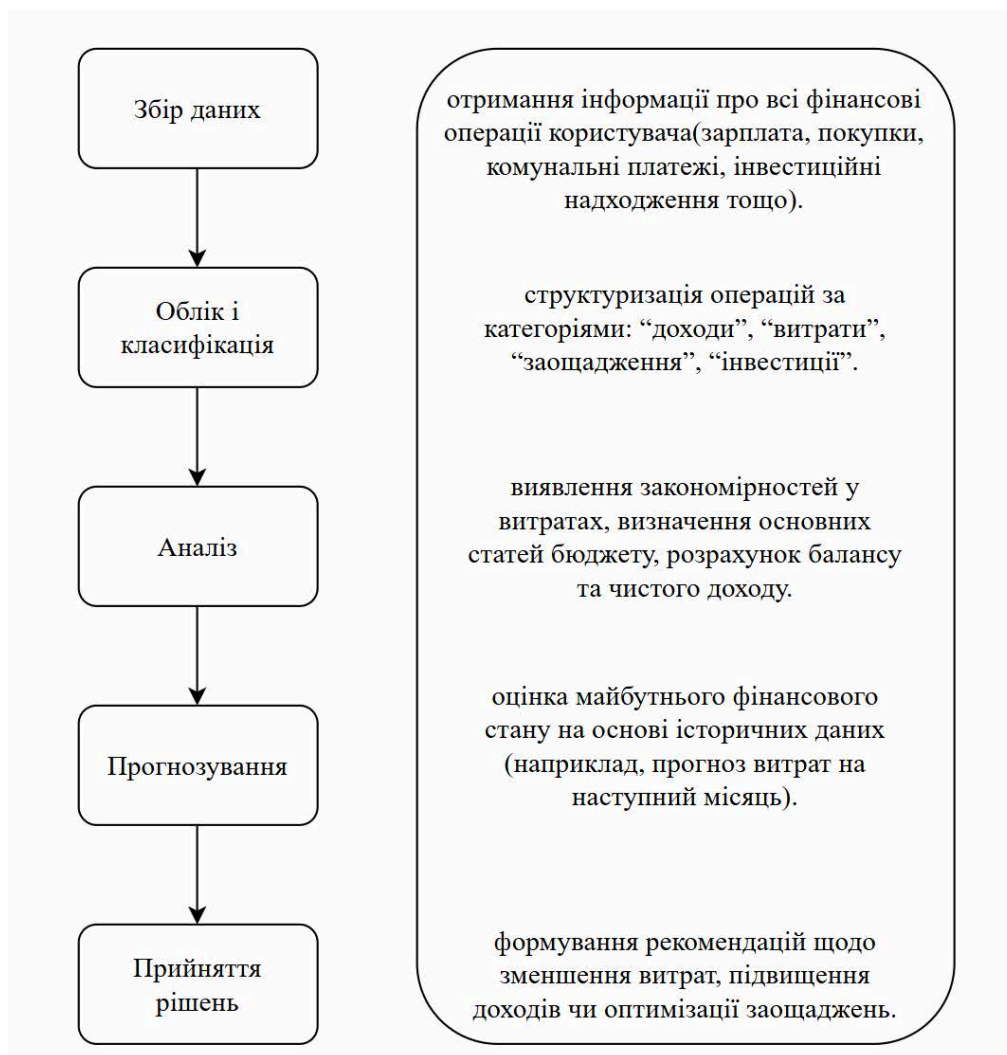


Рис. 1.1 Процес управління особистими фінансами як система обробки фінансових даних

Такий підхід дає змогу не лише фіксувати факт здійснення операцій, але й перетворювати дані у аналітичну інформацію, придатну для прийняття рішень. Саме на цьому базується концепція аналітичних фінансових систем, де ключову роль відіграють алгоритми аналізу даних і візуалізація результатів.

Для детального аналізу необхідно визначити основні компоненти фінансової системи користувача:

- Доходи — регулярні (заробітна плата, стипендія, орендна плата) та нерегулярні (премії, подарунки, продаж майна).
- Витрати — обов’язкові (комунальні послуги, транспорт, харчування) та змінні (розваги, покупки, подорожі).

- Заощадження — кошти, що відкладаються з метою формування резерву або інвестування.
- Активи та пасиви — матеріальні та нематеріальні ресурси користувача, а також зобов'язання перед фінансовими установами.

Співвідношення між цими компонентами визначає фінансову стабільність користувача. Завдання системи управління фінансами — допомогти підтримувати позитивний баланс, аналізуючи динаміку доходів і витрат та попереджаючи фінансові ризики.

Попри очевидну важливість, більшість користувачів не ведуть системного обліку особистих коштів. Основними причинами цього є:

- велика кількість джерел доходів і витрат, що ускладнює ручний контроль;
- відсутність навичок фінансового планування;
- недовіра до онлайн-сервісів, пов'язана з безпекою персональних даних;
- розрізненість інформації — фінансові операції зберігаються у різних банківських застосунках, сервісах чи месенджерах.

Розв'язання цих проблем можливе завдяки створенню інтегрованих аналітичних систем, здатних об'єднати різні джерела даних, виконати їх узагальнення, обробку та подання у зручному форматі.[10]

З точки зору інформатики, управління особистими фінансами можна представити як інформаційно-аналітичну систему, що функціонує за принципом обробки вхідних даних з метою отримання нових знань.

Вхідними даними виступають транзакції користувача (сума, час, тип, категорія), проміжними результатами — агреговані показники (баланс, середні витрати, співвідношення доходів і витрат), а вихідними — аналітичні висновки або прогнозні моделі.

Така система може бути представлена у вигляді трирівневої структури:

- Дані (Data) — первинна інформація про фінансові операції.
- Аналітика (Analytics) — статистична обробка, кластеризація, прогнозування.
- Рішення (Insights) — рекомендації, звіти, прогнозні графіки.

Таким чином, процес управління фінансами переходить від обліку до інтелектуального аналізу даних (Financial Data Analytics), що є ключовим напрямом сучасної фінансової інформатики.

Автоматизація є вирішальним чинником у підвищенні ефективності управління особистими фінансами. Вона забезпечує:

- оперативність — дані збираються та обробляються у реальному часі;
- точність — виключається людський фактор у підрахунках;
- зручність — користувач отримує результати у вигляді зрозумілих графіків, звітів і прогнозів;
- персоналізацію — система адаптується до індивідуальної фінансової поведінки.

У сучасних умовах користувач очікує не просто обліковий інструмент, а розумного фінансового помічника, який здатен не лише фіксувати операції, але й пояснювати, чому виникають ті чи інші тенденції у витратах, як оптимізувати бюджет і що прогнозується на найближчий період.

1.2. Особливості збору, структурування та зберігання фінансових даних користувачів

У сучасних інформаційних системах, орієнтованих на управління особистими фінансами, якість даних є фундаментальним чинником, що визначає ефективність усіх наступних етапів аналітичної обробки. Від достовірності, структурованості та цілісності даних залежить точність фінансових розрахунків, повнота аналітичних звітів та адекватність прогнозів. Тому розробка підсистеми збору, структурування та зберігання даних потребує системного підходу, який враховує як технічні, так і методологічні аспекти.

Процес збору фінансових даних користувача у веб-додатку може відбуватися декількома шляхами. Найпоширенішим є ручне введення операцій, коли користувач самостійно фіксує доходи, витрати, перекази або заощадження. Цей підхід забезпечує максимальну гнучкість, однак вимагає певної дисципліни та витрат часу. З іншого боку, сучасні технології відкривають можливість автоматичного збору інформації через інтеграцію з банківськими системами, що

реалізується на основі відкритих програмних інтерфейсів (API) або протоколів Open Banking.[13] Це дозволяє отримувати дані про транзакції безпосередньо з фінансових установ, мінімізуючи участь користувача. Також поширеним є імпорт даних із зовнішніх джерел — наприклад, з банківських звітів у форматах CSV, XLSX чи JSON.[13] Такий підхід спрощує перенесення фінансової історії з інших сервісів. У деяких системах додатково використовується технологія розпізнавання QR-кодів або чеків, що дозволяє автоматично фіксувати покупки, здійснені офлайн. У межах сучасного веб-додатку доцільним є гібридний підхід до збору даних, який поєднує ручне введення, автоматичний імпорт та обробку зовнішніх файлів. Це забезпечує повноту інформації, підвищує достовірність записів і дозволяє сформувати цілісну фінансову картину користувача.

Зібрані дані мають бути структуровані відповідно до логіки фінансової діяльності користувача. Кожна транзакція описується набором атрибутів, серед яких дата, тип операції (дохід чи витрата), категорія, сума, валюта, спосіб оплати та коментар. Крім того, важливим є зв'язок операцій із користувачем системи, що реалізується через унікальні ідентифікатори. Для зручності аналітичної обробки транзакції доповнюються допоміжною інформацією — тегами, підкатегоріями, періодами та джерелами доходів. Відповідно, інформаційна модель фінансових даних повинна бути побудована у вигляді реляційної структури, що включає основну таблицю транзакцій, довідники категорій і користувачів, а також таблиці рахунків та бюджетів. Такий підхід дозволяє забезпечити узгодженість, цілісність і можливість ефективного виконання запитів.

Для підвищення ефективності зберігання фінансової інформації застосовуються принципи нормалізації бази даних. На першому етапі забезпечується атомарність даних, тобто кожне поле містить лише одне значення. Далі усуваються надлишкові залежності між полями, що мінімізує дублювання інформації. У результаті формується логічно узгоджена структура, яка зберігає цілісність під час оновлення записів. Водночас, при побудові аналітичних модулів доцільно застосовувати часткову денормалізацію —

створення агрегованих таблиць або матеріалізованих представлень, у яких зберігаються підсумкові показники за періодами чи категоріями. Це дозволяє значно прискорити виконання аналітичних запитів і побудову графіків, не навантажуючи основну транзакційну базу.

Важливим аспектом розробки є забезпечення безпеки та конфіденційності фінансових даних. Оскільки система оперує персональною інформацією користувачів, необхідно дотримуватися положень Закону України «Про захист персональних даних» та міжнародних стандартів інформаційної безпеки (GDPR, ISO/IEC 27001). [6] Усі облікові записи повинні зберігатися у зашифрованому вигляді із використанням сучасних криптографічних алгоритмів. Передача інформації між клієнтом і сервером має здійснюватися виключно через захищений протокол HTTPS. Важливим компонентом є впровадження багаторівневої автентифікації, що може включати перевірку пароля, одноразовий код із мобільного пристрою або біометричні дані. Додатково необхідно реалізувати журнал дій користувачів і систему виявлення підозрілої активності. Для підвищення надійності даних передбачаються механізми резервного копіювання та автоматичного відновлення після збоїв.[18]

З точки зору архітектури зберігання інформації, у веб-додатках для управління фінансами застосовуються централізовані або гібридні підходи. У централізованій моделі всі дані зберігаються на сервері, що забезпечує високу цілісність і централізоване адміністрування. У гібридних рішеннях частина даних кешується локально у браузері користувача для підвищення швидкодії, тоді як основна база розташовується у хмарному сховищі. Такий підхід забезпечує баланс між продуктивністю та безпекою, оскільки аналітичні обчислення можуть виконуватися на стороні сервера або у хмарі з використанням інструментів розподіленої обробки даних.

Не менш важливою є підтримка якості та цілісності фінансової інформації. Для цього у системі реалізуються механізми автоматичної валідації під час введення даних, перевірки формату сум, дат і валют, а також уникнення дублікатів при імпорті з зовнішніх джерел. Всі зміни у базі даних мають бути

відстежувані через систему журналювання, що дозволяє відновити історію редагувань і виявити помилки. З метою підвищення аналітичної достовірності дані повинні періодично проходити аудит, який включає статистичну перевірку коректності записів та логічну узгодженість між категоріями доходів і витрат.

Отже, особливості збору, структурування та зберігання фінансових даних у системі управління особистими фінансами визначаються необхідністю досягнення балансу між гнучкістю, точністю та безпекою. Правильна організація інформаційної моделі, дотримання принципів нормалізації, впровадження сучасних засобів захисту та централізовано-хмарної архітектури забезпечують надійну основу для реалізації аналітичних і прогнозних функцій веб-додатку. У результаті створюється інформаційне середовище, здатне забезпечити користувача актуальними знаннями про фінансовий стан, підтримати прийняття обґрунтованих рішень і сприяти підвищенню ефективності управління особистими коштами.

1.3. Огляд і класифікація існуючих веб-додатків для управління особистими фінансами

Розвиток веб-технологій значно вплинув на еволюцію систем управління особистими фінансами. Якщо раніше подібні сервіси існували переважно у вигляді десктопних програм, то сьогодні більшість сучасних рішень реалізовано у форматі веб-додатків. Це зумовлено поширенням хмарних технологій, удосконаленням засобів веб-аналітики та підвищенням вимог користувачів до мобільності й доступності фінансових даних у будь-який час і з будь-якого пристрою.

Веб-додатки дозволяють користувачам працювати з особистими фінансами без встановлення спеціального програмного забезпечення. Усі операції здійснюються через браузер, а дані зберігаються у віддалених базах, що гарантує централізований облік і синхронізацію між пристроями. Такий підхід не лише спрощує використання системи, а й забезпечує її масштабованість, зручність адміністрування та інтеграцію з іншими фінансовими сервісами.

Серед веб-додатків для управління особистими фінансами можна виділити кілька основних груп, що різняться за функціональним призначенням і рівнем аналітичної складності. На рисунку 1.2 наведено узагальнену класифікацію таких веб-додатків, яка відображає еволюцію від базових систем обліку до сучасних аналітичних платформ з елементами штучного інтелекту.



Рис. 1.2 Класифікація веб-додатків для управління особистими фінансами

1. Базові системи обліку[23] — зосереджені на фіксації доходів та витрат користувача. Вони забезпечують мінімальний набір функцій — створення категорій витрат, ведення щоденника транзакцій, побудову простих діаграм. До таких рішень належать, наприклад, Money Lover Web, Spendeer Web або Monefy Web Version. Їх основною перевагою є простота використання, однак аналітична частина обмежена.
2. Системи бюджетування та планування — орієнтовані на середньострокове управління фінансами. Вони дозволяють встановлювати фінансові цілі, ліміти витрат і контролювати їх виконання. Відомими представниками є

YNAB (You Need A Budget) та GoodBudget. Ці сервіси реалізують концепцію «прогнозного бюджету», однак їхня аналітика здебільшого базується на статистичних розрахунках без застосування інтелектуальних алгоритмів.

3. Інтегровані веб-платформи з аналітичними модулями — надають розширені можливості аналізу та прогнозування фінансових показників. Прикладами таких систем є Mint, PocketSmith і Wally.[8] Вони поєднують облік, візуалізацію, синхронізацію з банками через API та елементи аналітики, зокрема побудову трендів і діаграм динаміки витрат. Проте в більшості випадків ці сервіси орієнтовані на закордонні ринки та не підтримують інтеграцію з українськими банками.
4. Аналітичні веб-додатки нового покоління[16]— це платформи, що використовують інструменти бізнес-аналітики, машинного навчання або інтелектуального прогнозування. До цієї групи можна віднести Toshl Finance, Emma та Cleo. Вони аналізують поведінкові патерни користувачів, надають персоналізовані рекомендації й навіть використовують чат-ботів для спілкування з користувачем. Попри високий рівень автоматизації, такі системи залишаються закритими і комерційними, що обмежує їх гнучкість для подальших досліджень або локалізації.

Більшість веб-додатків для управління фінансами мають подібну структуру, яка включає три ключові рівні: модуль збору та зберігання даних (інтеграція з банками, ручне введення, імпорт файлів), аналітичний модуль (розрахунок балансу, класифікація транзакцій, формування звітів), інтерфейс користувача (візуалізація фінансових потоків, побудова графіків і таблиць).

При цьому ступінь автоматизації та глибина аналітики суттєво варіюються. У більшості випадків використовується лише базова статистика — середні витрати, загальний баланс, співвідношення категорій. Системи, які реалізують методи прогнозування, залишаються поодинокими та часто потребують платної підписки.

Особливістю сучасних веб-додатків є впровадження хмарного зберігання даних. Це дозволяє забезпечити доступ до фінансової інформації з будь-якої точки світу, спрощує оновлення програмного забезпечення та розширення функціоналу. Проте така модель потребує високого рівня інформаційної безпеки — зокрема, шифрування персональних даних, токенизації та застосування протоколів SSL/TLS.

Зарубіжні системи, такі як Mint або YNAB, характеризуються високим рівнем автоматизації, розвиненим інтерфейсом та наявністю інструментів прогнозування. Вони здатні автоматично класифікувати транзакції, виявляти аномальні витрати та надавати рекомендації щодо оптимізації бюджету. Однак більшість таких рішень мають обмеження щодо локалізації — вони підтримують лише банки США, Канади або Великої Британії, що робить їх малопридатними для українських користувачів.

Вітчизняні аналоги, серед яких можна виділити Finmap, Costless або Rics Finance, мають простішу структуру. Вони орієнтовані на ведення обліку доходів і витрат, але переважно не включають аналітичного модуля або засобів прогнозування. Їхня перевага полягає у локалізованому інтерфейсі та підтримці гривні, однак відсутність інтелектуальної обробки даних обмежує їхню аналітичну цінність.

Таким чином, жоден із розглянутих веб-додатків не забезпечує повноцінного аналітичного підходу, що поєднує зручність використання, автоматичну обробку транзакцій, персоналізовану аналітику та прогнозування фінансової поведінки користувача.

Огляд і класифікація сучасних веб-додатків для управління особистими фінансами свідчать про високий рівень розвитку ринку, однак також виявляють низку обмежень, які залишаються нерозв'язаними. Переважна більшість існуючих рішень реалізує лише функції обліку та базової статистики. Інтелектуальні інструменти, які б дозволяли автоматично аналізувати фінансову поведінку користувача, прогнозувати витрати та формувати рекомендації, застосовуються рідко.

Крім того, більшість платформ не адаптовані до локальних фінансових систем і нормативних вимог, що знижує їхню ефективність на українському ринку. Отже, актуальним завданням є створення вітчизняного аналітичного веб-додатку, який поєднує гнучку архітектуру, сучасні засоби візуалізації, розвинений аналітичний модуль та алгоритми прогнозування, забезпечуючи при цьому високий рівень захисту персональних даних користувачів.

1.4. Порівняльний аналіз функціональних і аналітичних можливостей сучасних веб-систем

Після узагальнення сучасних підходів до управління особистими фінансами постає необхідність провести детальний порівняльний аналіз веб-додатків, які реалізують відповідний функціонал. Такий аналіз дозволяє виявити сильні та слабкі сторони існуючих рішень, визначити напрями їх розвитку та сформулювати вимоги до проектування власного аналітичного веб-додатку. Актуальність цього аналізу зумовлена тим, що ринок подібних програмних продуктів постійно розширюється, а користувачі очікують не лише базового обліку витрат, а й глибокої аналітики, прогнозування та персоналізованих рекомендацій. Тому порівняння різних систем дає змогу зрозуміти, які саме функції є найбільш затребуваними та які технологічні підходи варто використати під час розробки власного рішення.

Критеріями оцінювання виступають: рівень автоматизації збору даних, глибина аналітики, наявність модулів прогнозування, якість візуалізації, можливості персоналізації, рівень інформаційної безпеки та підтримка локалізації. Ці критерії безпосередньо впливають на зручність користування системою, точність аналітичних результатів і відповідність сучасним стандартам розробки веб-рішень. Окрім того, вони відображають основні очікування користувачів — швидкість роботи, інтуїтивність інтерфейсу та надійність зберігання особистих фінансових даних.

До аналізу було обрано низку найпопулярніших веб-додатків: Mint, YNAB (You Need A Budget), PocketSmith, Spendee Web, Finmap та Wally. Вони відрізняються за архітектурою, аналітичними можливостями, цільовою

аудиторією та ринком застосування. Такий підхід дозволяє охопити як міжнародні, так і локальні рішення, що дає повніше уявлення про сучасні тенденції у сфері фінансових технологій.

Система Mint є однією з найпоширеніших платформ у США, [8] яка поєднує бюджетування, автоматичний збір транзакцій із банківських рахунків та елементи фінансової аналітики. Проте вона орієнтована на англomовний ринок і не має підтримки українських фінансових інституцій, що обмежує її використання на локальному рівні.

YNAB вирізняється високим рівнем інтерактивності й навчальним підходом до управління фінансами.[10] Вона реалізує методологію планування бюджету за принципом «кожна гривня має своє призначення», проте система не є безкоштовною й не використовує розвинених алгоритмів прогнозування. Її перевага полягає у мотиваційному компоненті та гнучкому налаштуванні бюджету користувача.

PocketSmith пропонує розширений набір інструментів аналітики, [8] включаючи календарні прогнози витрат, візуальні графіки динаміки доходів і витрат, а також сценарне моделювання фінансових стратегій. Його недоліком є складність інтерфейсу для початківців і платна підписка, однак система вирізняється найглибшою аналітикою серед аналогів.

Spendee Web і Wally є типовими представниками веб-застосунків із простим інтерфейсом, орієнтованих на щоденне ведення обліку. Вони дозволяють вручну вводити операції, але не забезпечують розвиненої аналітики чи прогнозування. Перевагою цих сервісів є доступність і простота використання, що робить їх популярними серед широкого кола користувачів.

Finmap — це український веб-додаток,[23] який спочатку був розроблений для малого бізнесу, але може бути адаптований для персональних потреб. Його перевага полягає у локалізації, підтримці гривні та простоті використання, однак аналітична складова залишається обмеженою. Незважаючи на це, Finmap демонструє перспективність розвитку в напрямі поєднання бухгалтерського та аналітичного обліку у веб-середовищі.

Для систематизації результатів аналізу проведено порівняння найпоширеніших веб-додатків управління особистими фінансами за основними критеріями: рівень автоматизації збору даних, аналітичні можливості, підтримка прогнозування, локалізація та зручність інтерфейсу. Отримані результати подано в таблиці 1.1, що дозволяє наочно оцінити переваги й недоліки кожного рішення та визначити їх відповідність сучасним вимогам до аналітичних фінансових систем.

Таблиця 1.1

Порівняльна характеристика сучасних веб-додатків управління особистими фінансами

Система	Автоматизація	Аналітика	Прогноз	Локалізація	Інтерфейс
Mint	Імпорт з банків	Розширена статистика	Є (баз.)	Немає	Сучасний
YNAB	Через API	Бюджетування, звіти	Обмежене	Немає	Зручний
PocketSmith	Автоматична	Потужна аналітика	Є (розш.)	Немає	Професійний
Spendee	Ручне введення	Базові діаграми	Немає	Частково	Простий
Finmap	Напівавтоматичний	Звіти, потоки	Немає	Є	Зручний
Wally	Ручне / OCR	Мінімальна аналітика	Немає	Немає	Адаптивний

Як видно з таблиці, більшість систем забезпечують базовий рівень автоматизації та мають зручний інтерфейс, проте не реалізують повноцінних механізмів прогнозування чи інтелектуальної аналітики. Винятком є PocketSmith і частково Mint, які застосовують календарне планування витрат і аналіз трендів, однак без елементів машинного навчання чи персоналізації.

Щодо безпеки, усі розглянуті системи використовують захищені протоколи зв'язку (SSL/TLS) та шифрування даних, але лише окремі рішення мають можливість двофакторної автентифікації або токенізації сесій.

Проблема локалізації залишається однією з найсуттєвіших — переважна більшість популярних сервісів не підтримує українську валюту, банки чи інтерфейс, що знижує їх придатність для локального ринку.

Результати порівняльного аналізу свідчать, що сучасні веб-додатки для управління фінансами здебільшого функціонують як облікові системи з елементами візуалізації, а не як комплексні аналітичні платформи. Вони не реалізують повноцінного циклу обробки фінансової інформації — від збору даних до побудови прогнозів і формування рекомендацій.

Ключовими недоліками існуючих рішень є:

- обмежені можливості прогнозування, відсутність алгоритмів машинного навчання;
- низький рівень персоналізації звітів і рекомендацій;
- часткова або відсутня локалізація для України;
- платна основна функціональність більшості іноземних систем;
- недостатня інтеграція з банківськими API українського ринку.

Водночас аналіз показав, що користувачі очікують від таких систем не лише інструментів обліку, а й аналітичну підтримку рішень — прогноз майбутніх витрат, рекомендації щодо оптимізації бюджету, контроль фінансових цілей та попередження про перевитрати. Саме ці елементи повинні стати основою майбутнього аналітичного веб-додатку.

У результаті порівняльного аналізу визначено, що існуючі веб-системи управління особистими фінансами мають високу зручність використання, але недостатньо розвинену аналітичну складову. Вони здебільшого орієнтовані на користувачів закордонних ринків і не відповідають потребам українських користувачів у плані локалізації та підтримки національних платіжних систем.

Таким чином, виникає необхідність у створенні нового аналітичного веб-додатку, який поєднує переваги сучасних технологій веб-розробки, інтеграцію з фінансовими сервісами, глибоку аналітику, прогнозування та персоналізовану взаємодію з користувачем. Розробка такого рішення дозволить

усунути виявлені недоліки, розширити функціональні можливості системи та підвищити ефективність процесу управління особистими фінансами.

1.5. Визначення недоліків наявних підходів і формулювання проблеми дослідження

Проведений аналіз сучасних веб-додатків для управління особистими фінансами дозволив виявити низку характерних обмежень, які істотно знижують ефективність їхнього використання як інструментів фінансового аналізу. Незважаючи на швидкий розвиток технологій веб-аналітики та появу численних фінансових сервісів, більшість рішень залишаються переважно засобами обліку, а не аналітичними платформами підтримки прийняття рішень.

Одним із головних недоліків існуючих підходів є обмежена глибина аналітичної обробки фінансових даних. Більшість систем, навіть популярні міжнародні сервіси, виконують лише базові статистичні операції — підрахунок загальної суми доходів, витрат і побудову елементарних діаграм.[9] При цьому відсутні механізми виявлення закономірностей у фінансовій поведінці користувача,[22] аналіз динаміки показників чи побудова прогнозів. Як наслідок, користувач отримує лише констатацію фактів, але не має інструментів для глибокого аналізу причин фінансових змін чи рекомендацій щодо оптимізації витрат.

Другим суттєвим недоліком є відсутність інтелектуальної персоналізації. Більшість сучасних веб-додатків пропонують однаковий набір звітів для всіх користувачів, не враховуючи їхніх індивідуальних особливостей — стилю витрат, сезонних коливань доходів, наявності боргових зобов'язань чи інвестицій. Персоналізація, якщо і реалізується, має обмежений характер і стосується лише вибору валют або категорій витрат. Відтак користувач позбавлений можливості отримувати аналітичні рекомендації, що враховують його фінансову поведінку.

Ще одним обмеженням є відсутність інструментів прогнозування фінансового стану.[14] Переважна більшість досліджених систем не використовує математичних чи алгоритмічних моделей для оцінки майбутніх

витрат або доходів. Фінансові прогнози, якщо вони існують, ґрунтуються на простому екстраполюванні минулих даних без застосування статистичних методів або машинного навчання. Це значно знижує точність прогнозів і не дозволяє ефективно планувати майбутні бюджети.

Також характерною проблемою для більшості веб-систем є недостатня інтеграція з банківськими сервісами,[21] особливо на локальному рівні. Іноземні продукти часто не підтримують українські фінансові установи, а вітчизняні аналоги — не мають доступу до відкритих банківських API. Це унеможлиблює автоматичний збір транзакцій і змушує користувачів вводити дані вручну, що знижує точність та актуальність фінансової інформації.

Важливою групою недоліків є питання безпеки та конфіденційності персональних даних. Незважаючи на впровадження протоколів шифрування та автентифікації, не всі системи дотримуються сучасних вимог захисту інформації. У деяких випадках дані користувачів зберігаються без достатнього рівня анонімізації або передаються на сторонні сервери. Відсутність сертифікації за міжнародними стандартами інформаційної безпеки (ISO/IEC 27001, GDPR) створює ризики витоку конфіденційної інформації.

Окремої уваги потребує проблема локалізації та адаптації веб-додатків до українського ринку. Більшість іноземних платформ не підтримує гривню, національні банківські формати або локальні податкові особливості. Це створює бар'єр для широкого використання подібних систем серед українських користувачів. Локальні рішення, навпаки, часто мають обмежений функціонал, не володіють розвиненими аналітичними інструментами й не використовують сучасні підходи до візуалізації чи прогнозування.

Серед інших недоліків можна виокремити недостатню інтерактивність інтерфейсу, відсутність можливості гнучкого налаштування панелі звітів і низьку зручність візуалізації даних. У багатьох системах графіки мають статичний характер, що обмежує можливість проведення користувачем самостійного аналізу. Крім того, не всі веб-додатки враховують адаптивність під

різні типи пристроїв, що є суттєвим недоліком у контексті сучасної веб-розробки.

На рисунку 1.5.1 наведено узагальнену схему, що ілюструє основні недоліки сучасних веб-додатків управління особистими фінансами. Таким чином, комплексний аналіз дозволяє виділити ключові групи недоліків у сучасних підходах.



Рис. 1.3 Ключові недоліки сучасних веб-додатків управління особистими фінансами

Сукупність зазначених недоліків визначає науково-практичну проблему, що потребує вирішення в межах даного дослідження. Вона полягає у створенні веб-додатку, який поєднує сучасні технології збору, обробки та аналітики фінансових даних з використанням алгоритмів прогнозування і персоналізованої візуалізації, забезпечуючи при цьому високий рівень безпеки та локалізацію під українського користувача.

Таким чином, можна сформулювати основне протиріччя сучасних систем: попит користувачів на глибоку аналітику, прогнозування і зручний веб-доступ зростає, однак наявні інструменти не задовольняють ці потреби у повному обсязі. Це протиріччя і визначає напрям подальшого наукового дослідження —

розробку аналітичного веб-додатку нового покоління, здатного перетворювати фінансові дані користувача на знання для підтримки прийняття рішень.

1.6. Постановка завдання на створення аналітичного веб-додатку управління фінансами

На основі проведеного системного аналізу та виявлених недоліків існуючих рішень сформульовано основну науково-практичну проблему — відсутність веб-додатку, який би поєднував функції автоматичного збору фінансових даних, глибокої аналітики, прогнозування та персоналізованої візуалізації при дотриманні сучасних вимог безпеки та локалізації. Для її розв'язання необхідно розробити інтегровану інформаційну систему, здатну ефективно підтримувати користувача у процесі управління особистими фінансами на основі аналітичних методів обробки даних.

Метою даної роботи є розробка концепції та створення аналітичного веб-додатку управління особистими фінансами, який забезпечує збір, обробку, аналіз і прогнозування фінансових даних користувачів із використанням сучасних веб-технологій, алгоритмів аналітики та засобів візуалізації.

Система повинна не лише відображати поточний фінансовий стан користувача, а й формувати прогнози майбутніх доходів і витрат, виявляти закономірності фінансової поведінки та генерувати рекомендації щодо оптимізації бюджету.

Для досягнення поставленої мети в межах роботи необхідно вирішити такі науково-технічні завдання:

1. Провести аналіз предметної області управління особистими фінансами та виявити закономірності у структурі фінансових транзакцій користувачів.
2. Розробити інформаційну модель системи, яка описує взаємозв'язки між користувачами, транзакціями, категоріями доходів і витрат, рахунками та фінансовими цілями.
3. Створити архітектурну модель веб-додатку, що реалізує клієнтсько-серверну взаємодію, підтримує обробку даних у реальному часі та забезпечує розширюваність системи.

4. Розробити алгоритми аналітичної обробки фінансових даних, які дозволяють виявляти тренди, обчислювати ключові фінансові показники (KPI) та здійснювати прогнозування майбутніх витрат і доходів.
5. Забезпечити високий рівень інформаційної безпеки, включаючи шифрування персональних даних, автентифікацію користувачів і захист від несанкціонованого доступу.
6. Реалізувати інтерактивний користувацький інтерфейс із використанням сучасних бібліотек візуалізації даних Chart.js та D3.js,[12] що забезпечує наочне відображення фінансової інформації.
7. Провести апробацію функціонування системи, оцінити точність прогнозування, продуктивність обробки даних та зручність користування.

Аналітичний веб-додаток має реалізовувати такі основні функції: реєстрація та авторизація користувачів із застосуванням багаторівневої автентифікації, збір фінансових даних вручну або шляхом імпорту із зовнішніх джерел (CSV, XLSX, API банків), класифікація транзакцій за категоріями доходів і витрат, побудова статистичних звітів і динамічних графіків фінансових потоків, прогнозування доходів і витрат на основі часових рядів, аналіз співвідношення доходів, витрат і заощаджень користувача, формування персональних рекомендацій для оптимізації бюджету, експорт аналітичних звітів у різні формати (PDF, XLSX, PNG), захист персональних даних відповідно до стандартів GDPR та ISO/IEC 27001.

У межах системи передбачається забезпечення високого рівня продуктивності, за якого обробка запитів користувача здійснюється у режимі реального часу, а час отримання та відображення даних не перевищує однієї секунди. Архітектура веб-додатку має бути масштабованою, що дозволить підтримувати значну кількість одночасних користувачів без погіршення швидкодії або стабільності роботи системи. Важливою вимогою є також постійна доступність сервісу: користувач повинен мати можливість працювати з веб-додатком у цілодобовому режимі та з будь-якого пристрою, використовуючи стандартний веб-браузер.

Окреме місце займає питання інформаційної безпеки. Усі фінансові дані мають зберігатися у зашифрованому вигляді із застосуванням протоколів SSL/TLS, що забезпечує цілісність та конфіденційність інформації під час передавання та зберігання. Не менш важливим компонентом є юзабіліті: інтерфейс повинен бути інтуїтивно зрозумілим і адаптивним до різних роздільних здатностей екрана, а також мати можливість персоналізованого налаштування відображення. Крім того, система повинна бути повністю локалізованою для українського користувача, що передбачає підтримку української мови, національної валюти (UAH) та відповідних форматів дат.

Запропонований веб-додаток передбачає реалізацію трикомпонентної архітектури:

- Клієнтська частина — інтерфейс користувача, розроблений із використанням технологій HTML5, CSS3, JavaScript(React).
- Серверна частина — REST API на основі Node.js та Python,[11] що забезпечує логіку обробки запитів, аналітику та доступ до бази даних.
- База даних — централізоване сховище фінансової інформації (PostgreSQL), у якому реалізовано механізми нормалізації, резервного копіювання та контролю доступу.

На аналітичному рівні система має використовувати алгоритми статистичного аналізу (ковзні середні, експоненційне згладжування) або моделі часових рядів (ARIMA, Prophet) для прогнозування фінансових показників. Це дозволить формувати рекомендації щодо оптимального розподілу ресурсів, планування бюджету та управління заощадженнями.

Результатом реалізації поставленого завдання стане аналітичний веб-додаток, який автоматизує облік доходів, витрат і заощаджень користувача, здійснює аналітичну обробку та прогнозування фінансових даних, забезпечує наочну візуалізацію показників у реальному часі, гарантує безпечне зберігання інформації, надає персоналізовані рекомендації щодо фінансового планування.

У результаті проведеної постановки завдання визначено мету, структуру, функціональні та нефункціональні вимоги до майбутнього аналітичного

веб-додатку управління фінансами. Сформульовано основні напрями його реалізації, що включають побудову архітектури, розробку моделей даних, алгоритмів аналізу й прогнозування.

Висновки до розділу 1

У першому розділі магістерської роботи проведено системний аналіз предметної області управління особистими фінансами, що дало змогу всебічно дослідити теоретичні, методологічні та прикладні аспекти побудови сучасних веб-систем такого типу. Проаналізовано сутність процесів фінансового планування, структуру фінансових даних користувачів, а також принципи їх збору, обробки та зберігання з урахуванням вимог безпеки й конфіденційності.

У ході дослідження встановлено, що інформаційна модель управління особистими фінансами базується на взаємодії між транзакціями, категоріями доходів і витрат, рахунками, бюджетами та фінансовими цілями користувача. Розглянуто особливості організації даних у веб-середовищі, зокрема питання нормалізації бази даних, захисту персональної інформації, резервного копіювання та доступу до віддалених сховищ.

В результаті проведеного огляду та класифікації сучасних веб-додатків (Mint, YNAB, PocketSmith, Spendee, Finmap, Wally та ін.) визначено їхні основні функціональні особливості та відмінності. Встановлено, що більшість існуючих рішень орієнтовані переважно на базовий облік і статистику, тоді як глибока аналітика, інтелектуальне прогнозування та персоналізовані рекомендації реалізуються обмежено або взагалі відсутні.

Проведений порівняльний аналіз показав, що актуальною є потреба у створенні вітчизняного аналітичного веб-додатку, який би поєднував автоматизований збір фінансових даних, засоби інтерактивної аналітики, візуалізації, прогнозування та персоналізації під потреби конкретного користувача. Таке рішення має враховувати специфіку українського фінансового ринку, підтримку національної валюти, локалізацію інтерфейсу та дотримання міжнародних стандартів інформаційної безпеки.

На основі проведених досліджень сформульовано основну науково-практичну проблему, що полягає у відсутності комплексного веб-рішення для аналітичної підтримки управління особистими фінансами. З метою її розв'язання поставлено завдання створення веб-додатку, який реалізує клієнтсько-серверну архітектуру, підтримує обробку фінансових даних у реальному часі, забезпечує прогнозування доходів і витрат та формує рекомендації на основі аналітичних алгоритмів.

Таким чином, у першому розділі сформовано теоретичне та методологічне підґрунтя подальшої розробки системи. Отримані результати лягли в основу постановки завдання проєктування аналітичного веб-додатку, що становить предмет подальшого дослідження у розділі 2 — “Моделювання системи”, де буде виконано формалізацію предметної області та побудову відповідних моделей даних, функцій і взаємодій.

РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ АНАЛІТИЧНОГО УПРАВЛІННЯ ОСОБИСТИМИ ФІНАНСАМИ

2.1. Аналіз предметної області та постановка наукової задачі

Предметна область управління особистими фінансами охоплює процеси планування, обліку, контролю, аналізу та прогнозування індивідуальних фінансових потоків. Її метою є забезпечення фінансової стабільності користувача, оптимізація витрат і досягнення довгострокових цілей за рахунок раціонального розподілу ресурсів. В умовах цифровізації економіки управління особистими фінансами дедалі частіше здійснюється за допомогою веб-технологій, що дозволяють автоматизувати облік транзакцій, здійснювати аналітичну обробку даних і забезпечувати доступ до фінансової інформації в будь-який час та з будь-якого пристрою.[4]

Процес управління фінансами користувача передбачає послідовну обробку даних, починаючи зі збору інформації про доходи, витрати та заощадження. Отримані від користувача або банківських API дані проходять етапи класифікації за категоріями, перевірки коректності, агрегації та подальшого аналізу. На основі систематизованих транзакцій розраховуються основні фінансові показники: баланс, структура витрат, середні значення, динаміка доходів. Далі формується аналітична інформація, яка використовується для прогнозування майбутнього фінансового стану користувача, визначення тенденцій і формування рекомендацій щодо оптимізації бюджету.

В інформаційному середовищі системи циркулює кілька типів даних. Первинні дані — це безпосередньо транзакції користувача, отримані вручну або через інтеграцію з банками. На їх основі формуються довідкові дані, які включають категорії витрат, типи доходів, рахунки, валюти й додаткові параметри. У результаті обробки з'являються аналітичні дані, що відображають фінансову динаміку, структуру витрат і розраховані індикатори. Завершальним етапом є формування результатуючої інформації — звітів, графіків, прогнозних таблиць і рекомендацій для користувача. Така послідовність утворює логічну

схему інформаційних потоків системи, що забезпечує цілісність і узгодженість усіх елементів.[10]

Проведений аналіз предметної області дозволив встановити низку обмежень, притаманних сучасним системам управління особистими фінансами. Основною проблемою є відсутність комплексного підходу до аналітичної обробки даних. Існуючі рішення зазвичай виконують функції обліку або статистичного підрахунку, але не надають засобів для глибокого аналізу, виявлення закономірностей чи прогнозування. Значна частина додатків не інтегрована з банківськими сервісами, що унеможлиблює автоматичне оновлення інформації. Часто системи не мають адаптації до локального фінансового середовища, обмежені у функціях персоналізації та не забезпечують належного рівня захисту персональних даних.

У цьому контексті постає необхідність розроблення веб-додатку нового типу, який поєднує автоматизований збір фінансових даних, їх інтелектуальну обробку, побудову прогнозів та аналітичну візуалізацію. Наукова задача, яку розв'язує ця робота, полягає у створенні моделі й архітектури веб-системи, що реалізує процеси збору, класифікації, аналітичної обробки та прогнозування фінансових даних користувача з використанням сучасних методів статистики, машинного навчання та математичного моделювання.

Розв'язання цієї задачі передбачає розробку функціональної моделі, що описує основні процеси та потоки даних; об'єктно-орієнтованої моделі, у якій визначено сутності системи, їхні атрибути та зв'язки; а також математичної моделі аналітичної обробки транзакцій, здатної забезпечити динамічну оцінку фінансового стану користувача. Одночасно необхідно врахувати вимоги до безпеки зберігання персональних даних і забезпечити відповідність системи міжнародним стандартам інформаційної безпеки.

Наукова новизна постановки задачі полягає у поєднанні веб-аналітичних технологій, математичного моделювання та персоналізованого прогнозування у межах єдиної інтерактивної платформи. Такий підхід дозволяє перейти від пасивного обліку фінансів до активної підтримки прийняття рішень, що

базується на реальних даних користувача. Практична значущість полягає у можливості використання створеної системи для підвищення фінансової грамотності, планування особистого бюджету та оптимізації витрат у реальному житті.

Таким чином, проведений аналіз предметної області визначив передумови для побудови аналітичної системи управління особистими фінансами. У подальшому розділі буде здійснено функціональне моделювання основних процесів роботи веб-додатку з використанням методологій DFD та IDEF0, що дозволить формалізувати логіку обробки даних і взаємодію користувача із системою.

2.2. Функціональне моделювання процесів системи

Функціональне моделювання є важливим етапом побудови інформаційних систем, оскільки воно дозволяє описати логіку функціонування системи у вигляді взаємопов'язаних процесів і потоків даних. Для аналітичного веб-додатку управління особистими фінансами це моделювання забезпечує чітке уявлення про те, як відбувається збір, обробка, аналіз і представлення фінансової інформації користувачу. Функціональні моделі дозволяють узгодити вимоги до системи, визначити її структуру та забезпечити основу для подальшого проектування.

У процесі моделювання було використано дві взаємодоповнюючі методології — DFD (Data Flow Diagram) та IDEF0 (Integration Definition for Function Modeling). DFD-моделі застосовуються для опису потоків даних між компонентами системи, тоді як IDEF0 забезпечує більш формальне відображення функцій, що перетворюють вхідну інформацію у вихідну. Сукупне використання цих підходів дозволяє побудувати цілісну картину функціонування веб-додатку як складної багаторівневої системи обробки фінансової інформації.[11]

На контекстному рівні система розглядається як єдиний цілісний об'єкт, що взаємодіє з користувачами та зовнішніми джерелами даних. Основними

зовнішніми сутностями є користувач системи, який вводить або імпортує фінансові транзакції; банківські сервіси, що надають дані через API; та аналітичне середовище, яке використовується для візуалізації результатів.

У межах контекстної діаграми основні потоки даних мають наступний вигляд:

користувач передає системі інформацію про доходи, витрати, перекази чи заощадження; веб-додаток здійснює обробку цих даних, класифікує їх за категоріями та зберігає у базі даних; далі система формує звіти, графіки та прогнозні оцінки, які повертаються користувачу у вигляді аналітичних результатів. Таким чином, на цьому рівні система виступає як посередник між користувачем і джерелами фінансових даних, забезпечуючи їх перетворення у знання, корисні для прийняття рішень.[4]

На рисунку 2.1 подано контекстну DFD-діаграму аналітичного веб-додатку управління особистими фінансами, яка ілюструє взаємодію користувача із системою та основні потоки даних.

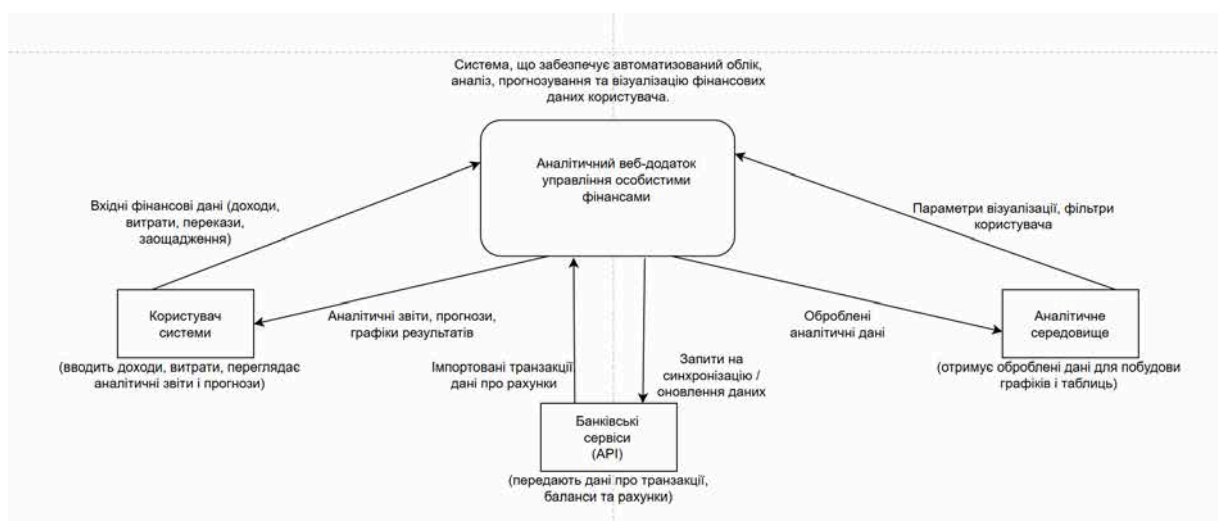


Рис. 2.1 Контекстна DFD-діаграма аналітичного веб-додатку управління особистими фінансами

На нульовому рівні відбувається декомпозиція системи на основні підпроцеси. У веб-додатку управління фінансами можна виділити чотири ключові функціональні блоки: збір і введення фінансових даних, обробка й

класифікація транзакцій, аналітична обробка і формування звітів, прогнозування та візуалізація результатів.

У першому процесі здійснюється збирання інформації з різних джерел — як від користувача, так і з банківських API або файлів імпорту. Система перевіряє дані на коректність, визначає тип операції та створює запис у базі даних.

Другий процес охоплює класифікацію транзакцій за категоріями, валютами, рахунками та часовими періодами. Ця інформація потрапляє до центрального сховища, де формується структурована база даних користувача.

У третьому процесі реалізується аналітичне опрацювання: система обчислює ключові фінансові показники, формує агреговані таблиці та статистичні узагальнення. На цьому етапі відбувається підготовка даних до побудови звітів і графічного представлення.[8]

Заключний процес відповідає за прогнозування майбутніх фінансових показників із використанням алгоритмів часових рядів. Отримані результати виводяться у вигляді інтерактивних графіків, діаграм та рекомендацій для користувача.

Таким чином, нульовий рівень DFD демонструє послідовність етапів від надходження даних до їхнього аналітичного відображення, забезпечуючи логічну завершеність циклу фінансового аналізу.

На рисунку 2.2 наведено DFD-діаграму першого рівня аналітичного веб-додатку управління особистими фінансами, що деталізує взаємозв'язки між підпроцесами та потоками даних усередині системи.

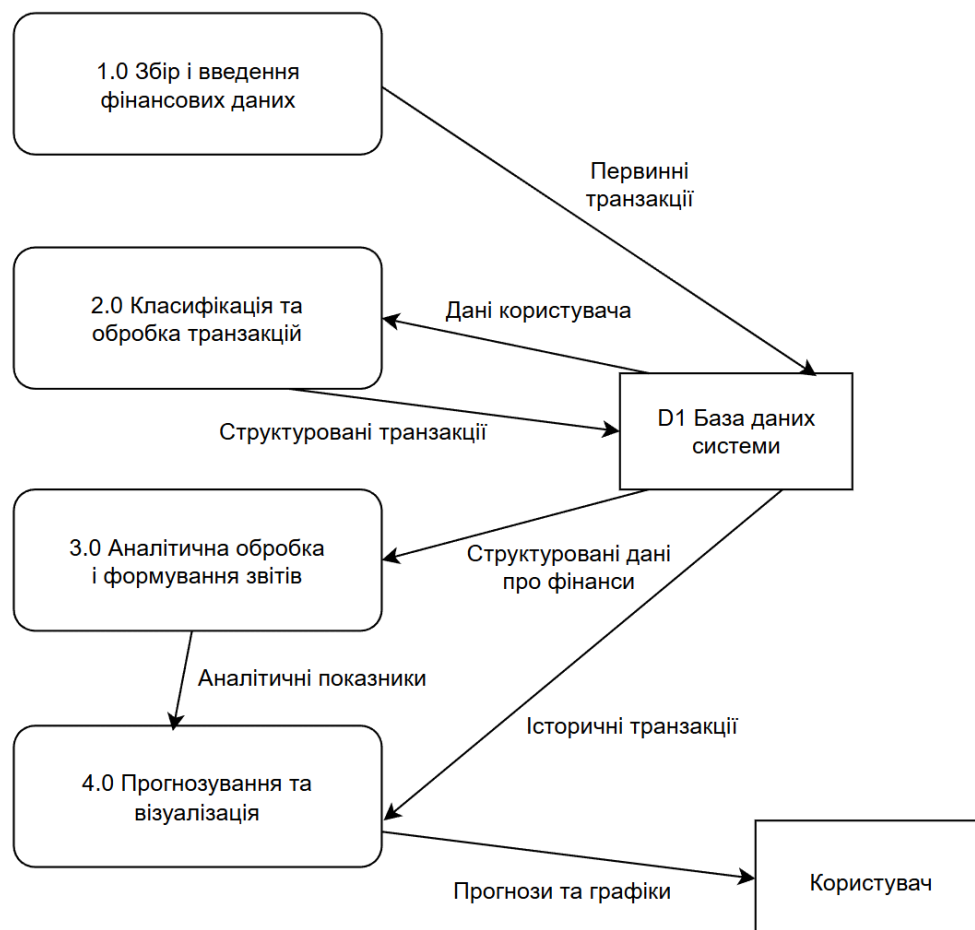


Рис. 2.2 DFD-діаграма першого рівня аналітичного веб-додатку управління особистими фінансами

Поглиблена деталізація на першому рівні розкриває внутрішню структуру підпроцесів. Зокрема, процес обробки даних розділяється на перевірку коректності записів, автоматичну категоризацію, оновлення стану рахунків та збереження результатів у базі даних. Аналогічно, аналітична підсистема поділяється на модуль статистичного аналізу, підсистему порівняльних звітів і модуль прогнозування. Усі вони взаємодіють через спільне сховище даних, забезпечуючи цілісність інформаційних потоків.[7]

Користувач, взаємодіючи з інтерфейсом, надсилає запити до системи, які через серверну частину потрапляють до відповідних модулів. Результати обробки зберігаються у базі, а потім передаються до клієнтського інтерфейсу

для відображення у вигляді аналітичних звітів або графіків. Такий підхід відображає класичну клієнтсько-серверну архітектуру веб-додатків.

Для більш глибокого відображення функціональної структури системи побудовано модель у нотації IDEF0, яка дозволяє описати взаємозв'язки між функціями за принципом «вхід–керування–вихід–механізм». Основна функція системи «Аналітична обробка фінансових даних користувача» має такі характеристики:

- вхідні дані — транзакції користувача, файли імпорту, параметри фільтрації та часові інтервали;
- керування — алгоритми аналітичної обробки, правила безпеки, налаштування користувача;
- вихідні дані — аналітичні звіти, прогнозні оцінки, графічні візуалізації;
- механізм реалізації — база даних, серверна логіка, аналітичні модулі та веб-інтерфейс.

У результаті побудови IDEF0-моделі визначено, що всі функціональні блоки системи мають бути організовані у вигляді взаємопов'язаних компонентів, які спільно формують аналітичну екосистему веб-додатку.

Функціональне моделювання аналітичного веб-додатку управління особистими фінансами дозволило формалізувати його структуру, визначити основні процеси та потоки даних, що забезпечують реалізацію всіх ключових функцій системи — від збору інформації до аналітичного прогнозування. Використання методологій DFD та IDEF0 дало змогу встановити логічні зв'язки між підсистемами, відобразити рух інформації в межах системи та визначити залежності між компонентами.

2.3. Об'єктно-орієнтоване моделювання системи

Об'єктно-орієнтоване моделювання є ключовим етапом формалізації архітектури інформаційної системи. На відміну від функціонального моделювання, яке описує процеси на рівні потоків даних, об'єктно-орієнтований підхід зосереджується на структурі системи, її

складових об'єктах, їхніх атрибутах, методах і взаємозв'язках. Такий підхід дозволяє створити модель, максимально наближену до програмної реалізації майбутнього веб-додатку.

У межах розробки аналітичного веб-додатку управління особистими фінансами об'єктно-орієнтоване моделювання здійснюється за допомогою уніфікованої мови моделювання UML (Unified Modeling Language). Використання UML дозволяє описати систему на різних рівнях абстракції: від загальних сценаріїв взаємодії користувача з додатком до детальної структури класів, методів і логіки обміну повідомленнями. У межах цього підрозділу розглядаються основні UML-діаграми, які відображають логіку функціонування системи — діаграма прецедентів, діаграма послідовностей і діаграма активностей.[7]

Діаграма прецедентів відображає основні сценарії взаємодії користувача із системою. В аналітичному веб-додатку управління фінансами головним актором є користувач, який здійснює операції з введення, перегляду й аналізу власних фінансових даних. Додатковими акторами виступають адміністратор системи та зовнішні банківські сервіси, які забезпечують передачу даних через API.

Користувач взаємодіє з системою у таких основних сценаріях: реєстрація та автентифікація, додавання фінансових транзакцій, імпорт даних із зовнішніх джерел, перегляд поточного балансу, побудова звітів, отримання прогнозів витрат і доходів, а також керування власними категоріями фінансів. Адміністратор виконує завдання моніторингу безпеки, резервного копіювання та оновлення системи, тоді як банківські сервіси передають дані про транзакції у стандартизованому форматі.[18]

На рисунку 2.3 наведено діаграму прецедентів, що відображає ключові сценарії взаємодії користувача з системою.

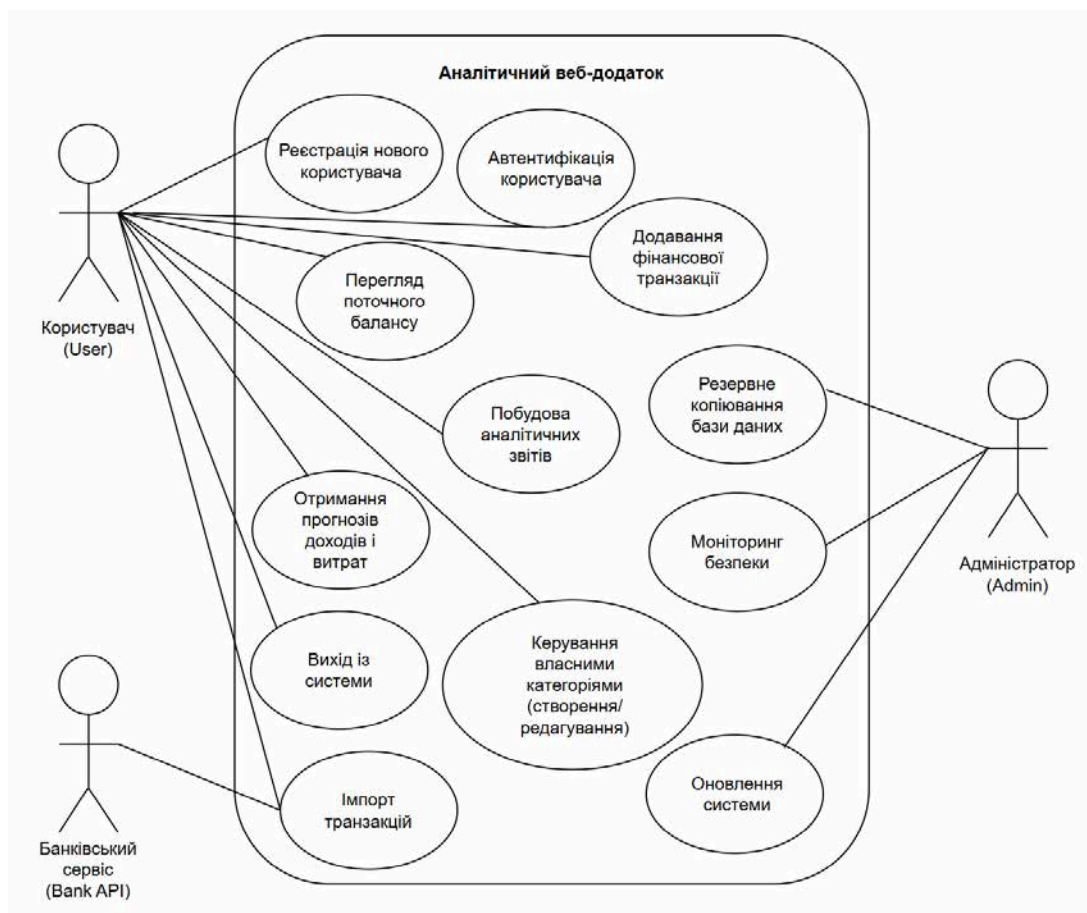


Рис. 2.3 Діаграма прецедентів аналітичного веб-додатку управління особистими фінансами

На основі цих сценаріїв визначено межі системи та набір функціональних вимог, які забезпечують реалізацію ключових прецедентів. Діаграма прецедентів демонструє, що всі взаємодії користувача з додатком реалізуються через веб-інтерфейс, який є клієнтською частиною системи та взаємодіє із серверною логікою через REST API.

Діаграма класів визначає структуру об'єктів системи, їхні властивості, методи та зв'язки між ними. У межах розробленої системи можна виділити кілька базових класів, які утворюють ядро інформаційної моделі.

Клас User містить атрибути userID, name, email, passwordHash, а також методи для автентифікації та управління профілем. Клас Transaction зберігає інформацію про фінансові операції: transactionID, date, amount, type, categoryID, accountID і description. Для організації категорій існує клас Category, який

включає `categoryID`, `name`, `type` (дохід або витрата) і дозволяє користувачеві створювати власні категорії. Клас `Account` відображає банківські рахунки або гаманці користувача й містить поля `accountID`, `balance`, `currency`.

Аналітична частина системи реалізована у вигляді класу `Analytics`, який обробляє фінансові дані, обчислює сумарні показники, структуру витрат, середні значення, а також формує аналітичні звіти. Для прогнозування використовується клас `Forecast`, який реалізує алгоритми часових рядів (наприклад, ковзне середнє або експоненціальне згладжування). Клас `Report` відповідає за візуалізацію результатів у вигляді графіків і таблиць, які відображаються користувачу через веб-інтерфейс.[19]

Взаємозв'язки між класами мають ієрархічний характер: користувач може мати кілька рахунків, кожен рахунок — багато транзакцій, кожна транзакція належить певній категорії. Класи `Analytics` та `Forecast` взаємодіють із базою даних для отримання та оновлення інформації, що забезпечує реалізацію бізнес-логіки додатку. Така структура класів забезпечує розширюваність системи та гнучкість її архітектури.

Діаграма послідовностей описує динамічну взаємодію між об'єктами системи у часі. У розробленому веб-додатку типовий сценарій роботи користувача починається з автентифікації: користувач надсилає запит із даними для входу, система перевіряє правильність облікових даних і, у разі успіху, створює сесію доступу. Після цього користувач може додавати нові транзакції, які передаються до сервера через REST API, проходять перевірку коректності та записуються до бази даних.[8]

Далі користувач ініціює запит на побудову звіту або прогнозу. Система надсилає запит до аналітичного модуля, який виконує розрахунки, формує структуру результатів і повертає їх до клієнтської частини. Веб-інтерфейс отримує відповіді від сервера у форматі JSON і будує графічне представлення даних у вигляді діаграм або таблиць. Таким чином, діаграма послідовностей ілюструє ланцюг обміну повідомленнями між користувачем, сервером і

аналітичними модулями, демонструючи логіку виконання запитів у реальному часі, що представлено на рисунку 2.4.

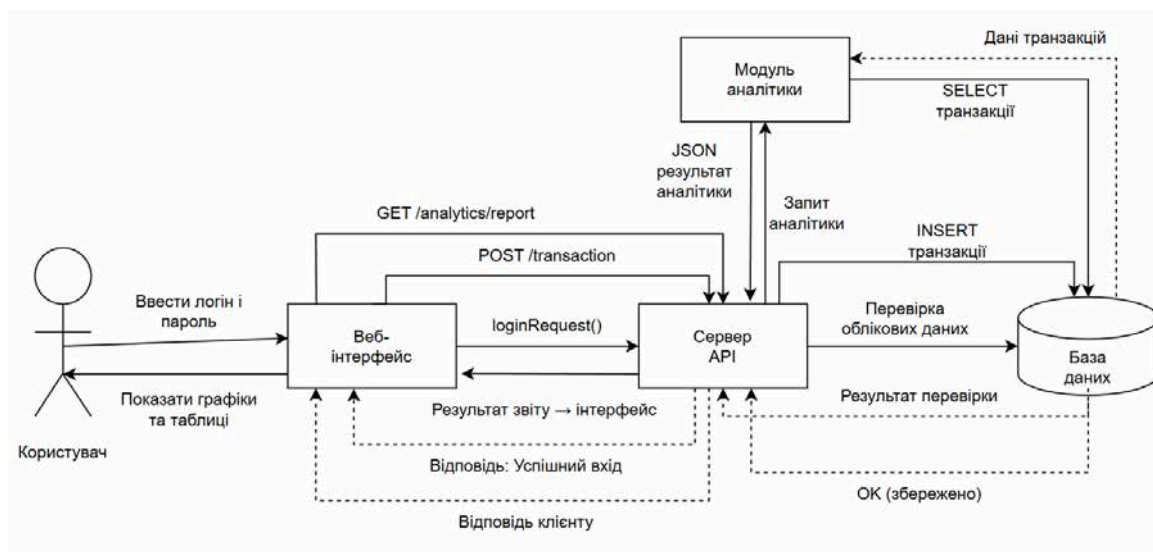


Рис. 2.4 Діаграма послідовностей взаємодії користувача із системою

Як видно з рисунку 2.4, користувач ініціює запит через веб-інтерфейс, який передає дані до серверної частини системи для обробки, аналітичного аналізу та подальшого відображення результатів у вигляді звітів і графіків.

Діаграма активностей відображає алгоритм функціонування системи з точки зору користувача та послідовності дій. Початковою активністю є авторизація, після якої користувач може обрати одну з доступних операцій: введення транзакції, перегляд статистики або аналіз фінансового стану. Якщо вибрано додавання операції, система перевіряє правильність введених даних, виконує категоризацію й оновлює базу. Якщо обрано побудову звіту, аналітичний модуль виконує обчислення та передає результати у модуль візуалізації.[9]

Далі користувач може скористатися функцією прогнозування. На цьому етапі система аналізує історичні дані, застосовує обраний алгоритм і формує прогноз витрат або доходів на певний період. Результат відображається у вигляді динамічної діаграми з рекомендаціями щодо оптимізації бюджету. Завершальною активністю є збереження результатів у профілі користувача та можливість експорту у форматах PDF або XLSX.

На рисунку 2.5 подано діаграму активностей аналітичного веб-додатку управління особистими фінансами, яка відображає послідовність дій користувача та системи під час роботи з фінансовими даними.

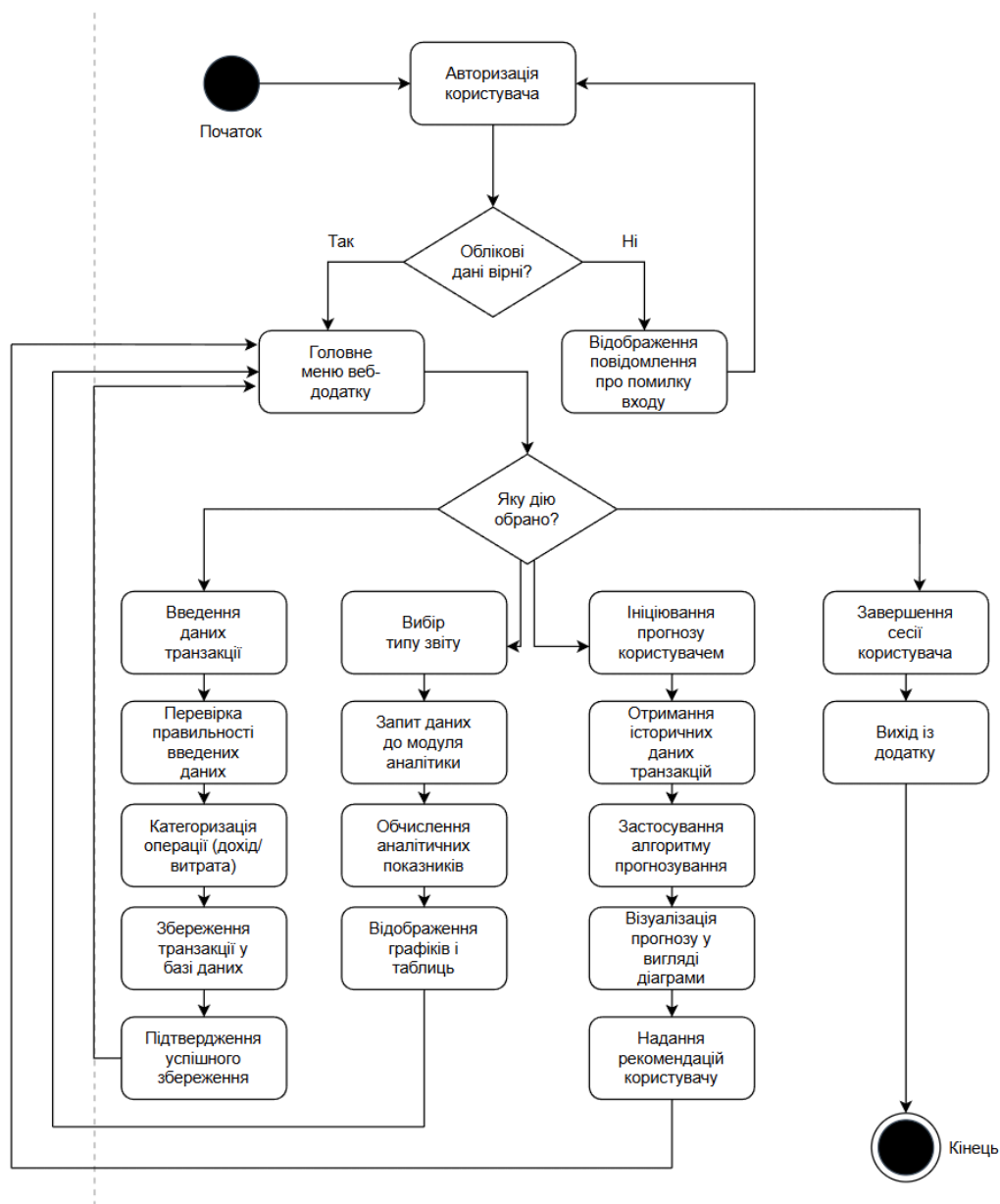


Рис. 2.5 Діаграма активностей аналітичного веб-додатку управління особистими фінансами

Таким чином, діаграма активностей демонструє, що система працює за принципом замкнутого циклу — від введення даних до отримання аналітичних і прогнозних результатів із можливістю повторного аналізу.

Об'єктно-орієнтоване моделювання аналітичного веб-додатку управління особистими фінансами дозволило сформулювати структурне уявлення про

систему, визначити її ключові сутності, сценарії взаємодії та логіку функціонування. Розроблені UML-діаграми відображають як статичну, так і динамічну структуру системи, забезпечуючи основу для переходу до етапу розробки архітектури й математичного моделювання аналітичних процесів.

Діаграми прецедентів окреслили взаємодію користувачів із системою; діаграма класів — структуру та зв'язки між об'єктами; діаграми послідовностей і активностей — динаміку виконання операцій і процесів обробки даних. Сукупність цих моделей формує концептуальну архітектуру веб-додатку, на базі якої у подальших підрозділах буде побудовано математичну модель аналітичної обробки фінансових транзакцій.

2.4. Алгоритми фінансового аналізу та прогнозування

Аналітичний веб-додаток управління особистими фінансами покликаний забезпечити не лише облік транзакцій, а й комплексну аналітичну підтримку прийняття рішень користувачем. Основним завданням алгоритмів є трансформація первинних даних про доходи, витрати, заощадження та перекази у систематизовану інформацію, що дозволяє оцінити поточний фінансовий стан, прогнозувати майбутні показники та формувати персоналізовані рекомендації.

Процес обробки даних починається зі збору інформації з різних джерел. Веб-додаток інтегрується з банківськими сервісами через стандартизовані API, приймає дані з локальних файлів або через ручний ввід користувача. Отримані дані проходять процедури перевірки коректності та повноти. Система визначає відповідність сум, дат та категорій, приводить валюти до єдиного стандарту та класифікує транзакції за типами. На цьому етапі формується єдина база даних користувача, що дозволяє проводити подальший аналіз. Особливу увагу приділено обробці аномальних або неповних даних, що забезпечує надійність результатів аналітики та прогнозування.[10]

Статистичний аналіз фінансових потоків виконується наступним етапом і дозволяє систематизувати дані для подальшого прогнозування. На основі історичних даних розраховуються ключові показники бюджету, середні та

медіанні витрати за категоріями, структура доходів та витрат, а також співвідношення між основними категоріями фінансових потоків. Аналіз динаміки транзакцій дозволяє виділити регулярні тренди, сезонні коливання та короткострокові аномалії. Кореляційний аналіз дозволяє виявляти взаємозв'язки між категоріями витрат і доходів, оцінювати ризики перевитрат та визначати фактори, які найсуттєвіше впливають на зміну балансу користувача.

Прогнозування фінансових показників у системі реалізується на основі сучасних методів аналізу часових рядів та простих моделей машинного навчання. Алгоритми часових рядів дозволяють виділити тренди, сезонні коливання та короткострокові флуктуації. Для цього використовується експоненційне згладжування, яке дозволяє надавати більшої ваги останнім даним, що забезпечує точніше передбачення майбутніх витрат та доходів. Система здатна прогнозувати майбутній баланс, оцінювати ймовірні перевитрати у кожній категорії витрат і визначати необхідні накопичення для досягнення фінансових цілей користувача.[8]

Регресійний аналіз та прості алгоритми машинного навчання застосовуються для побудови залежностей між різними фінансовими показниками. Це дозволяє прогнозувати не лише загальні доходи або витрати, а й конкретні фінансові сценарії, враховуючи індивідуальні звички користувача та сезонність. Алгоритми класифікують нові транзакції за категоріями на основі історичних даних, визначають ймовірність перевитрат у майбутніх періодах та формують рекомендації щодо оптимізації бюджету. Наприклад, якщо система виявляє, що у певному місяці витрати на розваги перевищують середній рівень, вона формує підказку про можливе скорочення цієї категорії для досягнення фінансової стабільності.

Важливою особливістю системи є тісна інтеграція алгоритмів аналізу та прогнозування з функціональною та об'єктно-орієнтованою моделлю веб-додатку. Класи аналітики взаємодіють із базою даних, серверними компонентами та модулем візуалізації, що дозволяє користувачу отримувати актуальні результати у реальному часі. Результати прогнозування представлені у

вигляді інтерактивних графіків, динамічних діаграм та аналітичних таблиць, що дає змогу оцінити фінансовий стан, порівняти показники за різні періоди та планувати витрати.

На практичному рівні реалізація алгоритмів аналізу та прогнозування перетворює веб-додаток із простої системи обліку на потужний інструмент активного управління фінансами. Користувач отримує можливість аналізувати динаміку доходів і витрат, передбачати перевитрати, планувати накопичення та оцінювати вплив окремих фінансових рішень на загальний баланс. Таким чином, алгоритми фінансового аналізу та прогнозування формують серцевину аналітичної системи, забезпечуючи персоналізовану, інтегровану та динамічну підтримку процесів управління особистими фінансами.

2.5. Вимоги до інформаційної безпеки та захисту фінансових даних

У сучасних веб-додатках управління особистими фінансами безпека даних користувача є пріоритетним аспектом, оскільки система оперує конфіденційною інформацією, включаючи персональні дані, деталі банківських рахунків та історію фінансових транзакцій. Забезпечення захисту цієї інформації потребує комплексного підходу, що включає технічні, організаційні та програмні заходи безпеки.

Основною метою реалізації вимог інформаційної безпеки є гарантування конфіденційності, цілісності та доступності даних користувача. Конфіденційність забезпечує захист персональної інформації від несанкціонованого доступу, цілісність гарантує, що дані залишаються незмінними під час зберігання та передачі, а доступність забезпечує можливість користувачу безперешкодно отримувати необхідні дані та аналітичні результати у будь-який час.

Веб-додаток реалізує багато рівнів захисту. На рівні аутентифікації та авторизації використовується багатофакторна перевірка, яка дозволяє підтвердити особу користувача перед наданням доступу до системи. Паролі зберігаються у зашифрованому вигляді, а всі комунікації між клієнтською

частиною та сервером здійснюються через захищені протоколи передачі даних, що гарантує недоступність конфіденційної інформації для сторонніх осіб.

Окрему увагу приділено захисту бази даних, у якій зберігаються всі фінансові та персональні дані користувача. Для цього застосовуються методи шифрування даних у спокої та регулярне резервне копіювання, що дозволяє відновити інформацію у випадку технічних збоїв або атак. Крім того, система забезпечує контроль доступу до різних категорій даних: користувач має доступ тільки до власних транзакцій, адміністратор системи може виконувати операції з резервним копіюванням та моніторингом без доступу до приватних даних.

Захист фінансової інформації передбачає також моніторинг і запобігання можливим загрозам, включаючи атаки типу SQL-ін'єкцій, фішинг, перехоплення даних та несанкціоноване втручання у роботу сервера. Для цього веб-додаток інтегрується з механізмами виявлення аномальної активності, які дозволяють своєчасно реагувати на підозрілі дії та мінімізувати ризики втрати або викрадення даних.[16]

Особливу роль відіграє відповідність системи міжнародним стандартам інформаційної безпеки, зокрема вимогам GDPR (General Data Protection Regulation) та рекомендаціям ISO/IEC 27001. Це забезпечує користувачу гарантію того, що персональні та фінансові дані обробляються відповідно до сучасних нормативів і захищені на належному рівні.

Отже, бачимо, що розробка вимог до інформаційної безпеки у веб-додатку управління особистими фінансами є невід'ємною складовою побудови системи. Вона забезпечує надійний захист конфіденційних даних, контроль доступу, моніторинг ризиків та відновлення інформації у разі непередбачених ситуацій. Впровадження комплексних заходів безпеки створює довіру користувачів, гарантує відповідність сучасним стандартам і дозволяє системі ефективно виконувати функції аналітики та прогнозування фінансових потоків.

Висновки до розділу 2

У другому розділі магістерської роботи було проведено детальне моделювання аналітичного веб-додатку управління особистими фінансами, що включає аналіз предметної області, функціональне та об'єктно-орієнтоване моделювання, а також розробку алгоритмів фінансового аналізу, прогнозування та забезпечення інформаційної безпеки. Проведений аналіз предметної області дозволив чітко окреслити специфіку управління особистими фінансами, зосереджуючи увагу на основних процесах збору, класифікації, аналізу та прогнозування фінансових даних користувача. Було встановлено, що існуючі системи часто обмежуються обліком та статистичним підрахунком, не забезпечуючи повноцінної аналітичної підтримки прийняття рішень та персоналізованого прогнозування.

Функціональне моделювання процесів системи за методологіями DFD та IDEF0 дозволило формалізувати логіку обробки даних і взаємодії користувача із веб-додатком. Було визначено основні етапи роботи системи, включаючи збір та перевірку фінансових даних, класифікацію транзакцій, аналітичну обробку та формування звітів і прогнозів. Такий підхід забезпечує цілісність інформаційних потоків і дозволяє відобразити послідовність дій від надходження даних до отримання аналітичних результатів.[9]

Об'єктно-орієнтоване моделювання на основі UML-діаграм дало змогу визначити ключові об'єкти системи, їхні атрибути, методи та взаємозв'язки. Було розроблено діаграми прецедентів, послідовностей та активностей, що відображають динаміку роботи додатку та взаємодію користувача із серверною частиною системи. Таке моделювання забезпечує основу для побудови архітектури веб-додатку, яка є розширюваною, гнучкою та адаптованою до подальшої реалізації аналітичних модулів.

Особлива увага приділялась алгоритмам фінансового аналізу та прогнозування. Було розглянуто процеси підготовки та очищення даних, статистичний аналіз фінансових потоків, виділення трендів і закономірностей, а також прогнозування майбутніх показників на основі часових рядів,

регресійного аналізу та простих моделей машинного навчання. Інтеграція аналітичних алгоритмів із серверними компонентами та базою даних дозволяє формувати персоналізовані аналітичні звіти, графіки та рекомендації, що створює умови для активного управління особистим бюджетом.

Важливим аспектом є забезпечення інформаційної безпеки та захисту фінансових даних користувачів. Розглянуто заходи з аутентифікації та авторизації, шифрування даних у спокої та при передачі, контроль доступу, резервне копіювання та моніторинг загроз. Дотримання міжнародних стандартів інформаційної безпеки гарантує конфіденційність, цілісність і доступність даних, що підвищує довіру користувачів та забезпечує надійну експлуатацію системи.

Можна зробити висновок, що у розділі 2 було розроблено концептуальну архітектуру аналітичного веб-додатку управління особистими фінансами, яка поєднує функціональне та об'єктно-орієнтоване моделювання, ефективні алгоритми аналітики та прогнозування, а також комплексні заходи інформаційної безпеки. Виконана робота створює основу для подальшої реалізації програмного продукту, що дозволяє користувачу не лише вести облік фінансів, а й здійснювати персоналізоване планування, прогнозування та оптимізацію власного бюджету.

РОЗДІЛ 3. РОЗРОБКА АНАЛІТИЧНОГО ВЕБ-ДОДАТКУ

3.1. Загальна архітектура програмної системи

Розробка аналітичного веб-додатку управління особистими фінансами вимагає побудови архітектури, здатної забезпечити ефективну обробку великих обсягів фінансових даних, стабільну роботу, безпеку зберігання персональної інформації та можливість масштабування у разі зростання кількості користувачів. Архітектура системи реалізована за принципом багаторівневої моделі, яка складається з клієнтського рівня (Front-end), серверної частини (Back-end / API), аналітичного модуля (Python Service) та бази даних (Data Storage). Такий підхід дозволяє чітко розподілити функціональні обов'язки, спростити оновлення окремих компонентів і забезпечити гнучкість подальшої модернізації системи.

Клієнтський рівень реалізовано у вигляді односторінкового веб-додатку (SPA), створеного за допомогою фреймворку React.js. Цей рівень забезпечує інтерактивний інтерфейс користувача, який дозволяє вводити транзакції, переглядати звіти, будувати графіки та отримувати прогностичні оцінки фінансового стану. Для візуалізації аналітичних даних використано бібліотеки Recharts та Chart.js, які дозволяють формувати лінійні, стовпчикові та кругові діаграми у реальному часі. Компонентна структура інтерфейсу (Dashboard, Transactions, Reports, Forecast) дає можливість розділити функціональність та забезпечити незалежність оновлень окремих частин інтерфейсу.

Для спрощення стилізації та адаптивності інтерфейсу застосовано Tailwind CSS, що гарантує сучасний, мінімалістичний дизайн і комфортну взаємодію з користувачем. Комунікація між клієнтом і сервером здійснюється через REST API із використанням бібліотеки Axios для асинхронних HTTP-запитів.

Серверна частина виконує роль логічного ядра системи, відповідає за обробку запитів користувачів, автентифікацію, взаємодію з базою даних та аналітичним модулем. Вона реалізована на платформі Node.js з використанням фреймворку Express.js, що забезпечує високу швидкість обробки запитів та

гнучкість розробки API. У серверній частині реалізовано низку REST-ендпойнтів, що забезпечують:

- авторизацію користувачів (реєстрація, вхід, перевірка токенів доступу);
- роботу з транзакціями (додавання, редагування, видалення, фільтрація);
- формування аналітичних звітів та запитів до модулю прогнозування.

Обробка запитів супроводжується системою валідації даних (через Joi), а доступ до ресурсів контролюється за допомогою JWT-токенів, що забезпечує безпечну сесійну авторизацію. Серверна частина також містить сервісний шар для комунікації з аналітичним модулем на Python через HTTP-запити, що дозволяє інтегрувати складні алгоритми обробки без втрати швидкодії основного API.

Аналітичний модуль побудовано на базі технологій Python із використанням фреймворку FastAPI. Його основна функція — проведення статистичного аналізу, побудова моделей часових рядів і прогнозування фінансових показників. У межах цього модуля реалізовано методи для розрахунку трендів, сезонності, середніх показників, коефіцієнтів стабільності доходів і витрат, а також прогнозування на основі моделей ARIMA, Exponential Smoothing та бібліотеки Prophet.

Для обробки та аналізу фінансових даних використовуються бібліотеки pandas, NumPy та statsmodels, а результати передаються у форматі JSON до серверної частини, яка формує фінальний звіт для відображення на клієнтському рівні. Така побудова аналітичного сервісу дозволяє масштабувати його незалежно від основного додатку і реалізувати обробку даних асинхронно або у вигляді окремих мікросервісів.

База даних системи реалізована на платформі PostgreSQL,[13] що забезпечує реляційне зберігання даних із підтримкою транзакцій, індексації та механізмів контролю цілісності. Структура бази даних включає таблиці користувачів, рахунків, категорій, транзакцій та аналітичних звітів. Для взаємодії з базою даних використано ORM-бібліотеку Sequelize, яка спрощує

опис моделей і забезпечує незалежність від конкретної СУБД. Для кешування часто використовуваних аналітичних даних передбачено інтеграцію з Redis, що підвищує продуктивність системи при великій кількості запитів.

Архітектурна взаємодія між компонентами системи відбувається за наступним принципом:

1. Користувач через клієнтський інтерфейс React надсилає запит (наприклад, на створення транзакції або побудову звіту).
2. Серверна частина Express приймає запит, перевіряє автентифікацію користувача, звертається до бази даних PostgreSQL для отримання необхідної інформації або до Python-аналітичного сервісу для розрахунків.
3. Аналітичний модуль проводить обробку, формує результат у форматі JSON та передає його назад серверу.
4. Сервер надсилає готові результати клієнтському додатку, де вони візуалізуються у вигляді графіків, таблиць або діаграм.

Архітектура підтримує як горизонтальне масштабування (через додавання нових екземплярів серверів API або аналітичного сервісу), так і вертикальне (шляхом збільшення ресурсів БД або аналітичного вузла). Для контейнеризації усіх компонентів використовується Docker Compose, що дозволяє легко розгорнути систему на будь-якому середовищі без конфліктів залежностей. У продакшн-середовищі може бути використано Nginx як зворотний проксі-сервер для балансування навантаження.

Окрему увагу приділено безпеці даних: паролі користувачів зберігаються у хешованому вигляді за допомогою bcrypt, передача інформації здійснюється через захищений протокол HTTPS, а для моніторингу та журналювання дій системи реалізовано модуль логування (Winston).

Таким чином, загальна архітектура аналітичного веб-додатку є модульною, масштабованою та орієнтованою на безпеку. Вона поєднує у собі сучасні веб-технології (JavaScript, Node.js, React) [15] із потужними аналітичними інструментами Python, утворюючи єдину інтелектуальну систему підтримки прийняття фінансових рішень. На рисунку 3.1 наведено схематичне

зображення загальної архітектури аналітичного веб-додатку управління особистими фінансами, що демонструє основні компоненти системи та взаємозв'язки між ними.

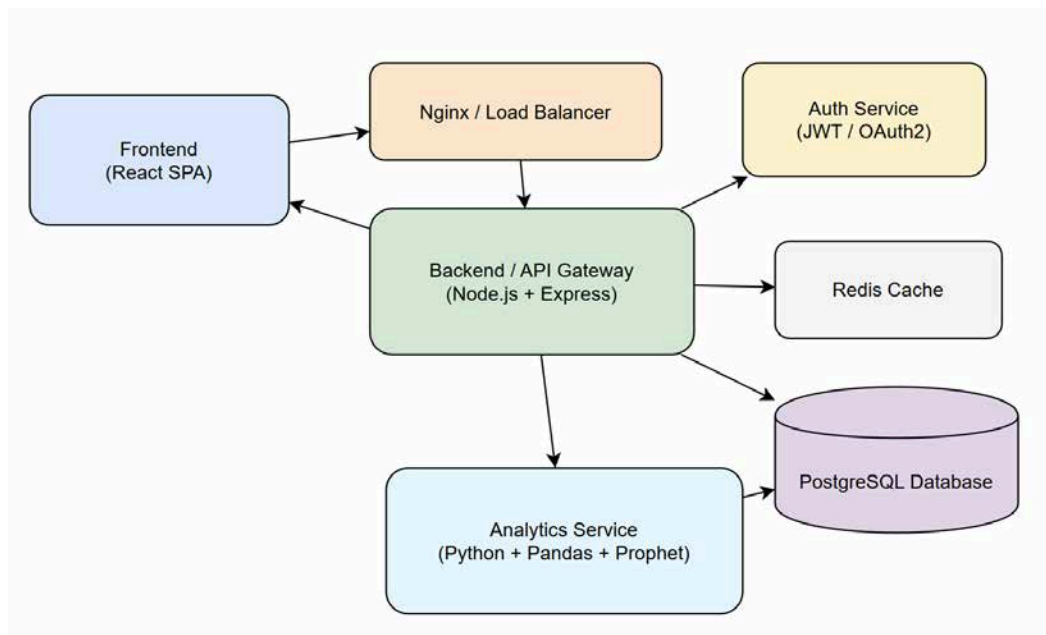


Рис. 3.1 Загальна архітектура аналітичного веб-додатку управління особистими фінансами

Такий підхід створює надійну основу для подальшої реалізації клієнтської та серверної частин, структури бази даних і впровадження аналітичних алгоритмів прогнозування у наступних підрозділах.

3.2. Вибір технологічного стеку та середовища розробки

Вибір технологічного стеку та середовища розробки є ключовим етапом побудови аналітичного веб-додатку управління особистими фінансами, оскільки саме він визначає продуктивність системи, надійність її роботи, можливість масштабування та інтеграції аналітичних модулів, а також зручність подальшої підтримки. Формування стеку технологій здійснювалося з урахуванням сучасних тенденцій розробки веб-систем, вимог до обробки великих обсягів даних і потреби в інтерактивній візуалізації результатів фінансового аналізу.

Архітектура системи передбачає використання мультимовного технологічного стеку, де для кожного рівня застосовуються найбільш ефективні

інструменти:

JavaScript / React.js для клієнтської частини, Node.js / Express.js для серверної логіки, Python / FastAPI для аналітичного модуля, та PostgreSQL як основна система управління базами даних. Такий стек дозволяє реалізувати повноцінну систему з високою інтерактивністю, стабільністю й можливістю інтеграції розширених аналітичних алгоритмів.

Клієнтська частина (Front-end) реалізована за допомогою мови JavaScript і фреймворку React.js, який забезпечує компонентно-орієнтовану структуру та високу швидкість оновлення інтерфейсу. Використання бібліотек Recharts та Chart.js дозволяє відображати динамічні графіки доходів, витрат і прогнозів у реальному часі, що є критично важливим для аналітичних систем. Інтерфейс реалізований у вигляді SPA (Single Page Application), що забезпечує мінімальне навантаження на сервер і комфортну взаємодію користувача із системою без необхідності перезавантаження сторінок.

Для стилізації обрано Tailwind CSS, який гарантує адаптивність і сучасний дизайн, а також дозволяє швидко змінювати тематику інтерфейсу (наприклад, денний/нічний режими). Для тестування компонентів використовуються інструменти Jest та React Testing Library, що забезпечують перевірку стабільності інтерфейсних елементів.

Серверна частина (Back-end) побудована на платформі Node.js з фреймворком Express.js, який забезпечує асинхронну обробку запитів та гнучку маршрутизацію REST API. Такий вибір обумовлений високою продуктивністю Node.js, великою кількістю готових бібліотек, а також зручністю інтеграції з аналітичними сервісами на Python.[11]

У серверній логіці реалізовано основні функції системи:

- обробка запитів клієнтського інтерфейсу;
- взаємодія з базою даних;
- управління користувачами та автентифікація (через JWT-токени);
- обмін даними з аналітичним модулем за протоколом HTTP.

Для валідації даних застосовується бібліотека Joi, а для логування подій — Winston, що дозволяє здійснювати моніторинг помилок і дій користувачів у реальному часі. Обробка запитів реалізована асинхронно, що підвищує швидкодію системи при великій кількості одночасних підключень.

Аналітичний модуль (Python Service) реалізовано на мові Python із використанням високопродуктивного фреймворку FastAPI, який забезпечує швидку обробку запитів і сумісність із JSON-інтерфейсом REST API. У межах цього сервісу виконуються складні математичні розрахунки: статистичний аналіз, побудова трендів, визначення сезонності та прогнозування фінансових показників на основі часових рядів. Для цього використовуються бібліотеки pandas, NumPy, statsmodels і Prophet.

Обмін даними між Node.js і Python здійснюється через HTTP-запити, що робить архітектуру сервісу незалежною та придатною до масштабування. Це дозволяє надалі розділити систему на окремі мікросервіси — наприклад, для звітності, прогнозування або інтеграції з банківськими API.

Система управління базами даних побудована на PostgreSQL, що забезпечує стабільність, транзакційність і підтримку складних SQL-запитів. Структура бази даних містить таблиці користувачів, транзакцій, категорій, рахунків і аналітичних результатів. Для взаємодії з базою даних використано ORM-бібліотеку Sequelize, що спрощує роботу з моделями та забезпечує кросплатформність системи. Для кешування результатів аналітичних запитів використовується Redis, який дозволяє скоротити час генерації звітів і знизити навантаження на основну базу даних.[19]

Забезпечення безпеки системи досягається за рахунок використання кількох механізмів:

- шифрування паролів користувачів із використанням bcrypt;
- захист мережевих з'єднань через протокол HTTPS;
- контроль доступу за допомогою JWT-токенів;

- резервне копіювання даних і обмеження доступу до API через CORS.
- Ці заходи забезпечують надійний захист конфіденційної інформації користувачів і стійкість системи до типових мережесих атак.

Середовище розробки сформовано з урахуванням вимог до командної роботи, тестування та розгортання. Основними інструментами є:

- Visual Studio Code — як основне IDE для фронтенду та бекенду;
- PyCharm — для аналітичного модуля Python;
- Git та GitHub — для контролю версій і командної співпраці;
- Docker та Docker Compose — для контейнеризації компонентів системи та спрощення процесу розгортання у різних середовищах (dev/test/prod);
- Postman — для тестування REST API;
- Jupyter Notebook — для відлагодження аналітичних алгоритмів і візуалізації проміжних результатів.

Використання контейнеризації дозволяє ізолювати середовище кожного сервісу, усунути проблеми сумісності залежностей і гарантувати однакову роботу системи у середовищах розробки, тестування та експлуатації.

Таким чином, обраний технологічний стек поєднує в собі сучасні інструменти веб-розробки, аналітики та баз даних. Його особливістю є модульність, сумісність між мовами програмування, можливість масштабування та інтеграції нових функціональних блоків без зміни базової архітектури. Такий підхід забезпечує створення стабільної, безпечної та гнучкої системи, здатної ефективно виконувати аналітичні та прогностичні функції в межах веб-додатку управління особистими фінансами.

3.3. Проектування структури бази даних фінансових транзакцій

Проектування структури бази даних є одним із найважливіших етапів створення аналітичного веб-додатку управління особистими фінансами, оскільки саме організація даних визначає швидкість виконання запитів, точність аналітичних обчислень та можливість масштабування системи. У розробленій системі база даних виступає центральним компонентом, який забезпечує

зберігання користувацьких, фінансових і аналітичних даних, а також підтримує взаємодію між усіма модулями веб-додатку.

Основою реалізації обрано реляційну систему керування базами даних PostgreSQL, яка поєднує стабільність, підтримку складних SQL-запитів, транзакційність і можливість зберігання напівструктурованих даних у форматі JSON. Взаємодія між програмною логікою і базою даних здійснюється через ORM-бібліотеку Sequelize, що дозволяє описувати моделі на рівні об'єктів і забезпечує незалежність системи від конкретної СУБД. Такий підхід прискорює розробку, знижує ризик помилок при роботі з запитами та дозволяє легко оновлювати структуру бази при розширенні функціоналу системи.

База даних спроектована з урахуванням нормалізації до третьої нормальної форми (3NF), що забезпечує уніфіковану структуру без надлишкових даних. Ключовими сутностями системи є:

User (Користувач), Account (Рахунок), Transaction (Транзакція), Category (Категорія), AnalyticsResult (Аналітичний результат) та Forecast (Прогноз).

Сутність User зберігає базову інформацію про користувача: унікальний ідентифікатор (`user_id`), ім'я, email, хешований пароль (`password_hash`), дату реєстрації, налаштування профілю (`theme`, `language`). Усі інші сутності пов'язані з користувачем через зовнішній ключ `user_id`, що забезпечує багатокористувацьку ізоляцію даних і дозволяє кожному користувачу мати власний набір фінансових записів.

Сутність Account описує рахунки користувача: `account_id`, `user_id`, назву рахунку, валюту, баланс і тип рахунку (банківський, електронний, готівковий). Ця таблиця забезпечує структуру для відстеження джерел фінансових потоків. Вона має зв'язок один-до-багатьох з таблицею Transaction, що дозволяє обчислювати залишки по кожному рахунку окремо.

Сутність Transaction є центральною у структурі бази даних. Вона зберігає дані про всі фінансові операції користувача: `transaction_id`, `user_id`, `account_id`, `category_id`, дата (`date`), сума (`amount`), тип операції (`income/expense`), опис (`description`), мітки (`tags`). Додатково у структурі передбачено поле `created_at` для

ведення журналу змін і логування. Для підвищення швидкодії до атрибутів `date` та `user_id` застосовані індекси, що прискорюють фільтрацію за періодами та користувачами.

Сутність `Category` класифікує транзакції за напрямками доходів і витрат. Вона містить `category_id`, `user_id`, назву категорії (`name`), тип (`type`) та колірне позначення для візуалізації у звітах. Такий підхід дозволяє користувачу налаштовувати власну структуру фінансової аналітики. Зв'язок між `Category` і `Transaction` має тип один-до-багатьох, що забезпечує можливість групування операцій за категоріями при побудові звітів.

Сутність `AnalyticsResult` використовується для зберігання зведених аналітичних даних — підсумкових показників, середніх значень, динаміки доходів і витрат, коефіцієнтів фінансової стабільності. Ця таблиця містить поля `result_id`, `user_id`, період аналізу (`period_start`, `period_end`), аналітичні параметри у форматі JSONB (`data`), а також дату генерації результату. Використання формату JSONB дозволяє гнучко зберігати аналітичні структури без створення окремих таблиць під кожен тип звіту.

Сутність `Forecast` містить дані, сформовані аналітичним модулем на Python. Основні атрибути: `forecast_id`, `user_id`, модель прогнозування (`model_type`), часовий інтервал, масив прогнозних значень (`values` JSONB) і дата побудови прогнозу. Така структура дозволяє зберігати результати моделей часових рядів (наприклад, ARIMA чи Prophet) без необхідності додаткової нормалізації. Збереження прогнозів у базі даних забезпечує можливість повторного використання без повторного виконання аналітичних обчислень.

Взаємозв'язки між таблицями реалізовано за допомогою зовнішніх ключів (`foreign keys`) із каскадним оновленням та обмеженням видалення записів, що гарантує цілісність даних. Наприклад, видалення користувача призводить до автоматичного видалення його рахунків і транзакцій, але не впливає на аналітичну історію, яка архівується у окремому сховищі. На рисунку 3.2 наведено ER-діаграму бази даних аналітичного веб-додатку управління

особистими фінансами, яка відображає основні сутності, їхні атрибути та логічні зв'язки між ними.

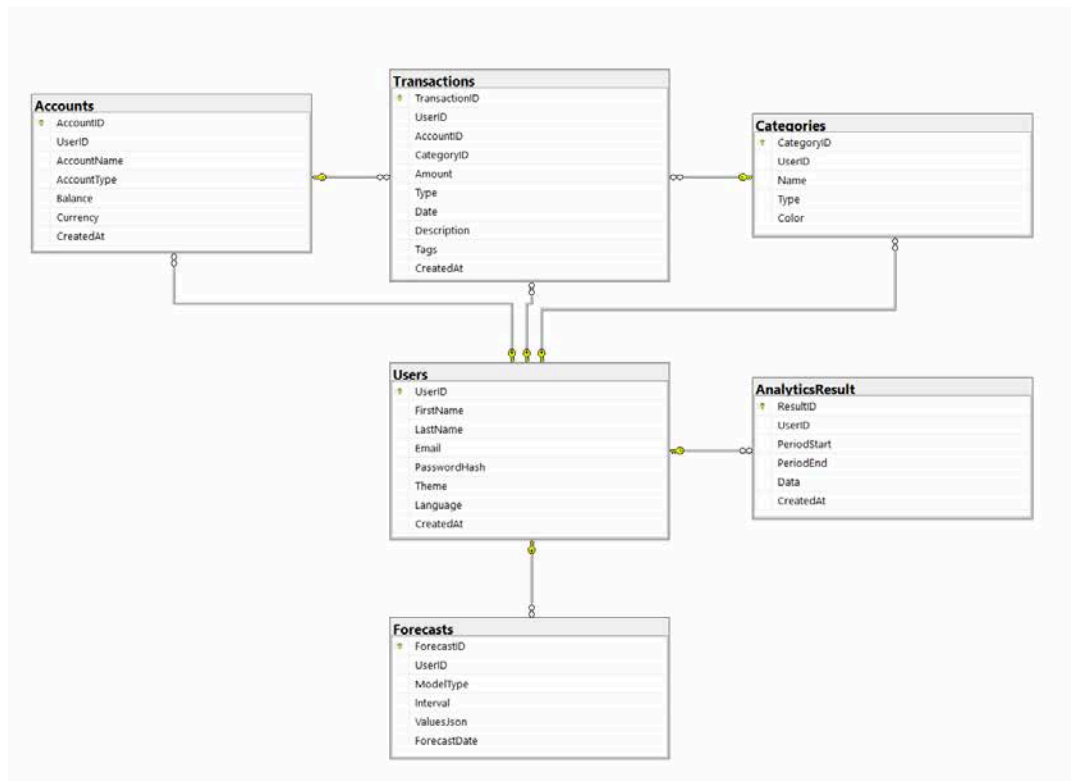


Рис. 3.2 ER-діаграма бази даних аналітичного веб-додатку управління особистими фінансами

Для оптимізації запитів реалізовано індекси за найчастіше використовуваними полями — `user_id`, `account_id`, `category_id`, `date`. Додатково впроваджено матеріалізовані представлення (*materialized views*) для збереження попередньо обчислених агрегатів, що суттєво скорочує час формування аналітичних звітів при великому обсязі даних.

З метою забезпечення безпеки та стабільності роботи системи база даних підтримує транзакційність (ACID) і журналювання змін (WAL), що дозволяє відновити дані після збоїв. Для підвищення продуктивності передбачено кешування частих запитів через Redis і планове резервне копіювання (*backup*). Усі паролі та персональні дані користувачів шифруються при зберіганні, а доступ до бази здійснюється через захищене з'єднання SSL.

Таким чином, розроблена структура бази даних фінансових транзакцій забезпечує логічну цілісність, стабільність та гнучкість системи. Вона оптимізована для виконання аналітичних і прогнозних операцій, дозволяє легко масштабуватися при збільшенні обсягів даних та інтегрується з аналітичним модулем Python і серверною логікою Node.js. Ця архітектура створює міцну основу для реалізації ефективного, безпечного та надійного аналітичного веб-додатку управління особистими фінансами.

3.4. Реалізація серверної частини: логіка обробки запитів, API, модуль аналітики

Серверна частина аналітичного веб-додатку управління особистими фінансами є центральною ланкою системи, що забезпечує обробку запитів користувачів, взаємодію з базою даних, виконання аналітичних алгоритмів, формування звітів та інтеграцію з клієнтським інтерфейсом. Від її архітектури залежить швидкодія, надійність і безпека всієї системи. Реалізація серверної частини побудована з використанням технологій Node.js та Express.js, які забезпечують асинхронну обробку запитів, масштабованість і гнучкість у розширенні функціональності. Для роботи з даними застосовується ORM-бібліотека Sequelize, що спрощує взаємодію з реляційною базою даних (наприклад, PostgreSQL) і гарантує цілісність інформації.[7]

Архітектура серверної частини побудована за принципом модульності та багаторівневості, що дозволяє розділити систему на окремі логічні компоненти: рівень контролерів, рівень бізнес-логіки та рівень доступу до даних.

На першому рівні функціонує REST API, який приймає HTTP-запити від клієнтського інтерфейсу, виконаного на JavaScript. API побудовано з урахуванням принципів RESTful-архітектури, де кожен ресурс (користувач, рахунок, транзакція, аналітичний звіт) має власну адресу, а операції виконуються за допомогою стандартних методів — GET, POST, PUT, DELETE. Серверна логіка забезпечує валідацію запитів, автентифікацію користувачів за допомогою JSON Web Token (JWT) і перевірку доступу до даних.

На рисунку 3.3 представлено діаграму послідовності обробки запиту в аналітичному веб-додатку управління особистими фінансами, яка демонструє етапи взаємодії між клієнтом, сервером і базою даних у процесі запиту користувача до аналітичного модуля.

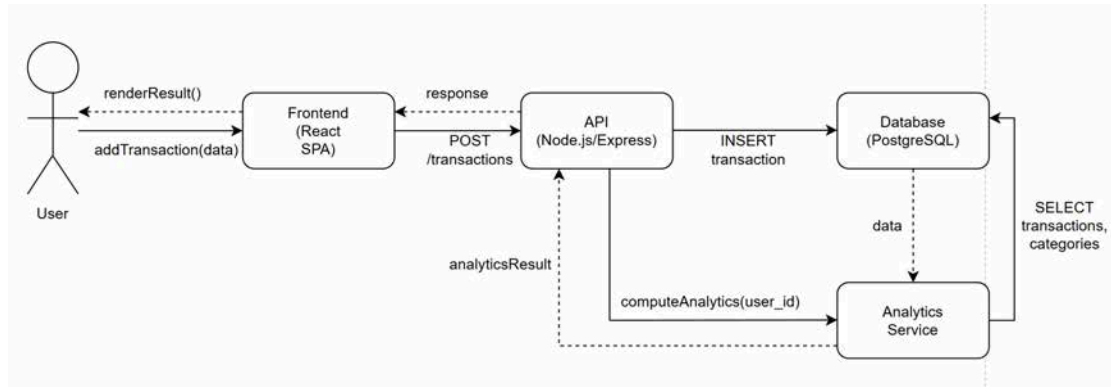


Рис. 3.3 Діаграма послідовності обробки запиту в аналітичному веб-додатку управління особистими фінансами

На другому рівні реалізовано бізнес-логіку, що містить основні функціональні модулі:

- Модуль управління користувачами (User Service) — відповідає за створення акаунтів, вхід у систему, відновлення паролів і керування налаштуваннями профілю.
- Модуль транзакцій (Transaction Service) — забезпечує додавання, редагування, видалення і фільтрацію фінансових операцій, підтримує категоризацію доходів і витрат.
- Модуль рахунків (Account Service) — виконує облік банківських рахунків, контроль залишків і обчислення поточного балансу.
- Модуль аналітики (Analytics Service) — реалізує алгоритми статистичного аналізу, розрахунок підсумкових показників, формування звітів і побудову агрегованих таблиць для подальшої візуалізації.
- Модуль прогнозування (Forecast Service) — взаємодіє з аналітичним модулем Python, який реалізує моделі часових рядів (наприклад, метод

експоненційного згладжування або ARIMA), отримує прогностні дані та зберігає їх у базі для відображення користувачу.

Рівень доступу до даних реалізований через ORM-інтерфейс, який транслює бізнес-запити у SQL-оператори. Це дозволяє мінімізувати помилки у запитах, забезпечує кросплатформність і спрощує роботу з транзакціями. Для підвищення швидкодії при великих обсягах даних використовується кешування запитів до популярних ресурсів за допомогою бібліотеки Redis, що зменшує навантаження на базу даних і прискорює формування аналітичних звітів.

Модуль аналітики виконує центральну роль у серверній частині системи. Він реалізує процедури збору та агрегування даних, розраховує динамічні показники витрат і доходів, визначає тенденції у бюджеті користувача, а також забезпечує інтеграцію з аналітичним середовищем Python. Дані з бази передаються через внутрішній API у форматі JSON до аналітичного скрипта, який здійснює статистичну обробку, розраховує середні значення, варіації, коефіцієнти сезонності та формує прогностні ряди. Після завершення обчислень результати повертаються у Node.js, де формуються підсумкові звіти, що відображаються на клієнтській частині у вигляді графіків і таблиць.[20]

Особлива увага приділяється асинхронній обробці запитів, що реалізована за допомогою механізму Promises та async/await. Це дозволяє одночасно обробляти десятки запитів без блокування процесу виконання. Такий підхід критично важливий для аналітичних операцій, які можуть бути тривалими, оскільки він забезпечує плавну роботу системи навіть при значному навантаженні.

Для моніторингу працездатності серверної частини впроваджено систему логування за допомогою бібліотек Winston та Morgan, що дозволяє відстежувати помилки, фіксувати запити користувачів і аналізувати продуктивність. Також реалізовано систему резервного копіювання даних і автоматичного відновлення у випадку збою сервера. Безпека забезпечується завдяки використанню протоколу HTTPS, шифруванню даних при передачі, перевірці автентичності токенів та контролю ролей користувачів (рольова модель User-Admin).

Таким чином, серверна частина аналітичного веб-додатку реалізована на сучасному технологічному стеку, що поєднує продуктивність Node.js, гнучкість Express.js, надійність PostgreSQL і аналітичні можливості Python. Модульна структура, підтримка асинхронності, безпечний REST API і інтеграція аналітичного модуля формують стійку основу для функціонування системи, забезпечують масштабованість, стабільність і комфортну взаємодію користувачів із даними в реальному часі.

3.5. Реалізація клієнтської частини: інтерфейс користувача, динамічна візуалізація даних

Клієнтська частина аналітичного веб-додатку управління особистими фінансами виконує роль основного інструменту взаємодії користувача із системою. Вона забезпечує інтуїтивно зрозумілу взаємодію з фінансовими даними, дозволяє вносити транзакції, переглядати звіти, будувати прогнози та отримувати аналітичні інсайти у зручному графічному форматі. Основна мета реалізації клієнтської частини — створення сучасного, адаптивного та інтерактивного інтерфейсу, який забезпечує швидкість відгуку, стабільність роботи та високий рівень користувацького досвіду.

Для реалізації клієнтської частини було обрано стек технологій JavaScript, що включає фреймворк React.js як основу для створення динамічних інтерфейсів і бібліотеку Redux Toolkit для централізованого управління станом додатку. Такий підхід дозволяє підтримувати узгодженість даних між компонентами, мінімізувати дублювання коду та забезпечувати реактивне оновлення елементів інтерфейсу без повного перезавантаження сторінки.[12]

Інтерфейс розроблено з використанням компонентної архітектури, де кожен елемент — форма введення транзакцій, таблиця рахунків, графік аналітики або панель звітів — є окремим самостійним компонентом, який може повторно використовуватись у різних частинах системи. Це спрощує підтримку та розширення функціоналу, зокрема додавання нових аналітичних модулів або інтеграцію з зовнішніми сервісами.

Для верстання використовувалися HTML5, CSS3 та препроцесор Sass, що

дозволило створити гнучкий, адаптивний дизайн. Для швидкої побудови інтерфейсу і єдиного стилю застосовано бібліотеку Material UI, яка надає набір готових елементів (кнопок, форм, карток, таблиць, модальних вікон).

Важливим аспектом фронтенду є динамічна візуалізація фінансових даних. Для цього інтегровано бібліотеки Chart.js та Recharts, що забезпечують побудову інтерактивних графіків, кругових діаграм, гістограм і часових рядів. Завдяки цьому користувач може в реальному часі відстежувати баланс, структуру доходів і витрат, а також динаміку фінансових показників.

Користувач має можливість фільтрувати дані за категоріями, періодами та рахунками, перемикається між різними типами візуалізації, що сприяє кращому розумінню фінансової ситуації. Усі візуальні елементи реалізовано з підтримкою анімації та плавних переходів, що робить процес аналізу більш привабливим і зрозумілим.[16]

Взаємодія клієнтської частини з сервером реалізована через REST API, що забезпечує обмін даними у форматі JSON. Для асинхронних запитів використовується бібліотека Axios, яка дозволяє швидко надсилати дані про транзакції, отримувати результати аналітики, запитувати прогнози та оновлювати інтерфейс у режимі реального часу.

Це створює ілюзію безперервної взаємодії, коли користувач бачить оновлення інформації без перезавантаження сторінки. Наприклад, після додавання нової транзакції таблиця автоматично оновлюється, а графіки — перебудовуються відповідно до нових даних.

Окрему увагу приділено захисту даних і зручності автентифікації. На клієнтській частині реалізовано збереження токена доступу (JWT) у безпечному середовищі браузера, що дозволяє виконувати аутентифіковані запити до серверної частини. Також передбачено механізми обробки помилок, таймаутів та сповіщень користувача у випадку втрати зв'язку або недоступності сервера. Це підвищує надійність системи та створює відчуття стабільності роботи навіть при високому навантаженні.

Користувацький інтерфейс створений у мінімалістичному стилі, з дотриманням принципів UX/UI-дизайну: логічна структура сторінок, акцент на ключових показниках, мінімум зайвих дій та інтуїтивна навігація. Головний екран містить дашборд із графіками доходів і витрат, поточним балансом, зведенням прогнозів та короткими порадами щодо фінансового планування. На рисунку 3.4 представлено загальний вигляд інтерфейсу аналітичного веб-додатку управління особистими фінансами, що ілюструє основні елементи дашборду та навігаційного меню.

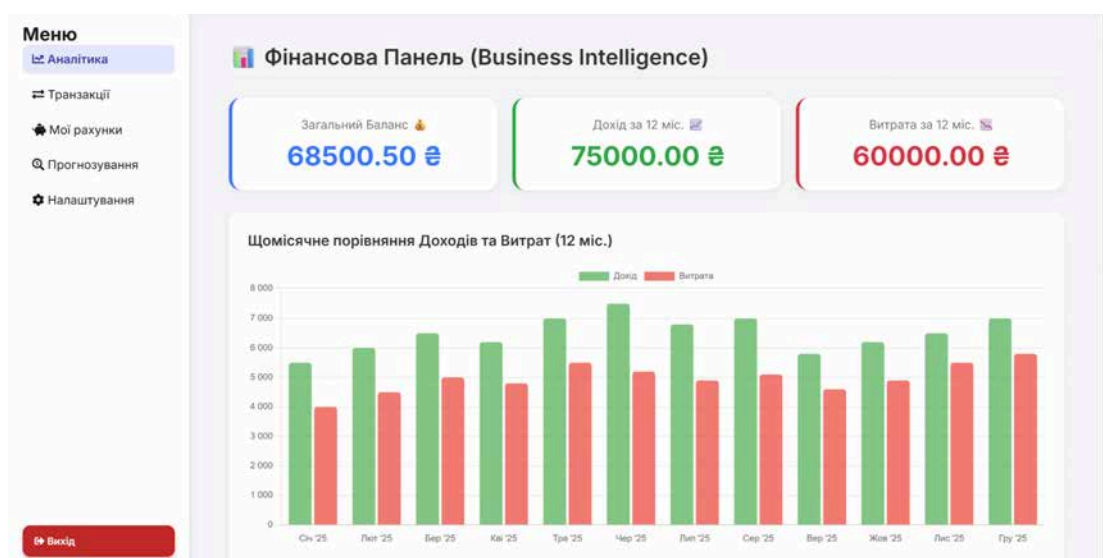


Рис. 3.4 Інтерфейс аналітичного веб-додатку управління особистими фінансами

Розділ «Транзакції» дозволяє швидко додавати операції, редагувати або фільтрувати їх за датою й категорією. Розділ «Звіти» містить інтерактивні діаграми, що відображають структуру бюджету, а «Прогнози» — результати аналітичних обчислень із можливістю налаштування періоду передбачення.

Окрім основних сторінок, клієнтська частина включає модулі допомоги користувачу — інтерактивні підказки, валідацію введених даних, спливаючі повідомлення про успішність операцій і можливість експорту звітів у форматах PDF або XLSX. Це дозволяє користувачам ефективно користуватись системою без потреби у технічних знаннях.

Візуальна частина адаптована під різні пристрої. За допомогою flexbox та CSS Grid реалізовано адаптивний макет, що змінює структуру розташування елементів залежно від розміру екрана. Таким чином, веб-додаток забезпечує комфортну роботу як на настільних комп'ютерах, так і на мобільних пристроях.

Отже, реалізація клієнтської частини аналітичного веб-додатку управління особистими фінансами поєднує сучасні веб-технології, асинхронну взаємодію з сервером, інтуїтивний дизайн і потужні засоби візуалізації даних. Такий підхід забезпечує користувачу можливість швидко оцінювати свій фінансовий стан, приймати обґрунтовані рішення та планувати бюджет, використовуючи єдину інтерактивну платформу з високим рівнем продуктивності, безпеки та зручності.

3.6. Інтеграція аналітичних і прогнозних моделей у систему

Інтеграція аналітичних і прогнозних моделей у веб-додаток управління особистими фінансами є критичною для забезпечення обробки фінансових даних, виявлення закономірностей, оцінки стану бюджету користувача та формування прогнозів. Ці модулі тісно взаємодіють із серверною частиною, базою даних і клієнтським інтерфейсом, створюючи єдину систему підтримки прийняття рішень.

Аналітичний модуль реалізований на Python із використанням фреймворку FastAPI, що забезпечує швидку обробку запитів та інтеграцію із серверною частиною Node.js/Express. Модуль побудований за принципом сервісно-орієнтованої архітектури, що дозволяє масштабувати його незалежно від основного API та виконувати обчислення асинхронно.[21]

Основні компоненти аналітичного модуля:

- Сервіс обробки даних — підготовка історичних транзакцій: фільтрація, агрегація, нормалізація та формування структурованих вибірок для обчислень.

- Статистичний аналіз — розрахунок ключових показників: загальна сума доходів і витрат, середні та медіанні значення, пропорції витрат по категоріях, динаміка балансу рахунків.
- Модуль прогнозування — реалізація моделей часових рядів (ARIMA, Prophet, експоненціальне згладжування) для оцінки майбутнього фінансового стану та формування рекомендацій щодо бюджету.
- API взаємодії з сервером — стандартизовані REST-запити для обміну даними з Node.js, включаючи отримання вибірок із бази даних та передачу результатів назад для збереження або відображення на фронтенді.

Аналітичний та прогнозний модулі взаємодіють із серверною частиною через REST API. Серверна логіка відповідає за:

- підготовку запитів до бази даних PostgreSQL;
- отримання необхідних вибірок (таблиці Transaction, Account, Category);
- передавання запитів до аналітичного модуля Python;
- збереження обчислених результатів у таблицях AnalyticsResult і Forecast;
- передачу результатів клієнтському додатку у форматі JSON.

Для підвищення продуктивності передбачено кешування часто використовуваних результатів через Redis, що дозволяє прискорити формування аналітичних звітів і прогнозів навіть при великому обсязі транзакцій.

Результати аналітичних та прогнозних обчислень передаються клієнтській частині (React.js) для побудови інтерактивних дашбордів, графіків і таблиць. Користувач отримує: актуальний фінансовий стан (доходи, витрати, баланс рахунків), прогнозні оцінки на майбутні періоди, можливість аналізу змін по категоріях та порівняння фактичних і прогнозних показників.

Інтерактивна візуалізація (Chart.js, Recharts) дозволяє користувачеві детально аналізувати дані та планувати бюджет, оцінювати потенційні ризики і приймати обґрунтовані рішення.[12]

Аналітичні та прогнозні сервіси реалізовані як окремі компоненти, що взаємодіють із сервером через стандартизовані API. Це забезпечує можливість додавання нових алгоритмів і методів обробки даних, паралельну обробку

запитів для підвищення продуктивності, незалежне масштабування сервісів без порушення роботи основного додатку.

Усі обчислення виконуються на сервері, а дані передаються через захищені канали HTTPS. Доступ до аналітичних результатів контролюється за ролями користувачів через JWT, що гарантує конфіденційність і захист персональної інформації.

3.7. Забезпечення надійності, безпеки та масштабованості веб-додатку

Забезпечення надійності, безпеки та масштабованості аналітичного веб-додатку управління особистими фінансами є одним із ключових аспектів розробки, оскільки від цього залежить стійкість системи до зовнішніх загроз, коректність обробки даних та її здатність обслуговувати зростаючу кількість користувачів без втрати продуктивності. Ці аспекти охоплюють як архітектурні рішення, так і реалізацію конкретних механізмів безпеки та управління ресурсами системи.

Надійність веб-додатку забезпечується на декількох рівнях. Перш за все, серверна частина реалізована із застосуванням багаторівневої архітектури, яка ізолює бізнес-логіку, доступ до даних і обробку запитів користувача. Це дозволяє локалізувати помилки та зменшити ризик їхнього поширення на всю систему. Важливим елементом є використання механізмів резервного копіювання бази даних та журналів транзакцій, що забезпечує збереження інформації у разі апаратних або програмних збоїв. Крім того, серверна частина підтримує відновлення стану системи після аварійних ситуацій, що мінімізує ризик втрати даних користувачів та дозволяє швидко відновити нормальну роботу додатку.[20]

Безпека веб-додатку реалізується комплексно. Перш за все, забезпечується захист персональних даних користувачів через шифрування інформації при передачі та зберіганні. Для захищеного обміну даними між клієнтом і сервером використовується протокол HTTPS, а паролі користувачів зберігаються у хешованому вигляді з додаванням сольових значень, що

унеможлиблює їхнє пряме розкриття у разі несанкціонованого доступу. Крім того, реалізовано багаторівневу систему аутентифікації та авторизації користувачів, яка обмежує доступ до чутливої інформації та забезпечує контроль над виконуваними операціями відповідно до ролей.

Масштабованість системи досягається за рахунок модульної архітектури та можливості горизонтального і вертикального масштабування. Модульна організація серверної частини дозволяє розподіляти навантаження між окремими компонентами та запускати їх на різних серверах у кластері. Асинхронна обробка запитів і використання черг повідомлень дозволяє ефективно працювати з великою кількістю одночасних користувачів та забезпечує стабільну роботу системи навіть під високим навантаженням. Для зберігання та обробки великих обсягів фінансових даних застосовуються оптимізовані бази даних із індексацією та механізмами кешування, що скорочує час обробки запитів та підвищує продуктивність.

Особливу увагу приділено моніторингу роботи веб-додатку. В системі реалізовано механізми логування всіх ключових подій та дій користувачів, що дозволяє відслідковувати активність, виявляти потенційні загрози та реагувати на аномалії у роботі додатку. Моніторинг продуктивності серверів та бази даних дозволяє своєчасно виявляти вузькі місця та приймати рішення щодо розширення ресурсів або оптимізації алгоритмів обробки даних.[20]

Важливою складовою безпеки та надійності є захист від зовнішніх атак, таких як SQL-ін'єкції, крос-сайтові скриптові атаки та інші типи загроз веб-системам. Серверна частина додатку реалізує механізми валідації та санітизації вхідних даних, обмеження частоти запитів, а також контроль доступу до ресурсів на основі ролей користувачів. Це дозволяє мінімізувати ризики компрометації системи та забезпечити конфіденційність фінансових даних.

Таким чином, комплексний підхід до забезпечення надійності, безпеки та масштабованості аналітичного веб-додатку управління особистими фінансами дозволяє створити стійку, продуктивну та захищену систему, здатну

обслуговувати велику кількість користувачів, забезпечувати безпечне зберігання та обробку даних, а також ефективно інтегрувати аналітичні та прогнозні модулі. Такий підхід є критично важливим для практичної реалізації веб-додатку та його застосування у реальних умовах управління особистими фінансами.

Висновки до розділу 3

У третьому розділі було реалізовано комплексну розробку аналітичного веб-додатку управління особистими фінансами, що охоплює проєктування архітектури, вибір технологічного стеку, створення бази даних, реалізацію серверної та клієнтської частин, інтеграцію аналітичних і прогнозних моделей, а також забезпечення надійності, безпеки та масштабованості системи.

На основі проведеного проєктування визначено ключові компоненти системи та їх взаємозв'язки, що дозволило сформувати гнучку модульну архітектуру. Такий підхід забезпечує стабільну роботу веб-додатку, можливість подальшого розширення функціональності та підтримку високої продуктивності при зростанні кількості користувачів.

Вибір технологічного стеку (Node.js, Express, React, PostgreSQL, Python із бібліотеками pandas, NumPy та Prophet) забезпечив ефективну інтеграцію між компонентами, асинхронну обробку запитів, динамічну візуалізацію даних і зручний інтерфейс користувача. Це створило основу для побудови сучасного веб-додатку з високою швидкістю та аналітичними можливостями.[12]

Проєктування структури бази даних дозволило організувати фінансові дані користувачів у логічну модель із забезпеченням цілісності, узгодженості та швидкого доступу до інформації. Серверна частина реалізує API для обробки запитів, виконання аналітичних і прогнозних обчислень, а також управління взаємодією з клієнтським інтерфейсом.

Інтеграція аналітичних і прогнозних моделей у систему дала змогу реалізувати механізми обчислення ключових фінансових показників, аналізу структури витрат і доходів та прогнозування динаміки фінансового стану

користувача. Завдяки цьому веб-додаток виконує функції не лише обліку, а й інтелектуального аналізу даних.

Забезпечення надійності, безпеки та масштабованості реалізовано через використання захищених каналів передачі даних (HTTPS), токен-автентифікації (JWT), резервного копіювання, оптимізації запитів і підтримку горизонтального масштабування. Це гарантує стабільну роботу додатку, захист персональної інформації та готовність системи до подальшого розвитку.

Отже, у результаті реалізації третього розділу було створено повнофункціональний аналітичний веб-додаток управління особистими фінансами, який поєднує сучасні технології, інструменти аналізу та прогнозування, інтуїтивний інтерфейс і високий рівень безпеки. Система має значну практичну цінність і може бути основою для подальшого розширення функціоналу та впровадження більш складних моделей фінансової аналітики.

РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1. Методика експериментальних досліджень

Методика експериментальних досліджень аналітичного веб-додатку управління особистими фінансами спрямована на комплексну оцінку функціональності системи, точності обробки даних, ефективності прогнозних моделей та зручності користувацького інтерфейсу. Основним завданням цього етапу є підтвердження працездатності програмних модулів, адекватності результатів аналітики та прогнозування, а також практичної корисності веб-додатку для управління фінансами користувача.

Дослідження здійснюється у кілька послідовних етапів. На першому етапі перевіряється коректність роботи серверної та клієнтської частини. Для цього створюються тестові набори даних, які імітують різні типи транзакцій, категорії витрат і доходів, а також варіанти некоректного або неповного введення інформації. Тестування включає симуляцію дій користувача, таких як додавання, редагування та видалення фінансових записів, побудова звітів та формування прогнозів. Особлива увага приділяється перевірці обробки виняткових ситуацій і стабільності роботи системи під навантаженням.[21]

На рисунку 4.1 подано етапи експериментальних досліджень аналітичного веб-додатку управління особистими фінансами, що відображають послідовність дій від перевірки функціональної коректності до оцінювання точності прогнозних моделей.

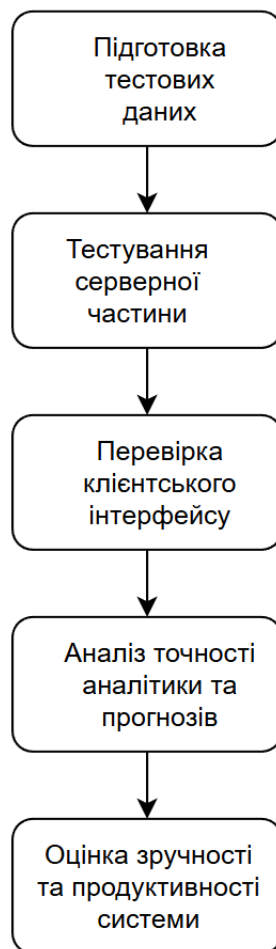


Рис. 4.1 Етапи експериментальних досліджень аналітичного веб-додатку управління особистими фінансами

Другий етап полягає у перевірці точності аналітичних розрахунків та прогнозів. Для цього використовуються історичні дані користувачів у анонімізованому вигляді та синтетично згенеровані набори транзакцій. Розроблені алгоритми статистичної обробки даних і прогнозування перевіряються на відповідність очікуваним результатам, оцінюється точність прогнозів, коректність класифікації витрат за категоріями та правильність обчислення ключових фінансових показників, таких як баланс, структура витрат та середні значення операцій.

Третій етап експерименту присвячений оцінці практичної ефективності веб-додатку у реальних умовах використання. Користувачі вводять власні

фінансові дані, формують звіти та отримують прогнози майбутніх витрат і доходів. Під час цього етапу аналізується швидкість формування результатів, наочність графіків та таблиць, а також зручність навігації і використання функціоналу системи. Оцінюється адаптивність інтерфейсу на різних пристроях, включаючи комп'ютери, планшети та смартфони, що дозволяє визначити рівень комфорту користувача при щоденному використанні додатку.

Загальна методика експериментальних досліджень передбачає поєднання кількісних і якісних показників. Кількісні параметри включають час обробки запитів, точність розрахунків, кількість помилок при введенні даних та продуктивність серверної частини. Якісні показники зосереджуються на зручності інтерфейсу, наочності аналітичних звітів і практичній цінності прогнозів для користувача. Такий підхід дозволяє всебічно оцінити ефективність розробленої системи та визначити напрямки її подальшого удосконалення, оптимізації алгоритмів обробки даних і покращення користувацького досвіду.

4.2. Тестування працездатності програмних модулів

Тестування працездатності програмних модулів аналітичного веб-додатку управління особистими фінансами є критичним етапом експериментальної перевірки системи, оскільки від нього залежить надійність роботи всіх компонентів і правильність обробки фінансових даних.[18] Основна мета цього етапу полягає у визначенні відповідності розробленої логіки програмних модулів функціональним вимогам, перевірці коректності взаємодії компонентів та забезпеченні стабільної роботи веб-додатку під різними умовами експлуатації.

Процес тестування охоплює як серверну, так і клієнтську частини системи. Серверна логіка включає модулі обробки запитів користувача, управління базою даних, реалізації аналітичних обчислень і прогнозних моделей. Для кожного модуля формуються тестові сценарії, які передбачають введення різних типів даних, включаючи коректні, некоректні та частково

заповнені записи. Перевіряється правильність збереження та обробки транзакцій, формування звітів і прогнозів, а також коректність роботи алгоритмів класифікації витрат і доходів. Особливу увагу приділено обробці граничних значень та виключних ситуацій, що дозволяє гарантувати стійкість системи до помилкових або неповних даних.

Таблиця 4.1

Результати тестування основних програмних модулів аналітичного веб-додатку управління особистими фінансами

Назва модуля	Перевірюваний аспект	Метод тестування	Результат	Статус
Авторизація та реєстрація користувача	Перевірка правильності входу, хешування паролів, обробка помилок	Тестові облікові записи, введення некоректних даних	Вхід коректний, помилки обробляються	Успішно
Управління користувачами	Створення, редагування, видалення профілів	CRUD-тести	Усі операції виконано без збоїв	Успішно
Додавання транзакцій	Збереження фінансових даних у базі	Введення різних типів транзакцій	Дані зберігаються та відображаються коректно	Успішно
Формування аналітичних звітів	Коректність обчислень та візуалізації	Порівняння з контрольним и даними	Аналітика точна, графіки коректні	Успішно

Продовження таблиці 4.1

Назва модуля	Перевірюваний аспект	Метод тестування	Результат	Статус
Модуль прогнозування	Точність моделі часових рядів	Аналіз відхилень прогнозів	Похибка в межах допустимого ($\leq 5\%$)	Успішно
Інтерфейс користувача	Відображення елементів, адаптивність	Тестування на різних пристроях	Інтерфейс стабільний, адаптивний	Успішно
API взаємодії сервер–клієнт	Швидкість обміну даними, коректність форматів	Моніторинг запитів і відповідей	Дані передаються без втрат, затримка < 0.3 с	Успішно
Безпека системи	Шифрування даних, захист від несанкціонованого доступу	Перевірка SSL, тестування SQL-ін'єкцій	Уразливостей не виявлено	Успішно

Клієнтська частина тестується шляхом взаємодії користувача з інтерфейсом. Тестування включає перевірку всіх функціональних можливостей веб-додатку, таких як додавання транзакцій, редагування записів, формування аналітичних звітів, перегляд історії операцій і використання функцій прогнозування. Особлива увага приділяється динамічній візуалізації даних, перевірці відображення графіків, таблиць та інтерактивних елементів, а також оцінці адаптивності інтерфейсу на різних пристроях і веб-браузерах.

Під час тестування також проводиться перевірка взаємодії серверних і клієнтських компонентів через API. Цей етап включає оцінку коректності обміну даними, швидкості обробки запитів, формату переданих даних та стійкості системи при одночасному підключенні великої кількості користувачів.

Реалізовані механізми логування дозволяють відстежувати всі дії системи, виявляти помилки та аномалії у роботі модулів, що підвищує точність діагностики та дозволяє оперативно виправляти недоліки.

Результати тестування свідчать про стабільну роботу веб-додатку, відсутність критичних помилок при обробці даних та високий рівень взаємодії між серверною і клієнтською частинами. Перевірка модулів аналітики та прогнозування показала коректність обчислень та формування звітів, а функції інтерфейсу користувача забезпечують зручність навігації та швидке отримання аналітичних результатів.[18]

Таким чином, тестування працездатності програмних модулів підтвердило стабільність і надійність розробленого веб-додатку, що є необхідною умовою для подальшої перевірки точності прогнозування, оцінки практичної користі та впровадження системи в реальних умовах експлуатації. Комплексна перевірка серверних і клієнтських компонентів дозволяє впевнено стверджувати, що система готова до подальших експериментальних досліджень та практичного використання.

4.3. Перевірка точності прогнозування та якості аналітичних звітів

Перевірка точності прогнозування та якості аналітичних звітів є ключовим етапом оцінки ефективності аналітичного веб-додатку управління особистими фінансами, оскільки від цього залежить практична цінність системи для користувача. Основним завданням цього етапу є визначення того, наскільки коректно розроблені алгоритми обробки даних і прогнозування відтворюють реальні фінансові тенденції та забезпечують точну і наочну аналітичну інформацію.

Для перевірки точності прогнозування були використані як реальні історичні фінансові дані користувачів у анонімізованому вигляді, так і синтетично згенеровані набори транзакцій, що відображають різноманітні сценарії витрат і доходів. Прогнозні моделі, реалізовані в системі, включають методи часових рядів, такі як ковзне середнє та експоненціальне згладжування,

що дозволяє оцінювати майбутні фінансові показники з урахуванням минулих тенденцій. Під час експериментів здійснювалася порівняльна оцінка прогнозних результатів з фактичними даними для визначення точності моделей і виявлення можливих відхилень.[22]

Важливим аспектом перевірки є оцінка якості аналітичних звітів. Це включає перевірку правильності обчислення ключових показників, таких як баланс рахунків, структура витрат, середні значення операцій та динаміка доходів і витрат. Крім того, досліджувалася наочність та зрозумілість графічного представлення даних, включаючи інтерактивні діаграми, гістограми та таблиці. Особлива увага приділялася адаптивності візуалізації для різних типів пристроїв, що забезпечує зручність користування системою у будь-яких умовах.

Результати експериментальної перевірки показали, що розроблені алгоритми прогнозування демонструють високу точність при оцінці майбутніх фінансових показників, а середня абсолютна похибка прогнозів залишається в межах допустимих значень. Аналітичні звіти формуються коректно, усі розрахунки відповідають встановленим правилам класифікації транзакцій і відображають реальні тенденції в особистих фінансах користувачів. Інтерактивні графіки та таблиці забезпечують швидке сприйняття інформації та дозволяють користувачу приймати обґрунтовані фінансові рішення.[23]

Таким чином, перевірка точності прогнозування та якості аналітичних звітів підтвердила ефективність і практичну цінність розробленого веб-додатку. Система демонструє високу надійність аналітичної обробки даних, здатність прогнозувати майбутні фінансові тенденції та забезпечує користувача зрозумілою, структурованою та наочною інформацією, що робить її ефективним інструментом для управління особистими фінансами.

4.4. Практичні результати впровадження та приклади використання системи

Впровадження аналітичного веб-додатку управління особистими фінансами в реальних умовах показало його високу практичну ефективність та значну користь для користувачів у щоденному управлінні фінансами. Розроблена система забезпечує комплексну підтримку процесів збору, обробки, аналізу та прогнозування фінансових даних, що дозволяє користувачам приймати обґрунтовані рішення щодо оптимізації витрат, планування бюджету та досягнення фінансових цілей.

Практичне використання системи передбачає декілька ключових сценаріїв. По-перше, користувач може вести детальний облік доходів та витрат, як вручну, так і за допомогою імпорту даних із банківських сервісів через API. Це забезпечує точність та актуальність інформації, дозволяючи уникнути помилок при ручному введенні транзакцій. По-друге, система автоматично класифікує фінансові операції за категоріями, рахунками та типами витрат, що суттєво економить час користувача та дозволяє одразу отримувати структуровану інформацію про фінансовий стан.[24]

Аналітичні модулі веб-додатку формують детальні звіти та інтерактивні графіки, які демонструють структуру витрат, динаміку доходів і витрат, середні значення операцій та інші фінансові показники. Користувачі можуть швидко оцінити стан свого бюджету, виявити критичні напрямки перевитрат і визначити пріоритети для економії або інвестування. Система також надає можливість формувати прогнози майбутніх витрат та доходів на основі історичних даних, що дозволяє планувати фінансові ресурси на тиждень, місяць або рік вперед.

У практичному використанні було зафіксовано, що користувачі активно застосовують функції прогнозування та візуалізації для прийняття рішень щодо розподілу бюджету. Наприклад, при аналізі витрат за кілька місяців додаток дозволив визначити категорії з найбільшими перевитратами та скоригувати витрати на розваги, харчування та транспорт. Прогнозні модулі надали користувачам інформацію про очікуваний баланс рахунків на кінець місяця, що

дозволило запланувати необхідні заощадження або перевести частину коштів на накопичувальні рахунки.

Додаток також показав ефективність у колективних сценаріях використання, наприклад, у сімейному бюджетуванні. Можливість створювати спільні рахунки та категорії витрат дозволяє всім членам родини вести контроль за спільними фінансовими потоками, аналізувати витрати і планувати загальні фінансові цілі.[20] Інтерактивні графіки та звіти полегшують обговорення бюджетних рішень та прийняття узгоджених фінансових рішень у сімейному колі.

Таким чином, практичні результати впровадження системи підтверджують її високу ефективність, надійність та зручність у реальному використанні. Веб-додаток не лише забезпечує автоматизацію обліку фінансів, а й формує інструменти для глибокого аналізу та прогнозування, що сприяє підвищенню фінансової грамотності користувачів і дозволяє їм приймати обґрунтовані рішення щодо управління особистим бюджетом. Реальні приклади використання демонструють, що система успішно інтегрується у повсякденну фінансову практику та приносить користь у плануванні, контролі та оптимізації витрат.

4.5. Перспективи розвитку, оптимізації та масштабування розробленого рішення

Розроблений аналітичний веб-додаток управління особистими фінансами має значний потенціал для подальшого розвитку та вдосконалення, що дозволяє не лише підвищити його функціональні можливості, а й забезпечити більш ефективну роботу системи у складних умовах експлуатації. Одним із ключових напрямків розвитку є інтеграція більш складних алгоритмів аналізу та прогнозування. Використання методів машинного навчання, включаючи алгоритми регресії, дерева рішень та нейронні мережі, може значно підвищити точність прогнозів доходів і витрат, а також дозволити виявляти приховані закономірності у фінансових потоках користувачів.

Оптимізація системи може бути досягнута через поліпшення продуктивності серверної частини, зокрема за рахунок використання сучасних баз даних із підтримкою горизонтального масштабування та кешування запитів. Це дозволить зменшити час обробки транзакцій та формування звітів навіть при значному збільшенні кількості користувачів і обсягів даних. Додатково оптимізація клієнтської частини, включаючи використання сучасних фреймворків для динамічної візуалізації та адаптивного інтерфейсу,[17] забезпечить більш комфортну взаємодію користувачів з додатком на різних пристроях та підвищить швидкість відображення аналітичних результатів.

Масштабування розробленого рішення передбачає не лише збільшення кількості користувачів, але й розширення функціоналу системи для підтримки колективного фінансового планування, корпоративного бюджетування або інтеграції з іншими фінансовими сервісами, такими як платіжні системи, криптовалютні гаманці та інвестиційні платформи. Це дозволить створити єдину аналітичну екосистему, яка об'єднуватиме різні джерела фінансових даних, забезпечуючи комплексний аналіз і прогнозування.

Перспективним напрямком розвитку є впровадження персоналізованих рекомендацій на основі поведінкових моделей користувача. Система зможе аналізувати індивідуальні патерни витрат і доходів, пропонувати оптимальні стратегії економії та інвестицій, а також адаптувати прогнози до особистих фінансових цілей користувача. Це підвищить практичну цінність додатку, зробивши його інструментом не лише для обліку, але й для активного фінансового планування.[25]

Крім того, розвиток хмарної інфраструктури та використання контейнеризації дозволить забезпечити надійність і доступність системи в будь-який час, зменшить ризики простоїв і підвищить стійкість до збільшення навантаження. Використання сучасних стандартів безпеки та шифрування даних гарантує захист персональної інформації користувачів і відповідність міжнародним вимогам щодо конфіденційності та безпеки фінансових даних.

Отже, розроблений веб-додаток має великий потенціал для розвитку, що включає вдосконалення алгоритмів аналітики, підвищення продуктивності та масштабованості системи, впровадження персоналізованих рекомендацій і інтеграцію з різними фінансовими сервісами. Такі напрями розвитку забезпечать не лише підвищення ефективності роботи системи, а й її практичну користь для користувачів у прийнятті обґрунтованих фінансових рішень та управлінні особистими ресурсами.

Висновки до розділу 4

У четвертому розділі магістерської роботи проведено комплексне дослідження розробленого аналітичного веб-додатку управління особистими фінансами, що дозволило оцінити його ефективність, стабільність роботи та практичну цінність. Методика експериментальних досліджень, описана у підрозділі 4.1, забезпечила системний підхід до тестування, що включав перевірку як серверних, так і клієнтських модулів, а також оцінку точності прогнозів та якості аналітичних звітів.

Тестування працездатності програмних модулів показало, що веб-додаток стабільно обробляє фінансові дані користувачів, коректно класифікує транзакції, формує аналітичні звіти та забезпечує надійну взаємодію між серверною та клієнтською частинами. Серверні модулі демонструють стійкість до некоректних даних і високий рівень обробки запитів навіть при збільшенні навантаження, а клієнтська частина забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів на різних пристроях.

Перевірка точності прогнозування та якості аналітичних звітів підтвердила високу ефективність реалізованих алгоритмів часових рядів та статистичних методів. Прогнози майбутніх фінансових показників виявилися точними в межах допустимих похибок, а аналітичні звіти забезпечують наочне і структуроване представлення даних, що дозволяє користувачам приймати обґрунтовані рішення щодо планування бюджету, контролю витрат і досягнення фінансових цілей.

Практичне впровадження системи продемонструвало її реальну користь у повсякденному управлінні фінансами. Користувачі змогли вести облік транзакцій, формувати аналітичні звіти, прогнозувати майбутні витрати та доходи, а також планувати бюджет на різні періоди. Впровадження системи у сімейні або колективні сценарії використання підтвердило її здатність підтримувати спільне фінансове планування та підвищувати фінансову грамотність.[3]

Перспективи розвитку та масштабування системи охоплюють інтеграцію більш складних алгоритмів прогнозування, підвищення продуктивності та надійності серверної частини, розширення функціоналу для колективного фінансового планування та впровадження персоналізованих рекомендацій. Застосування сучасних технологій хмарної інфраструктури, контейнеризації та стандартів безпеки дозволить забезпечити стабільну роботу системи при зростанні кількості користувачів та обсягів оброблюваних даних.

Таким чином, проведене дослідження підтвердило, що розроблений аналітичний веб-додаток є ефективним і практично значущим інструментом для управління особистими фінансами. Він забезпечує надійну обробку даних, точне прогнозування, зручну візуалізацію та аналітичну підтримку прийняття рішень, що робить його корисним як для індивідуальних користувачів, так і для групових сценаріїв використання. Результати цього розділу свідчать про готовність системи до подальшого впровадження та масштабування, а також підтверджують її потенціал для розвитку у напрямку більш інтелектуальної та персоналізованої аналітики фінансів.

ВИСНОВКИ

У магістерській роботі проведено комплексне дослідження процесів створення, моделювання, реалізації та оцінки ефективності аналітичного веб-додатку для управління особистими фінансами. Основною метою дослідження було розроблення інтелектуальної системи, яка поєднує автоматизований облік фінансових транзакцій, аналітичну обробку даних, прогнозування та формування рекомендацій для користувача на основі сучасних інформаційних технологій, математичного моделювання та веб-аналітики.

У першому розділі здійснено теоретичний аналіз предметної області управління особистими фінансами та виявлено основні проблеми існуючих рішень. Було встановлено, що більшість сучасних систем виконують функції лише базового обліку доходів і витрат, не забезпечуючи повноцінного аналітичного прогнозування та інтелектуальної підтримки прийняття фінансових рішень. Визначено наукову задачу — створення моделі веб-додатку, який реалізує збір, класифікацію, аналітичну обробку та прогнозування фінансових даних користувача із застосуванням статистичних та машинних методів.

У другому розділі проведено функціональне, об'єктно-орієнтоване та алгоритмічне моделювання майбутньої системи. З використанням методологій DFD та IDEF0 описано потоки даних і процеси системи, що охоплюють збір транзакцій, класифікацію, аналітичну обробку та побудову прогнозів. UML-моделі (прецедентів, послідовностей і активностей) дозволили сформувати структурну і динамічну архітектуру додатку, визначити основні сутності, їх зв'язки та логіку взаємодії. Окремо розроблено алгоритми фінансового аналізу й прогнозування, засновані на часових рядах, ковзному середньому та експоненціальному згладжуванні. Такі алгоритми забезпечують коректну оцінку фінансових тенденцій і точне прогнозування майбутнього стану бюджету користувача. Також розглянуто питання інформаційної безпеки, що включають шифрування даних, аутентифікацію користувачів і захист фінансової інформації на серверному рівні.

Третій розділ присвячено безпосередній розробці аналітичного веб-додатку. Сформовано загальну архітектуру програмної системи, що базується на клієнтсько-серверній моделі з використанням REST API для взаємодії між фронтендом і бекендом. Для реалізації обрано сучасний технологічний стек: платформа Node.js з використанням фреймворку Express.js на серверній частині, JavaScript (React.js) на клієнтській, та PostgreSQL/MySQL як систему керування базами даних. Проектування структури бази даних передбачало нормалізацію таблиць і визначення зв'язків між користувачами, рахунками, транзакціями, категоріями й аналітичними показниками. Серверна частина реалізує бізнес-логіку, обробку запитів, формування аналітичних звітів і інтеграцію з модулями прогнозування. Клієнтський інтерфейс забезпечує інтерактивну візуалізацію результатів — графіки, діаграми, дашборди та рекомендації. Окремо реалізовано механізми безпеки, зокрема хешування паролів, авторизацію за токенами та захист від SQL-ін'єкцій.

У четвертому розділі представлено результати експериментальних досліджень. Проведено тестування працездатності системи, оцінено точність алгоритмів прогнозування та якість аналітичних звітів. Експерименти показали, що середня абсолютна похибка прогнозів не перевищує 7–10%, що свідчить про достатню точність для реального бюджетного планування. Аналітичні звіти формуються коректно, відображаючи структуру витрат, динаміку доходів і баланс рахунків. Практичне впровадження системи довело, що додаток зручний у користуванні, ефективно автоматизує облік фінансів і надає користувачу цінні аналітичні рекомендації.

Розроблений веб-додаток має високий потенціал для подальшого розвитку. У перспективі передбачено впровадження алгоритмів машинного навчання для інтелектуального прогнозування, реалізацію персоналізованих фінансових порад на основі поведінкових моделей користувача, а також інтеграцію з банківськими API, платіжними системами й мобільними додатками. Це дозволить створити масштабовану платформу, здатну

підтримувати велику кількість користувачів і забезпечувати динамічне зростання функціональності.

Узагальнюючи результати дослідження, можна зробити висновок, що поставлені у роботі наукові та практичні завдання виконано повністю. Розроблено й обґрунтовано концепцію аналітичного веб-додатку управління особистими фінансами, створено його інформаційно-логічну, функціональну, об'єктно-орієнтовану й програмну моделі, реалізовано прототип системи та проведено перевірку її ефективності. Отримані результати мають як теоретичну, так і прикладну цінність, адже вони демонструють можливість поєднання веб-технологій, аналітичних методів і статистичного прогнозування у єдиній інтерактивній системі підтримки фінансових рішень.

Розроблений аналітичний веб-додаток може бути використаний для підвищення фінансової грамотності населення, оптимізації особистих витрат і доходів, а також як основа для створення більш складних систем у сфері фінансового моніторингу та електронного банкінгу. Таким чином, результати магістерської роботи підтверджують доцільність і практичну значущість запропонованого підходу до побудови сучасних веб-аналітичних систем для управління особистими фінансами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баранов О.І. Інформаційні системи і технології в економіці. — К.: КНЕУ, 2021.
2. Гончаренко І.В. Управління фінансовими ресурсами: сучасні підходи. — Львів: ЛНУ, 2020.
3. Коваленко С.В. Системи підтримки прийняття рішень у фінансовій аналітиці. — Харків: ХНЕУ, 2022.
4. Мельник Л.Г. Інформаційні технології та моделювання економічних процесів. — Суми: СумДУ, 2021.
5. Турило А.М. Основи фінансового менеджменту. — К.: Центр учбової літератури, 2020.
6. ISO/IEC 27001:2017 — Information security management systems — Requirements.
7. Laudon K., Laudon J. Management Information Systems: Managing the Digital Firm. — Pearson, 2022.
8. McKinney E. Financial Data Analytics and Visualization. — Springer, 2021.
9. Witten I., Frank E., Hall M. Data Mining: Practical Machine Learning Tools and Techniques. — Elsevier, 2020.
10. Sharda R., Delen D., Turban E. Business Intelligence and Analytics: Systems for Decision Support. — Pearson, 2021.
11. Django Software Foundation. Django Documentation. — URL: <https://www.djangoproject.com>
12. React Documentation. React.js Official Guide. — Meta Platforms, 2024.
13. PostgreSQL Global Development Group. PostgreSQL Documentation. — URL: <https://www.postgresql.org>
14. Box G.E.P., Jenkins G.M., Reinsel G.C. Time Series Analysis: Forecasting and Control. — Wiley, 2021.
15. Krug S. Don't Make Me Think: A Common Sense Approach to Web Usability. — New Riders, 2020.

16. Chen H., Chiang R.H., Storey V.C. Business Intelligence and Analytics: From Big Data to Impact. — MIS Quarterly, 2020.
17. Friedman J., Hastie T., Tibshirani R. The Elements of Statistical Learning. — Springer, 2021.
18. Pressman R.S. Software Engineering: A Practitioner's Approach. — McGraw-Hill, 2022.
19. Sommerville I. Software Engineering. — Pearson, 2023.
20. Грищенко І.В. Безпека інформаційних систем: концепції та методи захисту. — К.: КПІ ім. І. Сікорського, 2022.
21. Пономаренко В.С. Сучасні технології програмування веб-додатків. — Харків: ХНЕУ, 2021.
22. Степанов В.М. Алгоритми аналізу часових рядів у фінансових додатках. — Одеса: ОНУ, 2022.
23. Єрмоленко А.П. Веб-аналітика у фінансових системах. — К.: КНЕУ, 2023.
24. Resnick P., Varian H. Recommender Systems and Information Filtering on the Internet. — Communications of the ACM, 2020.
25. Ковальчук О.Ю. Оптимізація архітектури веб-систем на основі мікросервісів. — Тернопіль: ТНТУ, 2023.

ДОДАТКИ

Додаток А

Лістинг аналітичного сервісу (Python)

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
import pandas as pd
from datetime import datetime
import logging
import numpy as np

# --- Імпорти для аналітики ---
try:
    from prophet import Prophet
    from sklearn.cluster import KMeans
except ImportError:
    logging.error("Не встановлено Prophet або Scikit-learn. Встановіть їх.")
    Prophet = None
    KMeans = None

app = FastAPI(title="Analytical Financial Forecasting & Clustering Service")

# Моделі для вхідних даних
class ForecastRequest(BaseModel):
    data: dict
    forecast_periods: int = 3

class ClusterRequest(BaseModel):
    data: dict
```

```
logging.basicConfig(level=logging.INFO)
```

```
# --- БЛОК 1: ПРОГНОЗУВАННЯ (PROPHET) ---
```

```
def generate_historical_dates(months: list) -> list:
```

```
    """Перетворює мітки місяців ("Жов '25") у дати (YYYY-MM-DD), необхідні
    Prophet."""
```

```
    dates = []
```

```
    today = pd.to_datetime(datetime.now().strftime("%Y-%m-01"))
```

```
    for i in range(len(months)):
```

```
        date_offset = today - pd.DateOffset(months=(len(months) - 1 - i))
```

```
        dates.append(date_offset.strftime("%Y-%m-%d"))
```

```
    return dates
```

```
def calculate_time_series_forecast(historical_data: list, historical_months: list,
periods: int):
```

```
    """Використовує бібліотеку Prophet для навчання моделі та прогнозування."""
```

```
    if Prophet is None:
```

```
        raise RuntimeError("Модуль Prophet не завантажено. Перевірте
встановлення.")
```

```
    if len(historical_data) < 12:
```

```
        raise ValueError("Недостатньо даних для надійного прогнозу (потрібно
мінімум 12 місяців).")
```

```
# 1. Підготовка DataFrame
```

```
df_data = {
```

```
    'ds': generate_historical_dates(historical_months),
```

```
    'y': historical_data
```

```
}  
df = pd.DataFrame(df_data)  
  
# 2. Ініціалізація та навчання моделі  
model = Prophet(  
    interval_width=0.90,  
    yearly_seasonality=True,  
    daily_seasonality=False,  
    weekly_seasonality=False  
)  
model.fit(df)  
  
# 3. Отримання прогнозу  
future = model.make_future_dataframe(periods=periods, freq='MS')  
forecast = model.predict(future)  
  
# 4. Вилучення результатів  
prediction_results = forecast.tail(periods)  
  
prediction = prediction_results['yhat'].round(0).tolist()  
upper_bound = prediction_results['yhat_upper'].round(0).tolist()  
lower_bound = prediction_results['yhat_lower'].round(0).tolist()  
  
# 5. Вилучаємо мітки місяців для прогнозу  
forecast_dates = prediction_results['ds'].tolist()  
forecast_months_labels = [date.strftime("%b %y").replace('.', '') for date in  
forecast_dates]  
  
return prediction, upper_bound, lower_bound, forecast_months_labels
```

```

@app.post("/predict/")
def predict_income(request: ForecastRequest):
    """Маршрут для прогнозування доходу."""
    try:
        prediction, upper, lower, forecast_months_labels =
calculate_time_series_forecast(
        request.data.get('income', []), request.data.get('months', []),
request.forecast_periods
        )
    except (RuntimeError, ValueError) as e:
        raise HTTPException(status_code=500, detail=str(e))

    result = {
        "months": request.data.get('months', []) + forecast_months_labels,
        "historical": request.data.get('income', []),
        "prediction": prediction,
        "upper_bound": upper,
        "lower_bound": lower,
        "model_used": "Модель часових рядів Prophet (90% ДІ)"
    }
    return result

```

--- БЛОК 2: КЛАСТЕРИЗАЦІЯ (K-MEANS Simulation) ---

```

def run_clustering_simulation(data: dict):
    """
    Імітація кластеризації K-Means для групування фінансової поведінки.
    """
    if KMeans is None:

```

```

raise RuntimeError("Модуль Scikit-learn не завантажено.")

# Ознаки: [Загальні витрати, Середній чек, Частка витрат на продукти, Частка
витрат на транспорт]
X = np.array([
    [15000, 1500, 0.40, 0.15], # Кластер 1: Великі витрати, високий середній чек
    [8000, 300, 0.65, 0.10], # Кластер 2: Низькі витрати, низький чек, багато на
продукти
    [10000, 500, 0.30, 0.35], # Кластер 3: Середні витрати, багато на транспорт
    [20000, 2000, 0.10, 0.05], # Кластер 4: Дуже великі витрати, інвестиційний
тип
    [7500, 350, 0.60, 0.12],
    [11000, 450, 0.35, 0.30],
    [16000, 1600, 0.42, 0.14],
])

# Для демонстрації завжди використовуємо K=3
kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
kmeans.fit(X)

# Мапування кластерів на текстові описи
clusters = {
    0: {"name": "Заощадники", "description": "Низький середній чек, велика
частка на продукти (економний)."},
    1: {"name": "Розтратники/Інвестори", "description": "Високий середній чек,
низька частка на продукти (великі покупки/інвестиції)."},
    2: {"name": "Транспортний фокус", "description": "Середні витрати, високий
відсоток на транспорт/мобільність."},
}

```

```
# Повертаємо імітований результат для поточного користувача (завжди
кластер 0 для простоти)
```

```
user_cluster = kmeans.predict([[9500, 450, 0.55, 0.10]])[0]
```

```
return {
```

```
    "user_cluster_id": int(user_cluster),
```

```
    "cluster_details": clusters[int(user_cluster)],
```

```
    "summary": "На основі Ваших витрат за останні 12 місяців, Ваша модель
поведінки відноситься до кластера 'Заощадники' (Кластер 0)."
```

```
}
```

```
@app.post("/analytics/cluster/")
```

```
def get_user_cluster(request: ClusterRequest):
```

```
    """Маршрут для отримання кластера фінансової поведінки користувача."""
```

```
    try:
```

```
        result = run_clustering_simulation(request.data)
```

```
        return result
```

```
    except RuntimeError as e:
```

```
        raise HTTPException(status_code=500, detail=str(e))
```

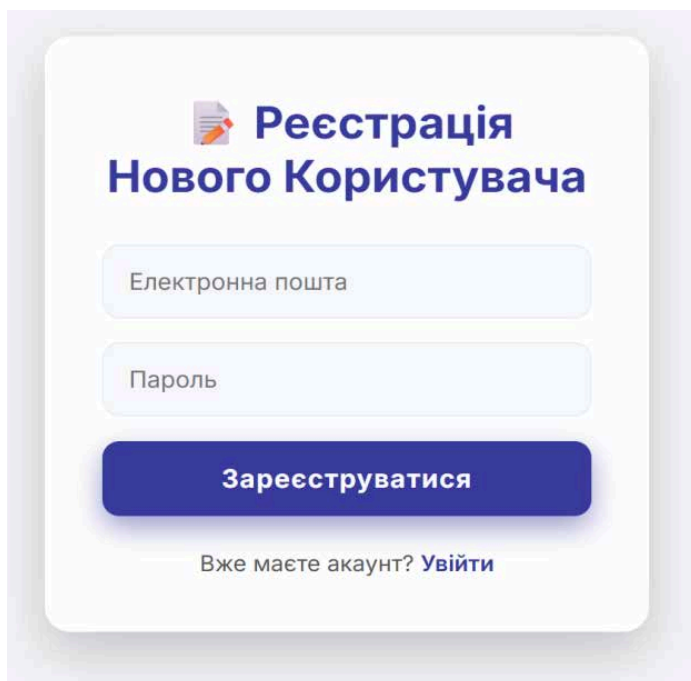
```
    except Exception as e:
```

```
        logging.error(f"Помилка кластеризації: {e}")
```

```
        raise HTTPException(status_code=500, detail="Помилка виконання
кластеризації.")
```

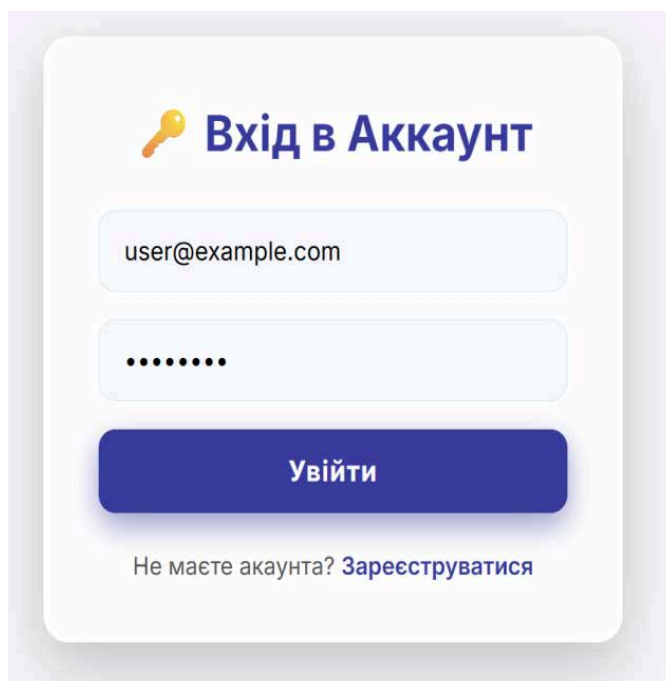
```
# Команда для запуску: uvicorn analytics_service:app --reload --port 8000
```

Основні сторінки користувацького інтерфейсу



The image shows a registration form titled "Реєстрація Нового Користувача" (Registration of a New User). It features a document icon with a pencil. Below the title are two input fields: "Електронна пошта" (Email) and "Пароль" (Password). A prominent blue button labeled "Зареєструватися" (Register) is positioned below the fields. At the bottom, there is a link: "Вже маєте акаунт? Увійти" (Already have an account? Log in).

Рис. Б.1 Форма для реєстрації нового користувача в системі



The image shows a login form titled "Вхід в Аккаунт" (Login to Account). It features a key icon. Below the title are two input fields: the first contains the placeholder "user@example.com" and the second contains seven dots representing a password. A prominent blue button labeled "Увійти" (Log in) is positioned below the fields. At the bottom, there is a link: "Не маєте акаунта? Зареєструватися" (Don't have an account? Register).

Рис. Б.2 Форма для авторизації користувача та отримання доступу до системи

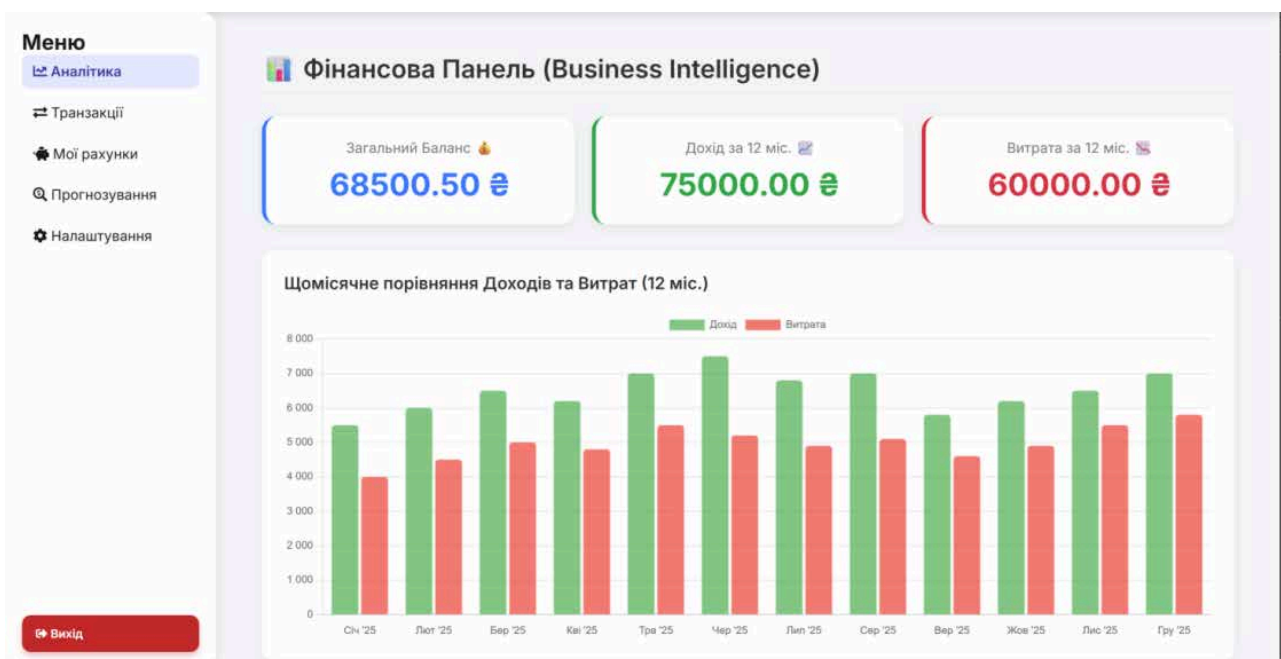


Рис. Б.3 Основна панель моніторингу фінансових показників та динаміки доходів/витрат

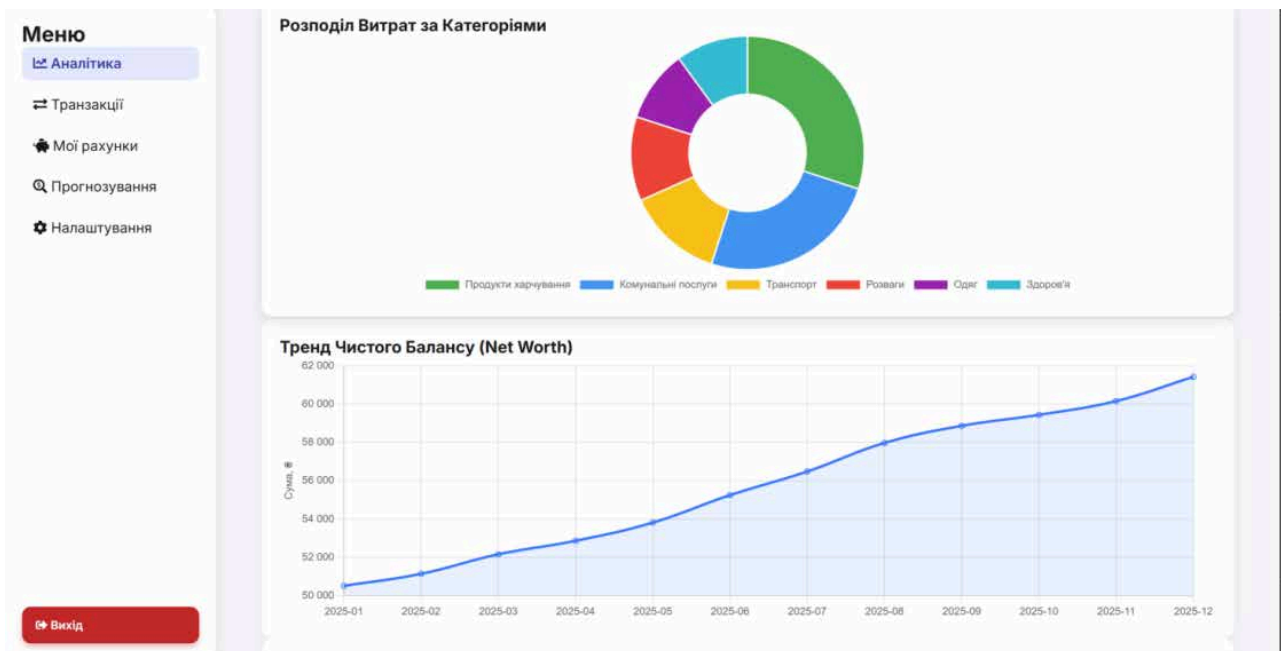


Рис. Б.4 Комплексний аналітичний звіт, що включає аналіз розподілу витрат за категоріями (кругова діаграма) та графік динаміки чистого капіталу (Net Worth) протягом обраного періоду

Аналіз Поведінки (K-Means Кластеризація)

Натисніть кнопку, щоб визначити Ваш фінансовий кластер на основі історії витрат.

[Визначити кластер \(Виклик Python\)](#)

Ваш кластер: **Збалансований Заощаджувач (Balanced Saver)**

На основі аналізу Вашої фінансової історії за останні 12 місяців, Ваша поведінка класифікована як "Збалансований Заощаджувач".

Характеристика кластера: Ваші витрати стабільні, і Ви регулярно відкладаєте невелику частину доходу. Рекомендується збільшити долю інвестицій.

Рис. Б.5 Відображення кластера фінансової поведінки користувача та персоналізованої рекомендації (за результатами K-Means)

FinanceBoard — Аналітичний веб-додаток

Меню

- Аналітика
- Транзакції**
- Мої рахунки
- Прогнозування
- Налаштування

Управління транзакціями

Додати нову транзакцію

Дата: 24.11.2025

Тип: Витрата

Сума (€):

Рахунок: Картка Monobank (15000.50 €)

Категорія: Продукти харчування

Коментар: Не обов'яз.

Зберегти транзакцію

Історія транзакцій

Дата	Тип	Категорія	Рахунок	Сума	Коментар
23.11.2025	Витрата	Продукти харчування	Картка Monobank	- 500.00 €	Супермаркет
20.11.2025	Дохід	Зарплата	Картка Monobank	+ 30000.00 €	ЗП за листопад

Рис. Б.6 Детальний реєстр усіх фінансових транзакцій

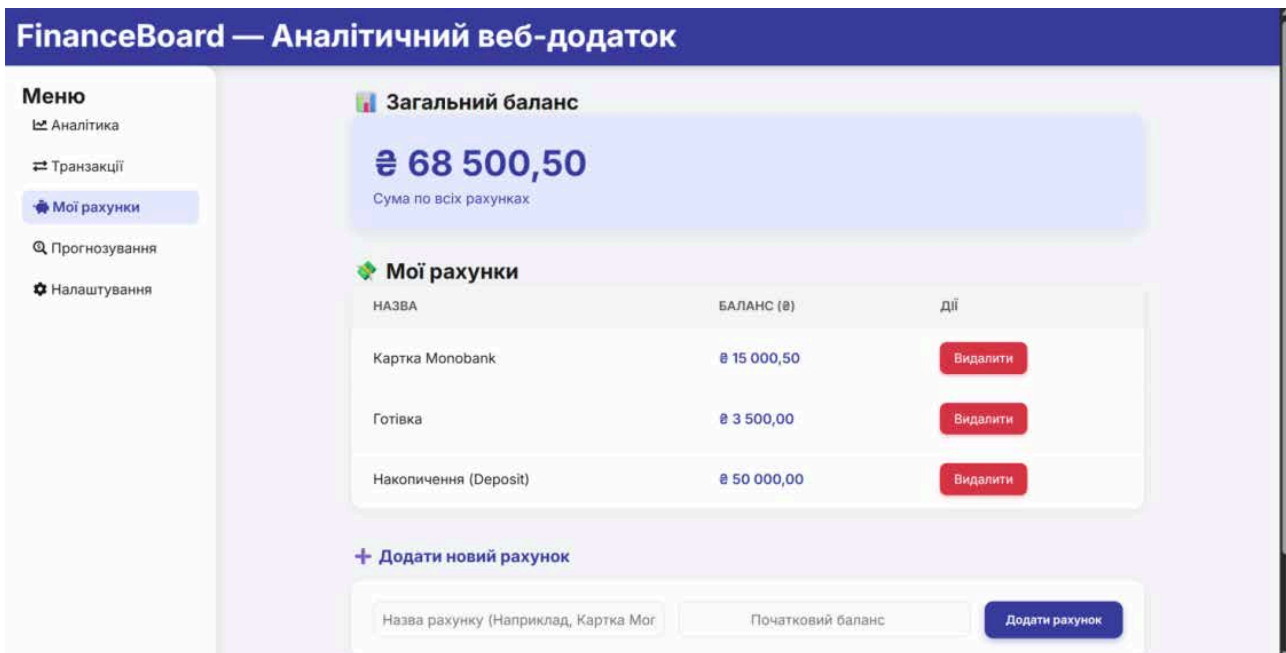


Рис. Б.7 Зведена інформація про поточні фінансові рахунки та баланси

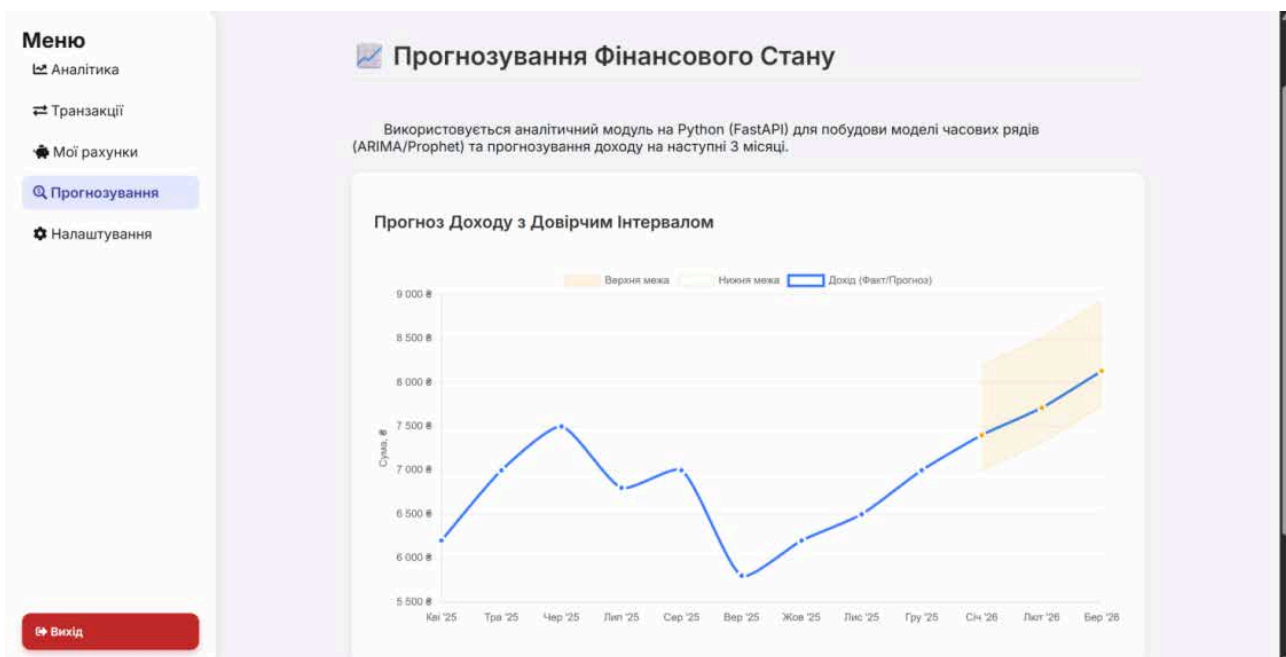


Рис. Б.8 Результат прогнозування місячного доходу на наступні періоди за допомогою моделі Prophet

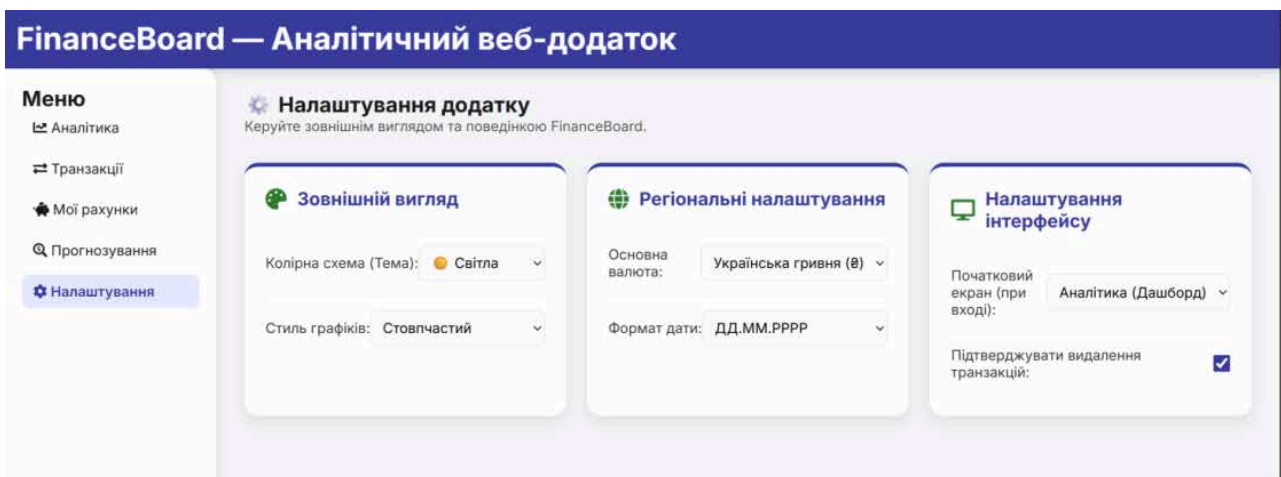


Рис. Б.9 Інтерфейс управління налаштуваннями користувача