**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ**
**ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ**
**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

# КВАЛІФІКАЦІЙНА РОБОТА

**на тему:** ДОДАТОК ДЛЯ ПЕРСОНАЛЬНОГО ФІНАНСОВОГО УПРАВЛІННЯ

Здобувача вищої освіти другого (магістерського) рівня,
ІІ курсу, групи КМІ-24-2, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології в економіці»


(підпис)

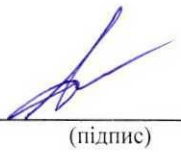
Мазурова Андрія Васильовича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Філоненко Михайло Миколайович, канд. фіз.-мат. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

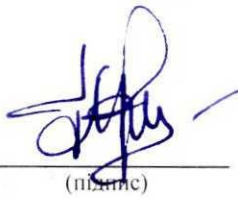
Керівник освітньо-професійної програми


(підпис)

Омельчук Антон Анатолійович, канд. техн. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту за рішенням кафедри протокол від «02» січня 2026 № 10

Завідувач кафедри


(підпис)

Філоненко Михайло Миколайович, канд. фіз.-мат. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Ірпінь – 2026

Реєстрація №28 22.01.2026 _____
(дата) (підпис відповідальної особи)

Результати перевірки до захисту 02.01.2026 _____
(до захисту, на доопрацювання) (дата) (підпис)

Результат захисту 4 (добре) 05.02.2026 _____
(оцінка) (дата) (підпис)

Голова комісії: _____ О.С. Дюбанов
(підпис)

Члени комісії: _____ А.Ю. Горбовий
(підпис)

_____ М.М. Філоненко
(підпис)

_____ І.Д. Погорєловська
(підпис)

## БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

| Об'єкти контролю | Діапазон оцінки в балах | Фактична кількість балів | Примітки | Дата, П.І.Б.: керівника, голови комісії членів комісії,секретаря, (підписи) |
|---|---|---|---|---|
| **ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ** | | | | |
| 1. Зміст кваліфікаційної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій) | від 0 до 30 | 23 | | 05.02.2026 Дюбанов О.С. |
| 2. Дотримання вимог щодо оформлення | від 0 до 10 | 7 | | Горбовий А.Ю. |
| 3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни | від 0 до 5 | 4 | | Філоненко М.М. Погорєловська І.Д. |
| 4. Рецензент | від 0 до 5 | 4 | | Суховецька Л.П. |
| РАЗОМ БАЛІВ | від 0 до 50 | 38 | | |

Оцінювання кваліфікаційної роботи науковим керівником - 40 балів.

### ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

| Оцінка за 100-бальною шкалою | Оцінка за шкалоюЄКТС | Оцінка за національною шкалою | Дата | П.І.Б.: голови, членів комісії, секретаря, (підпис) |
|---|---|---|---|---|
| 78 | C | 4 (добре) | 05.02.2026 | Дюбанов О.С. Горбовий А.Ю. Філоненко М.М. Погорєловська І.Д. Суховецька Л.П. |

### ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

| Оцінка за 100-бальною шкалою | Оцінка за шкалою ЄКТС | Оцінка за національною шкалою |
|---|---|---|
| 90-100 | A | 5 (відмінно) |
| 80-89 | B | 4 (добре) |
| 70-79 | C | |
| 60-69 | D | 3 (задовільно) |
| 50-59 | E | |
| 35-49 | FX | 2 (незадовільно) |

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Інформаційні управляючі системи і технології в економіці»

**Затверджую**
Завідувач кафедри _____ Філоненко М.М.
«11» грудня 2024

**Завдання на кваліфікаційну роботу**
**здобувачу вищої освіти**
Мазурову Андрію Васильовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Додаток для персонального фінансового управління
Затверджена наказом ректора від «10» грудня 2024 № 1852
2. Строк здачі закінченої роботи «02» січня 2026
3. Мета роботи Розробити веб-орієнтовану систему обліку персонального бюджету, яка дозволить користувачам ефективно вести контроль над своїми фінансами та забезпечити зручний інструмент для аналізу витрат
Об'єкт дослідження Процес обліку персонального бюджету.
Предмет дослідження Інформаційне забезпечення обліку персонального бюджету. Розглядаються аспекти технічної реалізації та функціональності системи для забезпечення її ефективності та зручності використання.
Орієнтовна база дослідження Системи обліку персонального бюджету, заснованої на сучасних технологіях та забезпеченої високою ступенем функціональності.
4. Вихідні дані до роботи вимоги до обсягу та структури;
5. Перелік графічного матеріалу графічна оболонка програмного забезпечення;
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ; 1.1. АНАЛІЗ ФІНАНСОВОЇ ГРАМОТНОСТІ ТА ПОВЕДІНКОВИХ ПАТЕРНІВ НАСЕЛЕННЯ УКРАЇНИ 1.2. ПРОБЛЕМАТИКА УПРАВЛІННЯ СІМЕЙНИМ БЮДЖЕТОМ ТА РОЛЬ ІТ-РІШЕНЬ; РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ; РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ; 3.1. ПРОЕКТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ 3.2. ТЕСТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.

7. Календар план виконання роботи:

| № п/п | Назва етапів кваліфікаційноїроботи | Строк виконання етапів роботи | |
|---|---|---|---|
| | | За планом | Фактично |
| 1 | 2 | 3 | 4 |
| 1 | Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи | До 15.12.2024 | До 15.12.2024 |
| 2 | **І розділ** (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень) | 30.05.2025 | 30.05.2025 |
| 3 | **ІІ розділ** (аналіз експериментальної частини про методики досліджень) | 10.09.2025 | 10.09.2025 |
| 4 | **ІІІ розділ** (проведені теоретичні і (або) експериментальні дослідження та проектні рішення) | 20.11.2025 | 20.11.2025 |
| 5 | **Висновки** | 11.12.2025 | 11.12.2025 |
| 6 | Доопрацювання роботи з урахуванням зауважень наукового керівника | 25.12.2025 | 25.12.2025 |
| 7 | Отримання відгуку керівника | 28.01.2026 | 28.01.2026 |
| 8 | Отримання зовнішньої рецензії | 30.01.2026 | 30.01.2026 |

8. Дата видачі завдання «11» грудня 2024

9. Керівник кваліфікаційної роботи _____ (підпис) Філоненко М.М. (прізвище, ініціали)

10. Керівник освітньо-професійної програми _____ (підпис) Омельчук А.А. (прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти _____ (підпис) Мазуров А.В. (прізвище, ініціали)

# АНОТАЦІЯ

ДОДАТОК ДЛЯ ПЕРСОНАЛЬНОГО ФІНАНСОВОГО УПРАВЛІННЯ

У процесі виконання кваліфікаційної роботи було спроєктовано та реалізовано веб-орієнтовану інформаційну систему обліку персонального бюджету.

Результатом роботи є повнофункціональна вебсистема, яка демонструє високу ефективність у сфері контролю особистих фінансів і має потенціал для подальшого масштабування, зокрема інтеграції з банківськими API та впровадження елементів інтелектуального прогнозування бюджету.

Кваліфікаційна робота містить 43 сторінок, 2 таблиці, 8 рисунків, список використаних джерел 45 найменувань.

Ключові слова: фінансове управління, інформаційна система, облік.

Рік захисту роботи 2026

# ABSTRACT

APPLICATION FOR PERSONAL FINANCIAL MANAGEMENT

During the qualification work, a web-based information system for accounting for a personal budget was designed and implemented.

The result of the work is a fully functional web system that demonstrates high efficiency in the field of personal finance control and has the potential for further scaling, in particular integration with banking APIs and the implementation of elements of intelligent budget forecasting.

The qualification work contains 43 pages, 2 tables, 8 figures, a list of sources used 45 items.

Keywords: financial management, information system, accounting.

The year of defense of the work is 2026

**Зміст**

**ВСТУП** ..... **3**

**РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ** ..... **6**

1.1. Аналіз фінансової грамотності та поведінкових патернів населення України ..... 6

1.2. Проблематика управління сімейним бюджетом та роль IT-рішень ..... 7

**РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ** ..... **13**

**РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ** ..... **17**

3.1. Проектування web-орієнтованої системи ..... 17

3.2. Тестування web-орієнтованої системи ..... 33

**ВИСНОВКИ** ..... **37**

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** ..... **39**

# ВСТУП

**Актуальність теми.** У сучасному світі, де швидкі технологічні зміни і зростання обсягу електронних фінансових транзакцій визначають нові вимоги до управління фінансами, здатність ефективно вести облік фінансів, розуміти та контролювати витрати стає критичною для багатьох людей, які прагнуть досягти фінансової стабільності та управління своїми ресурсами. Розвиток технологій, зокрема веб- розробки, надає унікальні можливості для створення зручних та ефективних інструментів управління фінансами, які легко доступні з будь-якого пристрою з підключенням до Інтернету. Система обліку персонального бюджету відповідає цим потребам та надає користувачам інструмент для досягнення фінансової відповідальності. Нижче наведено ключові аспекти актуальності розробки веб- орієнтованої системи обліку персонального бюджету:

1. Зростання інтересу до фінансової грамотності:

У сучасному світі існує зростаюча необхідність в розвитку фінансової грамотності серед населення. Веб-орієнтована система обліку персонального бюджету надасть користувачам зручний інструмент для вивчення та аналізу їхніх фінансів.

2. Необхідність ефективного управління фінансами:

З ростом складності фінансового життя і нестабільністю економічного середовища, особистий контроль над фінансами стає ключовим елементом стабільності та благополуччя. Веб-орієнтована система допоможе користувачам здійснювати ефективний контроль над своїми витратами та доходами.

3. Зручний доступ до інформації:

Оскільки сучасне суспільство характеризується великою мобільністю та високим рівнем використання Інтернету, веб-орієнтована система обліку фінансів забезпечить користувачам зручний доступ до інформації з будь-якого пристрою та місця.

4. Потреба в інструменті для аналізу витрат:

Користувачі все частіше виявляють бажання не лише вести облік своїх

фінансів, але й отримувати аналітичні дані та рекомендації стосовно оптимізації витрат. Розробка веб-системи, яка надає інструменти для аналізу витрат, відповідає цій потребі.

5. Розвиток технологій та електронних сервісів:

Завдяки стрімкому розвитку технологій та високій доступності Інтернету, веб-орієнтовані фінансові сервіси стають все більш популярними серед користувачів, які шукають зручні та інноваційні рішення.

Отже, розробка веб-орієнтованої системи обліку персонального бюджету є актуальним та важливим напрямком, який відповідає потребам сучасного суспільства в ефективних інструментах фінансового управління.

**Об’єкт дослідження** – процес обліку персонального бюджету.

**Предмет дослідження** – інформаційне забезпечення обліку персонального бюджету. Розглядаються аспекти технічної реалізації та функціональності системи для забезпечення її ефективності та зручності використання.

**Метою роботи** є розробка та впровадження веб-орієнтованої системи обліку персонального бюджету, яка дозволить користувачам ефективно вести контроль над своїми фінансами та забезпечити зручний інструмент для аналізу витрат.

Для досягнення поставленої мети проекту необхідно реалізувати ряд ключових завдань:

- Проаналізувати предметну область для визначення актуальності та цільової аудиторії використання системи, провести порівняльний аналіз існуючих інформаційних систем.
- Визначити технології для розробки системи.
- Розробити модель та структуру інформаційної системи.
- Здійснити програмну реалізацію структури та функціоналу системи.
- Провести тестування.

Практичне цінність полягає в забезпеченні зручного та ефективного контролю над фінансами, сприянні точному відстеженню витрат, а також допомозі у плануванні власного бюджету користувача через доступ до системи з

будь-якого пристрою з Інтернет-підключенням..

**Гіпотеза дослідження:**

Припускається, що розробка та впровадження веб-орієнтованої системи обліку персонального бюджету, заснованої на сучасних технологіях та забезпеченої високою ступенем функціональності, сприятиме підвищенню фінансової грамотності та ефективному управлінню фінансами користувачів. При цьому, оптимальний вибір технологій, структури та функціоналу системи дозволить створити зручний та доступний інструмент для ведення обліку та аналізу витрат, що сприятиме формуванню позитивної фінансової поведінки серед користувачів. Таким чином, гіпотеза передбачає позитивний вплив веб-орієнтованої системи на фінансову дисципліну та здатність користувачів до ефективного управління своїм персональним бюджетом.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1. Аналіз фінансової грамотності та поведінкових патернів населення України

Фінансова поведінка населення є фундаментальним чинником, що визначає динаміку національного фінансового ринку, масштаби тіньового сектору та ефективність впровадження державних реформ. Згідно з дослідженнями ОЕСР, для українських домогосподарств характерною є орієнтація на короткострокове споживання. Схильність до витрат замість заощадження зумовлена як специфічним історичним контекстом, так і високим рівнем економічної невизначеності, посиленим наслідками російської агресії.

**Статистичні показники.** Відповідно до інституційних звітів [1], індекс фінансової поведінки в Україні становить **5,2 бала з 9**. Попри те, що цей показник перевищує результати ряду сусідніх країн (4,8), він залишається нижчим за середньосвітове значення серед тридцяти країн-учасниць опитування (**5,4**).

**Фактори фінансової інклюзії.** Рівень залученості громадян до фінансової системи напряму корелює з їхньою здатністю аналізувати складні фінансові продукти, рівнем довіри до банківських установ та впевненістю у правовому захисті інтересів споживачів [2]. Когнітивні бар’єри та дефіцит довіри до державних і приватних фінансових інститутів є основними перешкодами для розширення фінансової інклюзії.

**Динаміка та стратегічне планування.** За даними дослідження Проєкту USAID та НБУ, за період з 2018 по 2021 рік загальний індекс фінансової грамотності українців продемонстрував позитивну динаміку, зрісши з **11,6 до 12,3 бала** [3]. У межах реалізації «Національної стратегії розвитку фінансової грамотності до 2025 року» було виявлено такі закономірності:

- **Демографічний розрив:** зберігається суттєва різниця в рівні грамотності між мешканцями міст та сільської місцевості.
- **Вікові особливості:** найвищий рівень компетенцій спостерігається у

віковій групі **25–59 років**, тоді як молодь та особи похилого віку (60+) потребують додаткових освітніх заходів.

- **Освітній ценз:** підтверджено пряму кореляцію між наявністю вищої освіти та рівнем фінансової обізнаності.

## 1.2. Проблематика управління сімейним бюджетом та роль ІТ-рішень

Значна частка українських родин стикається з фінансовими труднощами, що часто є наслідком не лише об'єктивного рівня доходів, а й неефективного їх розподілу. Дефіцит коштів до моменту отримання наступної заробітної плати змушує громадян звертатися до послуг мікрокредитних організацій, що створює ризики потрапляння у боргову залежність.

Вирішення цієї проблеми лежить у площині стратегічного планування **сімейного бюджету**. Бюджетування дозволяє сформувати цілісну картину фінансового стану, мінімізувати безконтрольні витрати та забезпечити фінансову стабільність.

**Ключові етапи контролю:**

1. **Моніторинг грошових потоків:** системна фіксація всіх транзакцій як основа для розумної економії.
2. **Спільне планування:** перехід від примусового контролю до колективного прийняття рішень у межах родини.
3. **Використання цифровізованих інструментів:** впровадження спеціалізованих ІТ-рішень (систем обліку персонального бюджету) значно спрощує процес ведення звітності та підвищує точність аналізу.

Таким чином, розробка та впровадження веб-орієнтованих систем для управління персональними фінансами є не лише технічним завданням, а й важливим соціально-економічним кроком. Такі сервіси є інструментом підвищення фінансової дисципліни, що в перспективі сприяє трансформації фінансової поведінки громадян та зміцненню їхнього добробуту.

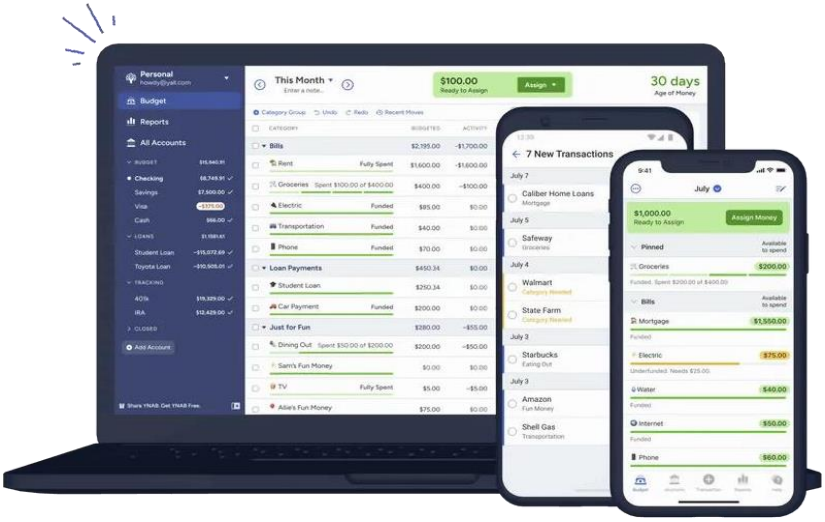



Рисунок 1.1 – YNAB – вигляд інтерфейсу рішення

У сучасній експертній спільноті стратегічне планування бюджету розглядається як фундаментальний інструмент контролю над власними фінансовими ресурсами. Дослідження [6] виокремлює ключові параметри, що є критичними при проєктуванні автоматизованих систем бюджетування, зокрема: гнучкість налаштувань, точність агрегації даних та ергономіку інтерфейсу.

Сучасний ринок FinTech-рішень пропонує широкий спектр систем, що мінімізують частку ручної праці завдяки автоматичному відстеженню транзакцій через інтеграцію з банківськими рахунками. Згідно з [7], використання таких інструментів дозволяє значно оптимізувати часові ресурси користувача, забезпечуючи при цьому високий рівень фінансової прозорості та стимулюючи ощадливу поведінку.

Проведемо порівняльний аналіз найбільш релевантних вебсистем та мобільних додатків:

**1. YNAB (You Need A Budget)**

Система базується на методології «нульового бюджету», де кожна грошова одиниця повинна мати чітке призначення.

- **Особливості:** Орієнтація на проактивне планування та формування цільових фондів [8].
- **Переваги:** Висока ефективність у зміні фінансових звичок користувача.
- **Недоліки:** Платна модель розповсюдження (підписка) та високий поріг

входження для початківців.

**2. Mint**

Один із найпопулярніших безкоштовних сервісів, що забезпечує глибоку інтеграцію з фінансовими інститутами.

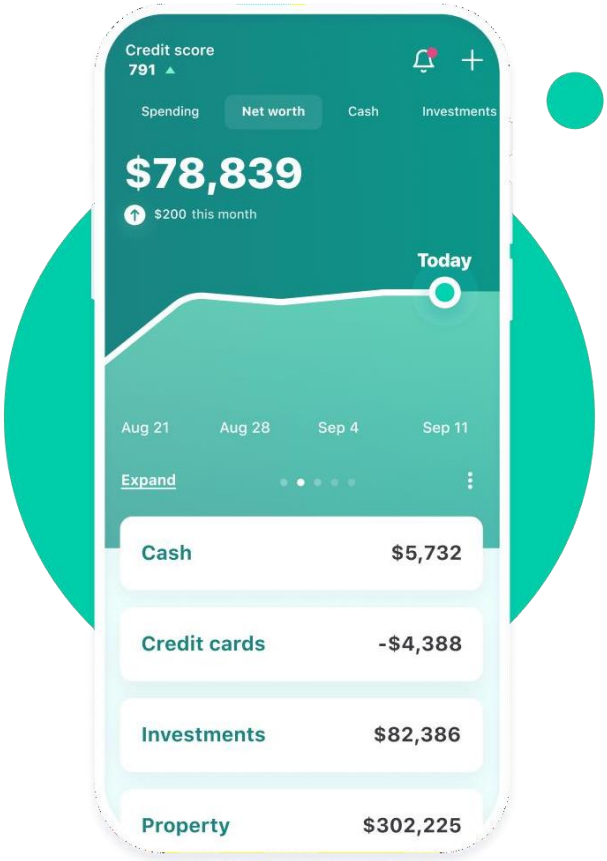



Рисунок 1.2 – Mint – вигляд застосунку

- **Особливості:** Автоматична категоризація витрат та моніторинг рахунків у реальному часі [9].
- **Переваги:** Відсутність плати за користування, автоматизація більшості процесів.
- **Недоліки:** Ризики, пов'язані з конфіденційністю даних, та похибки в алгоритмах автоматичної класифікації витрат.

**3. PocketGuard**

Програмне рішення, що фокусується на поточному стані ліквідності користувача.

- **Особливості:** Обчислення вільного залишку («In My Pocket») після врахування всіх обов'язкових платежів [10].

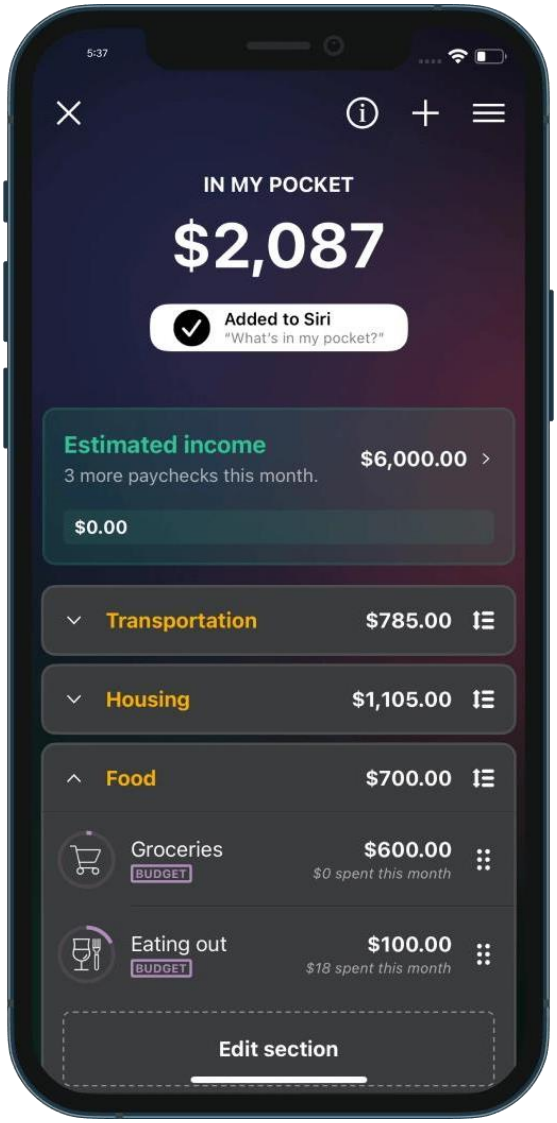



Рисунок 1.3 – PocketGuard – вигляд застосунку

- **Переваги:** Лаконічність інтерфейсу та висока швидкість оновлення даних.
- **Недоліки:** Критична залежність від стабільного інтернет-з'єднання та обмежений функціонал для складного фінансового аналізу.

**4. GoodBudget**

Цифрова реалізація класичного методу «віртуальних конвертів».

- **Особливості:** Користувач розподіляє кошти за лімітованими категоріями без прямої синхронізації з банківськими рахунками [11].

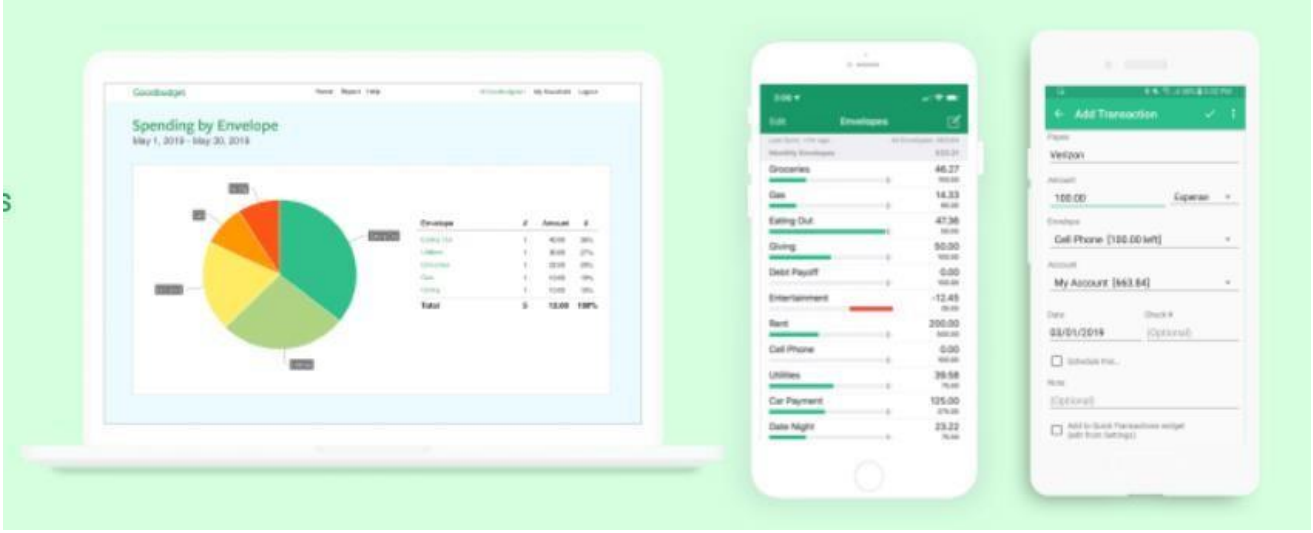



Рисунок 1.4 – GoodBudget – вигляд інтерфейсу рішення

- **Переваги:** Простота та стимулювання свідомого розподілу грошей.
- **Недоліки:** Висока частка ручного введення даних та обмежені можливості безкоштовної версії.

**Висновки щодо аналізу та обґрунтування розробки**

Проведений аналіз демонструє, що попри розмаїття існуючих рішень, більшість із них мають суттєві обмеження для вітчизняного ринку:

1. **Відсутність мовної адаптації:** Складність термінології в англомовних інтерфейсах створює бар'єр для широкого загалу українських користувачів.
2. **Вартість:** Висока ціна закордонних підписок часто нівелює вигоду від економії бюджету.
3. **Специфіка банківської інтеграції:** Не всі іноземні сервіси коректно взаємодіють з API українських банківських установ.

З огляду на це, розробка локалізованого програмного продукту, адаптованого до економічних реалій та потреб українського споживача, є актуальним науково-практичним завданням. Створення системи з інтуїтивним інтерфейсом та вбудованим освітнім модулем дозволить підвищити ефективність управління персональним бюджетом та сприятиме зростанню

загального рівня фінансової грамотності в країні.

Детальний аналіз існуючих аналогічних рішень дозволив виокремити низку функціональних недоліків, усунення яких стало пріоритетним завданням при розробці власної системи. Для об’єктивної оцінки та систематизації отриманих даних було проведено порівняльний аналіз за визначеними критеріями, результати якого наведено в таблиці 1.1

Таблиця 1.1 – Порівняння досліджених інформаційних систем.

| **Назва** | **YNAB** | **Mint** | **PocketGuard** | **GoodBudget** | **Розроблювана система** |
|---|---|---|---|---|---|
| **Вид підписки** | платна | безкоштовна | умовно безкоштовна | умовно безкоштовна | безкоштовна |
| **Складність інтерфейсу** | дуже складна | складна | складна | складна | легко |
| **Швидкість освоєння користувачем** | дуже повільно | повільно | повільно | повільно | швидко |
| **Наявність вибору валют** | присутня | відсутня | відсутня | відсутня | присутня |
| **Наявність української мови** | відсутня | відсутня | відсутня | відсутня | присутня |

# РОЗДІЛ 2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

**Метою роботи** є проєктування та програмна реалізація веб-орієнтованої інформаційної системи обліку персональних фінансів. Розробка спрямована на створення високоефективного інструментарію, що забезпечує користувачам можливість систематичного моніторингу, детального аналізу та стратегічного планування витрат, сприяючи впровадженню принципів раціонального управління особистим бюджетом.

Досягнення мети передбачає застосування передового технологічного стеку для побудови відмовостійкої, безпечної та масштабованої платформи. Особлива увага приділяється реалізації сучасних практик UI/UX-дизайну для забезпечення інтуїтивної взаємодії, а також впровадженню надійних протоколів захисту конфіденційної фінансової інформації.

Крім технічної складової, проєкт має **соціально-освітню спрямованість**: впровадження системи покликане підвищити рівень фінансової грамотності та свідомості користувачів через візуалізацію аналітичних даних та інтерактивні засоби планування ресурсів.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. **Комплексний аналіз предметної області та обґрунтування актуальності:**
   - Дослідження сучасних проблем та тенденцій у сфері управління персональними фінансами.
   - Огляд фахової літератури та актуальних статистичних даних для ідентифікації потреб цільової аудиторії.
2. **Порівняльний аналіз існуючих ІТ-рішень:**
   - Вивчення функціональних можливостей та архітектурних особливостей наявних вебсистем обліку бюджету.
   - Виявлення переваг та критичних недоліків конкурентних продуктів для формування конкурентних переваг власної розробки.
3. **Обґрунтування та вибір технологічного стеку:**

- Порівняльне дослідження сучасних фреймворків та інструментів веб-розробки.
- Вибір оптимальних технологій (Frontend, Backend, Database) з урахуванням вимог до швидкодії, безпеки та масштабованості.

4. **Проєктування архітектури інформаційної системи:**
   - Розробка логічної та фізичної моделей бази даних.
   - Проєктування архітектури системних модулів та створення прототипів користувацького інтерфейсу.
5. **Програмна реалізація функціоналу системи:**
   - Розробка клієнтської та серверної частин додатка відповідно до спроєктованої моделі.
   - Імплементація ключових модулів: обліку транзакцій, аналітичної обробки даних, генерації статистичної звітності та механізмів автентифікації.
6. **Верифікація та тестування:**
   - Проведення комплексного тестування (функціонального, безпекового, юзабіліті) для виявлення та усунення дефектів.
   - Оцінка відповідності розробленого продукту вихідним вимогам.

Для досягнення поставленої мети та забезпечення наукової обґрунтованості результатів у роботі застосовано комплекс методів дослідження:

1. **Теоретичний метод.** Полягає у системному аналізі предметної області, вивченні фахової літератури та документації. Метод використано для дослідження архітектурних паттернів вебдодатків, протоколів безпеки фінансових даних та сучасних концепцій UI/UX-дизайну.
   - *Інформаційна база:* ресурси MDN [12], W3Schools [13] та Stack Overflow [14] (технологічний аспект); Harvard Business Review [15] та Investopedia [17] (економічний аспект); стандарти OWASP [18] (безпека).
2. **Емпіричний метод.** Спрямований на збір та аналіз практичних даних щодо фінансової поведінки користувачів. Використовувався для

виявлення споживчих переваг та потреб, що лягли в основу порівняльного аналізу існуючих систем.

- *Джерела даних:* статистичні звіти Світового банку [20] та МВФ [21], аналітичні огляди Deloitte [27] та McKinsey [28].

3. **Метод моделювання.** Застосовано для візуалізації архітектури та функціональних зв'язків системи. Зокрема, використано методологію **IDEF0** [29] для декомпозиції функцій системи, ідентифікації інформаційних потоків та керуючих впливів.

4. **Системно-функціональний метод.** Дозволив розглянути вебсистему як сукупність взаємопов'язаних модулів (компонентів), визначити їхні ролі та забезпечити ефективну взаємодію користувача з інтерфейсом додатка.

**Технологічний стек та інструменти реалізації**

Для реалізації вебсистеми обліку персонального бюджету визначено наступні ключові напрями розробки та відповідні інструментарії:

1. **Управління даними:** Проєктування реляційної бази даних на базі **PostgreSQL**. Структура БД включає оптимізовані таблиці для зберігання транзакцій, категорій та облікових записів користувачів.

2. **Візуалізація та аналітика:** Інтеграція бібліотеки **Chart.js** для трансформації текстових фінансових даних у динамічні графіки та діаграми.

3. **Безпека та тестування:** Впровадження механізмів автентифікації, валідації вхідних даних та проведення Unit-тестування для перевірки цілісності коду.

**Обґрунтування вибору технологій**

Вибір обраного стеку (PERN: PostgreSQL, Express, React, Node) обумовлений наступними чинниками:

- **Material UI:** дозволяє скоротити терміни розробки завдяки використанню готових, стилізованих за стандартами Google компонентів, зберігаючи при цьому можливість глибокої кастомізації.

- **Express JS:** легковажний фреймворк з розвиненою системою Middleware, що ідеально підходить для створення RESTful API.
- **PostgreSQL:** обрана як одна з найбільш надійних СУБД з підтримкою складних запитів та високим рівнем безпеки даних, що є критичним для фінансових застосунків.
- **Chart.js:** надає оптимальний баланс між простотою впровадження та потужними можливостями інтерактивної візуалізації даних.

Інтеграція обраних технологій дозволяє створити стабільну, кросплатформну та безпечну систему. Масштабованість архітектури забезпечує можливість подальшого розширення функціоналу (наприклад, додавання мультивалютності або прогнозної аналітики) без радикальної переробки ядра системи.

# РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ

## 3.1. Проектування web-орієнтованої системи

На основі проведеного аналізу предметної області, обґрунтування актуальності теми, а також чіткого визначення мети та завдань дослідження, наступним логічним етапом роботи є безпосереднє проєктування архітектури веб-орієнтованої системи

Проект "Фінансовий менеджер" - це веб-додаток для персонального управління особистими фінансами, створений для спрощення процесу ведення обліку доходів та витрат користувача.

Додаток дозволяє користувачам:

- Відстежувати свої фінансові операції (доходи та витрати)
- Категорізувати транзакції для кращої організації
- Аналізувати фінансові тенденції за допомогою графіків та статистики
- Отримувати швидкий огляд фінансового стану на дашборді
- Фільтрувати та шукати транзакції за різними критеріями

Основна мета проекту - надати простий, інтуїтивний інструмент для ефективного управління особистим бюджетом без необхідності використання складних комерційних рішень.

Створити доступний, зрозумілий веб-додаток для персонального фінансового обліку, який дозволить користувачам мати повний контроль над своїми фінансами та приймати обґрунтовані фінансові рішення.

Допоміжні цілі:

- Автоматизація процесу ведення фінансових записів

- Візуалізація фінансових даних для кращого розуміння
- Простота використання без необхідності спеціальних навичок
- Швидкий доступ до фінансової інформації
- Можливість аналізу фінансових тенденцій у часі

## ІДЕЯ ТА КОНЦЕПЦІЯ

Основна ідея

Ідея проекту базується на необхідності простого та ефективного інструменту для ведення персонального бюджету. Замість використання складних фінансових програм або звичайних електронних таблиць, проект пропонує спеціалізоване веб-рішення, оптимізоване саме для задач персонального фінансового обліку.

Концепція розробки:

- Модульність: кожен компонент виконує конкретну функцію
- Простота: мінімалістичний інтерфейс без зайвих елементів
- Швидкість: швидка робота з даними та генерація звітів
- Доступність: працює в будь-якому сучасному веб-браузері
- Масштабованість: можливість розширення функціоналу в майбутньому

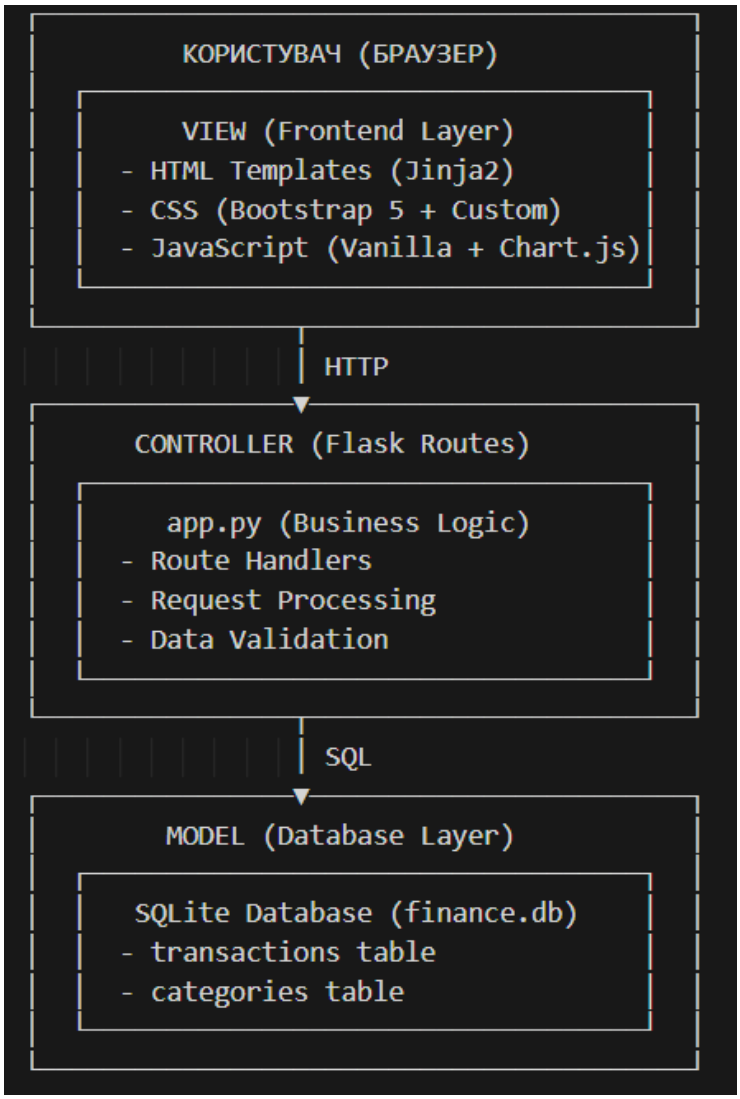
Цільова аудиторія:

- Особи, які хочуть контролювати свої фінанси
- Студенти та молоді сім'ї з обмеженим бюджетом
- Фрілансери та підприємці з нерегулярними доходами
- Будь-хто, хто хоче краще розуміти свої фінансові звички

## АРХІТЕКТУРА ПРОЕКТУ

Загальна архітектура:

Проект побудований за архітектурою "Client-Server" з використанням веб-технологій. Архітектура дотримується паттерну MVC (Model-View-Controller):




Компоненти архітектури:

- Backend (Серверна частина):

- Framework: Flask 2.3.3

- Мова: Python 3.8+

- Основні модулі:

  * Flask - веб-фреймворк

  * sqlite3 - робота з базою даних

  * datetime - робота з датами

* os - системні операції

Frontend (Клієнтська частина):

- HTML5 - структура сторінок
- CSS3 - стилізація (Bootstrap 5 + власні стилі)
- JavaScript (ES6+) - інтерактивність
- Chart.js - візуалізація даних
- Font Awesome - іконки

База даних:

- Система: SQLite 3
- Структура:
  * Таблиця 'transactions' - зберігає всі фінансові транзакції
  * Таблиця 'categories' - зберігає категорії доходів та витрат

Структура файлів проекту:

code/

```
├── app.py                  # Головний файл Flask-додатку
│   ├── Функції ініціалізації БД (init_db)
│   ├── Функції роботи з БД (get_db_connection)
│   ├── Маршрути (routes):
│   │   ├── / (головна сторінка - дашборд)
│   │   ├── /add_transaction (додавання транзакції)
│   │   ├── /transactions (список транзакцій)
│   │   ├── /analytics (аналітика)
│   │   ├── /delete_transaction/<id> (видалення)
│   │   └── /api/* (API endpoints)
│   └── Конфігурація (secret_key, DATABASE)
```

```
│
├── templates/              # HTML шаблони (Jinja2)
│   ├── base.html           # Базовий шаблон з навігацією
│   ├── index.html          # Головна сторінка (дашборд)
│   ├── add_transaction.html  # Форма додавання транзакції
│   ├── transactions.html     # Список всіх транзакцій
│   └── analytics.html        # Сторінка аналітики
│
├── static/                 # Статичні файли
│   ├── css/
│   │   └── style.css         # Власні стилі
│   └── js/
│       └── main.js           # JavaScript функціональність
│
├── demo_data.py            # Скрипт для створення тестових даних
├── requirements.txt        # Python залежності
├── README.md               # Документація проекту
└── finance.db              # База даних SQLite (створюється автоматично)
```

## ПЛАН РОЗРОБКИ

Етап 1: Планування та проектування

- Визначення вимог до системи
- Проектування структури бази даних

- Вибір технологій та інструментів
- Створення макетів інтерфейсу

Етап 2: Налаштування середовища
- Встановлення Python та необхідних бібліотек
- Налаштування віртуального середовища
- Створення структури проекту
- Ініціалізація репозиторію

Етап 3: Розробка бази даних
- Створення схеми бази даних
- Реалізація функції init_db()
- Додавання базових категорій
- Тестування операцій з БД

Етап 4: Розробка Backend
- Створення Flask додатку
- Реалізація маршрутів для основних операцій:
  * Головна сторінка з дашбордом
  * Додавання транзакцій
  * Перегляд списку транзакцій
  * Видалення транзакцій
- Реалізація бізнес-логіки:
  * Розрахунок статистики
  * Фільтрація та пошук
  * Аналітика та звіти
- Створення API endpoints для AJAX запитів

Етап 5: Розробка Frontend

- Створення базового шаблону (base.html)
- Розробка шаблонів для всіх сторінок
- Інтеграція Bootstrap для стилізації
- Додавання інтерактивних елементів
- Реалізація графіків за допомогою Chart.js
- Адаптація під мобільні пристрої

Етап 6: Функціонал аналітики

- Реалізація розрахунків статистики
- Створення графіків:
  * Графік доходів/витрат за місяцями
  * Кругові діаграми по категоріях
- Додавання фільтрів для аналітики
- Експорт даних (майбутнє розширення)

. Етап 7: Тестування та налагодження

- Модульне тестування окремих компонентів
- Інтеграційне тестування
- Тестування користувацького інтерфейсу
- Виправлення помилок та оптимізація

Етап 8: Документування

- Написання README
- Коментування коду
- Створення інструкцій користувача
- Підготовка документації для розробників

Етап 9: Оптимізація та покращення

- Оптимізація запитів до БД
- Покращення продуктивності

- Додавання кешування (майбутнє)
- Покращення UX

Етап 10: Розгортання (майбутнє)

- Підготовка до production
- Налаштування сервера
- Додавання безпеки (аутентифікація, шифрування)
- Моніторинг та логування

## БАЗОВА ЛОГІКА РОБОТИ ПРОЕКТУ

Логіка роботи з транзакціями:

Додавання транзакції:

1. Користувач відкриває форму додавання транзакції (/add_transaction)
2. Система завантажує доступні категорії з БД (розділені на доходи/витрати)
3. Користувач заповнює форму:
   - Вибирає тип (дохід/витрата)
   - Вибирає категорію (динамічно оновлюється залежно від типу)
   - Вводить суму
   - Встановлює дату
   - Додає опис (необов'язково)
4. При відправці форми (POST):
   - Дані валідуються на сервері
   - Транзакція зберігається в таблиці 'transactions'
   - Система генерує повідомлення про успіх
   - Користувач перенаправляється на головну сторінку

Перегляд транзакцій:

1. Користувач відкриває сторінку списку транзакцій (/transactions)

2. Система завантажує транзакції з БД:

- Застосовує фільтри (якщо вказані): тип, категорія, діапазон дат
- Сортує за датою (нові спочатку)
- Реалізує пагінацію (20 записів на сторінку)

3. Дані відображаються в таблиці з можливістю:

- Перегляду всіх деталей
- Видалення транзакції
- Застосування фільтрів

Видалення транзакції:

1. Користувач натискає кнопку видалення
2. Система виконує DELETE запит до БД
3. Транзакція видаляється з таблиці 'transactions'
4. Користувач перенаправляється на сторінку списку транзакцій

Логіка роботи дашборду:

Розрахунок статистики:

1. При завантаженні головної сторінки (/) система виконує наступні запити:

- Розраховує суму доходів за поточний місяць:

```
SELECT SUM(amount) FROM transactions
WHERE type='income' AND strftime('%Y-%m', date) = поточний_місяць
```

- Розраховує суму витрат за поточний місяць:

```
SELECT SUM(amount) FROM transactions
WHERE type='expense' AND strftime('%Y-%m', date) = поточний_місяць
```

- Обчислює баланс: доходи - витрати
- Отримує останні 10 транзакцій:

```
SELECT * FROM transactions
ORDER BY date DESC, created_at DESC LIMIT 10
```

- Розраховує статистику по категоріях:

```
SELECT category, type, SUM(amount) as total
FROM transactions
WHERE strftime('%Y-%m', date) = поточний_місяць
GROUP BY category, type
ORDER BY total DESC
```

Всі дані передаються в шаблон index.html

Шаблон відображає:

- Картки з основними показниками (доходи, витрати, баланс)
- Таблицю останніх транзакцій
- Статистику по категоріях

Логіка роботи аналітики:

Розрахунок місячної статистики:

1. Система запитує дані за останні 6 місяців:

```
SELECT strftime('%Y-%m', date) as month,
    SUM(CASE WHEN type='income' THEN amount ELSE 0 END) as income,
    SUM(CASE WHEN type='expense' THEN amount ELSE 0 END) as expenses
FROM transactions
WHERE date >= date('now', '-6 months')
GROUP BY strftime('%Y-%m', date)
ORDER BY month ASC
```

2. Дані конвертуються у формат, зручний для графіків
3. За допомогою Chart.js створюється графік:
   - Вісь X: місяці
   - Вісь Y: суми
   - Дві лінії: доходи (зелена) та витрати (червона)

Топ категорій:

1. Для витрат за поточний місяць:

```
SELECT category, SUM(amount) as total
FROM transactions
WHERE type='expense' AND strftime('%Y-%m', date) = поточний_місяць
GROUP BY category
HAVING SUM(amount) > 0
ORDER BY total DESC
LIMIT 10
```

2. Для доходів за поточний місяць:

```
SELECT category, SUM(amount) as total
FROM transactions
WHERE type='income' AND strftime('%Y-%m', date) = поточний_місяць
GROUP BY category
HAVING SUM(amount) > 0
ORDER BY total DESC
LIMIT 10
```

3. Дані відображаються у вигляді кругових діаграм (pie charts)

Логіка роботи з базою даних:

Ініціалізація:

При першому запуску додатку:

1. Викликається функція init_db()
2. Створюються таблиці (якщо не існують):
   - transactions: id, type, category, amount, description, date, created_at
   - categories: id, name, type
3. Додаються базові категорії:

- Доходи: Зарплата, Підробіток, Інвестиції, Інше
- Витрати: Продукти, Транспорт, Розваги, Оренда, Комунальні, Одяг, Здоров'я, Освіта, Інше

Підключення до БД:

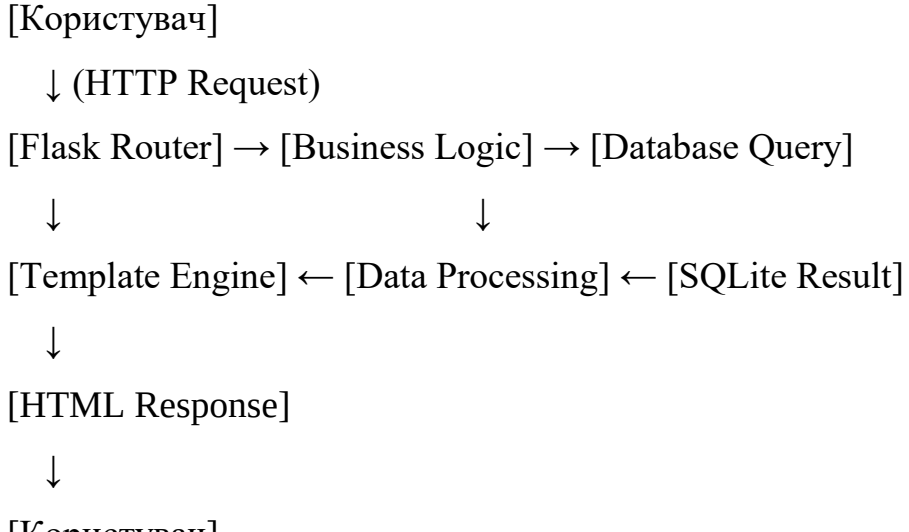
- Використовується функція get_db_connection()
- Налаштовано row_factory = sqlite3.Row для зручної роботи з результатами
- Після виконання запитів з'єднання завжди закривається

Логіка роботи API endpoints:

API endpoints дозволяють отримувати дані у форматі JSON для використання в AJAX запитах:

- /api/transactions - повертає список останніх 50 транзакцій у JSON
- /api/analytics/monthly - повертає місячну статистику у JSON
- /api/analytics/expenses - повертає топ витрат за місяць у JSON
- /api/analytics/incomes - повертає топ доходів за місяць у JSON

Потік даних в системі:



Обробка помилок:

- Валідація даних на стороні сервера
- Обробка помилок БД (try-except блоки)
- Повідомлення користувача через Flask flash messages
- Логування помилок (у debug режимі)

# ТЕХНІЧНІ ДЕТАЛІ

Структура бази даних:

Таблиця 'transactions':

- id (INTEGER PRIMARY KEY AUTOINCREMENT) - унікальний ідентифікатор
- type (TEXT NOT NULL) - тип транзакції: 'income' або 'expense'
- category (TEXT NOT NULL) - назва категорії
- amount (REAL NOT NULL) - сума транзакції
- description (TEXT) - опис транзакції (може бути NULL)
- date (DATE NOT NULL) - дата транзакції
- created_at (TIMESTAMP DEFAULT CURRENT_TIMESTAMP) - час створення запису

Таблиця 'categories':

- id (INTEGER PRIMARY KEY AUTOINCREMENT) - унікальний ідентифікатор
- name (TEXT NOT NULL UNIQUE) - назва категорії (унікальна)
- type (TEXT NOT NULL) - тип: 'income' або 'expense'

Маршрути (Routes):

GET / - Головна сторінка (дашборд)

GET /add_transaction - Форма додавання транзакції

POST /add_transaction - Обробка форми додавання

GET /transactions - Список транзакцій з фільтрами

GET /analytics - Сторінка аналітики

GET /delete_transaction/<id> - Видалення транзакції

GET /api/transactions - API: список транзакцій (JSON)

GET /api/analytics/monthly - API: місячна статистика (JSON)

GET /api/analytics/expenses - API: статистика витрат (JSON)

GET /api/analytics/incomes - API: статистика доходів (JSON)

Залежності проекту:

Flask==2.3.3 - веб-фреймворк

Werkzeug==2.3.7 - WSGI утиліти

Jinja2==3.1.2 - шаблонізатор

MarkupSafe==2.1.3 - безпека шаблонів

itsdangerous==2.1.2 - підписування даних

click==8.1.7 - CLI утиліти

blinker==1.6.3 - сигнали

pJDYSIYS залежності (CDN):

- Bootstrap 5.1.3 - CSS фреймворк
- Chart.js 3.9.1 - бібліотека графіків
- Font Awesome 6.0 - іконки
- Google Fonts (Inter) - шрифти

## МАЙБУТНІ ПОКРАЩЕННЯ

Функціональні покращення:

- Додавання системи аутентифікації користувачів
- Підтримка множинних валют
- Експорт даних у формати CSV, Excel, PDF
- Імпорт транзакцій з банківських виписок
- Встановлення бюджетів для категорій
- Нагадування про регулярні платежі
- Додавання тегів до транзакцій
- Можливість редагування транзакцій
- Пошук по опису транзакцій
- Календарний вигляд транзакцій

Технічні покращення:

- Міграція на PostgreSQL або MySQL
- Додавання кешування (Redis)
- RESTful API з повною документацією
- Unit та інтеграційні тести
- CI/CD пайплайн
- Docker контейнеризація
- Додавання логування
- Моніторинг продуктивності
- Безпека: HTTPS, CSRF захист, SQL injection захист

UX покращення:

- Темна тема
- Більше типів графіків
- Детальніші звіти
- Мобільний додаток (React Native / Flutter)
- PWA (Progressive Web App)
- Офлайн режим

Проект "Фінансовий менеджер" представляє собою повнофункціональний веб-додаток для персонального фінансового управління, реалізований з використанням сучасних веб-технологій. Архітектура проекту дозволяє легко розширювати функціонал та адаптувати систему під конкретні потреби користувачів.

Основні переваги проекту:

- Простота використання
- Швидкість роботи

- Візуалізація даних
- Гнучкість у налаштуванні
- Відкритий вихідний код.

## 3.2. Тестування web-орієнтованої системи

Розглянемо роботу web-орієнтованої системи з точки зору користувача. Вигляд головної сторінки наведено на рис. 3.1.

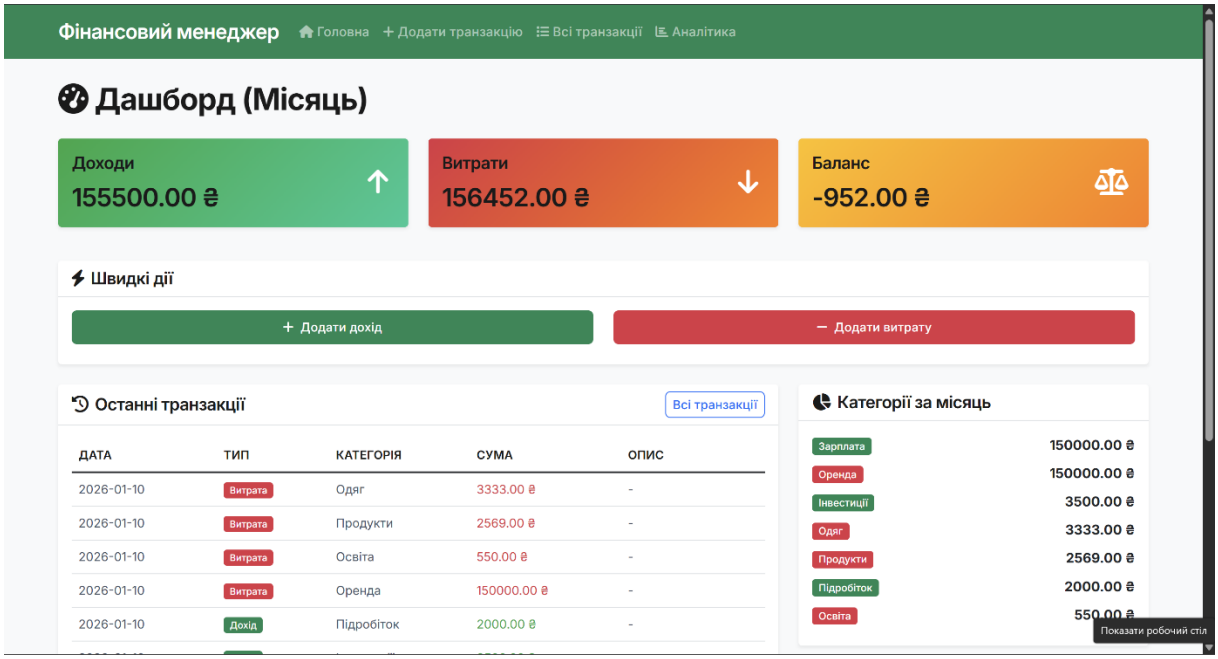



Рисунок 3.1.

Центральна панель керування (Dashboard) дозволяє оперативно вносити фінансові транзакції, здійснювати їх параметричний пошук та відстежувати динаміку витрат за допомогою інтерактивної статистики за обраний період (рис. 3.2.-3.4.)

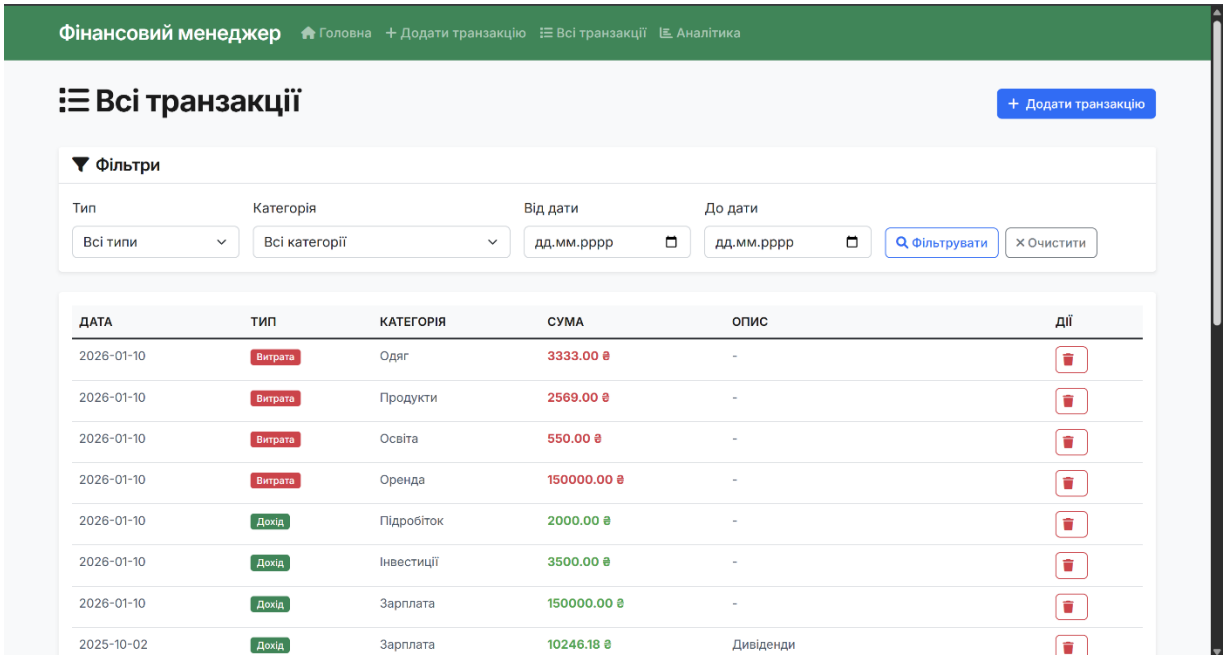



Рисунок 3.2.

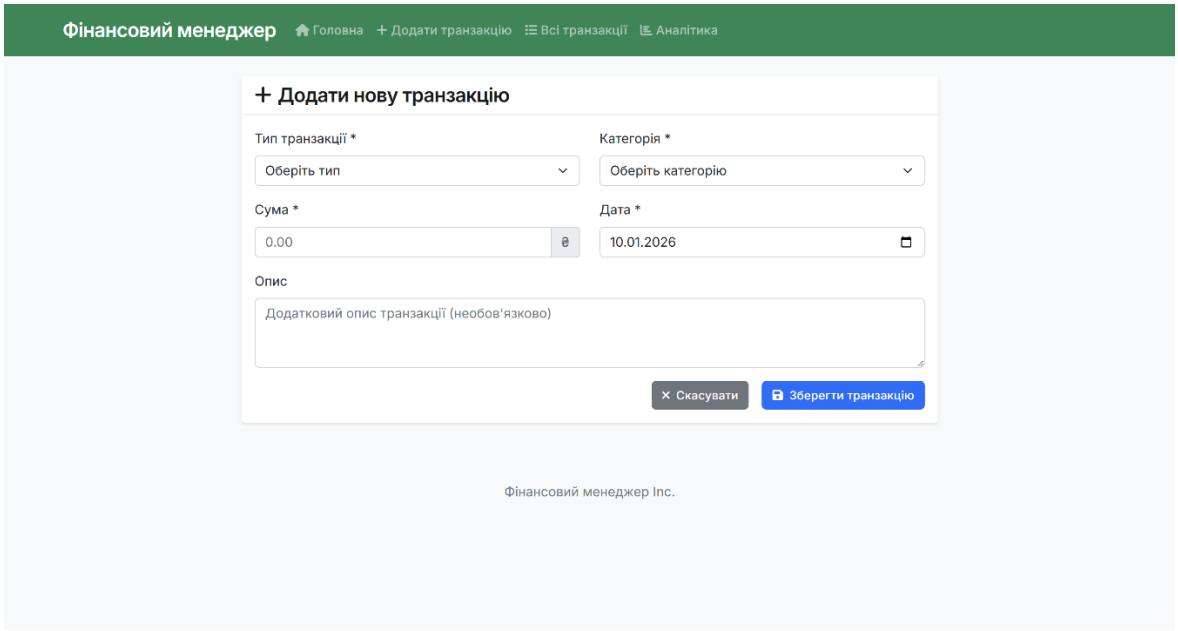



Рисунок. 3.3.

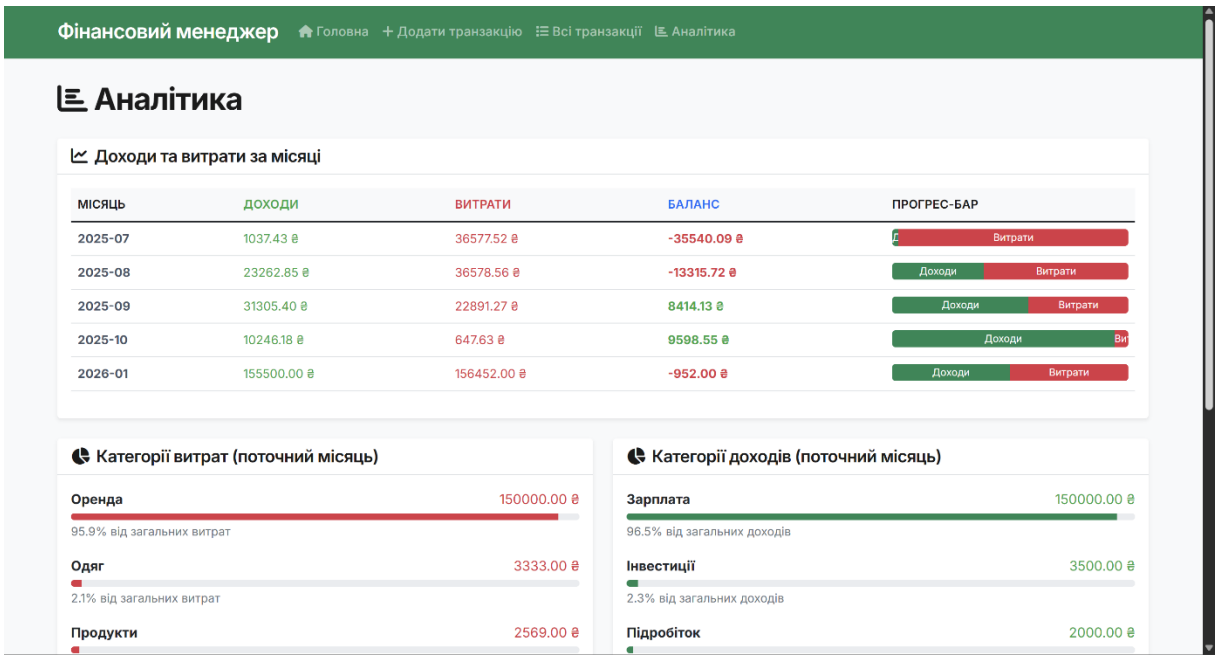



Рисунок 3.4.

Для підтвердження працездатності розробленого програмного продукту та перевірки його відповідності технічному завданню обов'язковим етапом розробки є проведення комплексного тестування.

У сучасній практиці інженерії програмного забезпечення виокремлюють декілька рівнів тестування, кожен з яких має свою специфічну мету:

- **Модульне тестування (Unit Testing)** — перевірка окремих компонентів або функцій коду на ізольованому рівні.
- **Інтеграційне тестування (Integration Testing)** — контроль коректності взаємодії між різними модулями системи.
- **Приймальне тестування (Acceptance Testing)** — фінальна оцінка продукту замовником або кінцевим користувачем на відповідність бізнес-вимогам та придатність до експлуатації в реальних умовах [45].

У рамках виконання кваліфікаційної роботи основну увагу приділено функціональному тестуванню. Цей вид спрямований на верифікацію програмного забезпечення з метою підтвердження того, що кожен елемент системи виконує свої функції згідно зі специфікацією вимог та очікуваннями користувачів. Функціональне тестування дозволяє переконатися у правильності

обробки даних, коректності навігації та стабільності виконання основних сценаріїв використання додатка.

За результатами тестувань, було сформовано таблицю 3.1, де відображено результати тестування.

Таблиця 3.1 Тестування функціоналу web-системи

| **Протестовано** | **Очікуваний результат** | **Фактичний результат** | **0/1** |
|---|---|---|---|
| **Створення витрат** | Витрати створюються, користувач бачить повідомлення про успіх | Після створення витрат, повідомлення про успіх | 1 |
| **Редагування витрат** | Зміни витрат відображаються в списку | Зміни витрат відображаються в списку | 1 |
| **Фільтрація витрат за допомогою форми** | Відображення витрат, що відповідають обраним параметрам | Витрати відобразились згідно обраним параметрам | 1 |
| **Відображення списку витрат** | Список витрат відображається коректно | Список витрат відображається коректно | 1 |
| **Аналітика прибутків і витрат** | Відображати суми витрат за категоріями | Відображає суми витрат за категоріями | 1 |

Тестовий репорт демонструє, що всі складові Web-орієнтованої системи пройшли успішне функціональне тестування. Критичних багів не виявлено, всі функції працюють правильно і відповідають очікуванням користувача

# ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було спроєктовано та реалізовано веб-орієнтовану інформаційну систему обліку персонального бюджету. За результатами проведеного дослідження та розробки можна зробити наступні висновки:

1. **Здійснено комплексний аналіз проблематики** управління персональними фінансами, що дозволило ідентифікувати ключові бар'єри на шляху до фінансової стабільності користувачів та обґрунтувати необхідність розробки автоматизованого рішення.
2. **Опрацьовано масив сучасних наукових публікацій** та прикладних досліджень, що підтвердило високу актуальність обраної теми та дозволило чітко сформулювати потреби цільової аудиторії в умовах динамічного фінансового ринку.
3. **Проведено компаративний аналіз ринкових аналогів**, за результатами якого виявлено функціональні дефіцити існуючих систем. Це стало підґрунтям для формування унікальних конкурентних переваг розробленого продукту.
4. **Обґрунтовано вибір технологічного стеку**, виходячи з критеріїв високої продуктивності, масштабованості та швидкості розробки. Обрана архітектура дозволяє системі ефективно обробляти запити та забезпечувати надійне зберігання даних.
5. **Спроектовано архітектурну модель системи**, яка включає деталізовану структуру реляційної бази даних, а також логічну та фізичну схеми взаємодії компонентів.
6. **Розроблено ергономічний інтерфейс користувача (UI/UX)**, орієнтований на мінімізацію когнітивного навантаження та забезпечення інтуїтивно зрозумілої взаємодії з функціоналом системи.
7. **Здійснено програмну реалізацію Full-stack рішення**, що охоплює клієнтську та серверну частини. Впроваджено ключові модулі: ведення

обліку транзакцій, категоризацію витрат, аналітичну обробку та динамічну візуалізацію фінансової статистики.

8. **Забезпечено верифікацію та валідацію системи** шляхом проведення функціонального тестування. Це підтвердило повну відповідність розробленого програмного забезпечення встановленим технічним вимогам та відсутність критичних помилок у логіці обробки даних.

**Результатом роботи** є повнофункціональна вебсистема, яка демонструє високу ефективність у сфері контролю особистих фінансів і має потенціал для подальшого масштабування, зокрема інтеграції з банківськими API та впровадження елементів інтелектуального прогнозування бюджету.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фінансова грамотність населення: зарубіжний досвід і вітчизняні реалії [Електронний ресурс] – Режим доступу до ресурсу: http://dspace.wunu.edu.ua/bitstream/316497/14651/1/%D0%A4%D0%86%D0%9D%2%D0%93%D0%A0%D0%90%D0%9C%D0%9E%D0%A2%20%D0%92%D1%96%D1%81%D0%BD%D0%B8%D0%BA%20%D0%A2%D0%9D%D0%95%D0%A3%20%D0%9A%D1%96%D0%B7%D0%B8%D0%BC%D0%B0%20%D0%A2.pdf

2. Фінансова грамотність, фінансова інклюзія та фінансовий добробут в Україні у 2021. Звіт за результатами дослідження [Електронний ресурс] – Режим доступу до ресурсу: https://bank.gov.ua/admin_uploads/article/Research_Financial_Literacy_Inclusion_Wel fare_2021.pdf

3. Національний банк України. За останні три роки рівень фінансової грамотності українців поліпшився – результати дослідження. [Електронний ресурс] – Режим доступу до ресурсу: https://bank.gov.ua/ua/news/all/za-ostanni-tri-roki-riven- finansovoyi-gramotnist-ukrayintsiv--polipshivsya--rezultati-doslidjennya

4. Сімейний бюджет. Як фінансова грамотність населення впливає на фінансовий добробут держави? [Електронний ресурс] – Режим доступу до ресурсу: https://simeinyi-budzhet.ua/familybudget/%D1%8F%D0%BA-%D1%84%D1%96%D0%BD%D0%B0%D0%BD%D1%81%D0%BE%D0%B2%D0% B0-%D0%B3%D1%80%D0%B0%D0%BC%D0%BE%D1%82%D0%BD%D1%96%D1% 81%D1%82%D1%8C-%D0%BD%D0%B0%D1%81%D0%B5%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F-%D0%B2/

5. ФГВФО. Сімейний бюджет: як і навіщо його планувати? [Електронний ресурс] – Режим доступу до ресурсу: https://www.fg.gov.ua/articles/48008-simeyniy- byudzhet-yak-i-navishcho-yogo-

planuvati.html

6. Фінансова грамотність. Що? Чому? Як? [Електронний ресурс] – Режим доступу до ресурсу: http://www.fst-ua.info/wp-content/uploads/2019/08/Financial_Literacy_Textbook_Aug2019.pdf

7. CNBC Select. These top budgeting apps sync with your bank accounts, are widely available and come highly rated [Електронний ресурс] – Режим доступу до ресурсу: https://www.cnbc.com/select/best-budgeting-apps/

8. YNAB. Change Your Relationship With Money [Електронний ресурс] – Режим доступу до ресурсу: https://www.ynab.com/

9. Mint. Managing money, made simple [Електронний ресурс] – Режим доступу до ресурсу: https://mint.intuit.com/

10. PocketGuard. Always know where your money goes [Електронний ресурс] – Режим доступу до ресурсу: https://pocketguard.com/

11. GoodBudget. Budget with a why Spend, save, and give toward what's important in life [Електронний ресурс] – Режим доступу до ресурсу: https://goodbudget.com/

12. Mozilla Developer Network (MDN) [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/

13. W3Schools Online Web Tutorials) [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/

14. Stack Overflow - Where Developers Learn, Share, & Build Careers [Електронний ресурс] – Режим доступу до ресурсу: https://stackoverflow.com/

15. Harvard Business Review - Ideas and Advice for Leaders [Електронний ресурс] – Режим доступу до ресурсу: https://hbr.org/

16. McKinsey & Company [Електронний ресурс] – Режим доступу до ресурсу: https://www.mckinsey.com/featured-insights

17. Investopedia [Електронний ресурс] – Режим доступу до ресурсу: https://www.investopedia.com/

18. OWASP Top Ten | OWASP Foundation [Електронний ресурс] – Режим доступу до ресурсу: https://owasp.org/www-project-top-ten/

19. JSON Web Tokens [Електронний ресурс] – Режим доступу до ресурсу: https://jwt.io/

20. World Bank Open Data [Електронний ресурс] – Режим доступу до ресурсу: https://data.worldbank.org/

21. International Monetary Fund (IMF) Data [Електронний ресурс] – Режим доступу до ресурсу: https://data.imf.org/

22. Statista - The Statistics Portal for Market Data, Market Research and Market Studies [Електронний ресурс] – Режим доступу до ресурсу: https://www.statista.com/

23. U.S. Securities and Exchange Commission (SEC) EDGAR Database [Електронний ресурс] – Режим доступу до ресурсу: https://www.sec.gov/edgar

24. Markets data - stock market, bond, equity, commodity prices - FT.com [Електронний ресурс] – Режим доступу до ресурсу: https://markets.ft.com/data

25. The Journal of Finance - Wiley Online Library [Електронний ресурс] – Режим доступу до ресурсу: https://onlinelibrary.wiley.com/journal/15406261

26. Journal of Financial Economics [Електронний ресурс] – Режим доступу до ресурсу: https://www.sciencedirect.com/journal/journal-of-financial-economics

27. Business insights, analysis & perspectives [Електронний ресурс] – Режим доступу до ресурсу: https://www2.deloitte.com/us/en/insights.html

28. Financial Services consulting | McKinsey & Company [Електронний ресурс] – Режим доступу до ресурсу: https://www.mckinsey.com/industries/financial- services/how-we-help-clients

29. А. В. Яковенко, О. О. Коновал Основи програмування: методичні вказівки до виконання комп’ютерних практикумів з дисципліни «Управління ІТ-проектами». Управління ІТ-проектами. НТУУ «КПІ ім. І. Сікорського», 2017. – 47с

30. React. The library for web and native user interfaces [Електронний ресурс] – Режим доступу до ресурсу: https://react.dev/

31. Ecma International [Електронний ресурс] – Режим доступу до

ресурсу: https://ecma-international.org/

32. MUI: The React component library you always wanted [Електронний ресурс] – Режим доступу до ресурсу: https://mui.com/

33. Chart.js. Open source HTML5 Charts for your website [Електронний ресурс] – Режим доступу до ресурсу: https://www.chartjs.org/

34. Express. Node.js web application framework [Електронний ресурс] – Режим доступу до ресурсу: https://expressjs.com/

35. PostgreSQL. The world's most advanced open source database. [Електронний ресурс] – Режим доступу до ресурсу: https://www.postgresql.org/

36. Хацько Н.Є., Гавриленко С.Ю. Методичні вказівки до виконання лабораторної роботи “Розробка діаграми варіантів використання у середовищі Umbrello UML Modeller” – Харків : НТУ «ХПІ», 2019. – 40 с

37. Data Management Information, News and Tips from TechTarget [Електронний ресурс] – Режим доступу до ресурсу: https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD

38. ClickiT. Web Application Architecture: The Latest Guide 2024 [Електронний ресурс] – Режим доступу до ресурсу: https://www.clickittech.com/devops/web-application-architecture/

39. IBM. What is a REST API? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ibm.com/topics/rest-apis

40. node-postgres [Електронний ресурс] – Режим доступу до ресурсу: https://node-postgres.com/

41. Node.js is an open-source, cross-platform JavaScript runtime environment [Електронний ресурс] – Режим доступу до ресурсу: https://nodejs.org/en

42. GitHub [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/

43. React Hook Form. Performant, flexible and extensible forms with easy-to- use validation. [Електронний ресурс] – Режим доступу до ресурсу: https://react-

hook- form.com/

44. Axios is a simple promise based HTTP client for the browser and node.js. [Електронний ресурс] – Режим доступу до ресурсу: https://axios-http.com/

45. QATestLab. Огляд видів тестування [Електронний ресурс] – Режим доступу до ресурсу: https://training.qatestlab.com/blog/technical-articles/review-the-types-of-testing/