

ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python»

на здобуття освітнього ступеня бакалавра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

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

\_\_\_\_\_ Євгенія ДУБОВА  
(підпис)

Виконала: здобувачка вищої освіти групи ПД-43

\_\_\_\_\_ Євгенія ДУБОВА

Керівник: \_\_\_\_\_ Володимир САДОВЕНКО  
к.ф.-м.н., доцент

Рецензент: \_\_\_\_\_

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Дубовій Євгенії Борисівні \_\_\_\_\_

1. Тема кваліфікаційної роботи: «Розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python»  
керівник кваліфікаційної роботи: к.ф-м.н., доцент Володимир САДОВЕНКО  
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи:

3.1 Середовище розробки PyCharm

3.2 Мова програмування Python та її бібліотеки

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд та аналіз існуючих мобільних застосунків.

2. Огляд та аналіз інструментів використаних для реалізації проекту.

3. Проектування та створення візуальної частини проекту.

4. Програмна реалізація мобільного застосунку.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Діаграма класів.
6. Екранні форми.
7. Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Дослідження програмних засобів	14.03-19.03.2024	
4	Проектування застосунку для інтерактивного вивчення основам фінансової грамотності	20.03-25.03.2024	
5	Програмна реалізація застосунку	26.03-23.04.2024	
6	Оформлення пояснювальної записки	24.04-28.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувачка вищої освіти

\_\_\_\_\_

(підпис)

Євгенія ДУБОВА

Керівник

кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Володимир САДОВЕНКО

## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 45 стор., 1 табл., 23 рис., \_ джерел.

*Мета роботи* – підтримка процесу вивчення основ фінансової грамотності серед користувачів за допомогою мобільного застосунку.

*Об'єкт дослідження* – процес вивчення основ фінансової грамотності.

*Предмет дослідження* – мобільний застосунок для вивчення основ фінансової грамотності.

*Короткий зміст роботи:* В роботі проаналізовано аналоги мобільних застосунків, їх функціонал та можливості. Також був проведений аналіз ІТ технологій які будуть використовуватися під час реалізації проекту, їх можливості, особливості та функціонал. Можливість перегляду уроків, проходження тестів, перегляд результатів. В роботі було використано мову програмування Python, та її бібліотеки KivyMD.

Сферою використання застосунку є фінанси: їх освоєння, грамотне управління

**КЛЮЧОВІ СЛОВА:** Python, KivyMD, PyCharm, фінансова грамотність.

## ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Визначення цільової аудиторії та основні аспекти фінансової грамотності, які будуть включені до застосунку.....	7
1.2 Сучасні підходи до освоєння фінансової грамотності .....	10
1.3 Інтерактивне навчання.....	11
2. АНАЛІЗ АНАЛОГІВ .....	13
2.1 Аналіз аналогів .....	13
2.2 Визначення основних функцій та вимог до застосунку.....	22
3. АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ .....	27
4. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	34
4.1 Діаграма прецедентів .....	35
4.2 Діаграма класів .....	37
4.3 Проектування графічного інтерфейсу користувача .....	38
4.4 Розробка програмного забезпечення.....	39
ВИСНОВКИ.....	50
ПЕРЕЛІК ПОСИЛАНЬ .....	51
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ .....	53
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ .....	60

## ВСТУП

У сучасному світі зростає необхідність у фінансовій грамотності серед населення. Кожна людина повинна мати базові знання з управління власними фінансами, розуміння основних понять і принципів, що стосуються бюджетування, інвестування, кредитів та інших фінансових аспектів. Ці навички є важливими для досягнення фінансового благополуччя та забезпечення стабільного майбутнього.

У цьому контексті розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності стає актуальною та значущою. Такий застосунок може стати ефективним інструментом для навчання та самонавчання, забезпечуючи користувачам доступ до освітнього контенту в будь-який зручний для них час і місце.

Метою даного дипломного проекту є розробка такого мобільного застосунку з використанням мови програмування Python. Python обраний для реалізації проекту через його простоту та гнучкість, а також через широкий набір бібліотек і фреймворків, що дозволять швидко та ефективно реалізувати необхідний функціонал. Мобільний застосунок буде спрямований на поширення базових знань з фінансів серед широкого кола користувачів, включаючи тих, хто має обмежений доступ до традиційної освіти. Застосунок буде надавати користувачам доступ до навчальних матеріалів у формі статей, інтерактивних вправ тощо. Крім того, буде реалізована можливість відстеження прогресу користувачів.

Підсумовуючи, розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python є актуальним та значущим проектом, який має на меті сприяти підвищенню рівня фінансової грамотності серед населення та сприяти досягненню фінансового благополуччя.

## 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Визначення цільової аудиторії та основні аспекти фінансової грамотності, які будуть включені до застосунку

Фінанси підтримують економіку, стимулюючи інновації та розвиток бізнесу. Вони є силою, яка може змінити життя людей та покращити їхнє матеріальне становище. Знання про фінансові принципи і уміння ефективно ними керувати дозволяють кожній особі планувати своє майбутнє, бути фінансово незалежною і досягати поставлених цілей. Крім того, фінансова стабільність має важливе значення для безпеки суспільства в цілому. Вона сприяє уникненню фінансових криз, забезпечує сталість економіки та дозволяє людям відчувати впевненість у майбутньому. Забезпечення доступу до фінансових послуг для всіх верств населення є ключовим аспектом соціальної справедливості, оскільки це допомагає зменшити нерівності і забезпечує рівні можливості для всіх.

Таким чином, фінанси мають величезне значення для сучасного суспільства і впливають на різні аспекти життя кожної людини. Розуміння їх ролі і важливості допоможе зрозуміти, як краще управляти фінансами, планувати майбутнє і забезпечити стабільність і розвиток для себе та для інших.

Національний банк України (НБУ) та Дитячий фонд ООН (ЮНІСЕФ) провели опитування, щоб оцінити рівень фінансової грамотності серед молоді. Участь взяли понад 12 тисяч молодих людей віком від 14 до 34 років. Результати показують, що більшість респондентів оцінюють свої знання як середні. Головними причинами низького рівня фінансової грамотності були вказані відсутність обговорення фінансових питань у сім'ї та обмежений досвід користування фінансовими продуктами через молодий вік. Більшість учасників опитування вважають, що навчання фінансової грамотності має бути обов'язковим у школах. Вони висловили інтерес до таких тем, як заощадження, планування витрат, інвестування та управління боргами. Опитування також

показало, що багато молодих людей стикаються з труднощами при використанні фінансових послуг через брак інформації та довіри до фінансових установ. Для підвищення рівня фінансової грамотності молодь пропонує організувати спеціальні курси та тренінги, проводити просвітницькі кампанії та залучати до цього процесу батьків та вчителів.

Ці результати показують необхідність активнішого впровадження програм фінансової освіти, щоб молоді люди могли ефективно управляти своїми фінансами та приймати зважені фінансові рішення.

Фінансова грамотність надає людям необхідні знання та навички для ефективного управління своїми фінансами, що включає кілька ключових аспектів:

**Особистий бюджет:** Фінансова грамотність допомагає людині керувати своїми доходами та витратами, складати бюджет, планувати витрати та зберігати гроші на майбутнє. Це дозволяє покращити фінансове становище та забезпечити фінансову стабільність.

**Планування майбутнього:** Фінансова грамотність допомагає людині планувати своє майбутнє, включаючи витрати на освіту, житло, медичні послуги та пенсію. Це дозволяє створювати цілі та працювати над їх досягненням.

**Управління кредитами та позиками:** Розуміння умов кредитів і позик, вибір вигідних умов, розрахунок витрат та своєчасне погашення зобов'язань – все це є важливими аспектами фінансової грамотності..

**Захист від фінансових ризиків:** Люди з фінансовою грамотністю здатні прогнозувати та захищати себе від різних фінансових ризиків, таких як втрата роботи, непередбачені витрати або зміни в ринкових умовах.

**Здатність до інвестування:** Розуміння основ інвестування допомагає людині збільшувати свої фінансові активи та забезпечує можливість збільшити свої доходи та відкрити нові можливості для фінансового зростання.

Всі ці аспекти підкреслюють важливість фінансової грамотності в сучасному світі. Люди, які мають ці навички, як правило мають кращі фінансові перспективи і можуть досягати більшого успіху в житті. Фінансова грамотність -важлива навичка, яка допомагає людям розуміти фінансові концепції і приймати



обґрунтовані рішення зі своїми грошима. Додаток для вивчення основ фінансової грамотності може бути корисним інструментом для тих, хто бажає покращити свої фінансові знання.

У сфері освіти додаток може бути використаний для навчання учнів та студентів фінансовим аспектам, надаючи їм доступ до теоретичних матеріалів та тестів для закріплення знань. Також підходить для самонавчання - це ще один можливий сценарій використання додатку, де він надасть доступну інформацію про особисті фінанси, інвестиції та борг.

Соціальні організації можуть використовувати додаток для навчання своїх клієнтів фінансовій грамотності, що сприятиме покращенню їх фінансового стану. Навіть фінансові установи, такі як банки та страхові компанії, можуть використовувати додаток для підвищення рівня фінансової грамотності серед своїх клієнтів і зменшення ризиків. У цілому, додаток для вивчення фінансової грамотності може стати корисним інструментом для розвитку фінансових знань та навичок у різних сферах життя.

Також важливо зазначити, що фінансова грамотність не тільки допомагає людям керувати своїми грошима більш ефективно, але й забезпечує їх фінансову стабільність у майбутньому. Інвестування, планування пенсії, управління боргами - все це вимагає розуміння фінансових концепцій і вміння приймати обґрунтовані рішення.

Застосунок для вивчення фінансової грамотності може також допомогти користувачам розуміти складні фінансові продукти, такі як акції, облігації, страхування і т.д. Це може збільшити їхню кваліфікацію в області фінансів і допомогти їм зробити вигідні інвестиційні рішення.

Необхідно також враховувати, що фінансова грамотність є важливою для всіх без винятку. Вона може допомогти знизити рівень фінансового стресу, покращити якість життя і забезпечити фінансову безпеку у майбутньому. Тому розвиток цієї навички через використання додатків для вивчення фінансової грамотності є важливим кроком для кожної людини.

## 1.2 Сучасні підходи до освоєння фінансової грамотності

Сучасні підходи до освоєння фінансової грамотності відзначаються інноваційністю, інтерактивністю та врахуванням потреб сучасного суспільства.

Персоналізоване навчання: більшість сучасних застосунків для фінансової грамотності намагаються створити персоналізовані навчальні шляхи для користувачів. Це означає адаптацію змісту та завдань до індивідуальних потреб, рівня знань та цілей користувача.

Використання ігрових елементів: багато застосунків використовують гейміфікацію для залучення користувачів. Сюди входять ігрові завдання, досягнення, таблиці лідерів та інші інтерактивні елементи, що полегшують процес навчання та роблять його цікавішим і заохочують до активної участі.

Застосунки з мобільною доступністю: з урахуванням широкого поширення смартфонів і планшетів, багато застосунків для фінансової грамотності доступні на мобільних пристроях. Це дозволяє користувачам вивчати фінансові концепції в будь-який час і в будь-якому місці. Використання інтерактивних методів навчання: застосунки надають можливість використання різноманітних інтерактивних методів навчання, таких як відеоуроки, аудіо-підкасти, інтерактивні тести та квізи. Це допомагає залучити увагу користувачів та полегшує засвоєння матеріалу.

Фокус на практичних аспектах: багато застосунків зосереджуються на наданні практичних навичок і порад з управління особистими фінансами. Це включає в себе створення бюджетів, планування витрат, розуміння опцій кредитування та інвестицій.

Моніторинг та аналіз результатів: деякі застосунки надають можливість користувачам відстежувати свій прогрес у навчанні та аналізувати свої фінансові звички. Це допомагає створити свідомість про власні фінанси та підвищити фінансову грамотність.

Навчання за допомогою реальних прикладів: багато застосунків надають навчальний матеріал, використовуючи реальні життєві ситуації та приклади. Це

дозволяє користувачам легше зрозуміти та застосовувати знання у практичних умовах.

Ці підходи допомагають створити ефективні та привабливі засоби для навчання фінансової грамотності, які сприяють залученню користувачів та підвищенню рівня їх знань у цій сфері.

### **1.3 Інтерактивне навчання**

Термін "інтерактивний" походить з англійської мови та має значення "взаємодіючий". Існують різні підходи щодо визначення інтерактивного навчання. Вчені визначають це як інтерактивне навчання: "Інтерактивність означає здатність взаємодіяти або переходити в режим розмови, діалогу з чимось (наприклад, з комп'ютером) або з кимось - не з людиною". Отже, інтерактивне навчання - це переважно навчання, під час якого відбувається взаємодія між учням і матеріалом..

У сучасному світі розважальні підходи до навчання стають дедалі популярнішими. У цифровому суспільстві стає необхідним використовувати програми для навчання. Такий інтерактивний підхід дозволяє створити електронну освітню систему, що забезпечує цікавий та ефективний процес навчання.

Великою перевагою використання навчальної програми є те, що вона дозволяє створити інтерактивне середовище, в якому люди можуть взаємодіяти з матеріалом. Це робить навчання більш доступним та зрозумілим для користувачів різних вікових груп. Крім того, досвід цифрового навчання дозволяє створювати ігрові завдання, які стимулюють творчий підхід, логічне мислення та проблемне мислення.

Однією з основних переваг використання програми навчання є те, що процес можна персоналізувати. Кожен може працювати у своєму власному темпі та на своєму рівні. Програма надає можливість відстежувати прогрес кожного та адаптувати завдання до його потреб. Це робить навчання більш ефективним та корисним для всіх. Все більше і більше навчальних закладів та вчителів використовують навчальні програми, щоб покращити якість освіти та зробити

навчання приємнішим. Досвід цифрової освіти стає невід'ємним елементом сучасної освіти, допомагаючи людям розвивати навички, необхідні для успіху у цифровому світі.

Завдяки використанню додатків у навчальному процесі, здобувачі освіти мають можливість самостійно вивчати та досліджувати різні предмети. Інтерактивні додатки допомагають зробити навчання цікавим, ефективним та доступним. Вони сприяють розвитку електронної освіти та стимулюють студентів до активного вивчення різних предметів. Використання додатків у навчанні відкриває нові можливості для розвитку освіти та покращення результатів.

Переваги використання цифрових програм для навчання:

- Можливість інтерактивної взаємодії з матеріалами навчання
- Стимулювання активності та зацікавленості студентів
- Можливість отримання миттєвого зворотного зв'язку
- Розвиток навичок та здібностей у більш ефективний спосіб
- Збереження уваги студентів на протязі тривалого часу

Завдяки ігровим елементам, викликам та досягненням, мобільні додатки можуть збільшити мотивацію користувачів до засвоєння матеріалу. Вони також можуть адаптуватися до різних стилів навчання, рівнів знань та індивідуальних потреб користувачів, забезпечуючи ефективніше сприйняття матеріалу. Деякі додатки надають можливість спілкування з іншими користувачами, вчителями або експертами у віртуальному середовищі, що дозволяє обмінюватися досвідом та отримувати додаткову підтримку. Загалом, інтерактивне навчання з використанням мобільних додатків створює динамічне та захопливе навчальне середовище, яке сприяє ефективному засвоєнню бажаних знань та навичок.

## 2. АНАЛІЗ АНАЛОГІВ

### 2.1 Аналіз аналогів

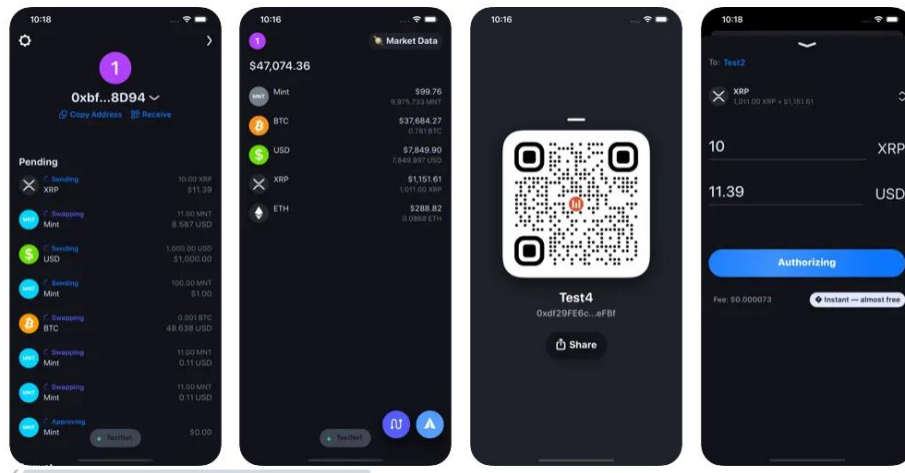


Рис. 2.1 Екранні форми застосунку Mint

**Mint** - це популярний застосунок для управління особистими фінансами, розроблений компанією Intuit. Він пропонує комплексний підхід до фінансового планування і контролю, що робить його корисним для користувачів, які прагнуть поліпшити свої фінансові звички та досягти фінансових цілей.

Основна функціональність Mint полягає у централізації фінансових даних. Користувачі можуть підключати свої банківські рахунки, кредитні картки, інвестиційні портфелі та інші фінансові акаунти, що дозволяє автоматично збирати інформацію про транзакції і баланс. Це забезпечує повну картину фінансового стану користувача в реальному часі. Застосунок надає можливість скласти бюджет і дозволяє користувачам планувати витрати за різними категоріями, такими як харчування, транспорт та розваги. Mint автоматично відстежує витрати і порівнює їх з встановленими бюджетними лімітами, надаючи користувачам вчасні сповіщення, якщо вони наближаються до перевищення бюджету. Ще однією важливою особливістю є можливість відстеження кредитного рейтингу. Mint надає безкоштовний доступ до кредитного скорингу, що дозволяє користувачам

слідкувати за своїм фінансовим здоров'ям і отримувати поради щодо його покращення.

З точки зору безпеки, Mint використовує багаторівневу систему захисту даних, включаючи шифрування на рівні банків, що забезпечує високий рівень захисту персональної інформації. Крім того, застосунок пропонує аналітику та звіти, які допомагають користувачам зрозуміти свої фінансові звички та знайти способи заощадження. Візуалізація даних у вигляді графіків та діаграм робить фінансову інформацію більш доступною і зрозумілою.

Mint має кілька значних переваг. Перш за все, його можливість централізації фінансових даних є великою перевагою для користувачів. Всі банківські рахунки, кредитні картки, інвестиції та інші фінансові активи можна підключити в одному місці, що дозволяє бачити повну картину фінансового стану в реальному часі. Це зручність робить фінансовий менеджмент значно легшим і менш стресовим. Застосунок також пропонує потужні інструменти для бюджетування. Користувачі можуть створювати детальні бюджети для різних категорій витрат і отримувати сповіщення, коли вони наближаються до перевищення встановлених лімітів. Це допомагає зберігати фінансову дисципліну і планувати витрати більш ефективно. Ще однією важливою перевагою є можливість відстеження кредитного рейтингу. Безкоштовний доступ до кредитного скорингу і рекомендації щодо його покращення надають користувачам цінну інформацію про їх фінансове здоров'я. Mint також вирізняється високим рівнем безпеки. Використання шифрування на рівні банків забезпечує захист персональних даних і фінансової інформації користувачів. Це дає впевненість у тому, що дані залишаються конфіденційними і захищеними від несанкціонованого доступу. Додатково, застосунок пропонує аналітичні інструменти та звіти, які допомагають користувачам зрозуміти їх фінансові звички та знайти способи заощадження.

Візуалізація даних у вигляді графіків та діаграм робить фінансову інформацію більш доступною і зрозумілою. Однак, Mint має і свої недоліки. Наприклад, інтерфейс може бути трохи перевантаженим, особливо для нових користувачів, що може ускладнити навігацію та використання всіх функцій.

Незважаючи на зручність централізації фінансових даних, інтеграція з деякими меншими банками або фінансовими установами може бути обмеженою або нестабільною, що може викликати певні незручності. Крім того, хоча базова версія Mint є безкоштовною, вона підтримується рекламою, що може відволікати користувачів. Деякі функції або звіти можуть вимагати преміум-підписки, що не завжди виправдовує свої витрати для кожного користувача. Нарешті, питання безпеки даних залишається актуальним. Хоча Mint використовує високий рівень шифрування, зберігання великої кількості фінансової інформації в одному місці може бути привабливою мішенню для кіберзлочинців, що викликає певні занепокоєння серед користувачів щодо безпеки їх даних.

Наступним чином, Mint є потужним і зручним інструментом для управління особистими фінансами, але не позбавленим деяких недоліків, які можуть впливати на досвід користувачів в залежності від їх індивідуальних потреб і уподобань.



Рис. 2.2 YNAB

**YNAB (You need a budget)** - це популярний застосунок для управління особистими фінансами, що має чітко визначену філософію бюджетування, спрямовану на допомогу користувачам в управлінні їхніми грошима і досягненні фінансових цілей. Його основною особливістю є система "zero-based budgeting", яка передбачає призначення кожного долара конкретному завданню. Застосунок YNAB побудований на чотирьох основних правилах: "Надайте кожному долару

завдання", "Підготуйтеся до майбутніх витрат", "Погасіть борги" і "Адаптуйтеся до змін". Ці правила допомагають користувачам встановлювати пріоритети у витратах, зберігати гроші на майбутнє, уникати заборгованості та гнучко реагувати на фінансові зміни. Одна з основних переваг YNAB – це його освітня складова. Застосунок надає широкий спектр навчальних матеріалів, включаючи вебінари, відеоуроки та статті, які допомагають користувачам зрозуміти основи фінансового управління та ефективно використовувати інструменти YNAB. Такий підхід сприяє розвитку фінансової грамотності та дисципліни.



Рис. 2.3 Екранні форми застосунку YNAB

Інтерфейс YNAB простий і зрозумілий, що полегшує його використання навіть для тих, хто не має глибоких знань у фінансах. Зручна система категорій дозволяє легко відстежувати витрати і коригувати бюджети в реальному часі. Застосунок підтримує синхронізацію з банківськими рахунками, що автоматизує процес введення транзакцій і зменшує ризик помилок. З точки зору безпеки, YNAB використовує шифрування на рівні банків та інші засоби захисту даних, що забезпечує високий рівень безпеки користувацької інформації. Однак YNAB має й деякі недоліки. По-перше, це платний сервіс, що може бути стримуючим фактором для деяких користувачів. Щомісячна або щорічна підписка може здатися занадто



дорогою, особливо якщо порівнювати з іншими безкоштовними альтернативами на ринку.

Крім того, інтерфейс, хоч і простий, вимагає певного часу для звикання, особливо для новачків у бюджетуванні. Деяким користувачам може здатися складним спочатку розібратися з усіма функціями та правилами, які пропонує YNAB. Ще один аспект, який варто зазначити, – це відсутність деяких функцій, які можуть бути корисними для досвідчених користувачів, наприклад, більш детальна аналітика або розширені інвестиційні інструменти. Застосунок орієнтований переважно на базове управління бюджетом і може не відповідати всім потребам користувачів, які шукають більш комплексне рішення. В цілому, YNAB є потужним інструментом для управління особистими фінансами, який допомагає користувачам встановлювати та досягати фінансових цілей завдяки чіткій системі бюджетування та навчальним матеріалам. Хоча це платний сервіс, багато користувачів вважають, що його вартість виправдовується ефективністю та результатами, які він приносить. YNAB (You Need a Budget) має кілька значних переваг, які роблять його привабливим для користувачів, які прагнуть покращити свої фінансові навички та дисципліну. Основна перевага полягає в системі "zero-based budgeting", яка дозволяє користувачам призначати кожному долару конкретне завдання, що допомагає точно контролювати витрати та заощадження. Це сприяє більш усвідомленому підходу до фінансового планування та допомагає уникати імпульсивних витрат.

Навчальна складова YNAB є ще однією значущою перевагою. Застосунок пропонує багатий набір навчальних матеріалів, включаючи вебінари, відеоуроки та статті, які сприяють підвищенню фінансової грамотності користувачів. Такий підхід допомагає не лише використовувати застосунок ефективно, але й розвивати загальні навички фінансового управління.

Інтерфейс YNAB простий і зрозумілий, що робить його зручним для користування навіть для новачків у бюджетуванні. Застосунок дозволяє легко відстежувати витрати за допомогою зручної системи категорій, що забезпечує прозорість і чіткість у фінансовому плануванні. Синхронізація з банківськими

рахунками автоматизує введення транзакцій, що зменшує ризик помилок і економить час.

Безпека даних у YNAB також на високому рівні. Застосунок використовує шифрування на рівні банків, що забезпечує надійний захист персональної та фінансової інформації користувачів. Однак, YNAB має й деякі недоліки. Головним серед них є те, що це платний сервіс. Щомісячна або щорічна підписка може бути значною витратою для деяких користувачів, особливо у порівнянні з іншими безкоштовними альтернативами на ринку. Ціна може стати бар'єром для тих, хто лише починає цікавитися фінансовим управлінням.

Загалом, YNAB є потужним і ефективним інструментом для управління особистими фінансами. Його система бюджетування та освітні матеріали сприяють підвищенню фінансової грамотності та дисципліни. Проте, вартість сервісу та певна складність на початковому етапі можуть бути перешкодами для деяких користувачів.



Рис 2.4 Acorns

**Acorns**— це інвестиційний застосунок, який робить акцент на мікроінвестиціях і автоматичному інвестуванні дрібних сум грошей. Його головна ідея полягає у використанні "здачі" з кожної покупки для інвестування в диверсифікований портфель ETF (біржових інвестиційних фондів). Однією з основних переваг Acorns є простота використання. Застосунок автоматично

округлює кожну покупку до найближчого цілого долара і інвестує різницю. Це дозволяє користувачам інвестувати без необхідності робити великі вкладення або мати глибокі знання у фінансовій сфері. Такий підхід робить інвестування доступним для широкого кола користувачів, включаючи молодь і тих, хто тільки починає цікавитися фінансами. Acorns також пропонує декілька додаткових функцій, таких як Recurring Investments (регулярні інвестиції), Found Money (кешбек від партнерів, що автоматично інвестується), і Retirement (рахунки IRA для пенсійних заощаджень). Ці функції допомагають користувачам розширити свої інвестиційні можливості і більш ефективно планувати майбутнє. З точки зору безпеки, Acorns використовує шифрування на рівні банків та інші передові заходи захисту даних, що забезпечує високий рівень безпеки фінансової інформації користувачів. Це важливо для збереження довіри користувачів і захисту їхніх активів.

Acorns має інтуїтивно зрозумілий інтерфейс, який легко освоїти. Користувачі можуть швидко налаштувати свої акаунти, підключити банківські рахунки і почати інвестувати за декілька хвилин. Застосунок також надає освітні матеріали, які допомагають користувачам зрозуміти основи інвестування і розвивати фінансову грамотність. Однак застосунок має недоліки. По-перше, щомісячна плата за обслуговування, хоч і невелика, може здаватися невиправданою для користувачів з низьким рівнем інвестицій. Зокрема, для тих, хто має невеликі суми на рахунках, щомісячні збори можуть суттєво впливати на загальну прибутковість інвестицій. Крім того, Acorns пропонує обмежені можливості для досвідчених інвесторів. Застосунок не дозволяє користувачам обирати конкретні акції чи інші інвестиційні інструменти, оскільки всі інвестиції автоматично розподіляються серед ETF на основі обраного портфеля. Це може не відповідати потребам тих, хто хоче мати більше контролю над своїми інвестиціями або прагне інвестувати в конкретні активи. Ще одним недоліком є те, що Acorns не надає можливості детального аналізу і моніторингу інвестицій. Хоча застосунок надає основні показники продуктивності портфеля, більш детальний аналіз або розширені інструменти для інвесторів відсутні.

У підсумку, Acorns є чудовим варіантом для тих, хто шукає простий спосіб почати інвестувати, особливо для новачків та молодих людей. Зручність автоматичного інвестування і доступність функцій для регулярних вкладень роблять його привабливим для багатьох користувачів. Проте досвідчені інвестори можуть вважати його обмеженим через відсутність можливості вибору конкретних інвестицій та детального аналізу.



Рис 2.5 Robinhood

**Robinhood** - це популярний застосунок для торгівлі акціями, ETF, опціонами і криптовалютами, який відомий своїм підходом до комісій: торгівля через Robinhood є безкоштовною, що значно відрізняє його від традиційних брокерських компаній. Це робить платформу особливо привабливою для новачків та молодих інвесторів, які прагнуть заощадити на комісіях.

Однією з головних переваг Robinhood є його простота у використанні. Інтерфейс застосунку є інтуїтивно зрозумілим і привабливим, що дозволяє легко орієнтуватися навіть тим, хто не має досвіду в торгівлі. Користувачі можуть швидко знаходити і купувати акції, ETF та інші фінансові інструменти. Також додаток пропонує реєстрацію та верифікацію акаунтів, які можна пройти всього за кілька хвилин.

Застосунок також надає корисні функції для трейдерів, такі як можливість торгівлі в режимі реального часу, сповіщення про зміну цін та новини ринку. Це

дозволяє користувачам швидко реагувати на ринкові зміни та приймати інформовані інвестиційні рішення. Robinhood також підтримує торгівлю криптовалютами, що додає ще більше можливостей для інвесторів.

Robinhood пропонує кілька додаткових функцій, таких як Robinhood Gold – преміум-сервіс, який надає доступ до маржинальної торгівлі, розширених аналітичних даних та інших переваг. Ця опція є корисною для більш досвідчених інвесторів, які хочуть мати більше інструментів для аналізу ринку.

Однак, Robinhood має і свої недоліки. Одним із головних є обмеженість освітніх ресурсів та аналітичних інструментів. У порівнянні з іншими брокерськими платформами, які надають широкий спектр освітніх матеріалів, детальних звітів та аналітики, Robinhood може здатися недостатньо інформативним для досвідчених інвесторів. Це може ускладнювати прийняття рішень для тих, хто потребує глибокого аналізу ринку.

Ще одним значним недоліком є кілька випадків технічних збоїв, які відбувалися в минулому. Наприклад, у періоди високої волатильності ринку, платформа зазнавала перебоїв, що призводило до неможливості проводити операції. Це викликало обурення серед користувачів, оскільки такі збої можуть призвести до значних фінансових втрат.

Крім того, Robinhood критикували за модель монетизації, відому як Payment for Order Flow (PFOF). Це означає, що компанія отримує компенсацію від маркет-мейкерів за напрямок ордерів користувачів до них. Деякі експерти вважають, що це може створити конфлікт інтересів і впливати на якість виконання ордерів.

Також варто зазначити, що Robinhood не пропонує можливості інвестування в більш складні фінансові інструменти, такі як ф'ючерси або форекс, що може обмежувати можливості для деяких інвесторів.

Підсумовуючи, Robinhood є відмінним вибором для початківців та тих, хто шукає простий спосіб торгівлі акціями, ETF та криптовалютами без комісій. Зручний інтерфейс і можливість швидкого входу на ринок роблять його привабливим для багатьох користувачів. Однак, обмежені освітні ресурси, технічні

збої і модель монетизації можуть стати недоліками для більш досвідчених або вимогливих інвесторів.

Таблиця 2.1

## Порівняння існуючих аналогів

Критерій / Додаток	Mint	YNAB	Acorns	Robinhood	Financial education
Перевірка рівня знань	-	+	-	-	+
Освітній контент	Навчальні статті, блоги	Онлайн курси, вебінари	Відеоуроки, статті	Фінансові новини, блоги	Навчальні матеріали
Різноманіття тем	Бюджетування, інвестиції, управління боргами	Управління особистими фінансами	Розуміння фінансів, інвестування	Інвестування, фінансові ринки	Кредитування, фінансування, інвестиції, страхування
Рівень складності	Підходить для початківців	Підходить для просунутих користувачів	Підходить для початківців	Підходить для середнього рівня	Підходить для початківців
Відстеження прогресу	-	+	-	+	+

## 2.2 Визначення основних функцій та вимог до застосунку

Основна функція застосунку для вивчення основ фінансової грамотності полягає у наданні користувачам інструментів і знань, необхідних для ефективного управління своїми фінансами. Це включає кілька ключових аспектів, які разом створюють всебічний освітній досвід.

Перш за все, застосунок пропонує освітні матеріали у різних форматах, таких як статті, відеоуроки, інтерактивні курси та вебінари. Ці матеріали охоплюють базові концепції фінансової грамотності, включаючи бюджетування, заощадження, інвестування, управління боргами та кредитними рейтингами. Мета полягає в тому, щоб користувачі отримали ґрунтовне розуміння фінансових принципів і змогли застосувати ці знання у своєму житті.

Крім теоретичних знань, застосунок пропонує інтерактивні інструменти, які допомагають користувачам застосовувати отримані знання на практиці. Наприклад, калькулятори бюджетів дозволяють створювати персоналізовані бюджети, враховуючи доходи та витрати. Інструменти для відстеження витрат допомагають користувачам бачити, куди йдуть їхні гроші, і знаходити способи скорочення непотрібних витрат.

Застосунок також включає симулятори інвестування, які дозволяють користувачам навчатися інвестуванню без ризику втратити реальні гроші. Ці симулятори імітують реальні ринкові умови і дозволяють користувачам експериментувати з різними інвестиційними стратегіями, розвиваючи свої навички і впевненість у прийнятті інвестиційних рішень.

Ще одним важливим компонентом є персоналізовані рекомендації та поради. Застосунок аналізує фінансові дані користувача і надає індивідуальні поради щодо управління фінансами. Наприклад, він може рекомендувати способи зниження боргів, збільшення заощаджень або оптимізацію витрат. Ці поради базуються на аналізі доходів, витрат і фінансових цілей користувача.

Важливою функцією є відстеження фінансових цілей. Застосунок дозволяє користувачам встановлювати короткострокові та довгострокові фінансові цілі, такі як накопичення на відпустку, купівля автомобіля або пенсійні заощадження. Інструменти для відстеження прогресу допомагають користувачам бачити, як вони просуваються до своїх цілей, і отримувати мотивацію для продовження роботи над ними.

Застосунок також може включати соціальні функції, які дозволяють користувачам обмінюватися досвідом і порадами з іншими людьми. Це можуть бути форуми, групи підтримки або можливість підключитися до фінансових коучів або менторів. Соціальна підтримка і взаємодія можуть значно підвищити ефективність навчання і мотивацію.

З точки зору технологій, застосунок використовує дані про фінансову поведінку користувачів для надання більш точних і релевантних порад. Це може включати аналіз витрат за допомогою підключення до банківських рахунків або

кредитних карток, а також використання алгоритмів машинного навчання для прогнозування фінансових потреб і ризиків.

Безпека є критично важливим аспектом, тому застосунок використовує сучасні методи шифрування і захисту даних для забезпечення конфіденційності та безпеки фінансової інформації користувачів. Це включає багаторівневу аутентифікацію і постійний моніторинг системи на наявність потенційних загроз.

У підсумку, основна функція застосунку для вивчення основ фінансової грамотності полягає в забезпеченні користувачів всебічною освітою та практичними інструментами для ефективного управління своїми фінансами. Це включає навчальні матеріали, інтерактивні інструменти, персоналізовані рекомендації, відстеження фінансових цілей і соціальну підтримку, що разом сприяє підвищенню фінансової грамотності та впевненості користувачів у прийнятті фінансових рішень.

Застосунок для вивчення основ фінансової грамотності повинен включати низку ключових вимог, щоб ефективно навчати користувачів і допомагати їм застосовувати знання на практиці. По-перше, освітні матеріали повинні бути доступні в різних форматах, таких як інтерактивні курси, відеоуроки, статті та вебінари. Це забезпечить різноманітність навчання і допоможе користувачам засвоювати інформацію найзручнішим для них способом.

Додатково, застосунок повинен пропонувати інтерактивні інструменти для практичного застосування знань. Наприклад, калькулятори для створення персональних бюджетів, інструменти для відстеження витрат та симулятори інвестування. Такі інструменти допоможуть користувачам зрозуміти, як фінансові принципи працюють у реальному житті, без ризику втратити реальні гроші.

Персоналізовані рекомендації є важливим аспектом, що робить застосунок більш корисним. На основі аналізу фінансової поведінки користувача, застосунок повинен надавати індивідуальні поради щодо управління грошима, зниження боргів, збільшення заощаджень та оптимізації витрат. Це допоможе користувачам отримати конкретні рекомендації, які враховують їхні унікальні фінансові ситуації.



Функція відстеження фінансових цілей також є необхідною. Користувачі повинні мати можливість встановлювати короткострокові та довгострокові фінансові цілі, такі як накопичення на відпустку або пенсійні заощадження, і відстежувати свій прогрес. Це підтримає мотивацію і допоможе зосередитися на досягненні фінансових цілей.

Застосунок повинен включати соціальні функції, які дозволяють користувачам спілкуватися з іншими, ділитися досвідом і отримувати підтримку. Це можуть бути форуми, групи підтримки або можливість підключення до фінансових коучів чи менторів. Соціальна підтримка може значно підвищити ефективність навчання.

Технологічна складова повинна забезпечувати високий рівень безпеки даних. Це включає використання сучасних методів шифрування, багаторівневу аутентифікацію і постійний моніторинг на наявність потенційних загроз. Безпека є критично важливою, оскільки застосунок обробляє конфіденційну фінансову інформацію.

Також важливо, щоб застосунок був сумісний з різними пристроями і операційними системами, щоб користувачі могли зручно використовувати його на своїх смартфонах, планшетах чи комп'ютерах. Інтерфейс повинен бути інтуїтивно зрозумілим і легко налаштовуваним, щоб користувачі могли швидко знаходити потрібну інформацію та інструменти.

Адаптованість до різних рівнів користувачів: Застосунок повинен бути придатним як для початківців, так і для досвідчених користувачів. Це означає, що він повинен надавати навчальний контент, який відповідає різному рівню знань і досвіду фінансів.

Підтримка міжнародних стандартів та законодавства: Якщо застосунок призначений для використання в різних країнах, він повинен дотримуватися міжнародних стандартів фінансової грамотності і враховувати особливості фінансового законодавства кожної країни.

Можливість синхронізації даних: Якщо застосунок доступний на декількох пристроях, він повинен підтримувати синхронізацію даних між ними. Це дозволить

користувачам зручно працювати зі своїми фінансовими даними на будь-якому пристрої. Підтримка мов: Застосунок повинен мати можливість вибору мови, щоб користувачі з різних країн могли використовувати його на своїй рідній мові.

Оновлення і підтримка: Регулярні оновлення та підтримка користувачів в разі виникнення питань або проблем є важливими аспектами, які допомагають забезпечити якість та ефективність застосунку впродовж тривалого часу.

Взаємодія з іншими фінансовими сервісами: Якщо користувачі вже використовують інші фінансові сервіси, застосунок може надавати можливість інтеграції з ними. Це дозволить зручно використовувати всі фінансові інструменти на одній платформі.

В цілому, застосунок для вивчення основ фінансової грамотності повинен забезпечувати всебічну освіту, практичні інструменти, персоналізовані рекомендації, відстеження фінансових цілей, соціальну підтримку, високий рівень безпеки та зручність використання. Це допоможе користувачам не тільки отримати знання, але й успішно застосовувати їх у своєму повсякденному житті.

### 3. АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

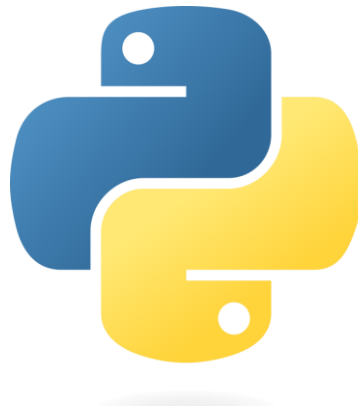


Рис. 3.1 Python

**Python** - це високорівнева мова програмування загального призначення, яка відома своєю простотою та легкістю використання. Вона має чистий і зрозумілий синтаксис, що робить її ідеальним вибором як для початківців, так і для досвідчених розробників. Одним з головних принципів Python є "читабельність коду", що сприяє його широкому поширенню серед програмістів.

Python підтримує об'єктно-орієнтоване, функціональне та структурне програмування, що надає розробникам гнучкість у виборі підходу до розв'язання конкретних завдань. Вона має велику стандартну бібліотеку, яка включає в себе різноманітні інструменти та модулі для роботи з файлами, мережами, базами даних, графікою та багато іншого. Однією з найбільш привабливих особливостей Python є його широка підтримка спільноти. Існує велика та активна спільнота розробників, яка створює багато корисних бібліотек та інструментів. Це дозволяє ефективно вирішувати різноманітні завдання та швидко знаходити відповіді на питання.

Однак, Python не є найшвидшою мовою програмування, особливо у порівнянні з мовами низькорівневого рівня, такими як C++ або Rust. Вона може бути менш ефективною для обробки великих обсягів даних або виконання високопродуктивних обчислень порівняно з іншими мовами.

У цілому, Python - це потужний і універсальний інструмент, який дозволяє швидко розробляти різноманітні програми та проекти за рахунок своєї простоти, читабельності та багатофункціональності.

Python має багато переваг, які роблять його популярним серед розробників. Однією з головних переваг є його простота і читабельність. Синтаксис Python нагадує звичайну англійську мову, що робить його легким для вивчення навіть для початківців. Ця властивість також сприяє зменшенню кількості помилок і полегшує підтримку та оновлення коду.

Python є високорівневою мовою програмування, що дозволяє розробникам писати код без необхідності турбуватися про низькорівневі деталі, такі як управління пам'яттю. Це підвищує продуктивність і дозволяє зосередитися на вирішенні бізнес-завдань. Крім того, Python є інтерпретованою мовою, що означає можливість швидкого тестування та налагодження коду, оскільки він виконується рядок за рядком.

Широка стандартна бібліотека Python забезпечує готові до використання модулі і функції для різних завдань, таких як робота з файлами, мережеве програмування, обробка даних і багато іншого. Це значно скорочує час розробки, оскільки багато завдань можна вирішити за допомогою стандартних інструментів.

Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне і функціональне програмування, що робить його дуже гнучким. Також він має великий і активний співтовариство, яке постійно розробляє нові бібліотеки та інструменти, а також надає підтримку і обмін знаннями через форуми, блоги і документацію.

Однак Python має і деякі недоліки. Одним з них є відносно повільна швидкість виконання порівняно з компільованими мовами, такими як C або C++. Це може бути критичним для застосунків, де продуктивність має вирішальне значення. Хоча існують способи оптимізації коду, Python все ще не підходить для високопродуктивних обчислень.

Ще одним недоліком є використання інтерпретованого виконання, що може призвести до більш високого споживання ресурсів. Для великих проектів це може стати проблемою, особливо в умовах обмежених ресурсів.

Python не підходить для всіх видів мобільних розробок. Хоча існують бібліотеки для створення мобільних додатків на Python, такі як Kivy або BeeWare, вони не є такими зрілими і продуктивними, як нативні інструменти для Android або iOS. Це обмежує використання Python у сфері мобільних додатків.

Залежність від глобального інтерпретатора (GIL) є ще одним обмеженням, яке може ускладнити паралельну обробку і багатопоточність, що також впливає на продуктивність у деяких сценаріях.

Таким чином, Python є чудовим інструментом для багатьох видів розробки завдяки своїй простоті, гнучкості і широкій бібліотеці. Однак його продуктивність, споживання ресурсів і обмеження у мобільній розробці можуть бути недоліками, які варто враховувати при виборі мови для конкретних завдань



Рис. 3.2 KivyMD

**KivyMD** - це фреймворк для розробки користувацьких інтерфейсів (UI) з використанням мови програмування Python та бібліотеки Kivy. Він спеціалізується

на створенні красивих та функціональних додатків з гнучкими можливостями налаштування та адаптацією до різних платформ.

Основні особливості KivyMD полягають у тому, що він базується на концепції матеріального дизайну, розробленої Google, і надає багато готових компонентів і елементів інтерфейсу, які відповідають стандартам та дизайну, що визначені Google Material Design. Крім того, завдяки використанню фреймворку Kivy, KivyMD дозволяє розробникам створювати крос-платформенні додатки, які можуть бути запущені на різних операційних системах, включаючи Windows, macOS, Linux, Android та iOS.

KivyMD надає широкий спектр можливостей для налаштування вигляду та поведінки елементів інтерфейсу. Розробники можуть змінювати кольори, розміри, типи шрифтів та багато іншого, щоб створювати унікальний інтерфейс для своїх додатків.

Код в KivyMD може бути написаний просто та зрозуміло завдяки пайтонівському синтаксису. Крім того, фреймворк надає багато прикладів та документації, що допомагає розробникам швидко оволодіти його функціями та можливостями.

KivyMD дозволяє створювати анімаційні ефекти та переходи між екранами, що додає інтерактивності та привабливості до додатків. Фреймворк є відкритим проектом, що означає, що розробники можуть вносити свої внески та покращення до фреймворку, а також існує активна спільнота користувачів, яка надає підтримку та спільний обмін знаннями.



### 3.3 PyCharm

**PyCharm** є високопродуктивним інтегрованим середовищем розробки (IDE), спеціально призначеним для роботи з мовою програмування Python. Він володіє рядом унікальних функцій і інструментів, які дозволяють розробникам ефективно створювати, тестувати та налагоджувати програми на Python будь-якої складності.

Однією з ключових переваг PyCharm є його потужний механізм автодоповнення, який допомагає розробникам швидко писати код, забезпечуючи доступ до методів, атрибутів та інших об'єктів з поточного контексту. Це дозволяє ефективно використовувати бібліотеки та фреймворки, що збільшує продуктивність розробки.

Потужний вбудований відлагоджувач є ще однією важливою функцією PyCharm. Він дозволяє розробникам встановлювати точки зупинки, аналізувати змінні та вираження, а також відстежувати виклики функцій, щоб ефективно виявляти та виправляти помилки в коді.

PyCharm підтримує роботу з різними версіями Python і забезпечує можливість переключення між ними з легкістю. Це особливо корисно, коли розробники працюють над проектами, які використовують різні версії мови або вимагають використання різних середовищ виконання.

Інтеграція з системами контролю версій, зокрема з Git, є ще однією важливою функцією PyCharm. Вона дозволяє розробникам легко працювати з командними проектами, здійснювати коміти, переглядати відмінені зміни та вирішувати конфлікти злиття, все це зручно з використанням інтерфейсу користувача PyCharm.

PyCharm також підтримує широкий спектр фреймворків та інструментів, що використовуються в розробці програм на Python, таких як Django, Flask, Pyramid, та інші. Це дозволяє розробникам легко створювати веб-додатки, настільні програми та інші проекти на основі популярних фреймворків.

Узагальнюючи, PyCharm є надійним, потужним та широко використовуваним інструментом для розробки програм на мові Python. Він надає розробникам всі необхідні засоби для створення якісного та ефективного програмного забезпечення.

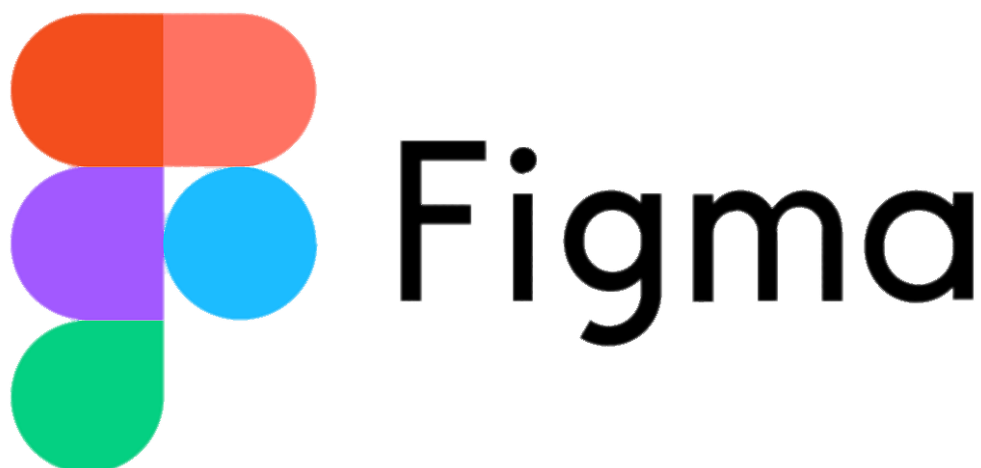


Рис. 3.4 Figma

**Figma** – це сучасний інструмент для дизайну, який набирає популярності завдяки своїй функціональності та зручності. Він є потужним векторним графічним редактором і прототипувальним інструментом, розробленим спеціально для спільної роботи в реальному часі. Основною перевагою Figma є її хмарна основа, що дозволяє командам дизайнерів працювати разом над проектами незалежно від їхнього місця розташування. Ця особливість спрощує процес колаборації, оскільки всі учасники можуть бачити внесені зміни миттєво і без необхідності постійного обміну файлами.

Крім того, Figma підтримує інтеграцію з багатьма іншими сервісами та інструментами, що робить її ще більш зручною для розробників і дизайнерів. З її допомогою можна створювати як прості макети, так і складні інтерфейси користувача, а також швидко тестувати ідеї за допомогою інтерактивних прототипів. Інструмент також надає можливості для створення дизайн-систем, що дозволяє підтримувати єдність і послідовність у проектах.

Figma має інтуїтивно зрозумілий інтерфейс і потужні інструменти для малювання, створення компонентів, роботи з текстом і стилями, що робить її ідеальним вибором як для початківців, так і для професійних дизайнерів. Вбудовані функції коментування і спільного редагування сприяють ефективному зворотному зв'язку і покращують процес ухвалення рішень.



Figma має багато переваг, що роблять її популярним вибором серед дизайнерів. Основною перевагою є хмарна основа, яка дозволяє працювати над проектами в режимі реального часу, забезпечуючи спільний доступ і редагування для всієї команди незалежно від їх місця знаходження. Це значно покращує співпрацю, дозволяючи бачити зміни миттєво і уникати проблем із синхронізацією файлів. Figma також інтегрується з багатьма іншими сервісами і інструментами, що спрощує робочі процеси і дозволяє зберегти послідовність у проектах завдяки можливості створення дизайн-систем.

Інтерфейс Figma є інтуїтивно зрозумілим, що полегшує процес навчання і роботи як для новачків, так і для досвідчених дизайнерів. Потужні інструменти для малювання, створення компонентів, роботи з текстом і стилями забезпечують високий рівень гнучкості та функціональності. Вбудовані можливості для коментування і спільного редагування сприяють ефективному зворотному зв'язку і швидкому ухваленню рішень.

Однак, Figma має і деякі недоліки. Залежність від інтернет-з'єднання може бути проблемою, оскільки робота в офлайн-режимі є обмеженою. Це може стати значною незручністю в умовах поганого або нестабільного інтернет-з'єднання. Крім того, хоча Figma пропонує безкоштовну версію, її функціональні можливості обмежені, і для доступу до більш просунутих функцій потрібна платна підписка.

Також деякі користувачі можуть відчувати певні труднощі з переходом на Figma з інших інструментів через різницю в інтерфейсі та робочих процесах. Хоча більшість цих проблем можна вирішити через навчання та адаптацію, вони можуть викликати певні незручності на початковому етапі.

Загалом, Figma – це інноваційне програмне забезпечення, яке поєднує в собі всі необхідні інструменти для створення сучасних веб-дизайнів і мобільних додатків, роблячи процес розробки більш швидким і ефективним..

#### 4. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Планування проекту з розробки програмного забезпечення є критично важливим етапом, що закладає основу для успішного виконання та завершення проекту. Воно починається з визначення мети та обсягу проекту, що включає розуміння вимог замовника, цілей бізнесу та кінцевих користувачів. На цьому етапі важливо залучити всіх зацікавлених сторін, щоб забезпечити повне і точне розуміння очікувань та обмежень.

Далі слідує етап аналізу та визначення вимог, де всі вимоги до програмного забезпечення документуються і уточнюються. Це включає функціональні та нефункціональні вимоги, такі як продуктивність, безпека та масштабованість. Після цього створюється детальний план проекту, який охоплює всі аспекти розробки, включаючи архітектуру системи, вибір технологій, розподіл ролей і відповідальностей у команді.

Наступним кроком є складання розкладу проекту, де визначаються етапи та завдання, встановлюються строки виконання і контрольні точки. Це допомагає відстежувати прогрес і своєчасно виявляти відхилення від плану. Важливо також враховувати ризики і розробляти стратегії їх мінімізації. Це може включати як технічні ризики, так і ризики, пов'язані з людськими ресурсами, бюджетом або зовнішніми чинниками.

Планування ресурсів є ще одним важливим аспектом, що включає визначення необхідних людських, фінансових і технічних ресурсів для успішної реалізації проекту. Тут враховуються кваліфікації та досвід членів команди, необхідні інструменти та програмне забезпечення, а також бюджетні обмеження. Ефективне управління ресурсами забезпечує їх оптимальне використання протягом усього проекту.

Комунікація та управління проектом є ключовими для забезпечення узгодженої роботи команди і своєчасного вирішення виникаючих проблем. Це включає регулярні зустрічі, звіти про прогрес, обмін інформацією та активне

залучення зацікавлених сторін у процес прийняття рішень. Інструменти управління проектами, такі як JIRA, Trello або Microsoft Project, можуть значно спростити цей процес, забезпечуючи прозорість і контроль над усіма аспектами проекту.

Завершальним етапом планування є оцінка і контроль якості, що передбачає встановлення критеріїв успішності та метрик для вимірювання якості програмного забезпечення. Це допомагає забезпечити відповідність кінцевого продукту вимогам і очікуванням замовника. Тестування, код-рев'ю, автоматизовані тести та інші методи контролю якості є важливими інструментами для досягнення високої якості продукту.

Планування проекту з розробки програмного забезпечення є комплексним процесом, що вимагає ретельної підготовки і системного підходу. Воно забезпечує чітку структуру і напрям для всіх учасників проекту, сприяючи ефективній і злагодженій роботі команди та успішному завершенню проекту.

#### Функціональні вимоги

- Можливість авторизації.
- Можливість проходження модулів з навчання.
- Можливість проходження тесту для визначення рівня знань.
- Можливість перегляду прогресу з навчання.

#### Нефункціональні вимоги:

- Наявність тесту на перевірку загальних знань в області фінансування.
- Зміна теми головного екрану застосунку.
- Українська локалізація контенту.

### **4.1 Діаграма прецедентів**

Діаграма прецедентів – дає можливість уявити типи ролей та їх взаємодію із системою. Проте діаграма не показує порядок виконання кроків. Вона зображує функціональні вимоги з точки зору користувача. Може описуватись текстом або у вигляді діаграми.

Діаграма складається з 4 об'єктів: актор, варіант використання, система та зв'язок. Актор це той хто взаємодіє з системою але не належить до неї, тобто знаходиться за межами системи. Позначається актор у вигляді стилізованого чоловічка. Прецеденти визначають очікувану поведінку та відповідають на питання що робить система. Прецеденти представляють набір можливих дій, функцій або завдань. Зображуються у вигляді еліпса з підписом дії у ньому. Прецедент вказує, що саме має трапитись проте не відповідає на питання як це має статися.

Система надає користувачу можливість виконувати такі сценарії використання:

- Реєстрація в системі
- Вхід в систему
- Перегляд наявних тем уроків
- Перегляд деталей уроків
- Проходження тестів
- Перегляд прогресу
- Перегляд результатів тестів

За допомогою даних пунктів та наявної інформації, ми можемо побудувати діаграму використання, що відображає взаємодію користувача з застосунком.

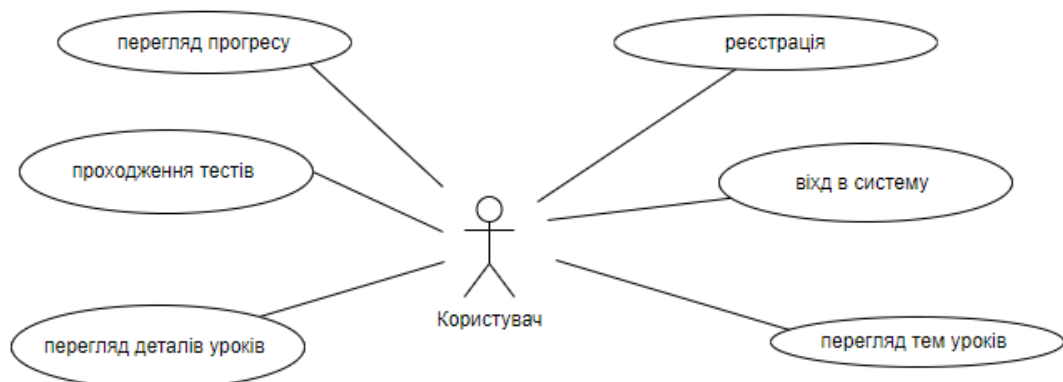


Рис.4.1 Use case діаграма

## 4.2 Діаграма класів

Діаграма класів - це графічне зображення структури програмного коду, яке використовується для візуалізації класів у системі програмування. Вона складається з класів та зв'язків між ними. Кожен клас представлений у вигляді прямокутника, в якому зазвичай зазначені його атрибути та методи. Зв'язки між класами показані за допомогою стрілок, що вказують напрямок взаємодії між класами. Діаграми класів використовуються для розробки, аналізу та документування програмних систем.

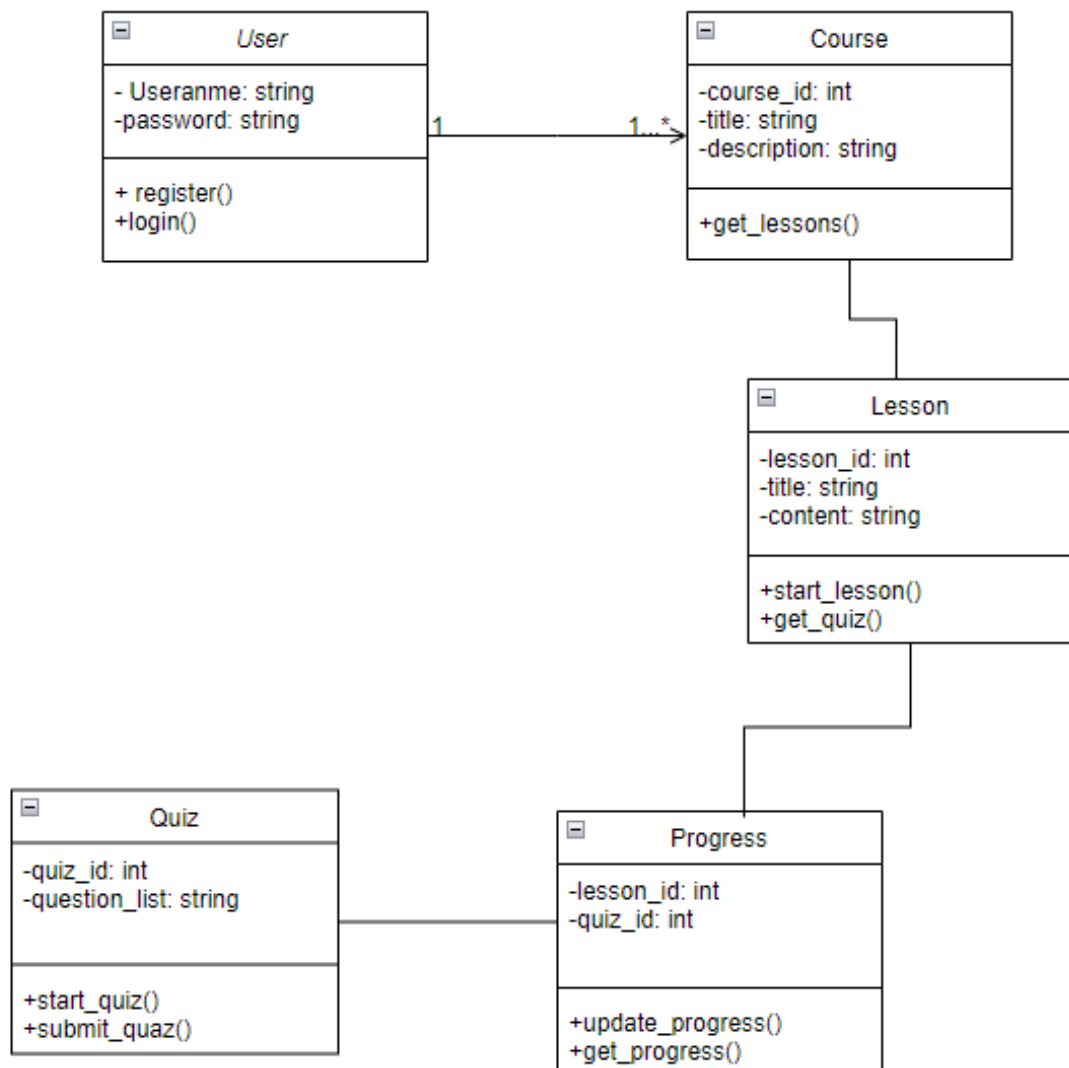


Рис.4.2 Діаграма класів

### 4.3 Проектування графічного інтерфейсу користувача

Принципи UX/UI дизайну відіграють ключову роль у створенні ефективних і зручних для користувачів інтерфейсів. Одним з основних принципів є орієнтація на користувача, що означає врахування потреб, поведінки та очікувань цільової аудиторії. Дизайн повинен бути інтуїтивно зрозумілим і легким у використанні, щоб користувачі могли швидко і без зусиль досягати своїх цілей. Важливо також забезпечити послідовність в дизайні, дотримуючись єдиних стилів, кольорів і шрифтів, щоб створити відчуття цілісності і передбачуваності.

Зручність використання також є важливим принципом. Інтерфейси повинні бути простими і зрозумілими, мінімізуючи кількість кроків, необхідних для виконання завдань. Важливо забезпечити зворотний зв'язок, щоб користувачі знали, що їхні дії були успішними або отримали інформацію про помилки.

Інтерактивні елементи відіграють ключову роль у UX/UI дизайні, забезпечуючи взаємодію користувача з системою. Кнопки є одним з основних інтерактивних елементів, які повинні бути достатньо великими, помітними і легко натискатися. Меню забезпечують структуру і навігацію, дозволяючи користувачам швидко знайти потрібну інформацію або функцію.

Інформаційна архітектура визначає, як інформація організована і представлена користувачам. Логічна структура, зрозуміла навігація і добре продуманий контент дозволяють користувачам легко знаходити потрібну інформацію.

Загалом, принципи UX/UI дизайну спрямовані на створення інтуїтивно зрозумілих, доступних і зручних для використання інтерфейсів, де інтерактивні елементи грають ключову роль у забезпеченні ефективної взаємодії користувача з системою.

## 4.4 Розробка програмного забезпечення

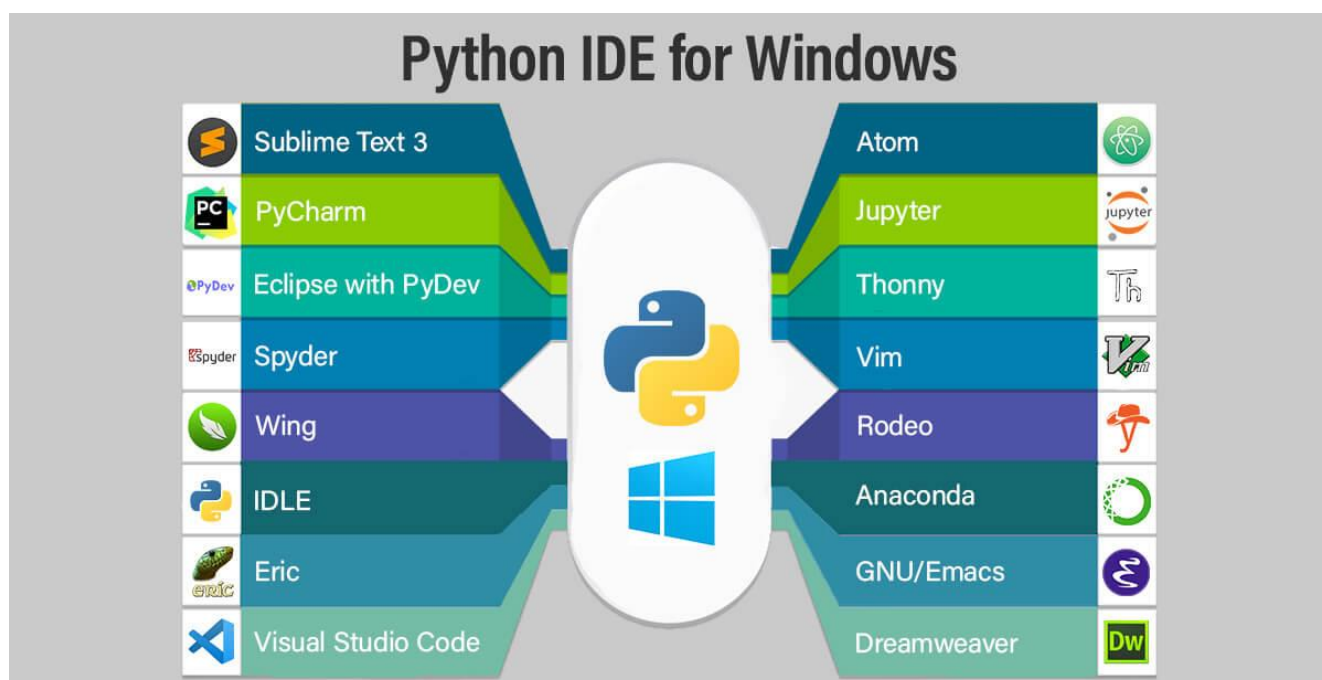


Рис. 4.3 Приклади інтегрованого середовища розробки

IDE, або інтегроване середовище розробки, є програмним інструментом, що надає програмістам все необхідне для розробки, тестування та налагодження програмного забезпечення в одному місці. IDE об'єднує текстовий редактор, компілятор або інтерпретатор, засоби автоматизації збірки, а також відладчик. Це значно спрощує процес написання коду, надаючи інструменти для автозавершення, підсвічування синтаксису, аналізу коду та рефакторингу.

Більшість IDE підтримують інтеграцію з системами контролю версій, такими як Git, що дозволяє відстежувати зміни в коді і працювати в команді. Вони також можуть містити вбудовані інструменти для тестування, які допомагають автоматизувати процеси юніт-тестування і тестування продуктивності. Деякі IDE надають графічні інтерфейси для налаштування проекту та управління залежностями, а також плагіни для підтримки різних мов програмування та фреймворків.

IDE спрощує керування великими проектами, забезпечуючи зручний доступ до всіх файлів і ресурсів проекту, інтеграцію з базами даних та веб-серверами. Це

особливо важливо в умовах сучасної розробки, де швидкість і ефективність роботи є критичними факторами. Популярні IDE, такі як Visual Studio, IntelliJ IDEA, Eclipse і PyCharm, пропонують широкий спектр можливостей і налаштувань, що дозволяє адаптувати середовище під конкретні потреби розробника чи команди.

PyCharm є потужним IDE для розробки на Python, створеним компанією JetBrains. Воно пропонує ряд переваг, що роблять його популярним вибором серед розробників. Однією з головних переваг є розширений набір інструментів для автозавершення коду, підсвічування синтаксису та інтелектуального аналізу коду, що допомагає швидше писати та виправляти код. PyCharm також інтегрується з популярними системами контролю версій, такими як Git, що спрощує командну роботу та управління змінами.

PyCharm надає вбудовану підтримку для тестування, зокрема, юніт-тестування та тестування продуктивності, а також підтримує популярні фреймворки для веб-розробки, такі як Django та Flask. Він включає зручні інструменти для роботи з базами даних, дозволяючи легко переглядати та змінювати структуру баз даних безпосередньо з IDE. Плагіни та розширення дозволяють користувачам налаштовувати PyCharm під свої специфічні потреби, додаючи нові функції та підтримку різних технологій.

Однак, є й деякі недоліки, які слід враховувати. Одним із них є високе споживання ресурсів, що може призводити до повільної роботи на старіших або менш потужних комп'ютерах. PyCharm також може здаватися складним для новачків через велику кількість функцій і налаштувань, що потребує часу для освоєння. Деякі користувачі можуть знайти інтерфейс надмірно перевантаженим, особливо якщо вони звикли до більш легких та мінімалістичних редакторів.

Також варто зазначити, що повна функціональність доступна лише у платній версії PyCharm Professional, тоді як безкоштовна версія Community Edition має обмежені можливості. Це може бути вирішальним фактором для розробників з обмеженим бюджетом або тих, хто не потребує всіх професійних функцій. Незважаючи на ці недоліки, PyCharm залишається одним із найпотужніших і



найзручніших інструментів для розробки на Python, особливо для великих проєктів і професійної розробки.

Після ознайомлення з основами та принципами UX/UI дизайну, починаємо створювати першу сторінку мобільного застосунку.

Для цього почнемо з імпорту необхідних бібліотек. Оскільки ми використовуємо бібліотеку Kivy, будемо імпортувати такі бібліотеки:

- `logging`: використовується для журналювання подій, зручний для відлагодження.
  - `kivy.lang.Builder`: дозволяє завантажувати Kivy файли `.kv`.
  - `kivy.uix.screenmanager`: для керування кількома екранами.
  - `kivy.metrics.dp`: для роботи з пікселями на екрані.
  - `kivymd.app.MDApp`: основний клас додатка KivyMD.
  - `kivymd.uix.dialog.MDDialog`,
  - `kivymd.uix.button.MDRaisedButton`, -
  - `kivymd.uix.list.TwoLineListItem`: -компоненти інтерфейсу KivyMD.
- `os`: для роботи з файловою системою.

Далі на першій сторінці ми розташуємо вікно авторизації. Вікно буде мати кнопку входу та реєстрації. Для входу необхідними полями зробимо «Логін» та «Пароль». При натиску на «Реєстрація» відкривається вікно з можливістю зареєструвати нового користувача.

Користувач починає взаємодію з додатком з екрану входу (`LoginScreen`), де він може ввести свої облікові дані (ім'я користувача та пароль). Якщо користувач ще не зареєстрований, він може перейти на екран реєстрації (`RegisterScreen`) і створити новий обліковий запис. Дані зберігаються у файлі `users.txt`, що дозволяє застосунку перевіряти правильність даних користувача при вході в застосунок.

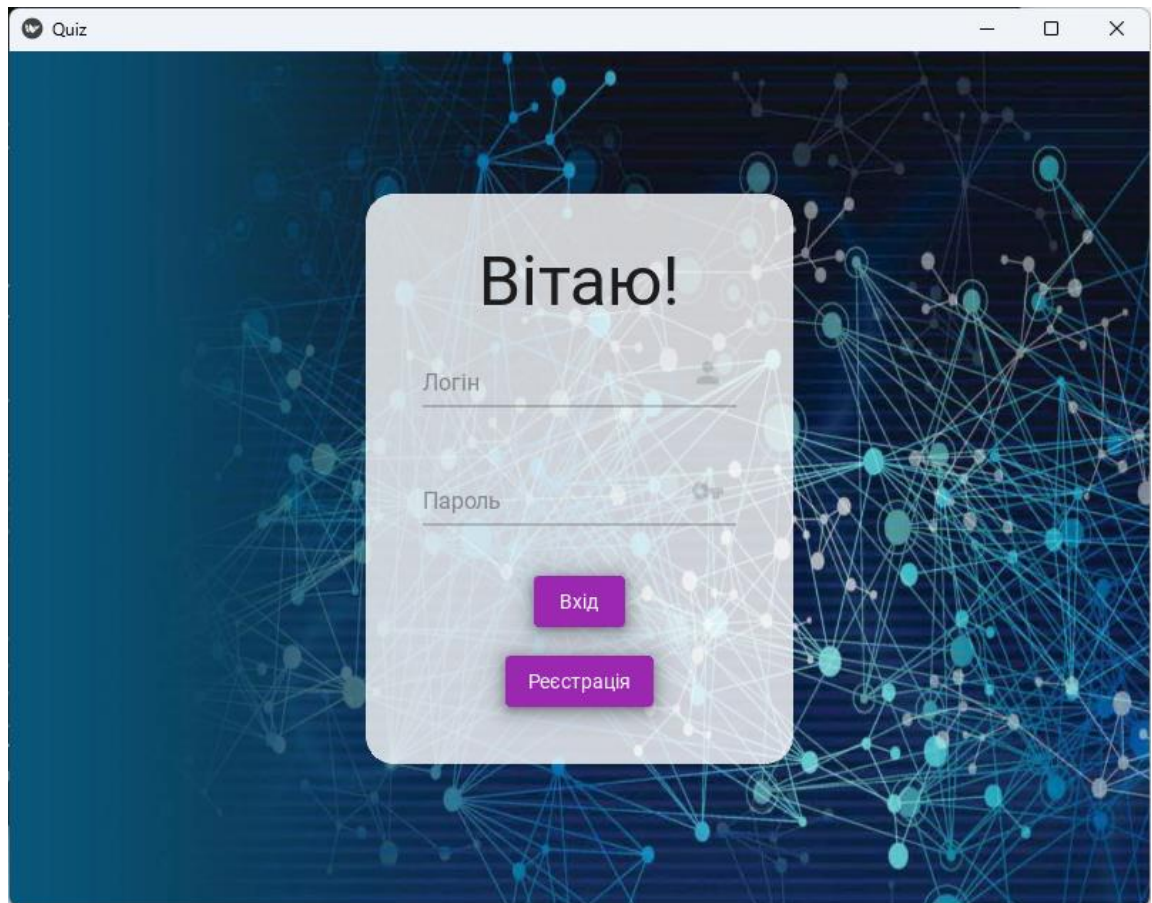


Рис. 4.4 Вікно авторизації

Далі створюємо «Головне меню» застосунку, розташовуємо основні кнопки які знадобляться користувачу для зручного користування ним, а саме: модулі, тестування, зміна кольору теми та вкладка «Профіль» в якій будуть відображені результати та прогрес навчання. Додаток включатиме екран уроків (LessonsScreen), де користувач може переглядати питання та відповіді у форматі уроків. Це допомагає користувачеві ознайомитися з матеріалом перед проходженням тесту. Уроки відображаються у вигляді списку, де кожен елемент містить питання та правильну відповідь. Користувач може вибрати урок для перегляду детальної інформації на екрані деталізації уроку (LessonDetailScreen).

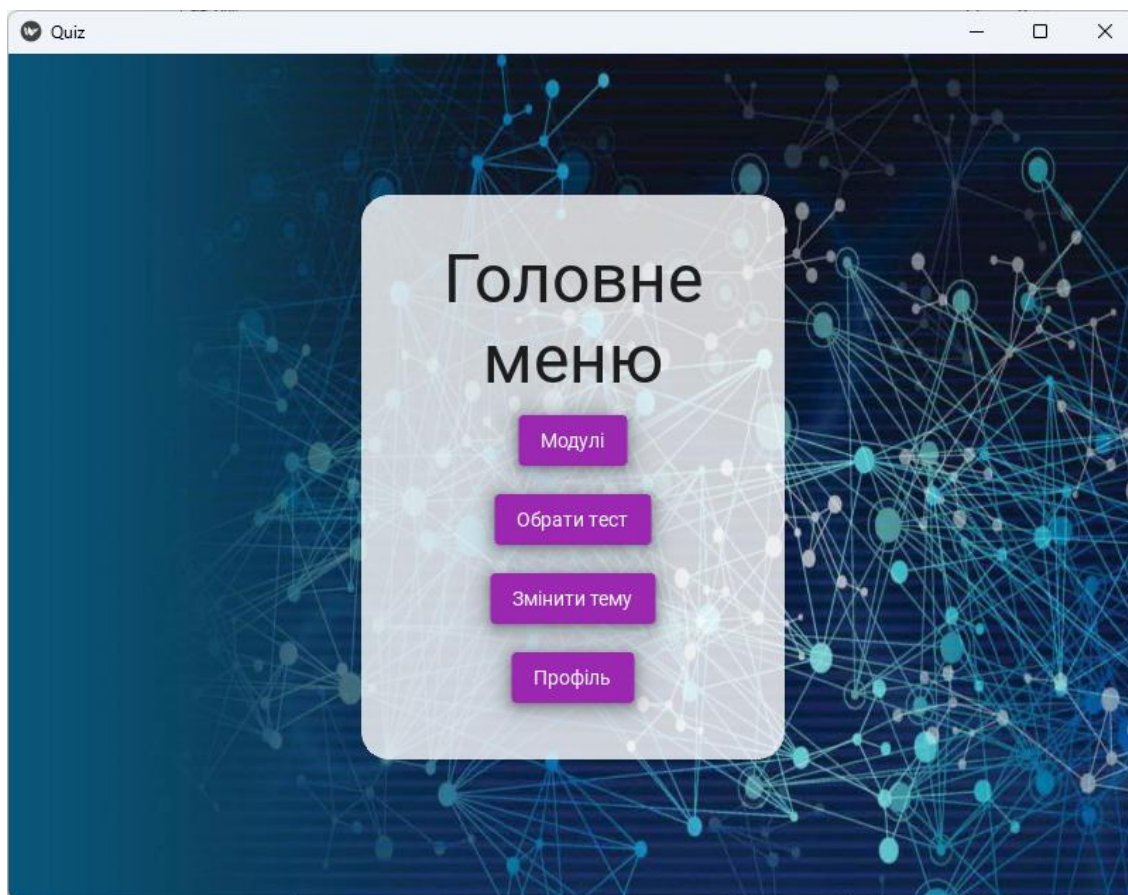


Рис. 4.5 Головне меню

На екрані вибору тесту (TestSelectionScreen) користувач може вибрати одну з кількох тем тестів, таких як фінансування, кредитування та ін. Кожна тема містить певну кількість питань з варіантами відповідей. Після вибору теми тестування користувач потрапляє на екран з питаннями (QuestionScreen), де він може почати тест. Програма підтримує як режим тестування, так і режим навчання, де користувач отримує детальну інформацію про кожну відповідь.

Розділ «Модулі» почнемо зі створення окремих уроків, в яких буде розташована інформація про основні теми та підказки до майбутніх тестів. При натисканні на конкретний модуль відкривається нове вікно з назвою модуля, відповіддю до нього та детальною інформацією.

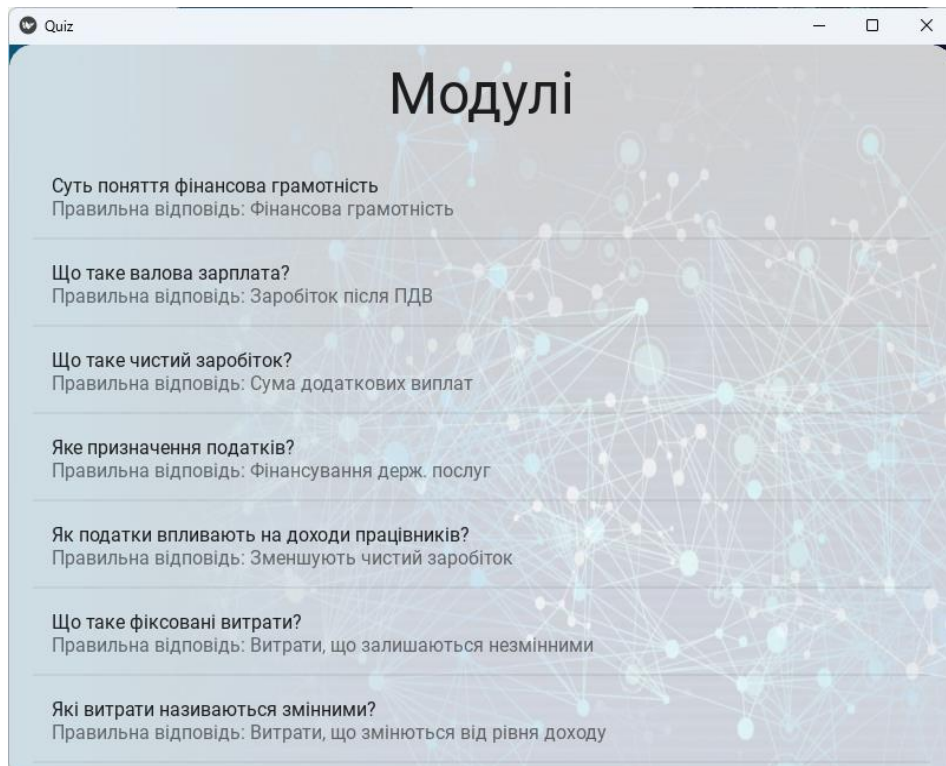


Рис. 4.6 Модулі

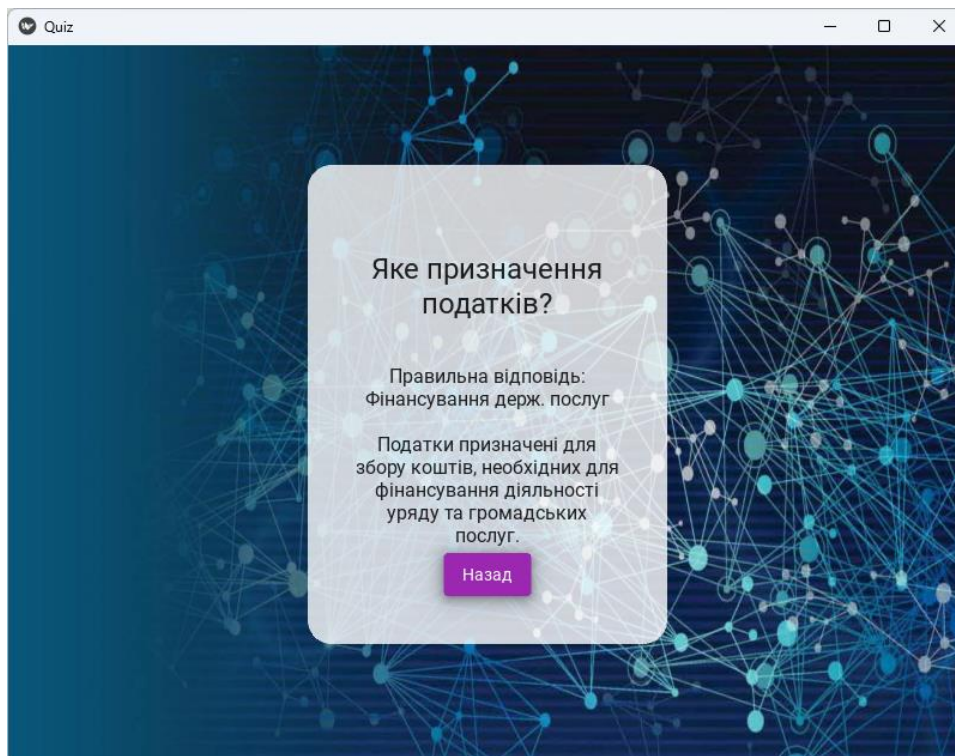


Рис. 4.7 Детальна інформація модулів

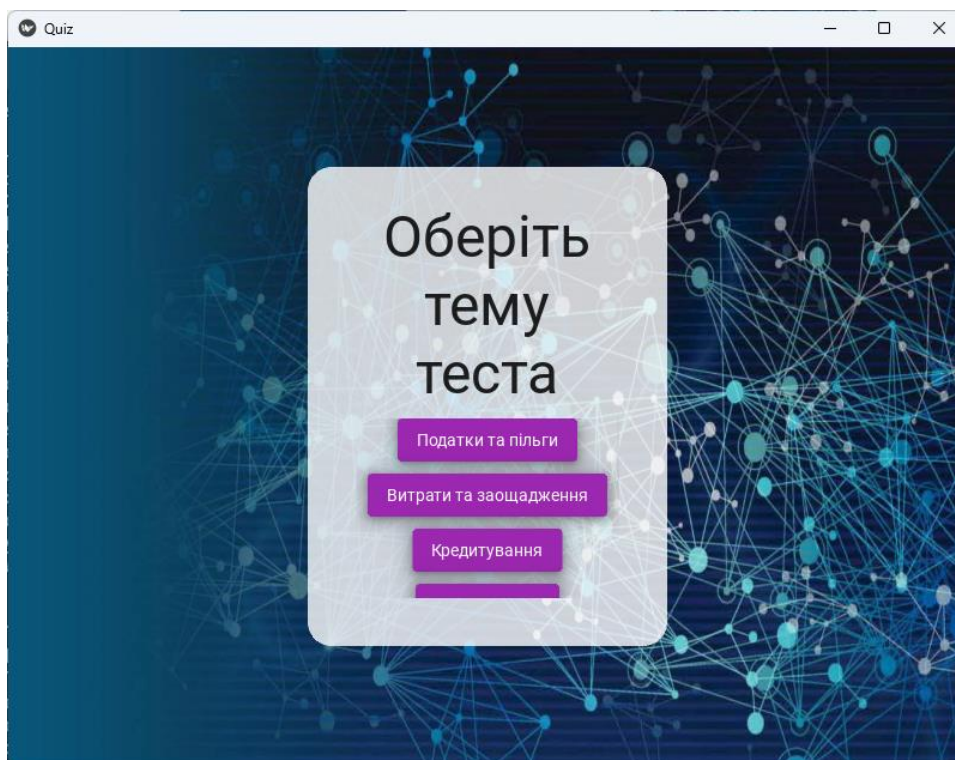


Рис. 4.8 Меню тестів

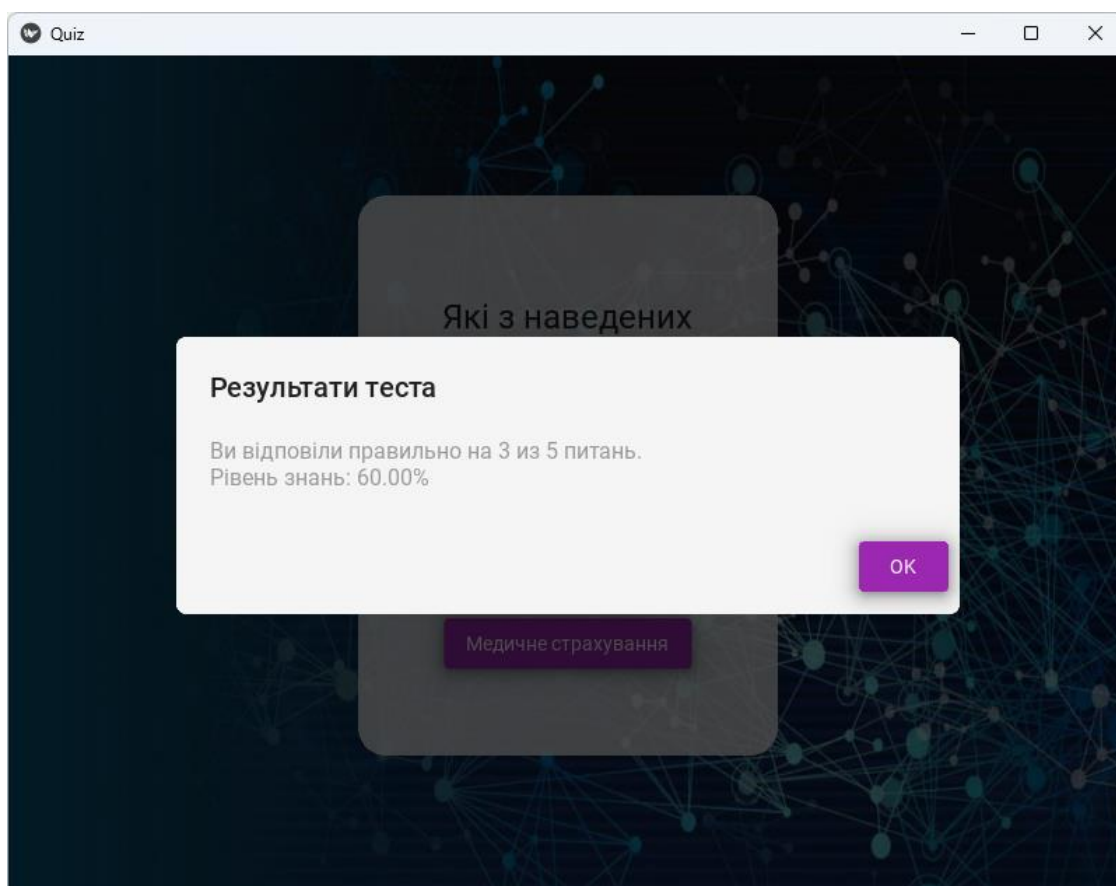


Рис. 4.9 Результат тестів

Також створюємо додатковий тест, для загальної перевірки знань з усіх модулів та надаємо йому аналогічні функції.

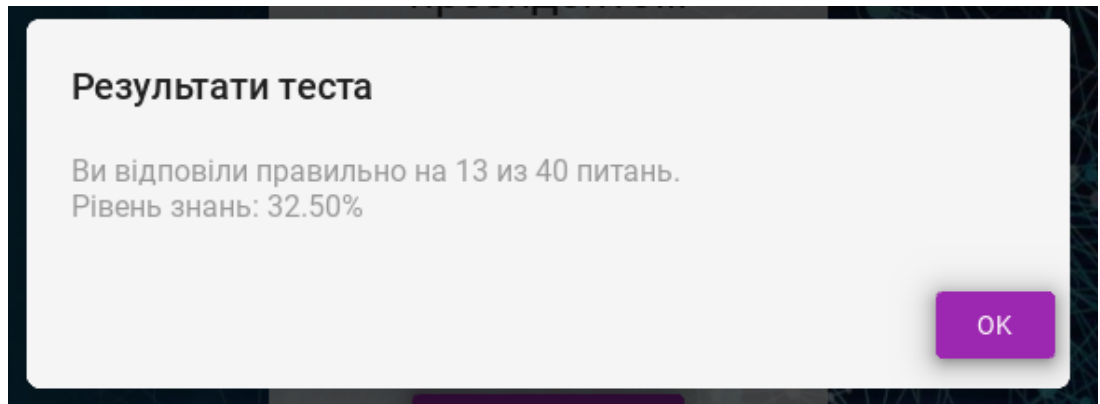


Рис. 4.10 Загальний тест

Повертаємось до головного екрану та оформлюємо кнопку «Змінити тему», надаємо їй функціонал зміни кольору обрамлення з світлої на темну та навпаки.

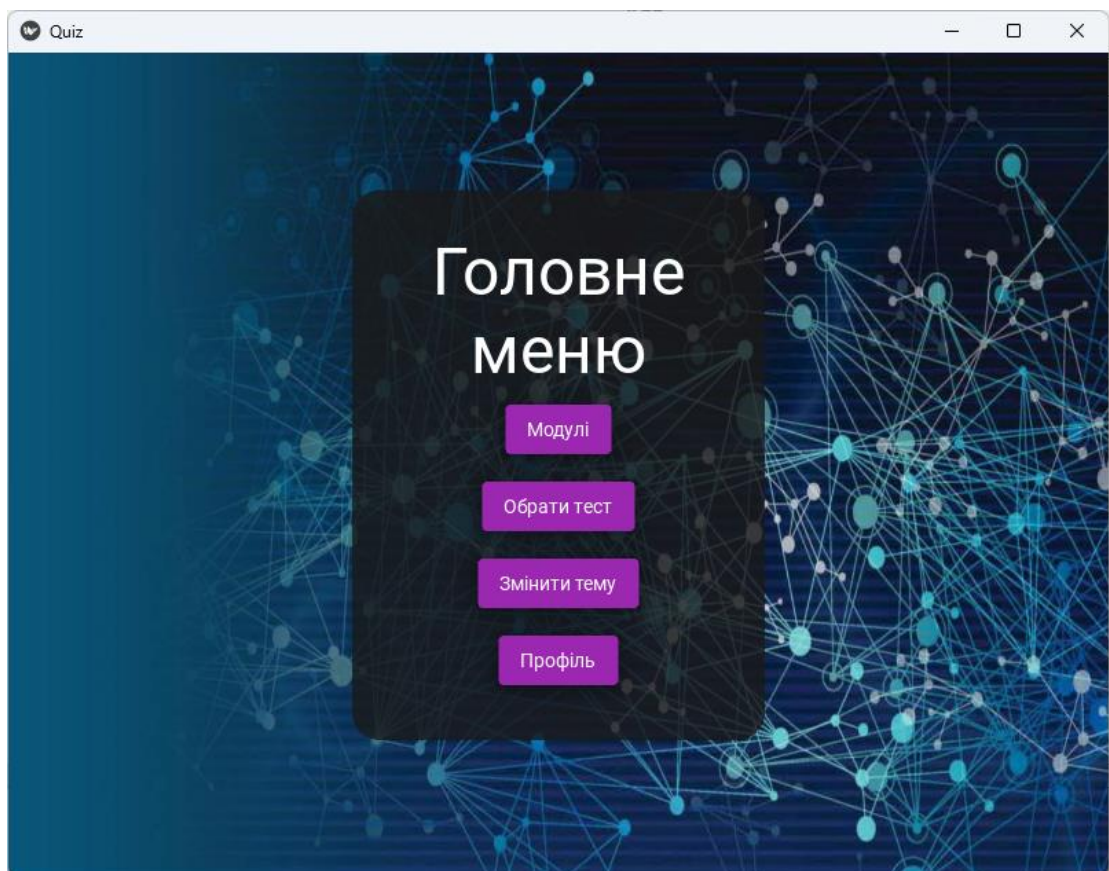


Рис. 4.11 Зміна теми

Створюємо кнопку «Профіль» та заповнюємо її інформацією у вигляді заповнюючої шкали відносно результатів пройдених тестів. Користувачі можуть переглядати свій прогрес у різних тестах.

- Функція `show_profile` оновлює дані про прогрес користувача для кожної теми тесту та відображає їх у вигляді прогрес-барів і відсотків.

- Прогрес розраховується на основі кількості правильних відповідей, що зберігаються у змінній `scores`.

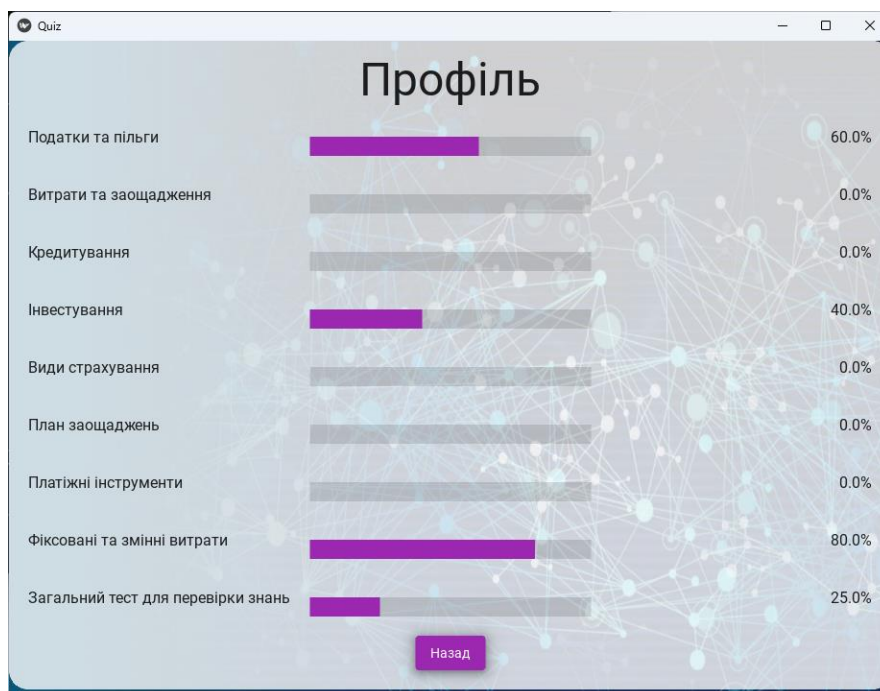


Рис. 4.12 Сторінка прогресу

Також перевіряємо можливість зміни теми в інших сторінках застосунку.

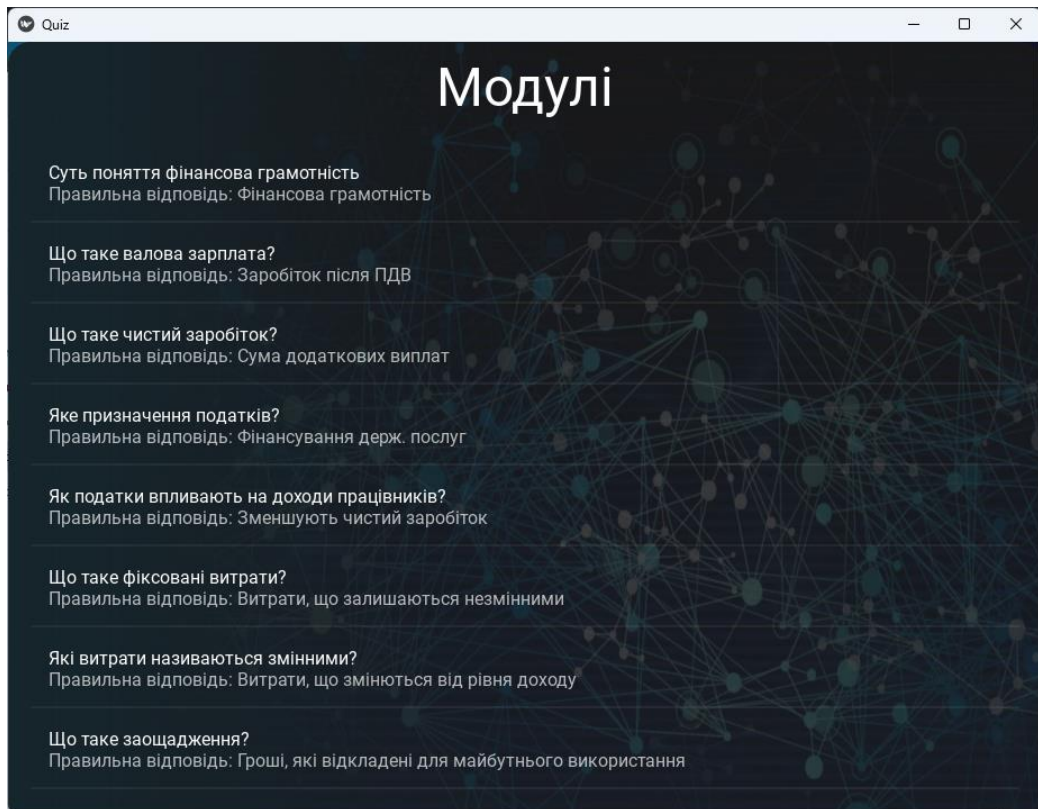


Рис. 4.13 Сторінка модулів з темною темою

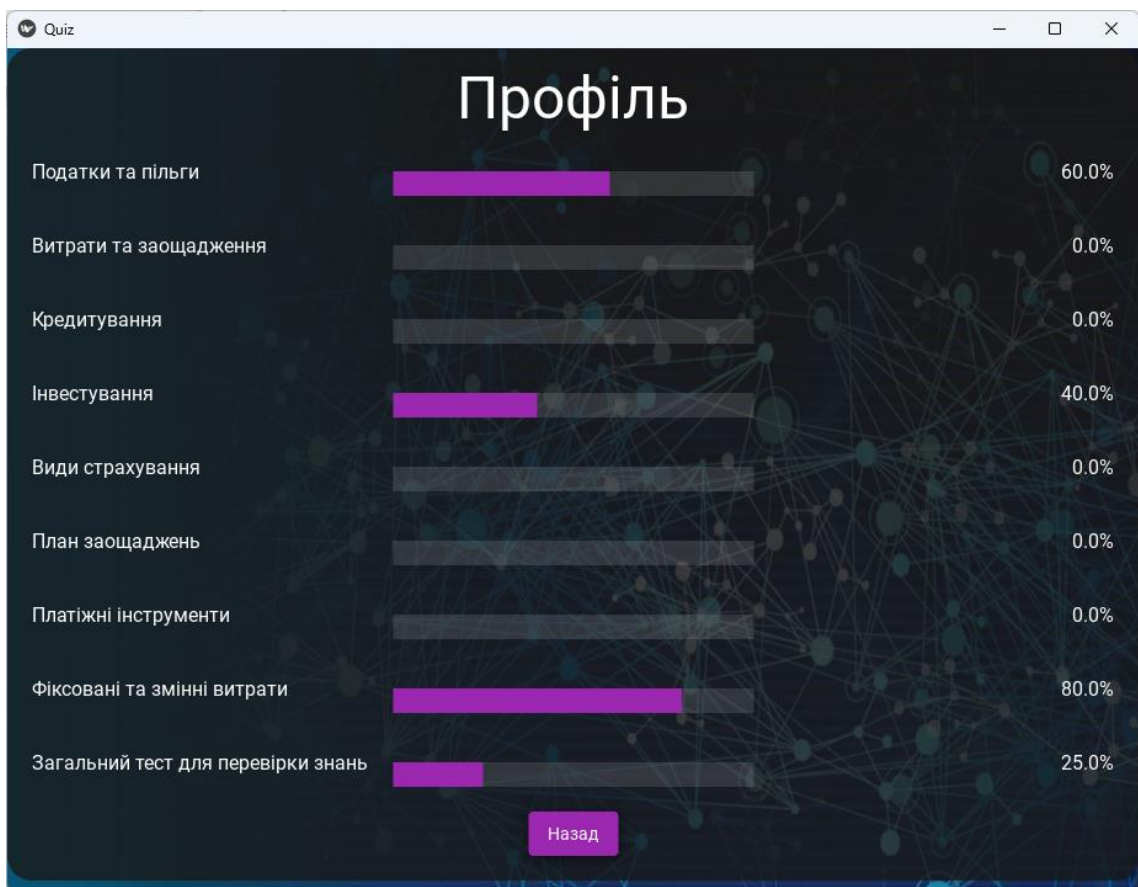


Рис. 4.14 Сторінка прогресу з темною темою



У розділі розробки програмного забезпечення для інтерактивного навчання фінансової грамотності було звернуто увагу на ряд ключових аспектів, спрямованих на поліпшення якості освіти у сфері фінансів. Перш за все, було використано сучасні технології та інструменти розробки, такі як фреймворк Kivy та його розширення KivyMD. Це дозволило створити додатки зі зручним та естетичним графічним інтерфейсом, що відіграє важливу роль у залученні та зацікавленні користувачів.

Одним з головних переваг такого підходу є можливість персоналізації навчального процесу. Додатки, розроблені з використанням цих технологій, дозволяють користувачам вивчати матеріал у зручному для них темпі, а також відповідно до їхніх особистих потреб і інтересів. Наприклад, системи відстеження прогресу, можливість налаштування складності завдань та індивідуальний підбір матеріалів забезпечують ефективне та цікаве навчання.

Ще однією важливою рисою розроблених додатків є їхня доступність. Завдяки використанню мобільних технологій, користувачі можуть отримувати доступ до навчального контенту в будь-який час та в будь-якому місці, що робить процес навчання більш гнучким та зручним.

У цілому, розробка програмного забезпечення для інтерактивного навчання фінансової грамотності відкриває нові можливості для підвищення рівня фінансової освіченості населення. Цей підхід сприяє залученню користувачів, надаючи їм інструменти для зручного та ефективного навчання, та сприяє формуванню стійких навичок управління фінансами, що є важливим елементом стабільного фінансового майбутнього.

## ВИСНОВКИ

У першому розділі цієї дипломної роботи був проведений аналіз предметної області. На основі научної літератури, пов'язаної з вивченням основ фінансової грамотності та інтерактивного навчання. Проведено аналіз потреб користувачів у підвищенні фінансової грамотності. Результати показують, що більшість респондентів оцінюють свої знання як середні. Було визначено цільову аудиторію застосунку, що включає підлітків, молодь та дорослих, які бажають покращити свої знання у сфері фінансів. У другому розділі виконаний аналіз аналогів кожного з додатків. Сформована таблиця-підсумок проведеного аналізу.

У третьому розділі дипломної роботи, орієнтуючись на поставлену задачу, було здійснено вибір технологій та платформи розробки. Було прийнято рішення розроблювати мобільний додаток для платформи iOS, з використанням мови програмування Python та його фреймворку KivyMD.

Далі були сформовані необхідні функціональні та нефункціональні вимоги за якими буде проводитись розробка програмного продукту. Завдяки програмному забезпеченню Figma було розроблено увесь інтерфейс мобільного застосунку з мінімалістичним дизайном для спрощення розуміння додатку користувачем.

У четвертому розділі, користуючись обраними інструментами та за допомогою аналізу аналогів, було розроблено програмне забезпечення.

Дубова Є.Б., Садовенко В.С. Розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python. Всеукраїнська науково-практична конференція молодих вчених та здобувачів вищої освіти. «Сучасна молодь в світі інформаційних технологій», 17 травня 2024 р., Херсон, Херсонський державний аграрно-економічний університет. Збірник тез. К.: ХДАЕУ 2024. С. 97.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Пітон, Ф. (2023). Документація Python. URL: [https://docs.python.org/3/](https://docs.python.org/3/) (дата звернення: 01.06.2024).
2. Алчин, М. (2010). Професійний Python. Apress.
3. Грінберг, М. (2018). Розробка веб-застосунків на Flask: Розробка веб-додатків з Python. O'Reilly Media.
4. Северенс, Ч. (2016). Python для всіх: Дослідження даних у Python 3. CreateSpace Independent Publishing Platform.
5. Свейгарт, А. (2015). Автоматизація рутинних завдань за допомогою Python: Практичне програмування для початківців. No Starch Press.
6. МакКінні, В. (2017). Python для аналізу даних: Обробка даних з Pandas, NumPy та IPython. O'Reilly Media.
7. Гровер, П. (2021). Практична розробка застосунків з PyQt5. Packt Publishing.
8. Берд, С., Клайн, Е., & Лопер, Е. (2009). Обробка природної мови за допомогою Python: Аналіз тексту з Natural Language Toolkit. O'Reilly Media.
9. Шо, З. А. (2017). Вивчення Python 3 шляхом важкої роботи: Дуже просте введення в чудовий світ комп'ютерів і коду. Addison-Wesley Professional.
10. Комісія з фінансової грамотності та освіти. (2020). Національна стратегія з фінансової грамотності 2020. Міністерство фінансів США.
11. Khan Academy. (2021). Курси фінансової грамотності. URL: [https://www.khanacademy.org/](https://www.khanacademy.org/) (дата звернення: 01.06.2024).
12. Coursera. (2023). Вступ до програмування на Python. URL: [https://www.coursera.org/learn/python](https://www.coursera.org/learn/python) (дата звернення: 01.06.2024).
13. Real Python. (2023). Реальні уроки Python. URL: [https://realpython.com/](https://realpython.com/) (дата звернення: 01.06.2024).

14. Apple Inc. (2021). Керівництво по інтерфейсу користувача. URL: [https://developer.apple.com/design/human-interface-guidelines/](https://developer.apple.com/design/human-interface-guidelines/) (дата звернення: 01.06.2024).

15. Google Inc. (2021). Керівництво по матеріальному дизайну. URL: [https://material.io/design/](https://material.io/design/) (дата звернення: 01.06.2024).

## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### Розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python

Виконала студентка 4 курсу

Групи ПД-43

Дубова Євгенія Борисівна

Керівник роботи

К.ф.м.н., доц., професор кафедри ІПЗ Садовенко Володимир Сергійович

Київ – 2024

## МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – підтримка процесу вивчення основ фінансової грамотності серед користувачів за допомогою мобільного застосунку.
- **Об'єкт дослідження** - процес вивчення основ фінансової грамотності.
- **Предмет дослідження** – мобільний застосунок для вивчення основ фінансової грамотності.

## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз існуючих мобільних застосунків з навчанням основ фінансової грамотності.
2. Визначити функціональні та нефункціональні вимоги до мобільного застосунку для навчання основ фінансової грамотності.
3. Проектування візуальної частини та інтерфейсу користова мобільного застосунку.
4. Програмна реалізація мобільного застосунку для навчання основ фінансової грамотності.

3

## АНАЛІЗ АНАЛОГІВ

Критерій / Додаток	Mint	YNAB	Acorns	Robinhood	Financial education
<u>Перевірка рівня знань</u>	-	+	-	-	+
<u>Освітній контент</u>	<u>Навчальні статті, блоги</u>	<u>Онлайн курси, вебінари</u>	<u>Відеоуроки, статті</u>	<u>Фінансові новини, блоги</u>	<u>Навчальні матеріали, тести</u>
<u>Різноманіття тем</u>	<u>Бюджетування, інвестиції, управління боргами</u>	<u>Управління особистими фінансами</u>	<u>Розуміння фінансів, інвестування</u>	<u>Інвестування, фінансові ринки</u>	<u>Кредитування, фінансування, інвестиції, страхування</u>
<u>Рівень складності</u>	<u>Підходить для початківців</u>	<u>Підходить для просунутих користувачів</u>	<u>Підходить для початківців</u>	<u>Підходить для середнього рівня</u>	<u>Підходить для початківців</u>
<u>Відстеження прогресу</u>	-	+	-	+	+

4

## ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### Функціональні вимоги:

1. Можливість авторизації.
2. Можливість проходження модулів з навчання.
3. Можливість проходження тесту для визначення рівня знань.
4. Можливість перегляду прогресу з навчання.

### Нефункціональні вимоги:

1. Наявність тесту на перевірку загальних знань в області фінансування.
2. Зміна теми головного екрану застосунку.
3. Українська локалізація контенту

5

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Python



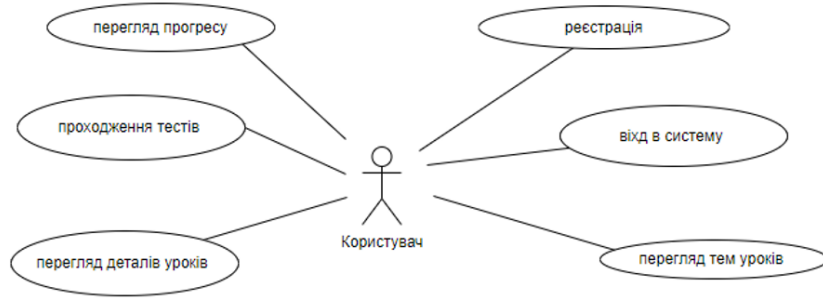
PyCharm



KivyMD

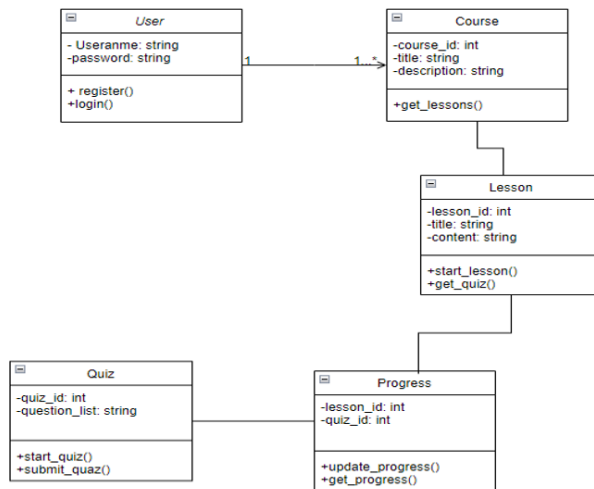
6

### Діаграма варіантів використання



7

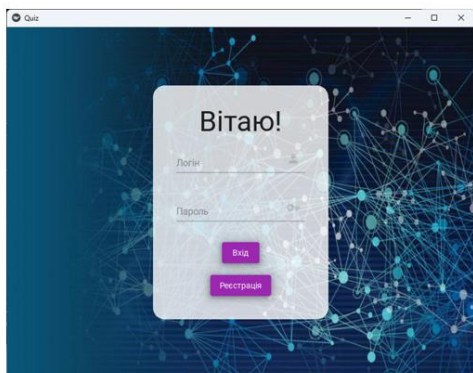
### Діаграма класів



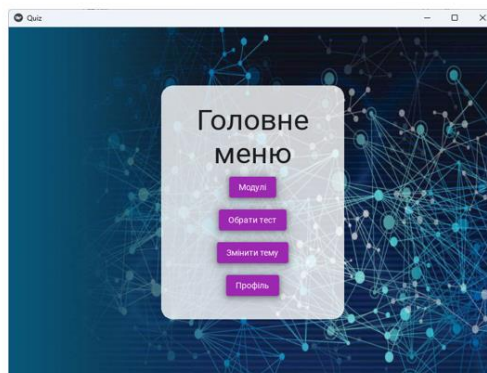
8



## ЕКРАННІ ФОРМИ



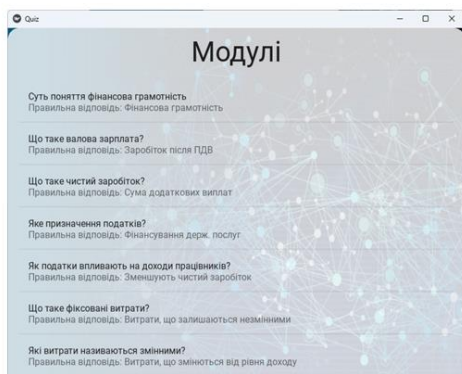
Вікно авторизації



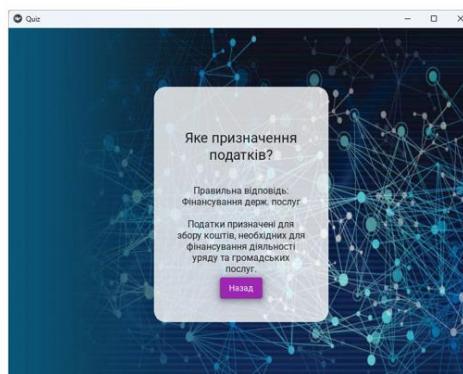
Головне меню

9

## ЕКРАННІ ФОРМИ



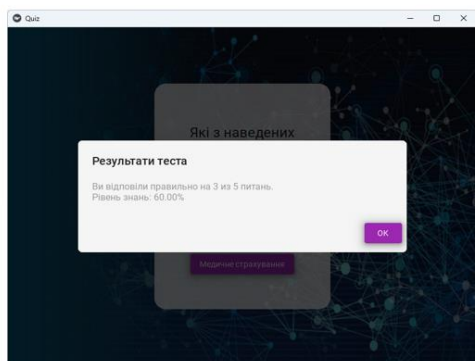
Сторінка уроків



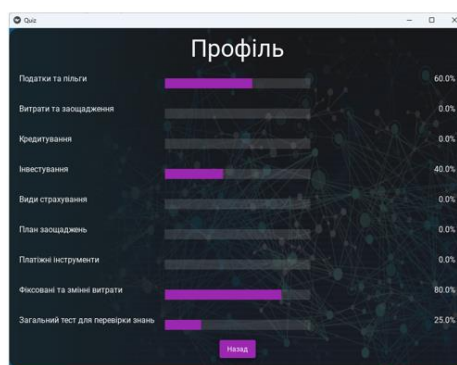
Детальна інформація модулів

10

## ЕКРАННІ ФОРМИ



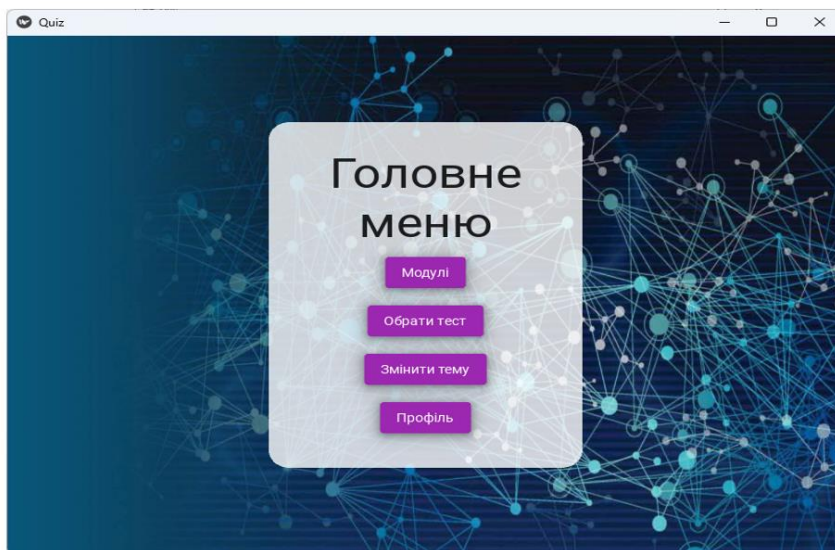
Результати тестування



Сторінка результатів тестування

11

## Відео роботи програми



12

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Дубова Є.Б. Садовенко В.С. . Розробка мобільного застосунку для інтерактивного навчання основам фінансової грамотності мовою Python. V Всеукраїнська науково-практична конференція молодих вчених та здобувачів вищої освіти «СУЧАСНА МОЛОДЬ В СВІТІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ». Збірник тез 17.05.2024 Херсон-Кропивницький.-С. 97.

13

## ВИСНОВКИ

1. Проведено аналіз існуючих мобільних застосунків для інтерактивного навчання.
2. Проведено аналіз потреб користувачів у підвищенні фінансової грамотності.
3. Проведено аналіз та огляд інструментів, які були використані під час виконання роботи, зокрема Python, PyCharm, KivyMD.
4. Розроблено функціональні та нефункціональні вимоги, головною з яких стала можливість перевірки знань за допомогою тестування.
5. Проаналізовані основні принципи UX/UI дизайну.
6. Розроблено застосунок на основі обраних технологій і з урахуванням вимог.

14

## ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ

```

import logging
from kivy.lang import Builder
from kivy.uix.screenmanager
import ScreenManager, Screen
from kivy.metrics import dp
from kivymd.app import MDApp
from kivymd.uix.dialog import
MDDialog
from kivymd.uix.button import
MDRaisedButton
from kivymd.uix.list import
TwoLineListItem
import os

# Налаштування логирования
logging.basicConfig(level=logging
g.DEBUG, format='%(asctime)s -
%(levelname)s - %(message)s')

class LoginScreen(Screen):
    pass

class RegisterScreen(Screen):
    pass

class MenuScreen(Screen):
    pass
class
TestSelectionScreen(Screen):
    pass
class QuestionScreen(Screen):
    pass

class ProgressScreen(Screen):
    pass

class ProfileScreen(Screen):
    pass

class LessonsScreen(Screen):
    pass

class LessonDetailScreen(Screen):
    pass

class QuizApp(MDApp):
    tests = {
        'finance_1': [
            {"question": "Суть
поняття фінансова грамотність",
"answers": ["Фінансова грамотність",
"Фінансова освіта", "Фінансова
поведінка"], "correct": "Фінансова
грамотність", "detail": "Здатність
ефективно планувати і
використовувати бюджет, приймати
рішення у сфері особистих фінансів,
орієнтуватися в послугах ."},
            {"question": "Що таке
валова зарплата?", "answers":
["Заробіток після ПДВ", "Загальний
заробіток", "Сума доп. виплат"],
"correct": "Заробіток після ПДВ",
"detail": "Валова зарплата — це
загальна сума доходу працівника до
будь-яких відрахувань"},
            {"question": "Що таке
чистий заробіток?", "answers":
["Заробіток до ПДВ", "Заробіток з
основної ставки", "Сума додаткових
виплат"], "correct": "Сума додаткових
виплат", "detail": "Чистий заробіток
— це сума грошей, яку працівник
отримує на руки після відрахування
всіх податків та інших утримань."},
            {"question": "Яке
призначення податків?", "answers":
["Підвищення доходу працівників",
"Зниження витрат уряду",
"Фінансування держ. послуг"],
"correct": "Фінансування держ.
послуг", "detail": "Податки призначені

```

для збору коштів, необхідних для фінансування діяльності уряду та громадських послуг."},

{"question": "Як податки впливають на доходи працівників?", "answers": ["Збільшують чистий заробіток", "Зменшують чистий заробіток", "Не впливають"], "correct": "Зменшують чистий заробіток", "detail": "Податки зменшують чистий заробіток, адже вони є обов'язковими відрахуваннями з валової зарплати.."},

],  
'finance\_2': [{"question": "Що таке фіксовані витрати?", "answers": ["Витрати що змінюються від рівня доходу", "Витрати, що залишаються незмінними", "Витрати на розваги"], "correct": "Витрати, що залишаються незмінними", "detail": "Фіксовані витрати — це регулярні витрати, сума яких залишається незмінною щомісяця."}],

{"question": "Які витрати називаються змінними?", "answers": ["Витрати на розваги", "Витрати, що завжди зростають", "Витрати, що змінюються від рівня доходу"], "correct": "Витрати, що змінюються від рівня доходу", "detail": "Витрати, які змінюються залежно від рівня доходу, споживання або інших обставин."},

{"question": "Що таке заощадження?", "answers": ["Гроші, які витрачаються на розваги", "Гроші, які відкладені для майбутнього використання", "Гроші, які отримують від інвестицій"], "correct": "Гроші, які відкладені для майбутнього використання", "detail": "Це частина доходу, яка не

витрачається, а відкладається для майбутнього використання.."}, {"question": "Яка мета резервного фонду?", "answers": ["Фінансування відпусток", "Інвестування в акції", "Покриття непередбачених витрат"], "correct": "Покриття непередбачених витрат", "detail": "Це грошові заощадження, призначені для покриття непередбачених витрат таких як медичні витрати або ремонт житла."},

{"question": "Які з наведених витрат належать до змінних?", "answers": ["Орендна плата", "Витрати на продукти", "Медичне страхування"], "correct": "Витрати на продукти", "detail": "Витрати на продукти та розваги є змінними, оскільки можуть значно варіюватися від споживчих потреб"},

],  
'finance\_3': [{"question": "Що таке кредит?", "answers": ["Гроші, які ви даруєте іншій особі", "Позика з поверненням", "Гроші, які не потрібно повертати"], "correct": "Позика грошей або товарів з обов'язковим поверненням через певний час і, як правило, з відсотками", "detail": "Він дозволяє отримати доступ до ресурсів негайно, а не чекати на їх накопичення."}],

{"question": "Які переваги кредиту?", "answers": ["Додатковий дохід", "Фінансова гнучкість", "Боргове навантаження"], "correct": "Фінансова гнучкість", "detail": "Фінансова гнучкість та можливість здійснювати великі покупки або інвестиції без необхідності відкладати кошти."},

{"question": "Що таке кредитна картка?", "answers": ["Для накопичення балів", "здійснення покупок в креди", "для оплати комунальних послуг"], "correct": "Пластикова картка, що дозволяє здійснювати покупки в кредит", "detail": "Дозволяє власнику здійснювати покупки в кредит в межах встановленого ліміту."},
 {"question": "Які правила користування кредитними картками?", "answers": ["Лише для великих покупок", "Витратити доступні кошти щомісяця", "Погашати заборгованість вчасно"], "correct": "Погашати заборгованість вчасно, не перевищувати кредитний ліміт, стежити за відсотковими ставками", "detail": "Недопущення перевищення кредитного ліміту, що може вплинути на кредитний рейтинг."},
 {"question": "Що таке кредитний рейтинг?", "answers": ["Оцінка фінансової стабільності країни", "Рейтинг популярності банку", "Оцінка кредитоспроможності"], "correct": "Оцінка кредитоспроможності особи", "detail": "Впливає на можливість отримання кредиту та умови кредитування."},
 ],
 'finance \_4': [
 {"question": "Що таке заощадження?", "answers": ["Інвестування грошей", "Витрачання грошей", "Відкладання грошей"], "correct": "Відкладання грошей", "detail": "Це процес відкладання грошей для майбутнього використання, зберігаючи їх у безпечних і надійних формах."},

{"question": "Що таке інвестування?", "answers": ["Витрачання грошей", "Вкладення грошей", "Зберігання під матрацом"], "correct": "Вкладення грошей", "detail": "Інвестування — це процес вкладення грошей у фінансові інструменти, такі як акції, облігації."},
 {"question": "Яка основна відмінність між інвестуванням та заощадженнями?", "answers": ["Інвестування менш ризиковане", "Заощадження завжди приносять дохід", "Інвестування має вищий ризик"], "correct": "Інвестування має вищий ризик", "detail": "Основна відмінність між інвестуванням та заощадженнями полягає у рівні ризику та потенційному доході."},
 {"question": "Які з наведених тверджень є правильною характеристикою заощаджень?", "answers": ["Низький рівень ризику, стабільний дохід", "Залежність від фондового ринку", "Необхідність моніторингу та аналізу"], "correct": "Низький рівень ризику, стабільний дохід", "detail": "Заощадження є низьким рівнем ризику та стабільним доходом, зазвичай зберігаються в банках"},
 {"question": "Які основні фактори впливають на вибір між інвестуванням та заощадженнями?", "answers": ["Лише фінансові цілі", "Лише горизонт інвестування", "Поточний рівень інфляції"], "correct": "Поточний рівень інфляції", "detail": "Вибір між інвестуванням та заощадженнями залежить від кількох факторів, таких як поточний рівень інфляції, рівень доходів, фінансові цілі та горизонт інвестування."},

```

    ],
    'finance _5': [
        {"question": "Що таке
медичне страхування?", "answers":
["Страхування, яке покриває медичне
обслуговування", "Страхування від
збитків, пов'язаних з транспортними
засобами", "Страхування від
фінансових втрат"], "correct":
"Страхування, яке покриває медичне
обслуговування", "detail": "Медичне
страхування покриває витрати на
медичне обслуговування, включаючи
консультації лікарів, лікування."},
        {"question": "Яке з видів
страхування захищає від фінансових
втрат, пов'язаних з аваріями?",
"answers": ["Страхування майна",
"Автомобільне страхування",
"Життєве страхування"], "correct":
"Автомобільне страхування", "detail":
"Автомобільне страхування
забезпечує захист від фінансових
втрат, пов'язаних з дорожньо-
транспортними пригодами."},
        {"question": "Що таке
страхування відповідальності?",
"answers": ["Страхування власного
майна від стихійних лих",
"Страхування, яке захищає від
фінансових втрат через шкоду,
завдану іншим особам", "Страхування
від втрати доходу через безробіття"],
"correct": "Страхування, яке захищає
від фінансових втрат через шкоду,
завдану іншим особам", "detail":
"Страхування відповідальності
покриває витрати, пов'язані з
відшкодуванням шкоди, завданої
іншим особам або їх майну."},
        {"question": "Яке
страхування покриває втрати через
знищення або пошкодження
будинку?", "answers": ["Життєве

```

```

страхування", "Страхування
подорожей", "Страхування майна"],
"correct": "Страхування майна",
"detail": "трахування майна
забезпечує фінансовий захист від
втрат через знищення або
пошкодження будинку."}],
    ],
}

# Смешанный тест
tests['general_knowledge'] = []
for key in tests:
    if key !=
'general_knowledge':

tests['general_knowledge'].extend(tests[
key])

current_question_index = 0
scores = {
    'finance _1': 0,
    'finance _2': 0,
    'finance _3': 0,
    'finance _4': 0,
    'finance _5': 0,
    'general_knowledge': 0,
}
is_learning_mode = False
learning_progress = []
selected_test = []
test_topic = ""

def build(self):
    logging.debug("Building the
application...")

self.theme_cls.primary_palette =
"Purple"

self.theme_cls.theme_style =
"Light"

return
Builder.load_file('main.kv')

```

```

def validate_login(self):
    username =
self.root.get_screen('login').ids.username
.text
    password =
self.root.get_screen('login').ids.password
.text
    if
self.check_credentials(username,
password):
        self.root.current = 'menu'
    else:
        dialog = MDDialog(
            title="Помилка входу",
            text="Користувач з
таким ім'ям вже існує.",
buttons=[MDRaisedButton(text="ОК",
on_release=lambda x: dialog.dismiss())]
        )
        dialog.open()

def register_user(self):
    username =
self.root.get_screen('register').ids.reg_us
ername.text
    password =
self.root.get_screen('register').ids.reg_pa
ssword.text
    if username and password:
        if not
os.path.exists('users.txt'):
            with open('users.txt',
'w') as f:
                pass
            with open('users.txt', 'r') as
f:
                users = f.readlines()
                for user in users:
                    if user.split(':')[0] ==
username:
                        dialog = MDDialog(

```

```

            title="Помилка
реєстрації",
text="Пользователь с таким логином
уже существует.",
buttons=[MDRaisedButton(text="ОК",
on_release=lambda x: dialog.dismiss())]
        )
        dialog.open()
        return
        with open('users.txt', 'a') as
f:
            f.write(f"{username}:{password}\n")
            dialog = MDDialog(
                title="Реєстрація
завершена",
                text="Ви успішно
zareєструвались. Вітаю.",
buttons=[MDRaisedButton(text="ОК",
on_release=lambda x:
self.go_to_login(dialog))]
            )
            dialog.open()
        else:
            dialog = MDDialog(
                title="Помилка
реєстрації",
                text="Будь ласка,
заповніть всі поля",
buttons=[MDRaisedButton(text="ОК",
on_release=lambda x: dialog.dismiss())]
            )
            dialog.open()

def check_credentials(self,
username, password):
    if not
os.path.exists('users.txt'):
        return False
    with open('users.txt', 'r') as f:

```



```

        users = f.readlines()
        for user in users:
            stored_username,
            stored_password = user.strip().split(':')
            if stored_username ==
            username and stored_password ==
            password:
                return True
        return False

    def go_to_register(self):
        self.root.current = 'register'

    def go_to_login(self,
    dialog=None):
        if dialog:
            dialog.dismiss()
            self.root.current = 'login'

    def go_to_test_selection(self):
        self.root.current =
        'test_selection'

    def go_to_lessons(self):
        self.root.current = 'lessons'
        self.display_lessons()

    def display_lessons(self):
        lessons_screen =
        self.root.get_screen('lessons')
        lessons_list =
        lessons_screen.ids.lessons_list
        lessons_list.clear_widgets()

        for test_topic, questions in
        self.tests.items():
            for question_data in
            questions:
                lessons_list.add_widget(
                TwoLineListItem(
                text=question_data["question"],

```

```

                secondary_text=f"Правильна
                відповідь: {question_data['correct']}",
                on_release=lambda
                x, q=question_data:
                self.show_lesson_detail(q)
            )
        )

    def show_lesson_detail(self,
    question_data):
        lesson_detail_screen =
        self.root.get_screen('lesson_detail')

        lesson_detail_screen.ids.lesson_detail_ti
        tle.text = question_data["question"]

        lesson_detail_screen.ids.lesson_detail_c
        ontent.text = f"Правильна відповідь:
        {question_data['correct']}\n\n{question_
        data.get('detail', 'Немає даних')}"
        self.root.current =
        'lesson_detail'

    def start_quiz(self, test_topic,
    learning_mode=False):
        self.current_question_index
        = 0

        self.is_learning_mode =
        learning_mode
        self.learning_progress = []
        self.test_topic = test_topic
        self.selected_test =
        self.tests[test_topic]
        self.show_question()
        self.root.current = 'question'

    def show_question(self):
        if
        self.current_question_index <
        len(self.selected_test):
            question_data =
            self.selected_test[self.current_question_i
            ndex]

```

```

        question_screen =
self.root.get_screen('question')

question_screen.ids.question_label.text
= question_data["question"]
        answers_layout =
question_screen.ids.answers_layout

answers_layout.clear_widgets()

        for answer in
question_data["answers"]:
            btn = MDRaisedButton(
                text=answer,

pos_hint={"center_x": .5},
                size_hint_y=None,
                height=dp(50),
                on_release=lambda
btn: self.check_answer(btn.text)
            )

answers_layout.add_widget(btn)
        else:
            self.show_result()

        def check_answer(self,
selected_answer):
            correct_answer =
self.selected_test[self.current_question_i
ndex]["correct"]
            if selected_answer ==
correct_answer:
                self.scores[self.test_topic]
+= 1

                if self.is_learning_mode:

self.show_learning_dialog(correct_answer)
            else:

self.current_question_index += 1
                self.show_question()

```

```

        def show_learning_dialog(self,
correct_answer):

self.learning_progress.append({"questio
n":
self.selected_test[self.current_question_i
ndex]["question"], "correct_answer":
correct_answer})
            dialog = MDDialog(
                title="Режим навчання",
                text="Правильна
відповідь: {}".format(correct_answer),
                buttons=[
                    MDRaisedButton(
                        text="Далі",
                        on_release=lambda x:
self.close_learning_dialog(dialog)
                    )
                ]
            )
            dialog.open()

        def close_learning_dialog(self,
dialog):
            dialog.dismiss()
            self.current_question_index
+= 1
            self.show_question()

        def show_result(self):
            percentage =
(self.scores[self.test_topic] /
len(self.selected_test)) * 100
            result_text = f"Ви відповіли
правильно на
{self.scores[self.test_topic]} из
{len(self.selected_test)}
питань.\nРівень знань:
{percentage:.2f}%"
            dialog = MDDialog(
                title="Результати теста",
                text=result_text,
                buttons=[

```

```

        MDRaisedButton(
            text="OK",
            on_release=lambda x:
self.go_to_menu(dialog)
        )
    ]
)
dialog.open()

def go_to_menu(self,
dialog=None):
    if dialog:
        dialog.dismiss()
        self.root.current = 'menu'

    def toggle_theme(self):
        self.theme_cls.theme_style =
"Dark" if self.theme_cls.theme_style ==
"Light" else "Light"

    def show_progress(self):
        progress_screen =
self.root.get_screen('progress')
        progress_list =
progress_screen.ids.progress_list
        progress_list.clear_widgets()

        for item in
self.learning_progress:
            progress_list.add_widget(
                TwoLineListItem(
text=item["question"],

secondary_text=f"Правильна
відповідь: {item['correct_answer']}"
            )
        )
        self.root.current = 'progress'

    def show_profile(self):
        profile_screen =
self.root.get_screen('profile')
profile_screen.ids.finance_1_progress.va
lue = self.calculate_progress('finance
_1')
        profile_screen.ids._finance
_1_percentage.text =
"{}%".format(self.calculate_progress('fi
nance_1'))

profile_screen.ids.finance_2_progress.va
lue = self.calculate_progress('finance_2')

profile_screen.ids.finance_2_percentage.
text =
"{}%".format(self.calculate_progress('fi
nance_2'))

profile_screen.ids.finance_3_progress.va
lue = self.calculate_progress('finance_3')

profile_screen.ids.finance_3_percentage.
text =
"{}%".format(self.calculate_progress('fi
nance_3'))

profile_screen.ids.finance_4_progress.va
lue = self.calculate_progress('finance_4')

profile_screen.ids.finance_4_percentage.
text =
"{}%".format(self.calculate_progress('fi
nance_4'))

profile_screen.ids.finance_5_progress.va
lue = self.calculate_progress('finance_5')

profile_screen.ids.finance_5_percentage.
text =
"{}%".format(self.calculate_progress('fi
nance_5'))

profile_screen.ids.general_knowledge_p
rogress.value =

```

```

self.calculate_progress('general_knowledge')

profile_screen.ids.general_knowledge_percentage.text =
    "{}%".format(self.calculate_progress('general_knowledge'))
    self.root.current = 'profile'

    def calculate_progress(self, test_topic):
        if test_topic not in self.scores:
            return 0

        app")

```

```

        total_questions = len(self.tests[test_topic])
        return (self.scores[test_topic] / total_questions) * 100 if total_questions > 0 else 0

    if __name__ == "__main__":
        try:
            logging.debug("Starting the application...")
            QuizApp().run()
        except Exception as e:
            logging.exception("Exception occurred while running the

```