

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

Кафедра цифрових технологій та методики навчання інформатики

«До захисту допущено»

Завідувач кафедри ЦТ та МНІ

_____Наталія ПАВЛОВА

«_____» «_____» 2025 р.

протокол № _____

ПІКУН ВЛАДИСЛАВ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**МЕТОДИЧНІ ПІДХОДИ ДО ОРГАНІЗАЦІЇ САМООСВІТИ ЗДОБУВАЧІВ
ОСВІТИ У ЗАКЛАДАХ ФАХОВОЇ ПЕРЕДВИЩОЇ ОСВІТИ З
ВИКОРИСТАННЯМ ГЕЙМІФІКОВАНИХ МОБІЛЬНИХ ТЕХНОЛОГІЙ**

Виконав:

здобувач другого (магістерського)
рівня вищої освіти

спеціальності 015.39 Професійна
освіта (Цифрові технології)

Владислав ПІКУН

Науковий керівник:

кандидат педагогічних наук, доцент

Наталія ПОЛЮХОВИЧ

Рівне – 2025

АНОТАЦІЯ

Пікун В. І. Методичні підходи до організації самоосвіти здобувачів освіти у закладах фахової передвищої освіти з використанням гейміфікованих мобільних технологій. – Кваліфікаційна робота на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 015 Професійна освіта (Цифрові технології). – Рівненський державний гуманітарний університет. – Рівне, 2025. – 76 с.

У кваліфікаційній роботі досліджено методичні підходи до організації самоосвіти здобувачів освіти у закладах фахової передвищої освіти з використанням гейміфікованих мобільних технологій, що дозволить підвищити навчальну та самоосвітню мотивацію студентів.

Проаналізовано теоретичні основи самоосвіти та гейміфікації, узагальнено вітчизняний і зарубіжний досвід використання гейміфікованих технологій у викладанні дисциплін комп'ютерного циклу. Визначено вимоги до ефективного гейміфікованого середовища для організації самоосвітньої діяльності здобувачів фахової передвищої освіти.

Практичним результатом дослідження є розробка гейміфікованого мобільного застосунку InformatykaGO, що забезпечує інтерактивне навчання, контроль результатів, мотивацію через систему балів, рівнів, бейджів і рейтингів. Запропоновано методичні рекомендації щодо впровадження розробленого застосунку в освітній процес закладів фахової передвищої освіти під час вивчення інформатики.

Результати дослідження можуть бути використані викладачами інформатики, методистами та здобувачами освіти для підвищення ефективності самоосвітньої діяльності.

Ключові слова: самоосвіта, гейміфікація, мобільні технології, фахова передвища освіта, дисципліни комп'ютерного циклу, інформатика, мобільний застосунок.

ABSTRACT

Pikun V. I. Methodological approaches to organizing self-education for students in institutions of professional pre-higher education using gamified mobile technologies. – Qualification thesis for obtaining the second (Master's) level of higher education in the specialty 015 Professional Education (Digital Technologies). – Rivne State University of the Humanities. – Rivne, 2025. – 76 p.

The qualification thesis investigates methodological approaches to organizing self-education for students of institutions of professional pre-higher education using gamified mobile technologies, which contributes to increasing students' learning and self-educational motivation.

The theoretical foundations of self-education and gamification are analyzed, and domestic and foreign experience in the use of gamified technologies in teaching computer-related disciplines is summarized. The requirements for an effective gamified environment for organizing self-educational activities of students in institutions of professional pre-higher education are defined.

The practical outcome of the research is the development of the gamified mobile application *InformatykaGO*, which provides interactive learning, learning outcomes monitoring, and motivation through a system of points, levels, badges, and rankings. Methodological recommendations for the implementation of the developed application in the educational process of institutions of professional pre-higher education during the study of informatics are proposed.

The research results can be used by informatics teachers, methodologists, and students to enhance the effectiveness of self-educational activities.

Keywords: self-education, gamification, mobile technologies, professional pre-higher education, computer-related disciplines, informatics, mobile application.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ I ТЕОРЕТИЧНІ ОСНОВИ САМООСВІТИ ТА ГЕЙМІФІКАЦІЇ В ОСВІТІ.....	8
1.1 Самоосвіта як складова навчального процесу в ЗФПО	8
1.2 Гейміфіковані технології у навчанні: сутність, види та переваги	12
1.3 Досвід використання гейміфікації у викладанні комп'ютерних дисциплін	17
РОЗДІЛ II АНАЛІЗ ТА ПРОЄКТУВАННЯ ГЕЙМІФІКОВАНОГО СЕРЕДОВИЩА ДЛЯ САМООСВІТИ З ІНФОРМАТИКИ	19
2.1 Аналіз існуючих гейміфікованих платформ для вивчення інформатики	19
2.2 Вимоги до ефективного гейміфікованого середовища для самоосвіти	22
2.3 Розробка концепції мобільного застосунку InformatykaGO	25
2.4 Функціональна структура та дизайн застосунку	28
РОЗДІЛ III РОЗРОБКА ТА ВПРОВАДЖЕННЯ ГЕЙМІФІКОВАНОГО ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНФОРМАТИКИ.....	36
3.1 Архітектура застосунку та інтеграція бази даних	36
3.2 Основні компоненти застосунку	39
3.3 Тестування та публікація застосунку в Google Play.....	52
3.4 Впровадження в освітній процес	54
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ.....	65

ВСТУП

Сучасний світ, зокрема освітній простір України, зіткнувся з безпрецедентними викликами, спричиненими пандемією COVID-19 та військовою агресією. Ці чинники не лише порушили традиційний навчальний процес, але й критично підкреслили важливість навичок самоосвіти для сучасної людини. З іншого боку стрімкий розвиток інформаційних технологій, безупинне оновлення та збільшення можливостей дистанційних технологій вимагають вироблення нових ефективних методик та їх впровадження в освітній процес освітніх закладів, зокрема й закладів фахової передвищої освіти (ЗФПО).

Зауважимо, що при таких умовах використання традиційних методів самоосвіти досить часто через брак мотивації виявляються не достатньо ефективними. Тому виникає потреба у підборі ефективних інструментів для структурування навчального процесу з використанням інтерактивних елементів, що допоможе підвищити інтерес та мотивацію до навчання.

Одним із таких методів є використання гейміфікації, що полягає у застосуванні елементів ігрового дизайну (системи балів, рівнів, нагород, змагань, зворотного зв'язку) у неігровому контексті, зокрема в освіті та сомоосвіті (Вербих, 2015). У роботі О. Дакалюк (Дакалюк, 2020) зазначається, що для підвищення мотивації, активізації самоосвіти варто запроваджувати різні форми інформаційно-комунікаційних технологій. Як один із методів ефективного залучення студентів до освітнього процесу І. Чижикова (Чижикова, 2023) пропонує використовувати гейміфікацію, що допомагає відчуті їм свою автономію під час навчання, допомагає відчувати зв'язок між вивченим матеріалом та реальними умовами використання, самостійно вирішувати який об'єм матеріалу за який час вивчати. Також питаннями гейміфікації займалися вчені: О. Кривонос, С. Горобець, М. Кривонос, К. Нехаєнко, Н. Арістова для підвищення мотивації при вивченні інформатики (Кривонос та ін., 2023; Арістова & Махович, 2023), О. Макаревич (Макаревич, 2015) для ефективності дистанційного навчання; О. Булгакова, В. Зосімов для кращих досягнень у

навчанні; Т. Чернова, А. Ольшевський, В. Іванов для мотивації до самоосвіти (Чернова та ін., 2025) та інші.

Мета дослідження. Дослідження гейміфікації як ефективного засобу організації самоосвіти здобувачів освіти у закладах фахової передвищої освіти та розробка методичного забезпечення для її реалізації засобами мобільних технологій.

Об’єкт дослідження – організація самоосвіти студентів у системі закладів фахової передвищої освіти.

Предмет дослідження – методичні підходи до організації самоосвітньої діяльності здобувачів освіти засобами гейміфікованих мобільних технологій в процесі вивчення інформатики в закладах фахової передвищої освіти.

Завдання дослідження:

- здійснення аналізу ключових понять за темою дослідження;
- узагальнення та систематизація теоретичного досвіду;
- вивчення структури та змісту гейміфікованих мобільних застосунків;
- розробка гейміфікованого мобільного застосунку «InformatikaGO» для організації самоосвіти здобувачів закладів фахової передвищої освіти;
- розробка методичних рекомендацій із впровадження застосунку у процес навчання інформатики у закладах фахової передвищої освіти.

Теоретичне значення дослідження полягає в аналізі поняття «самоосвіта» як важливої складової освітнього процесу закладів фахової передвищої освіти, на основі вивчення досвіду використання гейміфікації у викладанні дисциплін комп’ютерного циклу визначено вимоги до ефективного гейміфікованого середовища.

Практичне значення отриманих результатів кваліфікаційного дослідження полягає у тому, що: розроблено концепцію гейміфікованого мобільного застосунку та його функціональну структуру ефективну для самоосвіти студентів; спроектовано та реалізовано архітектуру та основні компоненти гейміфікованого мобільного застосунку «InformatikaGO» як

ефективного засобу організації самоосвіти здобувачів закладів фахової передвищої освіти.

Апробація роботи. Результати дослідження у вигляді доповідей представлено на звітних науково-практичних конференціях співробітників, студентів та аспірантів РДГУ, XVII Всеукраїнській науково-практичній конференції «Інформаційні технології в професійній діяльності» (м. Рівне, 2024), IV Всеукраїнській науково практичній конференції «Підготовка педагогів до професійної діяльності в умовах змішаного навчання» та на пленарному засіданні XVIII Всеукраїнської науково-практичної конференції «Інформаційні технології в професійній діяльності» (м. Рівне, 2025). За результатами опубліковано двоє тез конференцій (Пікун, 2024; Пікун, 2025).

Структура та обсяг роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Обсяг роботи складає 76 сторінок, в тому числі основного тексту 54 сторінки, 22 рисунків та 7 таблиць.

РОЗДІЛ I

ТЕОРЕТИЧНІ ОСНОВИ САМООСВІТИ ТА ГЕЙМІФІКАЦІЇ В ОСВІТІ

1.1 Самоосвіта як складова навчального процесу в ЗФПО

На четверту частину XXI століття припадає велика кількість змін в освіті. Світ не стоїть на місці, те що вивчалось вчора, сьогодні вже може бути не актуальним. Причинами цих змін стали значна кількість чинників. Одним із таких чинників можна вважати пандемію COVID-19, яка глибоко порушила глобальну систему освіти, спричинивши значні виклики та трансформації. Оскільки школи та університети по всьому світу були вимушені припинити очне навчання, швидкий перехід до онлайн-навчання виявив значні прогалини в технологічній інфраструктурі, цифровій грамотності та рівному доступі до освіти (Стефанська, 2023).

Варто не забувати про повномасштабне вторгнення в Україну, коли учні та студенти були змушені вчитися в умовах повного відключення світла, та переховуватись у бомбосховищах заради збереження власного життя. Аналізуючи дані пошукового запиту «Самоосвіта» в Google Trends (рис. 1.1), можна побачити що інтерес із часом став більш активним починаючи з 2022 року та досягає піку у вересні місяці кожного наступного року. Ці дані свідчать про актуальність теми самоосвіти в Україні (Google trends, 2025).

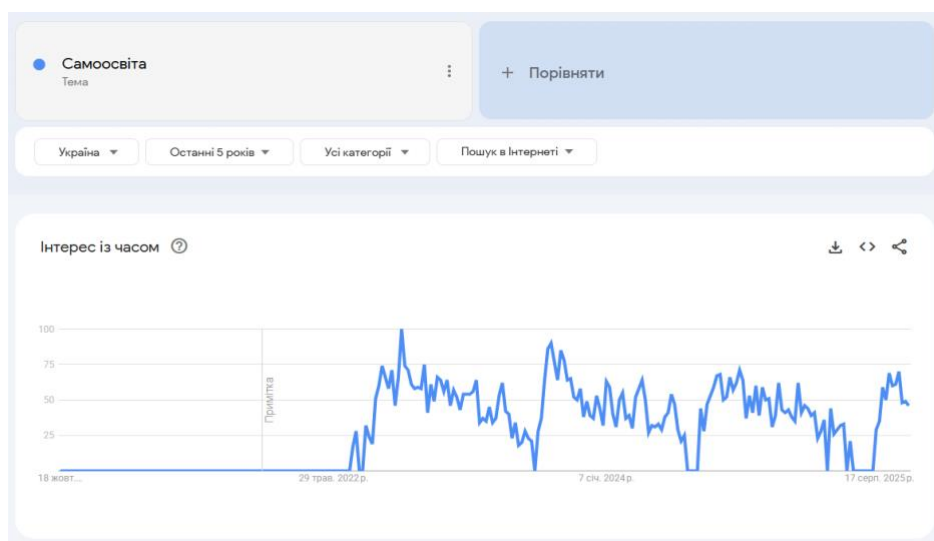


Рис. 1.1 Динаміка активності запиту протягом 2020-2025 років

Такі загрози ставлять під питання існування освіти в цілому та лише підкреслює важливість навичок самоосвіти для сучасної людини. Якщо на початку століття самоосвіта займала місце як додатковий елемент для навчання, то тепер це невід’ємна, свідомо організована та цілеспрямована пізнавальна діяльність.

Дослідження показують, що самоосвіта – це процес, у якому індивід активно займається власним навчанням та розвитком без прямого керівництва вчителя або навчальної програми (Підгорна, 2024). Тобто у самоосвіті роль суб’єкта та об’єкта освіти відіграє одна особа.

Дуже важливо не прирівнювати самоосвіту до самостійної роботи. Самоосвіта: мета визначається самим студентом, без контролю з боку викладача. Самостійна робота: виконується за завданням та під керівництвом викладача.

Самоосвіта – це не просто накопичення інформації. Це є багатокомпонентна діяльність, яка охоплює низку взаємопов’язаних процесів. І. Жорова та інші науковці пропонують розглядати її як сукупність декількох ключових «само-» компонентів (Жорова та ін., 2012):

- Самооцінка – це здатність особи, об’єктивно оцінювати власні можливості та потреби.
- Самооблік – вміння враховувати не тільки наявність своїх професійних якостей а і їх якість.
- Самовизначення – здатність усвідомити свої інтереси та обрати своє місце в житті й суспільстві.
- Самоконтроль – здатність керувати власну діяльність, а також її регулювати.

Самоосвіта є ключовою складовою навчального процесу в закладах фахової передвищої (професійної) освіти, оскільки вона забезпечує постійне професійне зростання педагогічних працівників. А також формує конкурентоспроможного, самостійного фахівця серед здобувачів освіти.

В умовах стрімкого розвитку науки та освітніх технологій, самоосвіта є об’єктивною необхідністю і головною формою підвищення професійної

компетентності для викладачів та майстрів виробничого навчання (Мельник, 2013). Як вважає К. Ушинський, «педагог живе до тих пір, поки він учиться, коли він перестає вчитися, у ньому вмирає педагог».

Причини які мотивують педагогів до самоосвіти, можуть бути різними. Одне із джерел (Жорова та ін., 2012) пропонує розглядати такі фактори, як зміни в змісті освіти, освоєння нових навчальних програм, запровадження нестандартних форм, пошук оригінальних методик викладання, конструювання нового педагогічного досвіду на підставі досягнень науки та практики з обраної проблеми, поглиблення знань із питань, які викликають інтерес, конкуренція, громадська думка, особистісні якості.

Основні напрями самоосвіти педагога:

Технологічний: Вивчення нового обладнання та технологічних процесів через стажування, вебінари від виробників.

Методичний: Опанування сучасних педагогічних технологій (проектне навчання, STEM-підходи).

Психолого-педагогічний: Розвиток soft-skills, необхідних для ефективної роботи з молоддю.

Головною метою закладів передвищої освіти є сформувати кваліфікованого робітника, який зможе самостійно вирішувати професійні завдання. Самоосвіта тут виступає як інструмент формування самоосвітніх компетентностей. Таких як вміння планувати, організовувати, здійснювати та оцінювати власну пізнавальну діяльність. Таким чином самоосвіту можна вважати не самоплинним процесом, а спланованою навчально-пізнавальною діяльністю, що спрямовується на досягнення певного результату (Шлеїна, 2023).

Аналіз наукових джерел засвідчує, що однією з найбільш поширених та обґрунтованих класифікацій є рівневий підхід для оцінки готовності студентів до самоосвіти. Зокрема, дослідник В. Буряк пропонує розділяти готовність студентів до самоосвіти на три ключові рівні, що різняться за мотиваційними, когнітивними та організаційно-діяльнісними характеристиками: початковий, середній та вищий (Буряк, 2002).

Початковий рівень відрізняється стихійністю мотивів та зовнішньою залежністю. В цей період мотиви самоосвіти студентів здебільшого стихійні, адже вони не пов'язують власні особисті потреби навчання з більш широкими суспільними інтересами та цілями. В області знань спостерігається їхній ізольований характер: студенти часто не бачать навіть міжпредметних зв'язків. Організаційно-діяльна частина також слабка: управління джерелами інформації не систематизується, а окремі джерела зовсім не використовуються.

Для середнього рівня готовності характерне бажання до становлення самостійності та системного осмислення навчального матеріалу. Мотиваційна сфера розширюється: студенти можуть самостійно встановлювати собі цілі самоосвіти та здійснювати їх якісно. Вони вже розуміють необхідність поєднання самоосвіти із інтересами суспільства, але ще не завжди знають, як саме чітко сформулювати відповідну соціально значиму мету. На когнітивному рівні знання з навчальних предметів вже систематизовані, але міжпредметні зв'язки ще не завжди добре усвідомлюються. У вже студенти знають як працювати з основними джерелами інформації і можуть самостійно організувати процес самоосвіти. Але їм може не вистачати навичок для раціонального планування цілісного засобу самоосвіти.

Вищий рівень відображає найкращий рівень розвитку особистості як суб'єкта самоосвіти. Основою мотивацією є глибоке сприйняття необхідності розуміння суспільно значущих цілей та володіння навичками, правильно їх формулювати і досягати оптимальним чином. Знання особистості набувають цілісного змісту, який будується на глибокому розумінні наявності міжпредметних зв'язків у науці. В організаційно-діяльному плані, самостійний студент вміє застосовувати різноманітні джерела інформації та здатний, незалежно на науковій основі, здійснювати оптимальний контроль всього процесу самоосвіти – від організації і здійснення ідеї до самоконтролю отриманих студентських результатів.

Шляхи інтеграції самоосвіти в навчальний процес:

Проблемне та проєктне навчання: Заохочення студентів до самостійного пошуку інформації та рішень для реалізації навчальних проєктів.

Цифровізація: Активне використання відкритих освітніх ресурсів (Coursera, Prometheus, YouTube-канали фахівців) для поглибленого вивчення фаху.

Виробнича практика: Створення умов, коли студенти на підприємстві мусять самостійно освоювати нові прийоми роботи, консультуючись із наставниками та майстрами.

Самоосвіта є стратегічною складовою навчального процесу в ЗФПО, що забезпечує синхронізацію освітньої підготовки з динамічними вимогами ринку праці. Інтеграція самоосвіти в освітній процес сприяє не лише оновленню професійного рівня педагогів, а й формує у здобувачів освіти ключову здатність – бути самодостатніми та конкурентоспроможними фахівцями.

На основі проведеного аналізу поняття можемо зробити висновок про психолого-педагогічні особливості самоосвіти, а саме: потрібно розвивати пізнавальні мотиви студентів, адже самоосвіта в першу чергу ґрунтується на внутрішній мотивації студента; викладач повинен здійснювати цілеспрямований педагогічний вплив не лише як джерело знань, а більш як наставник, тьютор; самоосвіта допомагає розвинути таку важливу компетентність як навчальна автономія, що формує готовність студентів до безперервного навчання впродовж життя.

1.2 Гейміфіковані технології у навчанні: сутність, види та переваги

Сучасна освіта орієнтована на підвищення мотивації та залученості здобувачів. Одним із ефективних засобів цього є гейміфікація – використання ігрових механік у неігровому середовищі, зокрема у процесі навчання (Zichermann & Cunningham, 2011).

За даними Google Trends, починаючи з 2022 року значно зріс відсоток пошукового інтересу до терміну «Гейміфікація» в Україні (рис. 1.2). На графіку видно регулярні коливання з яскраво вираженими піками інтересу, які часто повторюються. Це може свідчити про початок навчальних семестрів (коли гейміфікація активно обговорюється в освіті), запуск великих маркетингових кампаній або бізнес-проектів із застосуванням гейміфікації.

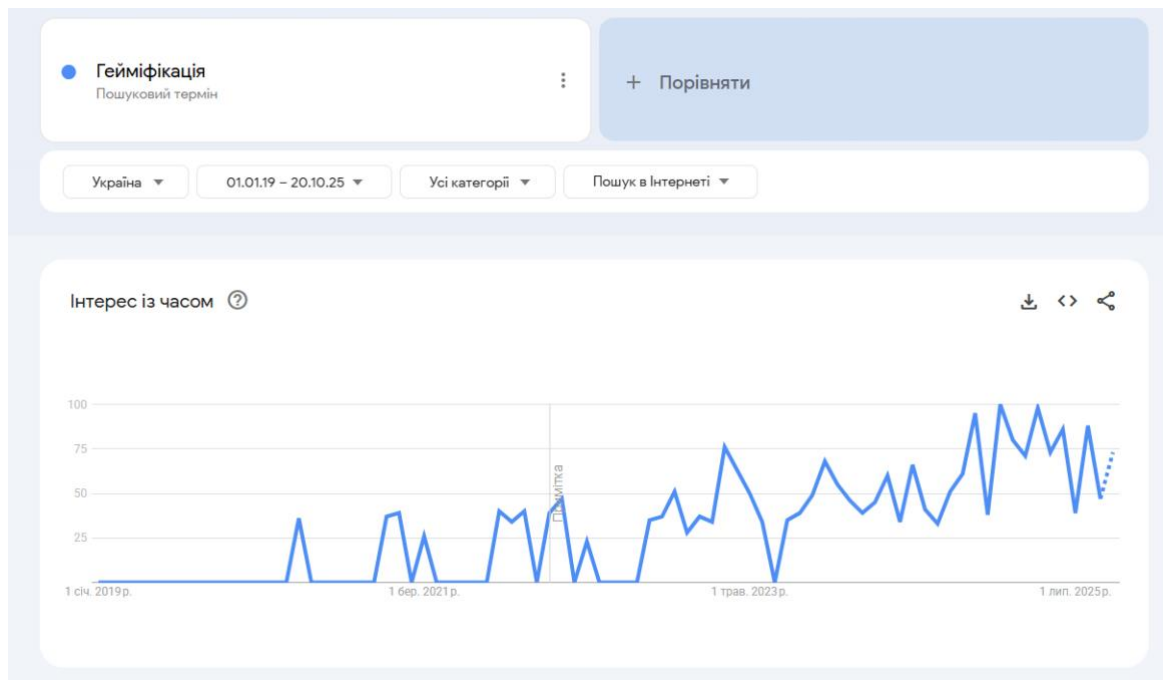


Рис. 1.2 Динаміка пошукового інтересу до терміну «Гейміфікація» в Україні

У сфері сучасної освіти часто зустрічаються два популярні слова: гейміфікація та ігрове навчання. Хоча поняття виглядають подібними, вони представляють собою кардинально різні методи до викладання та навчання.

Як було зазначено вище, гейміфікація – це використання ігрових механік у неігровому середовищі. Тобто беруться конкретні елементи гри (бали, квести або бейджі), та вводяться в процес навчання. Приклад: викладач створює квест, кожна тема це місія, правильні відповіді відкривають наступний рівень.

З іншого боку «Ігрове навчання» передбачає розгляд гри, як моделі реального світу. За допомогою ігрового навчання, ігрове середовище моделює

умови та концепції які можна побачити в реальному світі. Наприклад гра «Kerbel Space Program» точно моделює ракетобудування та реальну фізику. Гра «PC Building Simulator» зосереджена на керуванні майстернею, яка створює та обслуговує комп'ютери, що моделює реальні основи володіння бізнесом будови комп'ютерів.

Тобто гейміфікація застосовує ігрові елементи або ігрову структуру до існуючих навчальних видів діяльності; ігрове навчання розробляє навчальні види діяльності, які за своєю суттю є ігровими (Gamification and game-based learning, б. д.)

На основі досліджень визначено основні елементи гейміфікації, які наведено в таблиці 1.1.

Таблиця 1.1

Основні елементи гейміфікації

Елемент	Опис	Приклад у навчанні
Бали (points)	Відображають результативність студента	Отримання балів за виконані завдання
Рівні (levels)	Показують прогрес навчання	Перехід на «наступний рівень» після тестів
Бейджі (badges)	Візуальні нагороди за досягнення	Бейдж «Перший вхід», «100% тесту»
Рейтинг / таблиця лідерів	Стимулює змагання між учнями	Лідери тижня у застосунку
Streak (послідовність днів)	Мотивує до регулярності	Вогник активності, як у Duolingo
Місії, квести	Створюють відчуття пригоди та цілі	Виконання тематичних завдань по розділах

Розглянемо детальніше ключові з них.

Бали (points). Бали є одним із базових елементів гейміфікації, який використовується для кількісної оцінки досягнень користувача. В освітньому контексті бали виступають показником прогресу у навчанні: за виконання певних

завдань, проходження тестів чи досягнення цілей студент отримує певну кількість балів.

Система балів забезпечує миттєвий зворотний зв'язок, дозволяючи студенту бачити результати своєї роботи в реальному часі. Це сприяє формуванню внутрішньої мотивації, оскільки навіть незначні досягнення винагороджуються.

Рівні (levels). Рівні відображають поступовий розвиток користувача, що просувається від простіших до складніших завдань. У навчальному середовищі вони формують структуру навчального процесу, розподіляючи матеріал на логічні етапи.

Перехід між рівнями створює у студентів відчуття прогресу, досягнення нових вершин і зростання компетентності. Це підсилює мотивацію до навчання, особливо коли кожен рівень супроводжується новими можливостями або винагородами (наприклад, відкриттям додаткових тем, доступом до складніших тестів чи нових бейджів). Рівнева система також допомагає індивідуалізувати навчання, оскільки кожен студент може рухатись у власному темпі, поступово вдосконалюючи свої навички.

Бейджі (badges). Бейджі – це візуальні значки та символи, що свідчать про досягнення користувача у певній сфері. Вони виконують функцію нематеріальної винагороди, яка має значну психологічну цінність, оскільки демонструє успіхи користувача перед іншими учасниками або самим собою.

У навчальному контексті бейджі можуть надаватись за:

- перше виконання завдання;
- завершення курсу або теми;
- досягнення високих результатів у тестах;
- щоденну активність упродовж певного часу.

Рейтинг або таблиця лідерів (leaderboard). Рейтинг або таблиця лідерів є потужним інструментом соціальної гейміфікації. Вона дозволяє порівнювати досягнення студентів між собою, що активізує здорове суперництво та підвищує рівень залученості. Рейтингова система може будуватися на основі набраних

балів, виконаних завдань або здобутих бейджів. Проте важливо зберігати баланс між мотивацією та психологічним комфортом, щоб уникнути відчуття тиску чи зневіри у тих, хто має нижчі результати.

Streak (послідовність днів). Streak (серія або послідовність днів) – це елемент гейміфікації, який стимулює регулярність навчання. Користувач отримує винагороду (візуальну або символічну) за безперервні дні активності в системі. Такий механізм базується на психологічному ефекті «ефекту серії» (streak effect): людина не хоче «перервати ланцюжок», тому намагається заходити до застосунку щодня. Це формує навичку систематичної самоосвіти і допомагає підтримувати постійний темп навчання.

Місії та квести (missions / quests). Місії або квести – це структуровані завдання з чітко визначеною метою, які створюють відчуття сюжетного розвитку в навчальному процесі. Вони дозволяють студентам відчути себе «героями», що проходять шлях до майстерності, виконуючи послідовні завдання.

Квести можуть бути індивідуальними або груповими, короткостроковими або тривалими. Їхня головна функція – перетворення навчального процесу на інтерактивну історію, де кожне завдання має значення для загального результату.

Варто зазначити, що на даний час є досить велика кількість джерел гейміфікації. І саме від технічних можливостей студента залежить вибір того чи іншого засобу (платформи, сайту, додатків чи ігор). Наведемо ефективні засоби гейміфікації освітнього процесу (рис. 1.3), що успішно використовуються на заняттях або для самоосвіти (Чижикова, 2023).



Рис. 1.3 Ефективні засоби гейміфікації

У праці А. Токаревої «Навчальні цифрові ігри: від теорії до практики» (Токарева, 2019, с. 103) розкривається п'ять стилів навчання: через передачу знань, імітацію освітнього процесу, експеримент, навчання через залучення, через відкриття.

В розрізі дослідження зосереджуємо увагу на особливості впровадження гейміфікації для самоосвіти, тобто за межами класу.

1.3 Досвід використання гейміфікації у викладанні комп'ютерних дисциплін

Вплив гейміфікації на мотивацію студентів важко переоцінити. Цей її аспект розглядається багатьма дослідниками. Так у праці українського вченого І. Ярича йдеться про те, що гейміфікація є одним із трендів сучасності (Ярич, 2025).

Дослідимо хто у своїх працях розкривав практичні аспекти використання гейміфікації при викладанні дисциплін комп'ютерного циклу.

У дослідженні зарубіжних вчених (Lampropoulos та ін., 2023) розкрито результати експериментального дослідження про вплив гейміфікації, доповненої реальності та ігор вцілому на вивчення комп'ютерних дисциплін. Розроблений та представлений мобільний застосунок, який проектувався разом зі студентами мав на меті покращення освітнього процесу через створення інтерактивного, інклюзивного та студентоцентрованого освітнього середовища. Зазначено, що залучення студентів до розроблення таких застосунків значно підвищує мотивацію студентів, розвиток їх комунікативних здібностей та критичного мислення.

У дослідженні Kashif Ishaq, Atif Alvi йдеться про те, що система навчання, яка на даний час застосовується більшістю закладів фахової передвищої та вищої освіти перенавантажують здобувачів освіти навчальним теоретичним матеріалом, що в свою чергу погіршує мотивацію до навчання

(Ishaq & Alvi, 2023). Вони пропонують використання персоналізованої гейміфікації, що поєднує гейміфікацію та персоналізоване навчання та дозволить враховувати індивідуальні особливості здобувачів освіти під час вивчення мов програмування.

Також для підвищення знань з програмування колективом авторів (Dissanayake та ін., 2024) запропоновано комплексну освітню платформу, що містить персоналізовані вікторини, практичні проекти та адаптивні моделі навчання. Платформа розроблена з використанням передових моделей штучного інтелекту, зокрема для реалізації індивідуальних траєкторій навчання кожного здобувача освіти після перевірки його успішності.

Авторами Nicha Sirikong, Porawat Visutsak (Sirikong & Visutsak, 2024) використано методи гейміфікації для покращення навчання програмуванню Python. Оскільки в учнів виникали проблеми з вивченням складних конструкцій через традиційні методи, то була розроблена гра, що включала бали, винагороди, таблиці лідерів, обмеження в часі для створення інтерактивного середовища. Результатом дослідження було визначення рівня задоволеності учнів таким форматом навчання. Було підкреслено, що в учнів зросла цікавість до вивчення Python зокрема, що підкреслило потенціал гейміфікації для сприяння ефективному та цікавому навчанні.

Аналіз праць українських дослідників свідчить про те, що гейміфікація розглядається як метод підвищення мотивації до навчання різних навчальних дисциплін. Це підтверджують праці І. Махович (Makhovych, 2025) та інших науковців.

РОЗДІЛ II

АНАЛІЗ ТА ПРОЄКТУВАННЯ ГЕЙМІФІКОВАНОГО СЕРЕДОВИЩА ДЛЯ
САМООСВІТИ З ІНФОРМАТИКИ

2.1 Аналіз існуючих гейміфікованих платформ для вивчення інформатики

Стрімкий розвиток ІТ-сфери та постійна потреба в актуалізації знань робить ефективні інструменти самоосвіти критично важливими. У цьому контексті гейміфікація визнана потужним засобом підвищення мотивації та залученості. Про це свідчить висока популярність освітніх застосунків у загальних рейтингах мобільних платформ. На рисунку 2.1 видно, що категорія «Освіта» займає перше місце за кількістю застосунків на Google Play та четверте на App Store (Google play vs ios app store, 2025).



Рис. 2.1 Графік порівняння категорій застосунків з ресурсу 42matters

Метою даного дослідження є проведення загального аналізу поширених гейміфікованих застосунків у сфері комп'ютерних дисциплін для ідентифікації

ключових гейміфікаційних механік та педагогічних обмежень. Отримані результати сформують основу для визначення вимог до розробки власного гейміфікованого середовищ

З огляду на необхідність визначення базових тенденцій та ефективних практик, був проведений пошук на платформах Play Market та App Store.

Критерії вибору застосунків для загального аналізу:

1. Тематична відповідність: Безпосередній зв'язок із комп'ютерними дисциплінами (програмування, алгоритми, ШІ, мережі).
2. Наявність гейміфікаційних елементів: Чітке використання принаймні трьох елементів (квізи, бали, рейтинги, бейджі, тощо).
3. Популярність та відгуки: Наявність значної кількості завантажень або високий рейтинг (вище 4.0).

В результаті пошуку за ключовими запитами («computer science learn,» «algorithms quiz,» «вивчення комп'ютера») було виявлено відносно невелику кількість застосунків, які одночасно відповідають критеріям (1) та (2). Для забезпечення репрезентативності первинної вибірки, було проаналізовано п'ять найбільш релевантних платформ.

У таблиці 2.1 наведено приклади поширених застосунків, на платформі Play Market, для навчання комп'ютерних дисциплін (Додатки android у google play, 2025)

Таблиця 2.1

Порівняльний аналіз застосунків

№	Назва платформ	Основні теми	Елементи гейміфікації	Переваги	Обмеження
1	Вивчити комп'ютер 2023	Будова комп'ютера, базові поняття	Квізи, бейджі, рейтинги	Доступ до курсів офлайн; фокус на базовій термінології.	Обмежена кількість тем та навчального матеріалу; модель пасивного навчання.

2	Informática Oposiciones Pro	ІКТ, основи комп'ютерної грамотності	Квізи, тести, відстеження прогресу	Інтуїтивний інтерфейс, орієнтація на тестування знань.	Відсутня українська мова; вузька спрямованість на підготовку до іспитів.
3	Computer Science Quiz Topics	Штучний інтелект, Мережі, Безпека, Алгоритми	Квізи, змагальний режим (PVP)	Пояснення відповідей; широке охоплення теоретичних дисциплін.	Відсутня українська мова; надмірний акцент на запам'ятовуванні теорії.
4	Computer Science Engg in 2024	Штучний інтелект, Алгоритми та структури даних	Тести, рейтинги, система рівнів	Велика кількість інформаційного наповнення, структуровані й контент.	Відсутня українська мова; брак практичних завдань (кодування, симуляції).
5	Brilliant: Learn by doing	Алгоритмізація, Програмування, Кібербезпека, ШІ	Рейтинг, бейджі, серія, внутрішньоігрова валюта, нагадування	Персоналізоване навчання; акцент на інтерактивних задачах та вирішенні проблем.	Переважно англomовний контент; більшість поглибленого матеріалу платна.

Проведений загальний аналіз дозволив ідентифікувати спільні ефективні механіки та критичні прогалини у гейміфікованих застосунках для вивчення комп'ютерних дисциплін:

Ключові елементи та переваги:

- Базові гейміфікаційні елементи: Абсолютна більшість застосунків активно використовує квізи/тести, рейтинги (Leaderboards) та бейджі (Badges). Це підтверджує їхню ефективність як мінімального набору інструментів для підтримки мотивації.
- Мотиваційні інструменти: Платформи, подібні до Brilliant (№5), демонструють вищу залученість завдяки персоналізації, серії (streaks) та використанню внутрішньоігрової валюти для створення економіки застосунку.

Критичні обмеження та прогалини:

- Пасивна модель навчання: Головним недоліком більшості проаналізованих застосунків (№1, №2, №3, №4) є надмірний акцент на тестуванні теоретичних знань (модель «запам'ятай-відтвори»). Це не сприяє

розвитку критично важливих практичних навичок у сфері ІТ, таких як кодування, налагодження чи системний аналіз.

- Недостатня практична орієнтація: Існує значний брак застосунків із вбудованими середовищами для практичного кодування або симуляції реальних ІТ-задач, які б дозволили учню застосовувати знання, а не лише їх перевіряти.
- Мовна та культурна адаптація: Застосунки з найширшим наповненням (№3, №4, №5) не мають україномовної локалізації, що обмежує їхнє використання у вітчизняному освітньому просторі.

2.2 Вимоги до ефективного гейміфікованого середовища для самоосвіти

На основі даних таблиці 2.1 та визначивши основні елементи, переваги, критичні обмеження та прогалини застосунків, можна сформулювати наступні вимоги для гейміфікованих платформ.

Вимоги до контенту

1. Чіткі цілі навчання та правила гри

Користувач має розуміти як йому вивчати нову тему(наприклад: здобути певну кількість балів за правильні відповіді і відкрити нові теми). Правила гри повинні бути простими, якщо вони надто складні користувач швидко втратить інтерес.

2. Ігрові механіки з елементами навчання

Правильно підібрані ігрові елементи додають цікавості у процес вивчення нового матеріалу, але й не відволікають від нього. Користувачі мають бачити свій прогрес. Для цього можна використовувати прогрес-бари або шкалу досвіду. За кожну дію повинна бути винагорода для підтримки цікавості. Винагороди мають бути цінними і пов'язані з навчальним процесом.

3. Соціальна взаємодія та змагання

Гейміфікація працює краще, коли присутня соціальна взаємодія. Чудовий приклад застосунок №3, він має змагальний режим який сприяє до вивчення нового. Невід’ємною частиною гейміфікації є рейтинг. Важливо щоб він не визначався тільки за одним критерієм, необхідно враховувати фактори швидкості виконання завдання, точністю та пройдених знань.

4. Гнучкість та персоналізація

Кожен студент вчиться по-своєму і враховувати це є однією з вимог ефективного середовища. Система має адаптовуватися під потреби кожного користувача. Якщо студент робить багато помилок, система запропонує легші завдання. Якщо навпаки студент швидко виконує все то система пропонує складніші завдання

5. Наратив та ігровий дизайн

Візуалізація інформації у вигляді віртуального світу допоможе глибше зануритись в ігровий процес. Додаток №5 має внутрішню історію, яка допомагає краще сприймати матеріал, а отже ефективніше навчатись. Візуальний стиль середовища повинен бути приємним та інтуїтивним.

Технічні вимоги:

1. Операційна система

Відповідно до статистики Statcounter, в Україні на період від червня 2024 року до червня 2025 року, середнє значення мобільної операційної системи Android становить 69,11%. Для IOS це значення складає 30,52% . Інші операційні системи займають 0,36% (*Mobile operating system market share ukraine, 2025*). Проаналізувавши дані з ресурсу Statcounter (рис. 2.2), визначили, що основними мобільними операційними системами для українських користувачів є Android та IOS. Тобто для вибору середовища розробки необхідно враховувати властивість кросплатформного експорту проєкту.

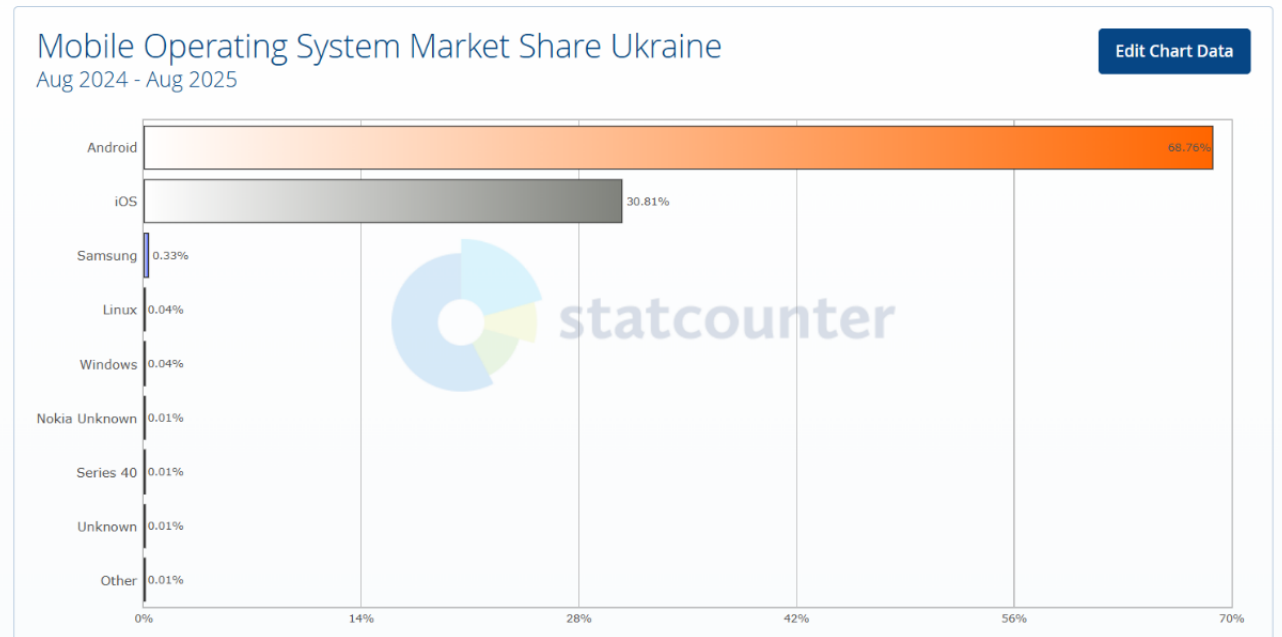


Рис. 2.2 Дані з ресурсу Statcounter про використання операційних систем у відсотках

2. Оптимізованість

Додаток повинний бути оптимізований для більшості пристроїв. Ефективне навчання можливе лише за нормальної взаємодії з застосунком. Неприпустимими є зависання, постійні збої та перевантаження системи користувача. Додаток має на меті вирішувати проблеми, а не створювати нові.

3. Безпека даних

При роботі з особистими даними користувачів, розробник зобов'язаний забезпечити надійне їх зберігання. Платформи на яких зберігаються дані не повинні передавати конфіденційну інформацію третім особам. Користувачі мають право надсилати запити на повне видалення особистих даних. Розробник програмного повинен вчасно оновлювати застосунок при виявленні загрози для даних користувачів.

4. Масштабованість

Застосунок має бути готовим для додавання нового функціоналу, та зростання аудиторії. Продуктивність додатку не повинна падати при стрімкому зростанню користувачів. Заздалегідь продумана масштабована архітектура

спростить інтеграцію нових можливостей, не впливаючи на стабільність і продуктивність існуючих.

5. Локалізація

На основі проведених досліджень з попереднього пункту, більшість застосунків не мають української мови. Однією з головних вимог застосунку є наявність такої мови за замовчуванням. Згідно закону України «Про забезпечення функціонування української мови як державної», зокрема статті 27 «Державна мова у сфері інформації для загального ознайомлення та у сфері інформаційних технологій», встановлено певні вимоги. Однією з таких вимог є те, що програмне забезпечення яке реалізується в Україні, повинно мати користувацький інтерфейс українською мовою. Якщо додаток має кілька мовних версій, україномовна версія за обсягом та змістом не може бути меншою за іншомовні версії (Про забезпечення функціонування української мови як державної, 2025).

6. Тестування

Готовий продукт повинен бути ретельно протестований перед загальною публікацією. Це є критично важливий етап у життєвому циклі розробки, оскільки допомагає виявити та виправити помилки, перш ніж додаток потрапить до кінцевих користувачів. Перевіряється коректність роботи функцій застосунку, сумісність з пристроями, безпека даних та продуктивність.

Вище описані вимоги є універсальними та можуть бути використані для розробки подібних платформ.

2.3 Розробка концепції мобільного застосунку InformatykaGO

Сучасний освітній процес характеризується високим рівнем динамічності та діджиталізації. Це вимагає створення нових засобів підтримки самоосвіти студентів. Одним із таких засобів є мобільний застосунок InformatykaGO, який поєднує принципи повсюдного навчання (Ubiquitous learning, U-learning) та

елементи гейміфікації для підвищення мотивації і залучення здобувачів освіти до самостійної роботи з інформатики.

Повсюдне Навчання (Ubiquitous Learning, U-Learning) є однією з найважливіших трансформаційних парадигм у сучасній освіті. Дослідження Ji-Ping Zhang показало, що воно виникло в процесі еволюції процесів у цифровій освіті двох парадигм: електронне навчання (U-Learning) та мобільне навчання (M-Learning). Повсюдне навчання визначається не просто як використання цифрових інструментів, а як нова освітня модель, яка активно використовує можливості, надані цифровими медіа, для забезпечення гнучкого навчального досвіду (Zhang, 2008).

Мобільний застосунок InformatykaGO, має на меті підвищити ефективність самоосвітньої діяльності студентів закладів передвищої освіти через інтеграцію навчального контенту, системи тестування, мотиваційних механізмів і персоналізованого відстеження прогресу у зручному мобільному середовищі.

Для реалізації поставленої мети визначено такі основні завдання:

- Розробити архітектуру мобільного додатку, яка надає доступ до навчального контенту з інструментами самооцінювання.
- Реалізувати механізми гейміфікації (бали, рівні, бейджі, рейтинг, streak, квести) для формування внутрішньої мотивації студентів.
- Інтегрувати принципи повсюдного навчання (U-learning) для забезпечення навчання будь-де і будь-коли.
- Забезпечити зручний інтерфейс і логічну навігацію, орієнтовану на студентів закладів професійної та передвищої освіти.
- Провести тестування застосунку й оцінити ефективність його використання у навчальному процесі.

Основні принципи, покладені в основу застосунку InformatykaGO, такі:

- Мобільність – можливість навчатися у будь-якому місці та в будь-який час за допомогою смартфона.
- Персоналізація – підлаштування контенту під потреби й темп навчання користувача; можливість бачити власний прогрес.

- Інтерактивність – активна взаємодія з матеріалом через відео, тести, практичні завдання.
- Доступність – простий і зрозумілий інтерфейс, що не вимагає спеціальних технічних знань.
- Безперервність навчання – підтримка регулярного залучення через механізм streak і нагадування про активність.

U-learning у InformatykaGO реалізується через поєднання коротких навчальних блоків (відео, пояснення, завдання), які користувач може виконувати в зручний для себе момент, із можливістю зберігати результати у хмарній базі даних MongoDB.

Цільова аудиторія мобільного застосунку охоплює здобувачів освіти закладів передвищої (професійної) освіти, які вивчають інформатику та суміжні дисципліни.

Основні групи користувачів:

1. Студенти та учні закладів фахової передвищої (професійної) освіти

- Віковий діапазон: 15–20 років.
- Потреби: зручний доступ до навчальних матеріалів, тестів і відеоуроків у будь-який час.
- Мотивація: підвищення рівня знань, підготовка до заліків і контрольних робіт, розвиток цифрової грамотності.
- Особливості поведінки: часто користуються смартфонами, віддають перевагу коротким інтерактивним форматам подачі матеріалу.

2. Викладачі інформатики та суміжних дисциплін

- Використовують застосунок як додатковий інструмент для організації самоосвітньої діяльності студентів.
- Мотивація: контроль прогресу учнів, гейміфікація навчання, підвищення зацікавленості здобувачів освіти.

3. Освітні адміністратори або методисти

- Можуть аналізувати ефективність цифрових засобів навчання.

- Мотивація: підвищення якості освітнього процесу та впровадження інновацій у навчання.

2.4 Функціональна структура та дизайн застосунку

Одним із невід’ємних етапів розробки мобільного застосунку є вибір середовища розробки. Цифрова індустрія програмування налічує велику їх кількість. Найпопулярнішими технічними рішеннями є *React Native*, *Flutter*, *Xcode* та *Android Studio*. Кожне з них має свої переваги та недоліки.

Таблиця 2.1

Порівняльна таблиця React Native, Flutter, Xcode та Android Studio

Характеристика	React Native	Flutter	Xcode	Android Studio
Тип	Кросплатформний фреймворк	Кросплатформний фреймворк	IDE / Нативний інструмент	IDE / Нативний інструмент
Основна мова	JavaScript/TypeScript	Dart	Swift/Objective-C	Java/Kotlin
Розробник	Meta (раніше Facebook)	Google	Apple	Google (на основі IntelliJ IDEA)
Цільові платформи	iOS, Android, Web, Desktop (з обмеженнями)	iOS, Android, Web, Desktop	iOS, macOS, watchOS, tvOS	Android, ChromeOS
Підхід до UI	Використовує нативні компоненти через міст (Bridge)	Власні віджети, що рендеряться через Skia (власний рушій)	Нативні UI компоненти	Нативні UI компоненти
Продуктивність	Добра, але може поступатися нативній	Дуже висока, близька до нативної	Найвища (нативна)	Найвища (нативна)

Крива навчання	Середня (якщо знайомі з React)	Помірна (потрібно вивчити Dart та концепції віджетів)	Середня/Висока (Swift/Kotlin/Java + особливості платформи)	Середня/Висока (Swift/Kotlin/Java + особливості платформи)
Швидке перезавантаження (Hot Reload)	Так (Hot Reloading)	Так (Hot Reload)	Ні (або обмежено для SwiftUI/Compose)	Ні (або обмежено для SwiftUI/Compose)
Екосистема	Велика, завдяки популярності JavaScript/React	Швидко зростаюча, сильна підтримка від Google	Зріла, добре документована Apple	Зріла, добре документована Google
Використання	Швидка розробка кросплатформних додатків	Високопродуктивні кросплатформні додатки зі складним UI	Розробка нативних додатків для екосистеми Apple	Розробка нативних додатків для екосистеми Android

Важливо ретельно підбирати середовище розробки, адже це вплине на ефективність процесу створення програмного забезпечення. Враховуючи це було зарання визначено вимоги до такого середовища в пункті 2.2. Провівши аналіз, результати якого наведено в таблиці 2.1, можна зробити наступні висновки:

- Для максимальної продуктивності та доступу до всіх функцій платформи, найкраще підходить використання нативної розробки (Xcode для iOS/macOS, Android Studio для Android). Це вимагає більших ресурсів та підтримки двох окремих кодових баз.
- Для кросплатформної розробки (одна кодова база для двох ОС) конкурентами є React Native та Flutter.

- Перевага Flutter – висока продуктивність та узгодженість UI: Flutter використовує мову Dart, власний рендеринговий рушій Skia, що дозволяє йому «малювати» віджети безпосередньо, минаючи «міст» (як у React Native). Це гарантує, що інтерфейс виглядатиме абсолютно однаково на різних пристроях, незалежно від версії ОС, що є дуже важливим для застосунків які будуть використовуватись для навчання.

Врахувавши висновки та вимоги, визначили що для розробки мобільного застосунку InformatykaGO, найкраще підходить Flutter.

Flutter – це фреймворк з відкритим кодом, розроблений компанією *Google*. Його мета, створення візуально гармонійних, нативно скомпільованих, кросплатформних застосунків з єдиним кодом для *Web*, *Android*, *iOS* та *Windows* (Flutter on mobile, 2025).

Так як InformatykaGO працює з різними типами інформації (навчальні матеріали, результати тестування, персональні досягнення та прогрес), було прийнято рішення використовувати дві спеціалізовані бази даних. Використання двох окремих сховищ забезпечує чітке розмежування контенту та користувацьких даних. Такі дії покращують безпеку даних, масштабованість та зручність оновлення системи.

Для зберігання навчальних матеріалів(лекцій, тестів), було використано MongoDB. Воно визначається як система керування базами даних (СКБД) з відкритим кодом, яка є нереляційною. Також вона є NoSQL рішенням для баз даних. Замість використання таблиць і рядків, як у реляційних СКБД, MongoDB використовує гнучкі документи для зберігання та обробки даних.

Документи або колекції документів MongoDB є основними одиницями даних. Ці документи, відформатовані як Binary JSON (Java Script Object Notation), можуть зберігати різні типи даних і розподілятися між декількома системами. Оскільки MongoDB використовує динамічну схему проектування, користувачі мають безпрецедентну гнучкість при створенні записів даних, запиті колекцій документів через агрегацію MongoDB та аналізі великих обсягів інформації. У таблиці 2.№ наведено переваги та недоліки цієї СКБД у порівнянні з MySQL.

Таблиця 2.2

Порівняння MongoDB та MySQL

Характеристика	MySQL (Реляційна)	MongoDB (NoSQL)
Структура даних	Використовує схеми та таблиці для стандартизації типів даних. Має жорстку природу.	Менш обмежувачий формат. Використовує гнучкі документи.
Ключовий пріоритет	Цілісність та ізоляція даних є важливими (наприклад, при керуванні транзакційними даними).	Доступність та швидкість є основними.
Продуктивність	Підходить для широкого спектру ситуацій.	Пропонує вищу продуктивність.

Не менш важливим є порівняння MongoDB з собі подібними NoSQL, базами даних. Cassandra (<https://cassandra.apache.org/>) і MongoDB обидві вважаються базами даних NoSQL, але мають різні сильні сторони (табл. 2.3).

Таблиця 2.3

Порівняння MongoDB та Cassandra

Характеристика	Cassandra (NoSQL)	MongoDB (NoSQL)
Структура даних	Використовує традиційну структуру таблиць із рядками та стовпцями. Допомогає підтримувати одноманітність та довговічність.	Використовує гнучкі документи. Пропонує неперевершену гнучкість .
Перехід та синтаксис	Може запропонувати легший перехід до NoSQL, оскільки має синтаксис, схожий на SQL .	Використовує мову запитів, схожу на SQL бази даних.
Обробка даних	Надійно керує розгортанням та реплікацією без великої конфігурації.	Переважає у гнучкості для обробки структурованих та неструктурованих наборів даних .
Продуктивність	Не може зрівнятися з MongoDB за продуктивністю та надійністю для критично важливих хмарних програм.	Вища продуктивність і надійність, особливо для місійно-критичних хмарних додатків

Таким чином, у ширшому контексті NoSQL, MongoDB вирізняється серед конкурентів завдяки своїй динамічній схемі, що дозволяє легко додавати нові функції та атрибути, і здатності легко обробляти JSON-подібні документи, що робить її високоефективною при аналізі даних у реальному часі та забезпеченні гнучкості мобільних додатків.

Формат JSON-подібних документів дозволяє:

- зберігати лекції та тести у гнучкій структурі без жорстких схем;
- легко розширювати чи змінювати структуру окремих документів;
- швидко оновлювати навчальний контент без потреби оновлювати сам застосунок;
- формувати складні об'єкти (наприклад, питання з варіантами відповідей) у природній вкладеній формі.

Firebase – це набір інструментів для розробки мобільних застосунків та керування інфраструктурою. У застосунку InformatykaGO він відповідає за авторизацію користувачів та елементи гейміфікації. На рисунку 2.3 можна побачити повний перелік інструментів Firebase.

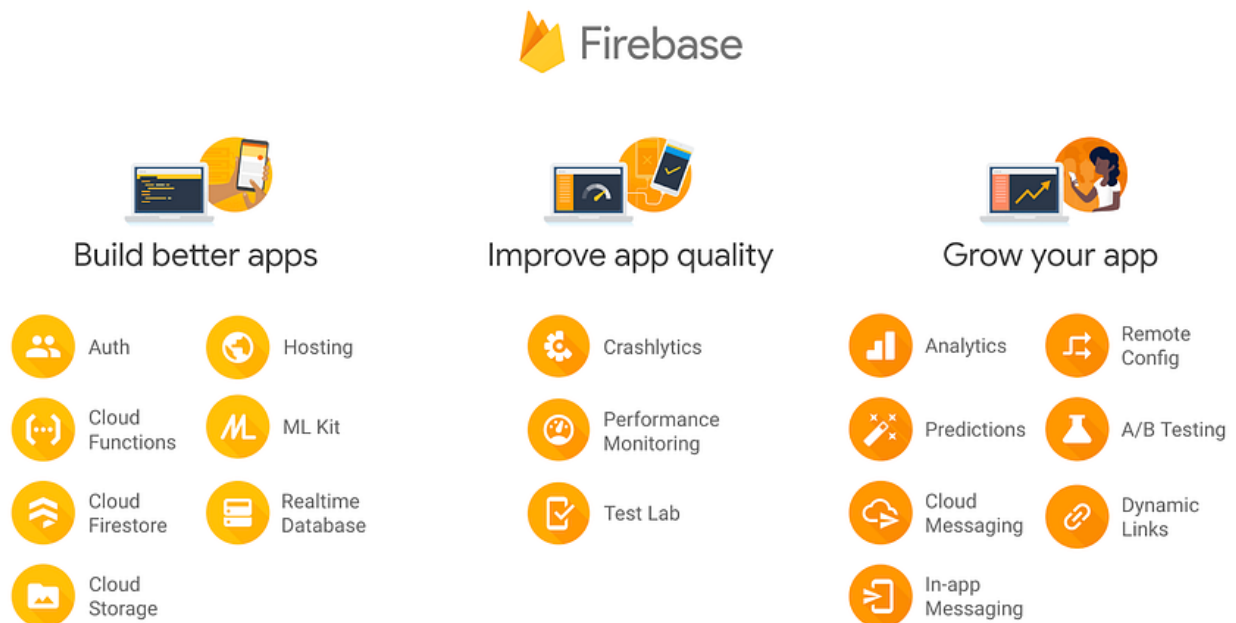


Рис. 2.3 Набір інструментів Firebase

Firebase виконує кілька ключових функцій у застосунку:

- *Авторизація користувачів.* Використовується модуль Firebase Authentication. Він надає готові механізми для створення та керування обліковими записами, таким чином дозволяючи користувачу легко відновити доступ і продовжити навчання на будь-якому пристрої.
- *Збереження прогресу та активності.* У хмарній базі Firestore зберігаються показники, що характеризують навчальні результати: кількість поінтів, виконані тести, поточний рівень прогресу
- *Відстеження streak (серії активних днів).* Firebase дозволяє фіксувати дату останнього входу та автоматично визначати, чи продовжується streak. Це важливо для побудови механізмів мотивації, характерних для освітніх застосунків.
- *Система бейджів.* У Firebase зберігаються унікальні ID відкритих користувачем бейджів, що дає змогу після оновлення або повторного входу відобразити актуальний стан досягнень.

Firebase інтегрується з *Flutter* через офіційні пакети, такі як *firebase_core*, *firebase_auth*, *cloud_firestore*. Це гарантує стабільну роботу на різних платформах без додаткових налаштувань. Такий підхід дозволяє зосередитися на логіці застосунку, та не витратити час на створення механізмів авторизації, чи синхронізації даних.

Дизайн InformatykaGO побудований на принципах мінімалізму, кольорів та інтуїтивності. Для комфортного перегляду контенту взято спокійну світло-синю палітру кольорів, зрозумілу типографіку, а також достатньо великий розмір тексту. Всі екрани побудовані в модульному підході: окремі компоненти реалізовані як багаторазові віджети, що спрощує подальше розширення застосунку та підтримує чистоту архітектури. Функціональна частина містить 4 основні екрани: Головна, Лекції, Тести і Профіль – такий розподіл є умовно-логічним та призначений для чіткої навігації.

На сторінці «Головна» відображується прогрес користувача, рекомендації та таблиця лідерів – це створює мотиваційний ефект.

«Лекції» представляють собою структурований список лекцій, з можливістю сортування за категоріями та пошуку.

Структура сторінки «Тести» містить список інтерактивних тестів, які також можна сортувати за категоріями (наприклад: програмування, IoT) та важкістю (легкий, середній, важкий).

Сторінка «Профіль» відображає досягнення, streak та бейджі, синхронізовані з Firebase. На рисунку 2.4 продемонстровано вигляд дизайну в програмі Figma.

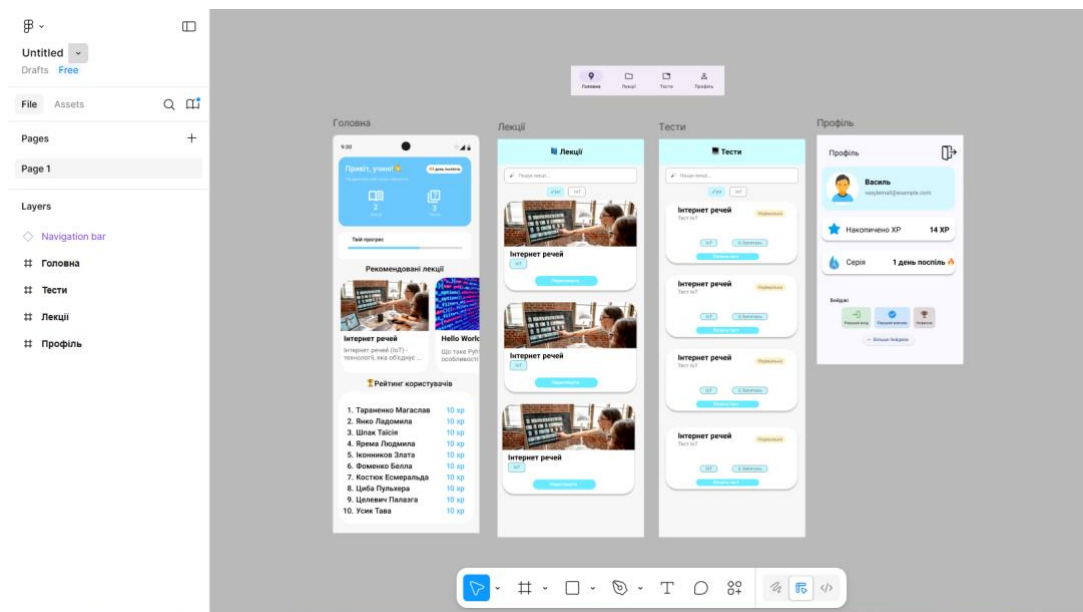


Рис. 2.4 Знімок екрану сторінки Figma з дизайном застосунку

Така побудова дизайну та функціональної частини забезпечує логічність, інтуїтивність та мотивацію, що є основними вимогами до сучасного мобільного освітнього програмного забезпечення. Брендбординг є ключем відновлення бренду застосунку та унікального стилю InformatykaGO – основні пункти включають палітру кольорів, шрифти, логотип, основні елементи дизайну та стиль ілюстрації (рис. 2.5).



Рис. 2.5 Брендборт застосунку InformatykaGO

Таким чином, функціональна структура та дизайн мобільного застосунку InformatykaGO забезпечують цілісність, зручність використання та відповідність сучасним вимогам до освітніх мобільних рішень. Використання фреймворку Flutter та хмарних технологій зберігання даних створює технічні й дидактичні передумови для ефективної організації самоосвітньої діяльності здобувачів освіти, що слугує основою для подальшого впровадження застосунку в освітній процес.

РОЗДІЛ III

РОЗРОБКА ТА ВПРОВАДЖЕННЯ ГЕЙМІФІКОВАНОГО ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ІНФОРМАТИКИ

3.1 Архітектура застосунку та інтеграція бази даних

Для розробки застосунку «InformatykaGO» обрано архітектурний патерн Client-Server для управління станом (рис. 3.1). Цей підхід дозволяє ефективно передавати дані про стан користувача (наприклад, його бали або прогрес) між різними віджетами (екранами).

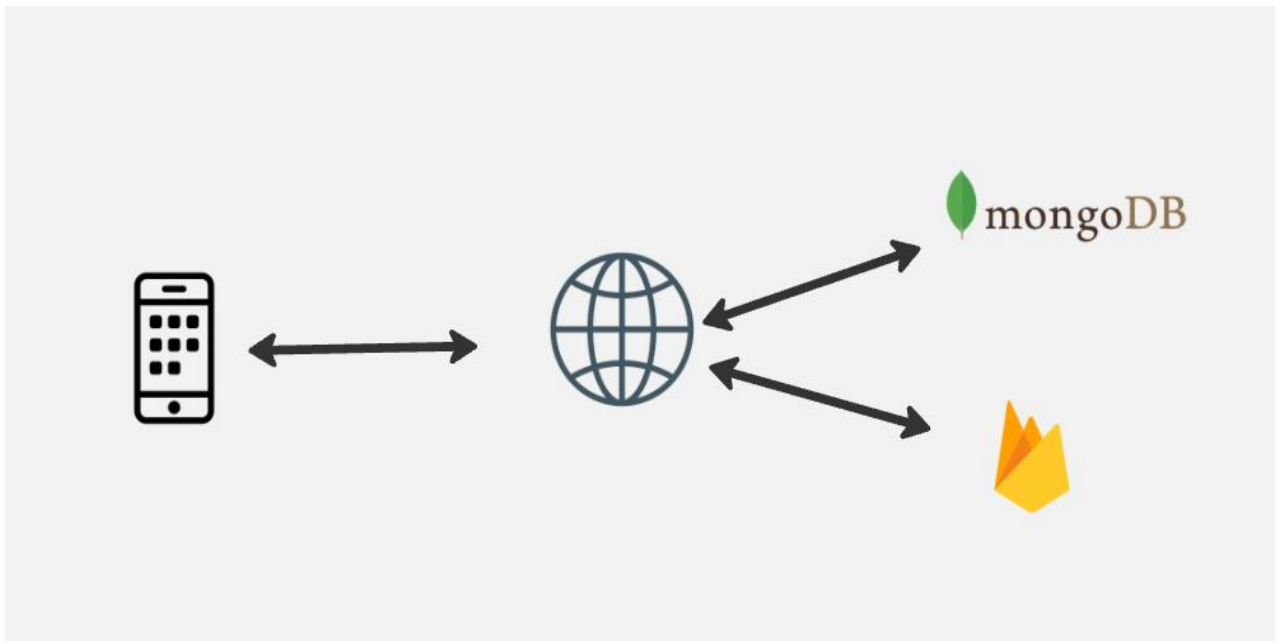


Рис. 3.1 Архітектурний патерн Client-Server

Платформа **Firebase**, у застосунку InformatykaGO, виконує функцію комплексного Backend-as-a-Service (BaaS). Вона повністю бере на себе завдання, пов'язані з ідентифікацією користувачів та зберіганням їхніх персоналізованих даних.

На відміну від традиційних монолітних бекендів, інтеграція з Firebase є декларативною та керується подіями.

Глобальна ініціалізація: з'єднання з Firebase встановлюється один раз при запуску застосунку. У коді це чітко видно у головній функції `void main()` через

виклик `await Firebase.initializeApp();`. Цей єдиний виклик налаштовує весь SDK Firebase для комунікації з проєктом у хмарі.

Потокове керування сесією: архітектура автентифікації побудована не на окремих запитах, а на прослуховуванні потоку (Stream). У віджеті `MyApp` використовується `StreamBuilder`, який слухає `FirebaseAuth.instance.authStateChanges()`. Цей потік миттєво повідомляє клієнта про будь-яку зміну стану (користувач увійшов, вийшов, чи сесія ще завантажується). Саме цей підхід визначає, чи бачить користувач `WelcomePage` (якщо `snapshot.hasData` повертає `false`) або `MainScreen` (якщо `true`), забезпечуючи реактивний та плавний користувацький досвід.

`Cloud Firestore` використовується як NoSQL база даних для збереження всієї інформації, що стосується конкретного користувача. Код (Додаток А) демонструє повний цикл операцій (CRUD):

- Створення (Create): Коли новий користувач вперше вказує свій нікнейм, `AuthService` викликає `_firestore.collection('users').doc(user.uid).set({...})` для створення нового документа, пов'язаного з його унікальним `uid`.
- Читання (Read): Численні компоненти зчитують дані. Наприклад, `HomePage` зчитує `passedTests` та дані для таблиці лідерів, а `ProfilePage` завантажує повний профіль для відображення.
- Оновлення (Update): Це ключова операція. Коли користувач проходить тест, функція `_saveScoreToFirebase` демонструє ефективні методи оновлення.

На рисунку 3.2 зображено алгоритм запуску застосунку та автентифікації користувача. Початком процесу є перевірка стану авторизації (чи користувач вже входив у систему раніше). Якщо ні, йому пропонується сторінка вітання (`WelcomePage`). Якщо так, система перевіряє, чи заповнено поле нікнейму, і, залежно від цього, спрямовує на сторінку створення нікнейму або на головний екран. Поле нікнейму перевіряється при кожному відкритті застосунку, у разі якщо користувач перервав реєстрацію облікового запису.

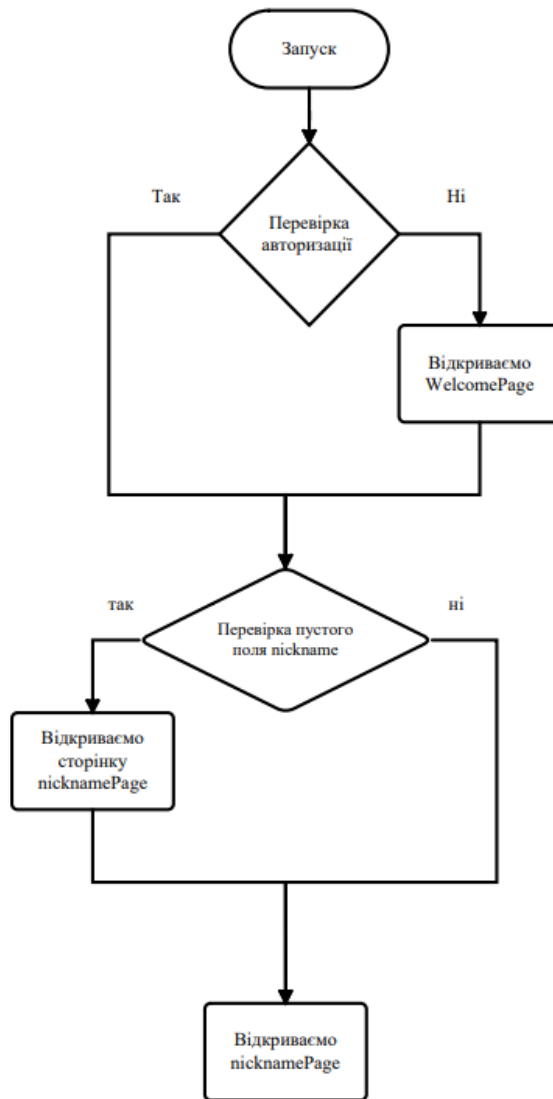


Рис. 3.2 Алгоритм запуску застосунку та автентифікації користувача

Таким чином, визначені архітектурні рішення та підходи до інтеграції бази даних забезпечують надійність, масштабованість і стабільну роботу мобільного застосунку InformatykaGO. Реалізована архітектура створює технічне підґрунтя для впровадження функціональних можливостей застосунку та його подальшого використання в освітньому процесі.

3.2 Основні компоненти застосунку

Точкою входу в застосунок є функція `main()`, яка виконує ініціалізацію Flutter (`WidgetsFlutterBinding.ensureInitialized()`) та Firebase (`Firebase.initializeApp()`) (рис. 3.3).

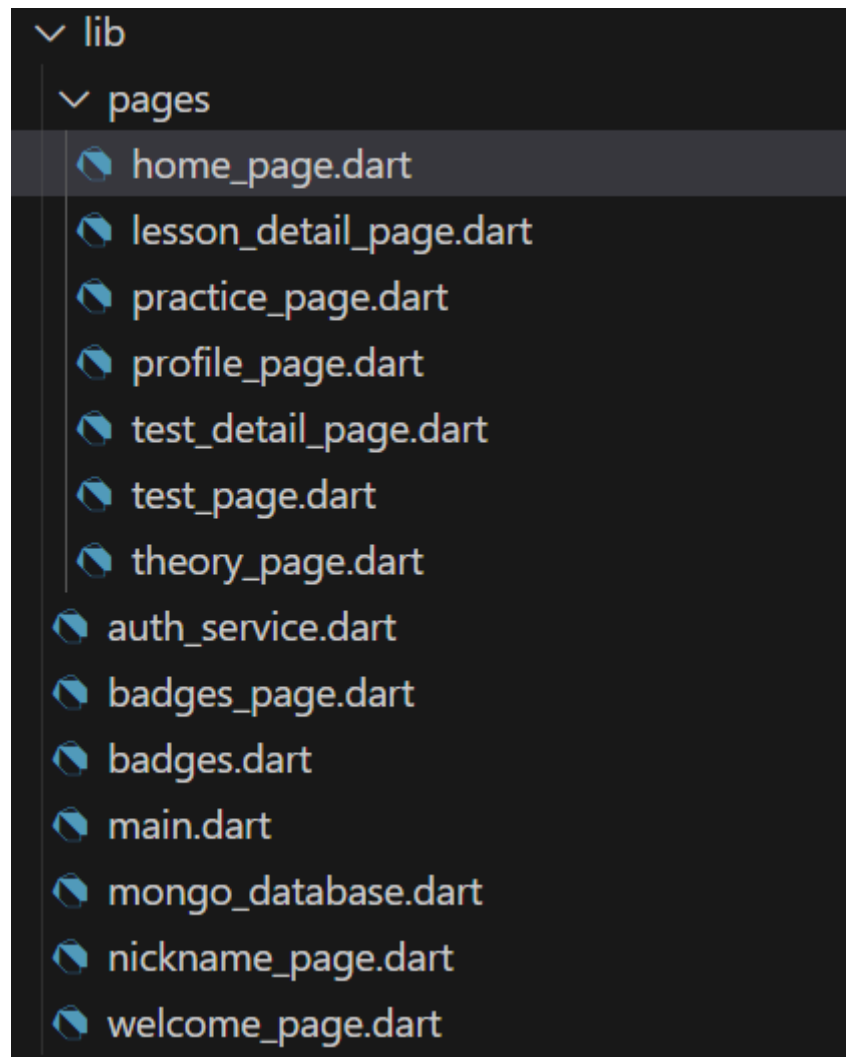


Рис. 3.3 Структура фалів мобільного застосунку

Корінь ієрархії віджетів, клас `MyApp`, використовує `StreamBuilder` для моніторингу стану автентифікації. Він слухає потік `FirebaseAuth.instance.authStateChanges()`, який автоматично сповіщає про зміни сесії. Залежно від стану потоку, `StreamBuilder` відображає один із чотирьох станів:

- Під час очікування: `CircularProgressIndicator`.
- У разі помилки: Екран з повідомленням про помилку (`snapshot.hasError`).
- Якщо є дані сесії: `MainScreen` для автентифікованих користувачів.
- Якщо даних немає: `WelcomePage` для неавтентифікованих користувачів.

Цей stream-based підхід в корені дерева віджетів є основною стратегією керування станом автентифікації застосунку, забезпечуючи, що весь інтерфейс реагує передбачувано та автоматично на зміни статусу сесії користувача без ручного втручання.

Основна навігація (MainScreen)

Віджет `MainScreen` є головним контейнером для автентифікованого користувача (рис. 3.4). Його структура складається зі `Scaffold` з `NavigationBar` внизу екрана. Для переміщення між основними розділами застосунку використовується віджет `IndexedStack`. Вибір `IndexedStack` замість перебудови сторінок при кожній зміні вкладки є свідомою оптимізацією продуктивності. Це гарантує збереження стану кожної з чотирьох основних сторінок (`HomePage`, `LecturesPage`, `TestsPage`, `ProfilePage`), включаючи позиції прокрутки, фільтри пошуку та завантажені дані, що забезпечує плавний та безперервний користувацький досвід.

При ініціалізації `MainScreen` викликається асинхронний метод `_checkNickname`. Ця функція перевіряє наявність та правильність поля `nickname` у документі користувача в `Firestore` (`data['nickname'].toString().trim().isEmpty`). Якщо нікнейм відсутній або є порожнім рядком, застосунок примусово перенаправляє користувача на `NicknamePage`. Цей крок є критичним елементом онбордингу, який гарантує, що кожен користувач має унікальний ідентифікатор у системі перед доступом до основного функціоналу.

Процес онбордингу нового користувача

Шлях нового користувача від першого запуску застосунку до отримання повного доступу складається з кількох логічних кроків:

1. Старт: Користувач відкриває застосунок і потрапляє на екран WelcomePage.
2. Вхід: Користувач ініціює вхід через Google, викликаючи метод AuthService.signInWithGoogle().
3. Перевірка профілю: Після успішної автентифікації система перевіряє, чи існує для цього користувача документ у колекції users Firestore та чи заповнене в ньому поле nickname.
4. Створення нікнейму: Якщо нікнейм відсутній, користувач автоматично перенаправляється на NicknamePage. На цьому етапі відбувається перевірка на унікальність нікнейму та фільтрація нецензурної лексики.
5. Завершення: Після успішного збереження нікнейму в Firestore користувач потрапляє на головний екран MainScreen і отримує повний доступ до функціональності застосунку.

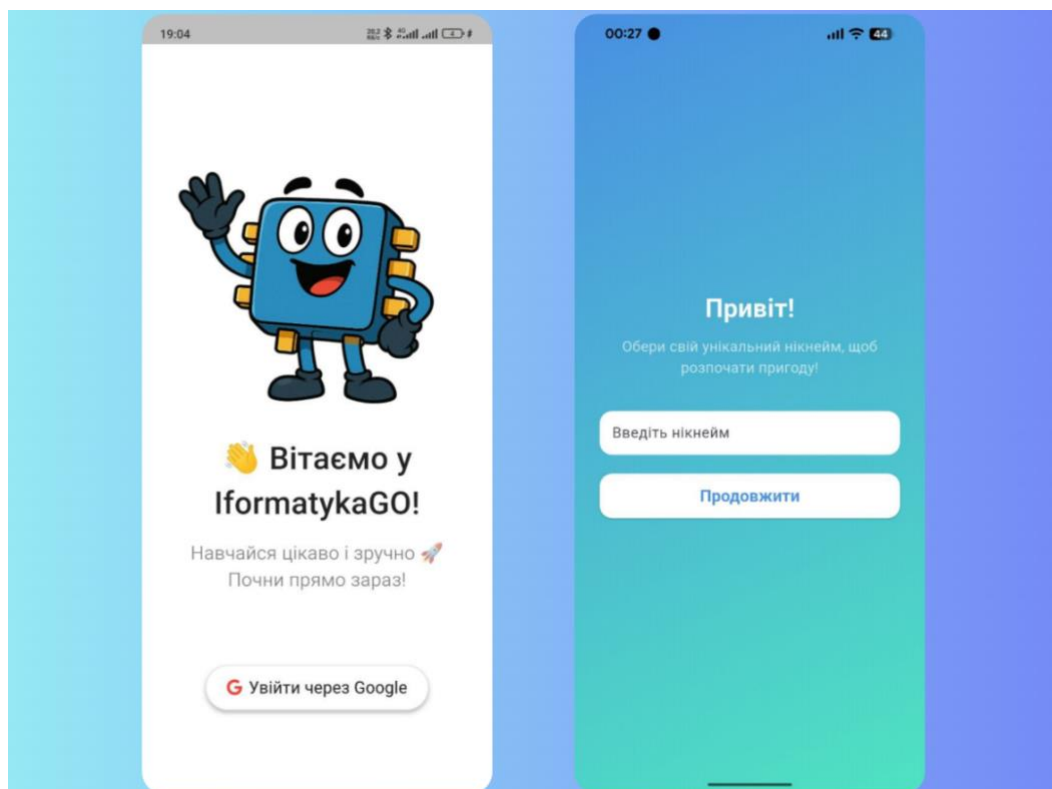


Рис. 3.4 Знімки екрана WelcomePage та NicknamePage

Головний екран та дашборд (HomePage)

HomePage функціонує як інформаційний хаб і демонструє архітектурний патерн агрегації даних з обох джерел (рис. 3.5):

- З Firebase Firestore він отримує персоналізовану статистику: streak, passedTests та дані для leaderboard.
- З MongoDB він завантажує список лекцій для відображення в блоці «Рекомендовані лекції».

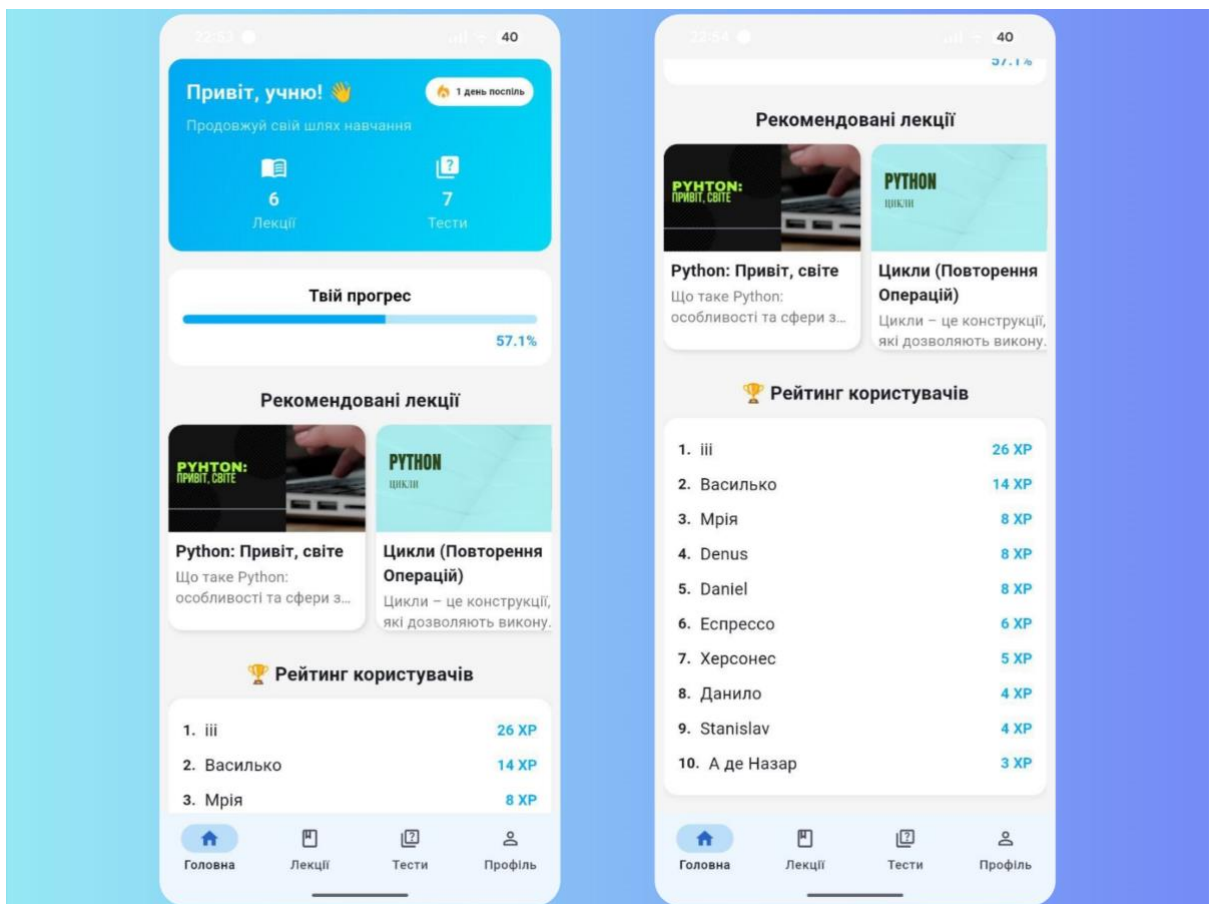


Рис. 3.5 Знімки екрана HomePage

Колекція *lessons* призначена для зберігання навчального контенту у вигляді окремих лекцій з комп'ютерних дисциплін. Кожен документ у цій колекції містить метадані про лекцію: назву, опис, посилання на відео, ілюстрацію, категорію та пов'язаний тест. На рисунку 3.6 можна побачити як виглядає панель керування документом в MongoDB Console. Ці дані завантажуються застосунком під час переходу користувача на сторінку «Лекції» (*LecturesPage*).

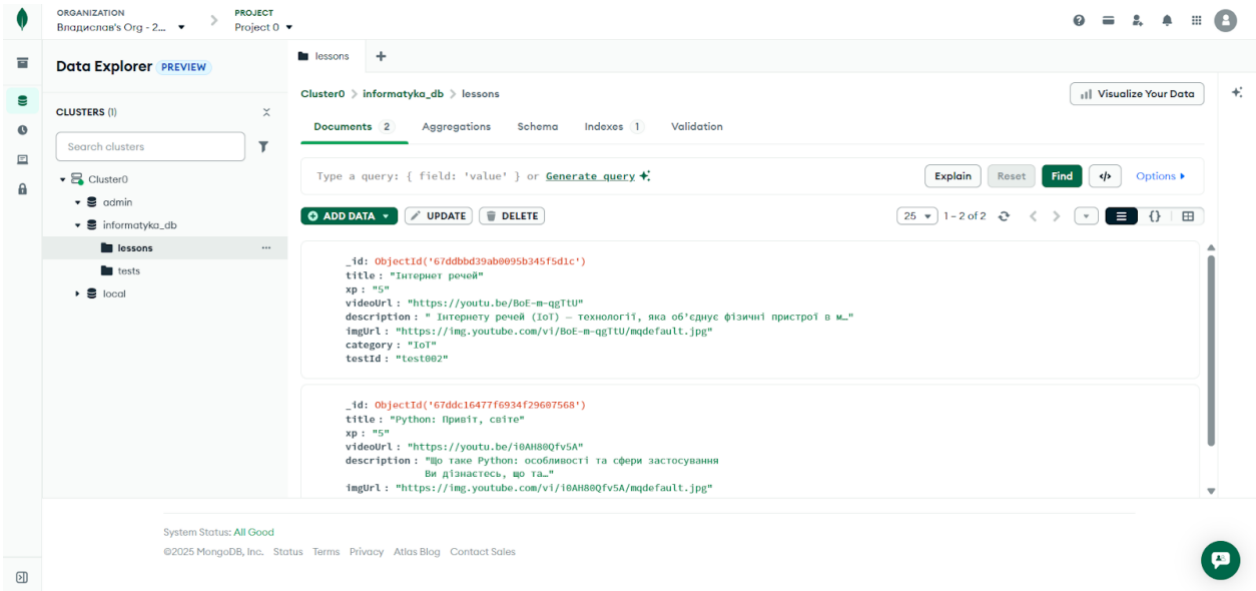


Рис. 3.6 Знімок екрана MongoDB console

Структура колекції наведена у вигляді таблиці (таб. 3.1), що містить назви полів, їх типи та призначення. Вона відображає модель даних, яка використовується в модулі *MongoDatabase* для вибірки контенту з бази MongoDB Atlas.

Таблиця 3.1

Структура документів колекції *lessons*

Поле	Тип	Призначення
_id	ObjectId	Унікальний ідентифікатор документа в MongoDB.
title	string	Назва лекції, яка відображається у списку.
xp	string / number	Кількість досвіду (XP), що нараховується за перегляд лекції або виконання пов'язаного тесту.
videoUrl	string	Посилання на відео-лекцію (наприклад, YouTube).
description	string	Короткий опис змісту лекції.
imgUrl	string	URL-зображення, яке використовується як обкладинка для лекції.

category	string	Тематична категорія (наприклад, “IoT», “Networking», “Cybersecurity»).
testId	string	Ідентифікатор тесту з колекції tests, який пов’язаний із даною лекцією.

Таке структурування забезпечує зручне завантаження тестів у застосунок *InformatykaGO* та дозволяє користувачеві проходити інтерактивні перевірки знань одразу після опрацювання лекційного матеріалу.

Таблиця 3.2

Структура документів колекції tests

Поле	Тип	Опис
_id	string	Унікальний ідентифікатор тесту (наприклад, «test002»). Використовується для зв’язку з лекціями (testId у колекції lessons).
title	string	Назва тесту, що відповідає назві лекції.
description	string	Короткий опис або мета тесту.
difficulty	string	Рівень складності тесту (наприклад, “Легко», “Нормально», “Складно»).
category	string	Тематична категорія, до якої належить тест (наприклад, “IoT», “Мережі», “Алгоритми»).
questions	array<map>	Масив об’єктів, кожен із яких описує окреме запитання тесту.
text	string	Текст запитання.
options	array<map>	Масив варіантів відповідей для запитання.
text	string	Текст варіанту відповіді.
isCorrect	boolean	Позначка правильності відповіді (true — правильна, false — неправильна).

У мобільному застосунку *InformatykaGO*, використовується хмарна база даних *Firebase Firestore*. Вона забезпечує персоналізацію, збереження прогресу користувачів та відстеження досягнень.

Для кожного нового користувача, автоматично створюється окремий документ у колекції *users*. Він містить основну інформацію про обліковий запис, статистику навчання, результати виконаних тестів, а також список отриманих бейджів. На рисунку 3.7 зображено як у базі даних, зберігається інформація про користувачів.

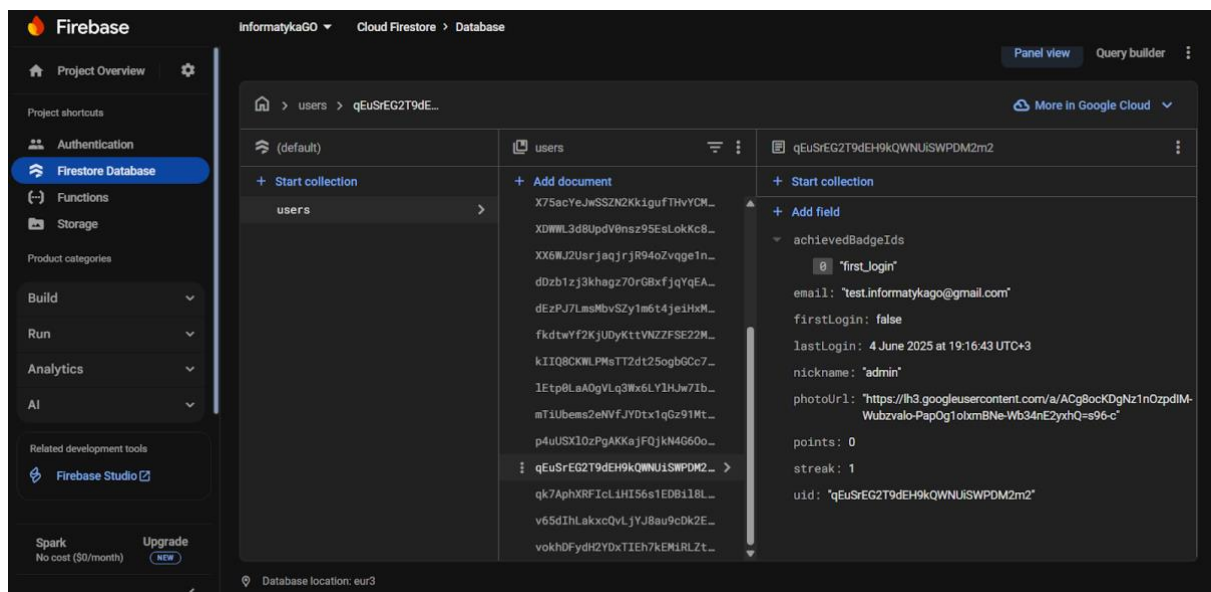


Рис. 3.7 Знімок екрану вікна Firebase Console

У таблиці 3.3 подано детальний опис структури документа користувача в колекції *users* бази даних *Firebase Firestore*. Кожен запис (документ) у колекції містить унікальний набір полів, що відображають персональні дані, прогрес навчання, досягнення та іншу службову інформацію

Структура документів колекції *users*

Поле	Тип	Опис
uid	string	Унікальний ідентифікатор користувача у Firebase Authentication.
email	string	Електронна адреса користувача (автоматично зчитується з облікового запису Google).
nickname	string	Ім'я користувача у системі (встановлюється при першому вході).
photoUrl	string	Посилання на аватар користувача з Google профілю.
firstLogin	boolean	Вказує, чи це перший вхід користувача (true — перший, false — наступні).
lastLogin	timestamp	Час останнього входу у застосунок.
updatedAt	timestamp	Час останнього оновлення даних користувача.
streak	number	Кількість днів поспіль, коли користувач заходив у застосунок.
score	number	Загальна кількість зароблених балів за тести (сума всіх проходжень).
achievedBadgeIds	array<string>	Масив ідентифікаторів бейджів, які користувач уже здобув. Наприклад: [«first_login», «first_test_passed», «night_owl», «streak_3», «xp_100»]
passedTests	array<map>	Масив об'єктів (map), де кожен елемент — це результат пройденого тесту:
id	string	Ідентифікатор тесту (наприклад, «test001»).
score	number	Кількість правильних відповідей.
total	number	Загальна кількість питань у тесті.

main.dart: Це головний файл застосунку, який відповідає за його первинну ініціалізацію та контроль стану. Він забезпечує ініціалізацію **Firestore** (`Firestore.initializeApp()`) і використовує віджет `StreamBuilder<User?>` для моніторингу змін стану автентифікації користувача.

Важливою деталлю реалізації є те, що `HomePage` завантажує всі лекції та тести з `MongoDB` через методи `_fetchLessons` та `_fetchTests` лише для того, щоб відобразити їх загальну кількість (`lectures`, `quizzes`) у віджеті `_buildWelcomeCard`.

Навчальний контент (Лекції та Уроки)

Потік роботи з навчальним контентом починається на `LecturesPage`, де користувач може переглядати повний список лекцій, отриманих з `MongoDB`, з можливістю фільтрації та пошуку (рис. 3.8). При виборі лекції користувач переходить на `LessonDetailPage`, який використовує пакет `youtube_player_flutter` для відтворення відео. Кожна лекція може бути пов'язана з тестом через поле `testId`. Цей `testId` утворює ключовий зв'язок між статичним контентом у `MongoDB` та динамічними даними про прогрес користувача у `Firestore`, дозволяючи застосунку відстежувати засвоєння конкретної теоретичної теми.

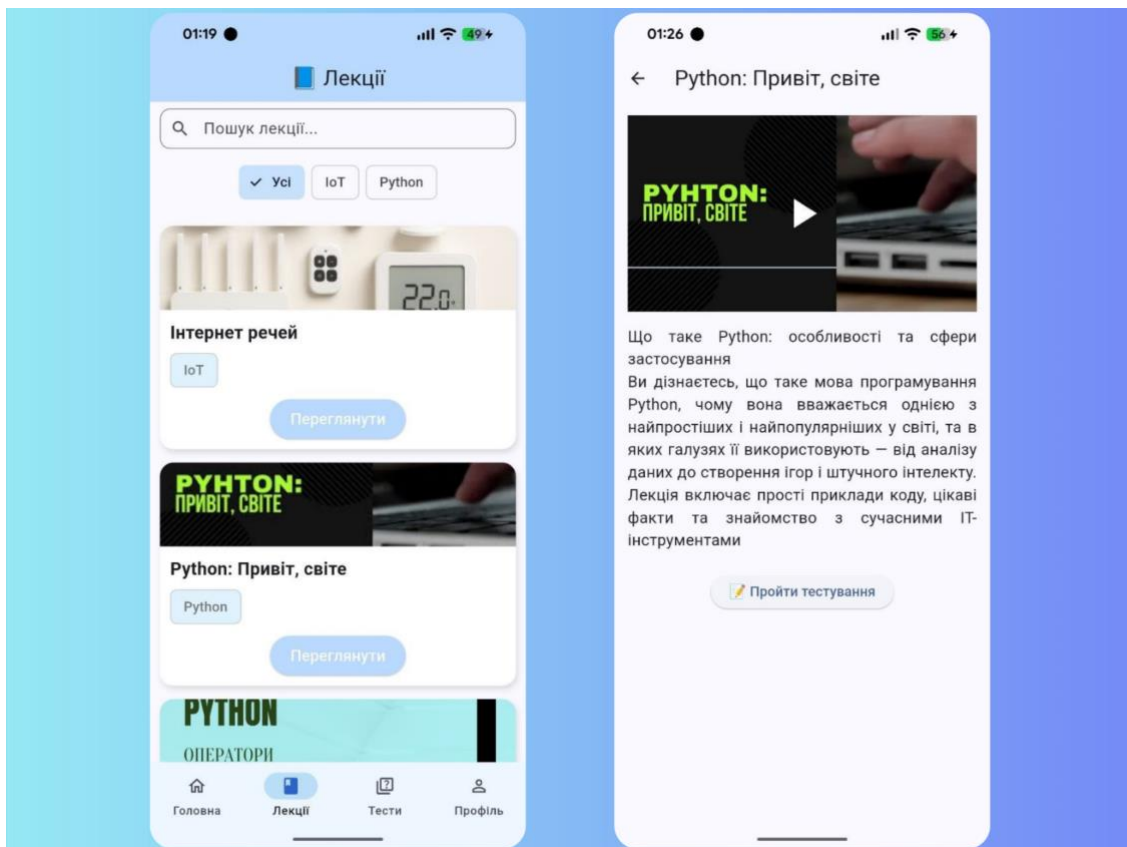


Рис. 3.8 Знімки екрана LecturesPage та LessonDetailPage

Система тестування та оцінювання (TestsPage та TestDetailPage)

TestsPage завантажує список доступних тестів з MongoDB та одночасно звертається до Firestore, щоб отримати масив `_passedTests` для поточного користувача (рис. 3.9). Це дозволяє візуально позначати завершені тести, вимикаючи кнопку входу (`onPressed: isPassed ? null : () {...}`). Основна логіка, що запобігає повторному проходженню, реалізована в LessonDetailPage та TestDetailPage.

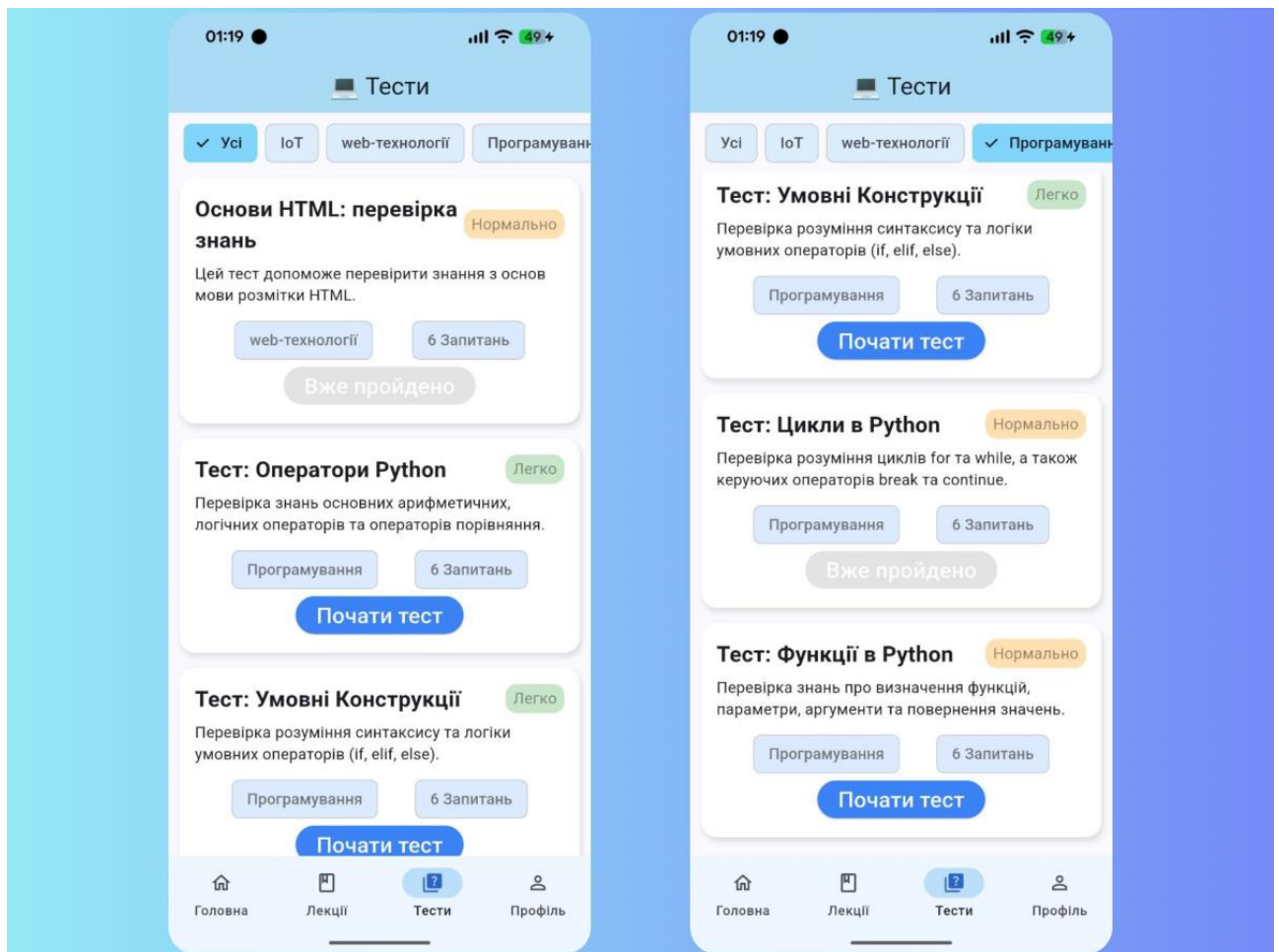


Рис. 3.9 Знімки екрана TestsPage

Логіка збереження результатів у `TestDetailPage`, реалізована у функції `_saveScoreToFirebase`, є основою для системи гейміфікації та працює за принципом «найкращого результату» (рис. 3.10):

1. Система перевіряє, чи існує запис для цього `testId` у масиві `passedTests` користувача.
2. Новий результат (`newScore`) порівнюється з попереднім (`previousScore`).
3. Оновлення даних у `Firestore` відбувається лише за умови, що `newScore` більший за `previousScore`.
4. Загальний рахунок користувача оновлюється за допомогою атомарної операції `FieldValue.increment(newScore - previousScore)`, а не перезаписуванням загальної суми.

Така реалізація забезпечує цілісність даних та заохочує користувачів покращувати свої результати.

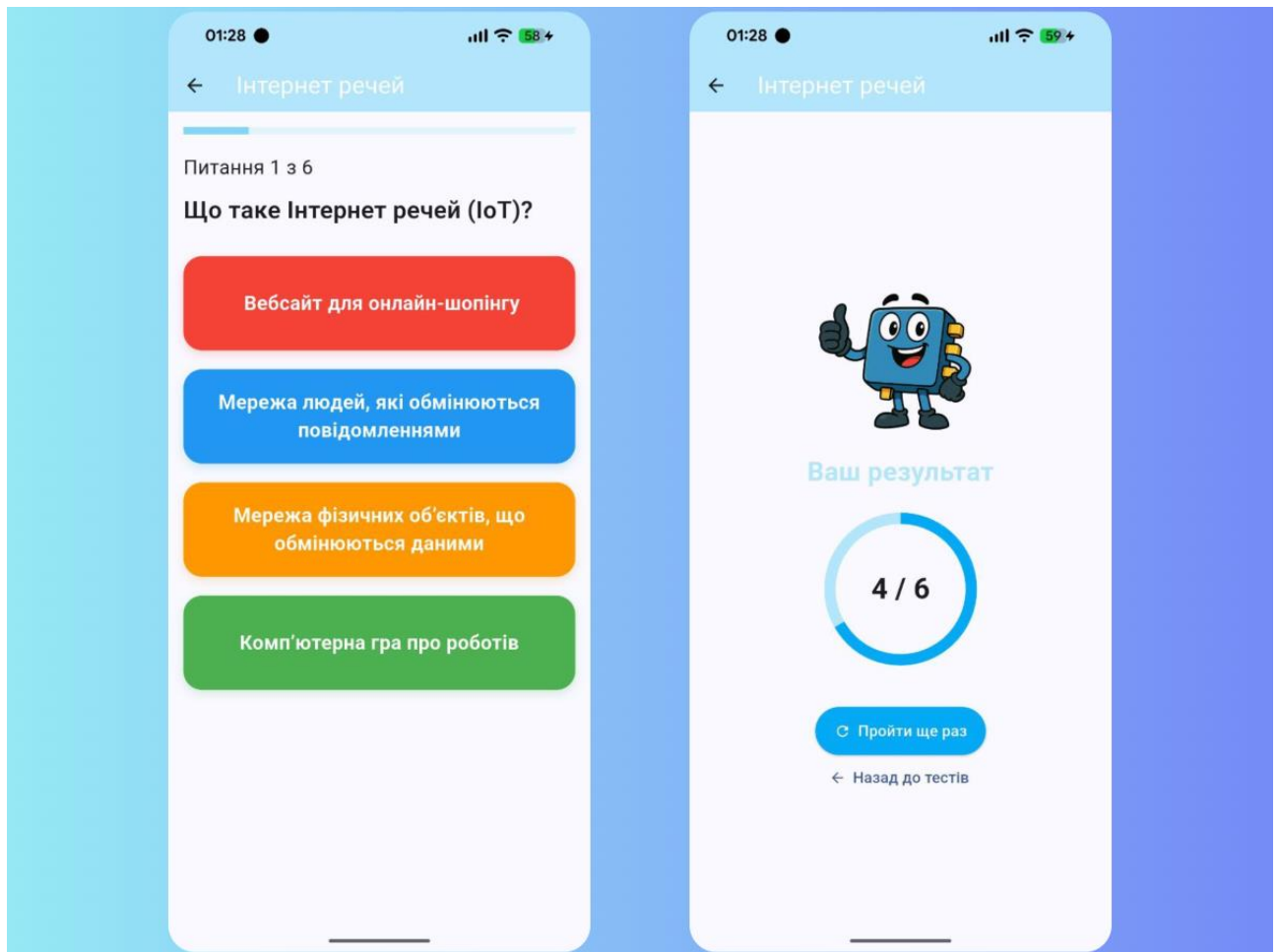


Рис. 3.10 Знімки екрана TestDetailPage

Система гейміфікації (Профіль та Бейджі)

Гейміфікація реалізована через систему балів, серій входів та бейджів. Сторінка ProfilePage відображає статистику користувача, а ключова логіка синхронізації досягнень інкапсульована в методі `_loadUserDataAndSyncBadges`. Система бейджів (`badges.dart`) визначає критерії для досягнень (наприклад, «набрати 50 балів (бейдж "Активний учень")»). ProfilePage при завантаженні порівнює поточні показники користувача з цими критеріями та синхронізує нові досягнення з Firestore, записуючи їх ID в `achievedBadgeIds` (рис. 3.11).

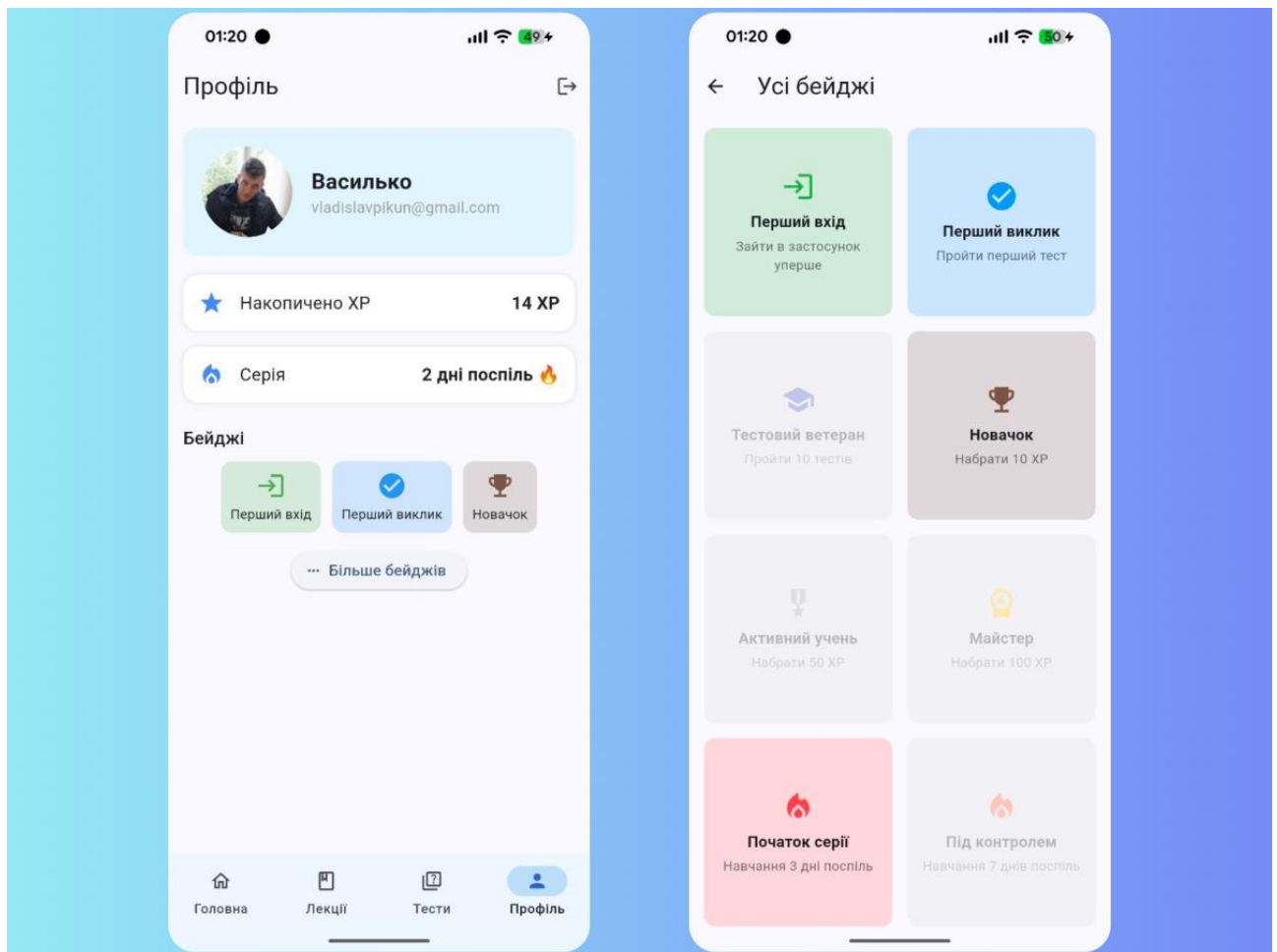


Рис. 3.11 Знімки екрана ProfilePage

Зовнішні залежності

Для реалізації функціоналу було використано зовнішні бібліотеки, які визначені у файлі `pubspec.yaml`, зокрема: `firebase_core` та `cloud_firestore` (інтеграція з backend), `mongo_dart` (робота з NoSQL БД), `Youtubeer_flutter` (відтворення відеоконтенту), `percent_indicator` (візуалізація прогресу) та `flutter_spinkit` (анімація завантаження).

Завдяки продуманому вибору технологій та чіткому розподілу логіки, архітектура «InformatykaGO» є добре структурованою, масштабованою та готовою до подальшого розвитку з впровадженням нових функцій.

3.3 Тестування та публікація застосунку в Google Play

Для розповсюдження мобільного застосунку було обрано платформу Google Play. З цією метою був створений обліковий запис розробника у сервісі Google Play Console (Рис. 3.12). Оскільки акаунт є новоствореним, Google Play висунув обов’язкову вимогу перед публікацією застосунку у відкритий доступ – проведення закритого тестування із залученням мінімум 12 тестувальників, які повинні встановити та використовувати застосунок (без видалення) протягом щонайменше 14 днів (Рис. 3.13).

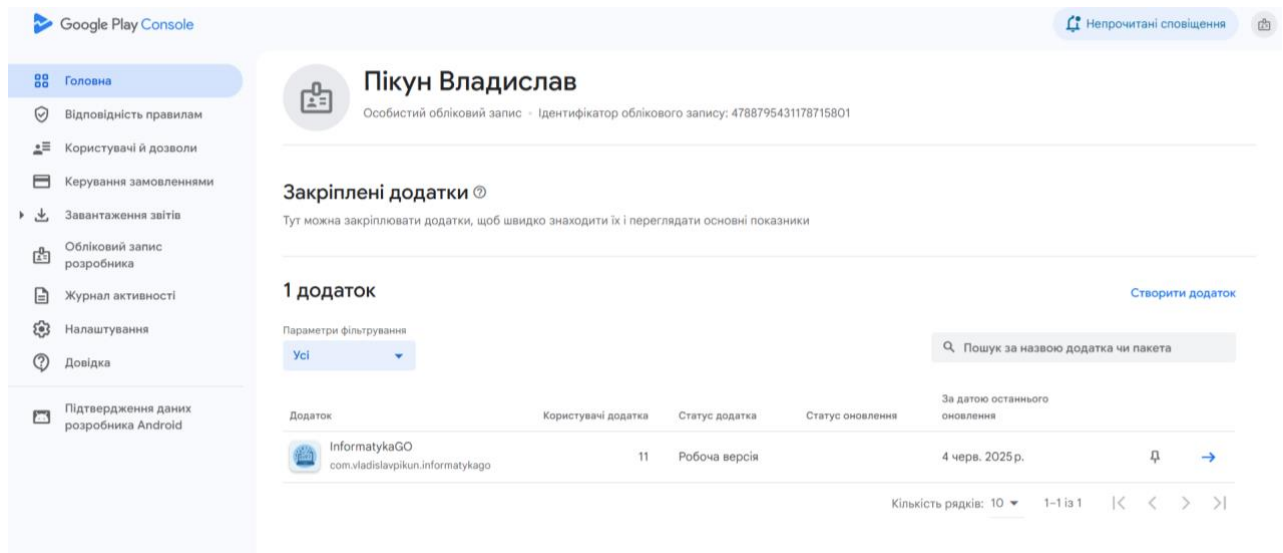


Рис. 3.12 Знімок екрану головної сторінки Google Play Console

У тестуванні взяли участь 12 користувачів, що дозволило зібрати перші відгуки, оцінити стабільність роботи застосунку та виявити помилки, пов’язані з інтерфейсом, продуктивністю та інтеграцією з Firebase і MongoDB.

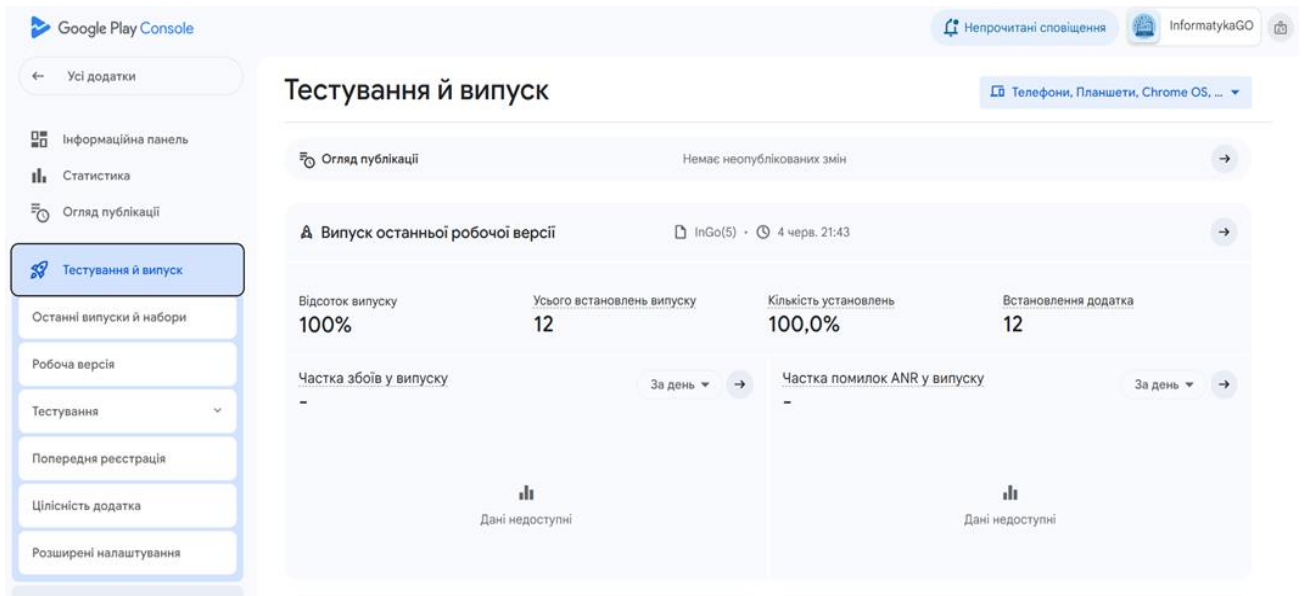


Рис. 3.13 Знімок екрану сторінки «Тестування та випуск»

Загальна кількість встановлень на етапі закритого тестування склала 12, що дало змогу оцінити роботу застосунку на різних пристроях і з різними версіями Android. Зібраний зворотний зв'язок став основою для подальшого виправлення помилок і підвищення якості користувацького досвіду.

Після успішного проходження етапу закритого тестування застосунок був підготовлений до публікації у Google Play. На цьому етапі виконувалась низка технічних та організаційних дій, необхідних для розміщення мобільного продукту в офіційному магазині.

По-перше, була сформована релізної збірка застосунку, оптимізована для поширення. До цього входили:

- увімкнення режиму мінімізації та обфускації коду;
- генерування підписаного APK/AAB-файлу;
- перевірка роботи застосунку саме у релізному режимі, оскільки його поведінка може відрізнятися від дебаг-версії.

По-друге, було підготовлено оформлення сторінки застосунку в Google Play Console (рис. 3.14). Зокрема, додано:

- опис застосунку, ключові особливості та призначення;
- знімки екрану інтерфейсу та іконку;

- категорію, вік обмеження, мову та політику конфіденційності;
- інформацію про обробку даних відповідно до вимог Google Play.

InformatykaGO

Пікун Владислав

10+
Завантаження

3+

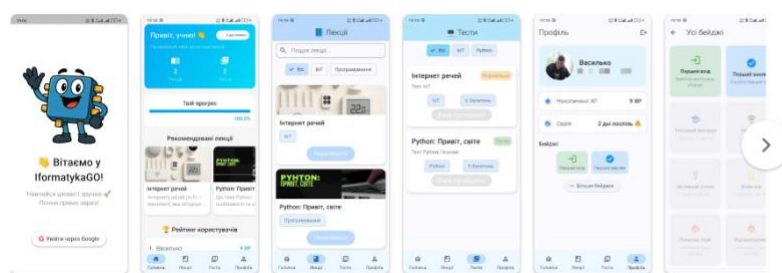
Від 3 років

Установити на інших пристроях

Надсилати

Цей додаток доступний для деяких ваших пристроїв

Цим контентом можна ділитися з учасниками сімейної групи. Докладніше про Сімейну бібліотеку.



Підтримка додатка

Додатки, варті уваги

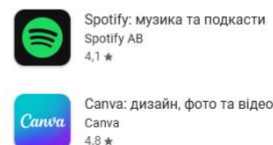


Рис. 3.14 Сторінка опублікованого застосунку в Google Play

Завантаживши релізний пакет, застосунок пройшов автоматичну перевірку Google Play Protect, а також модераційний аналіз щодо відповідності політикам безпеки, UI-доступності, роботи з персональними даними та технічним вимогам. Після активації релізу застосунки стає доступним для завантаження користувачами через Google Play.

3.4 Впровадження в освітній процес

З метою забезпечення ефективного впровадження мобільного гейміфікованого застосунку InformatykaGO в освітній процес закладів фахової передвищої освіти доцільним є формулювання методичних рекомендацій, орієнтованих на викладачів комп'ютерних дисциплін та здобувачів освіти. Запропоновані рекомендації ґрунтуються на принципах повсюдного навчання (U-learning), компетентнісного підходу та дидактичних можливостях мобільних

технологій, що дозволяє інтегрувати застосунок у традиційні та змішані форми навчання.

Ефективне використання InformatykaGO передбачає його розгляд не як самостійного заміни навчального курсу, а як допоміжного цифрового засобу, що доповнює навчальні програми з інформатики та інформаційних технологій. У зв'язку з цим рекомендовано здійснювати впровадження застосунку з урахуванням навчальних цілей дисципліни, рівня підготовки здобувачів освіти та особливостей організації освітнього процесу в конкретному закладі освіти.

Для викладачів доцільним є використання застосунку як інструменту підтримки самостійної роботи студентів, організації формувального оцінювання та підвищення навчальної мотивації. Рекомендується залучати InformatykaGO для попереднього опрацювання теоретичного матеріалу перед аудиторними заняттями, а також для закріплення вивчених тем шляхом виконання тестових і практикоорієнтованих завдань. Такий підхід сприяє оптимізації навчального часу та активізації пізнавальної діяльності здобувачів освіти. Крім того, результати тестування можуть використовуватися викладачем для моніторингу індивідуального прогресу студентів і коригування змісту навчального матеріалу.

Важливою методичною умовою є систематичність використання застосунку. Рекомендується інтегрувати його у структуру навчального курсу на регулярній основі, а не епізодично, що забезпечить формування у здобувачів освіти навичок самоорганізації та відповідального ставлення до навчання. Елементи гейміфікації, реалізовані у застосунку (рейтинги, бейджі, система заохочень), доцільно використовувати як додатковий мотиваційний чинник, не замінюючи ними традиційні форми оцінювання, а доповнюючи їх.

Для здобувачів освіти застосунок рекомендується як засіб організації індивідуальної траєкторії навчання та самоконтролю рівня засвоєння навчального матеріалу. Регулярне використання InformatykaGO сприяє формуванню навичок самостійної навчальної діяльності, розвитку відповідальності за власні результати та підвищенню рівня цифрової грамотності. Особливу увагу слід приділяти плануванню навчального часу та

систематичному виконанню завдань, що відповідає концепції безперервного навчання.

Ефективність впровадження мобільного застосунку значною мірою залежить від створення відповідних педагогічних умов, зокрема позитивної навчальної мотивації, методичної підтримки з боку викладача та узгодженості змісту застосунку з освітніми програмами. Важливим чинником є також готовність учасників освітнього процесу до використання мобільних технологій як повноцінного інструменту навчання.

Таким чином, запропоновані методичні рекомендації забезпечують науково обґрунтоване та педагогічно доцільне впровадження мобільного застосунку InformatykaGO в освітній процес. Їх реалізація сприятиме підвищенню якості самоосвітньої діяльності здобувачів освіти, модернізації навчального середовища та інтеграції сучасних цифрових технологій у підготовку фахівців у галузі інформаційних технологій.

Як приклад практичної реалізації запропонованих методичних рекомендацій можна розглянути використання мобільного застосунку InformatykaGO під час вивчення дисципліни *«Бази даних»*. Застосунок може бути інтегрований у процес опанування тем, пов'язаних з основами проєктування баз даних, моделями даних, мовою структурованих запитів SQL та принципами роботи з системами управління базами даних.

На етапі попереднього опрацювання навчального матеріалу здобувачам освіти доцільно запропонувати використання застосунку для ознайомлення з теоретичними відомостями щодо реляційної моделі даних, понять таблиці, поля, ключа та зв'язків між сутностями. Короткі теоретичні блоки та мультимедійні матеріали сприяють формуванню цілісного уявлення про структуру баз даних ще до проведення аудиторного заняття.

Під час практичних занять застосунок InformatykaGO може використовуватись як засіб формувального оцінювання шляхом виконання тестових завдань з тем нормалізації баз даних, створення таблиць та написання простих SQL-запитів. Отримані результати дозволяють викладачеві оперативно

оцінити рівень засвоєння матеріалу та визначити теми, які потребують додаткового пояснення.

У межах самостійної роботи студентів застосунок доцільно використовувати для закріплення знань з дисципліни та підготовки до контрольних заходів. Регулярне виконання завдань у застосунку сприяє формуванню навичок логічного мислення, системного аналізу даних та підвищує готовність здобувачів освіти до виконання практичних лабораторних робіт у середовищах управління базами даних.

Таким чином, використання InformatykaGO під час викладання дисципліни «Бази даних» демонструє можливість його ефективної інтеграції у навчальний процес та підтверджує доцільність застосування мобільних технологій для підтримки формування професійних компетентностей у сфері інформаційних технологій.

Апробація мобільного застосунку під час педагогічної практики в приватному вищому навчальному закладі «Рівненський економіко-гуманітарний та інженерний коледж», засвідчила його ефективність як для організації самостійної роботи студентів, так і для дидактичної підтримки занять. Студенти активно взаємодіяли з тестами, переглядали навчальні матеріали та демонстрували зростання інтересу до навчання. Використання елементів гейміфікації, зокрема системи досягнень, очок та streak, сприяло формуванню регулярності навчальної діяльності та підвищенню внутрішньої мотивації.

ВИСНОВКИ

У кваліфікаційній роботі здійснено комплексне дослідження проблеми організації самоосвіти здобувачів закладів фахової передвищої освіти з комп'ютерних дисциплін в умовах цифровізації освітнього середовища. Актуальність обраної теми зумовлена зростанням ролі самостійної навчальної діяльності, необхідністю формування цифрової компетентності майбутніх фахівців та потребою впровадження інноваційних освітніх технологій, здатних забезпечити безперервність і доступність навчання. У цьому контексті концепція повсюдного навчання (U-learning) розглянута як ефективна педагогічна основа для організації сучасного освітнього процесу.

У теоретичній частині роботи проаналізовано наукові підходи до поняття самоосвіти, її структури, функцій та місця в системі професійної підготовки здобувачів освіти. Визначено, що самоосвітня діяльність є важливим чинником формування здатності до безперервного навчання, саморозвитку та професійної мобільності. Окрему увагу приділено аналізу мобільних освітніх технологій і гейміфікації як засобів активізації пізнавальної діяльності, підвищення навчальної мотивації та формування позитивного ставлення до навчання.

У процесі дослідження здійснено порівняльний аналіз сучасних інструментів розробки мобільних застосунків та обґрунтовано доцільність використання фреймворку Flutter для створення освітнього програмного продукту. Визначено його переваги з точки зору продуктивності, кросплатформності, масштабованості та зручності підтримки, що є важливими критеріями для розробки мобільних навчальних застосунків, орієнтованих на тривале використання в освітньому процесі.

Практична частина роботи передбачала проектування, розробку та реалізацію мобільного гейміфікованого застосунку InformatykaGO, спрямованого на підтримку самоосвіти з інформатики. У межах розробки визначено функціональну структуру застосунку, реалізовано модулі теоретичного навчання, практичних завдань, тестування та персонального

профілю користувача. Для забезпечення збереження, обробки та синхронізації навчальних даних використано сучасні хмарні технології, що забезпечують стабільність роботи та можливість подальшого розширення функціоналу.

Особливу увагу приділено впровадженню елементів гейміфікації, зокрема системи балів, рейтингу, бейджів і streak, які спрямовані на формування регулярної навчальної активності та підтримку внутрішньої мотивації здобувачів освіти. Реалізовані механізми персоналізації дозволяють користувачам відстежувати власний прогрес, аналізувати результати навчальної діяльності та коригувати індивідуальну освітню траєкторію.

Практичну значущість дослідження підтверджено результатами апробації розробленого застосунку під час педагогічної практики у закладі фахової передвищої освіти. Використання InformatykaGO в освітньому процесі сприяло підвищенню навчальної активності студентів, зростанню інтересу до дисциплін комп'ютерного циклу та покращенню організації самостійної роботи. Спостереження засвідчили позитивний вплив мобільного застосунку на регулярність виконання навчальних завдань і загальний рівень залученості здобувачів освіти.

Узагальнюючи результати дослідження, можна стверджувати, що поставлену мету кваліфікаційної роботи досягнуто, а всі визначені завдання виконано в повному обсязі. Розроблений мобільний гейміфікований застосунок є ефективним засобом організації самоосвіти з комп'ютерних дисциплін в умовах повсюдного навчання та може бути рекомендований до використання в освітньому процесі закладів фахової передвищої та професійної освіти.

Перспективи подальших досліджень полягають у розширенні функціональних можливостей застосунку, впровадженні адаптивних навчальних алгоритмів, удосконаленні аналітичних інструментів моніторингу навчальних досягнень, а також у проведенні експериментальних досліджень з метою кількісної оцінки впливу мобільних гейміфікованих технологій на результати навчання здобувачів освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Арістова, Н. О., & Махович, І. А. (2023). Гейміфікація як засіб підвищення мотивації навчання студентів комп'ютерних спеціальностей. У *Світ дидактики: Дидактика в сучасному світі : Збірник матеріалів II Міжнародної науково-практичної інтернет-конференції, м. Київ, 22-23 листопада 2022 року* (с. 201–204). Видавництво Людмила.
2. Буряк, В. К. (2002). *Умови та засоби самоосвіти студентів*. [б. в.]. <https://doi.org/10.31812/123456789/4773>
3. Вербак, К. (2015). *Gamification*. <https://gameit.tech/tag/kevin-verbakh/>
4. Дакалюк, О. (2020). Інформаційно-комунікаційні технології як засіб активізації самостійної роботи студентів в умовах дистанційного навчання. *Modern Information Technologies and Innovation Methodologies of Education in Professional Training Methodology Theory Experience Problems*, 35–42. <https://doi.org/10.31652/2412-1142-2020-58-35-42>
5. *Додатки android у google play*. (2025). Android Apps on Google Play. <https://play.google.com/store/apps?hl=uk>
6. Жорова, І. Я., Кузьмич, Т. О., & Назаренко, Л. М. (2012). *Самоосвіта педагога як умова підвищення його професійної компетентності : Методичні рекомендації*. РІПО.
7. Кривонос, О. М., Горобець, С. М., Кривонос, М. П., & Нехаєнко, К. П. (2023). Використання стратегій гейміфікації для мотивації школярів до вивчення інформатики. *Вісник Житомирського державного університету імені Івана Франка. Педагогічні науки*, (115), 65–74.
8. Макаревич, О. О. (2015). Гейміфікація як невід'ємний чинник підвищення ефективності елементів дистанційного навчання. *Молодий вчений*, 275–278.
9. Мельник, Н. (2013, грудень). *Роль самоосвіти у підвищенні професійної майстерності педагога*. Сайт методиста Камінь-Каширського ВПУ. https://metodvpu.blogspot.com/2013/12/blog-post_11.html

10. Підгорна, А. (2024, 25 квітня). *Що таке самоосвіта?* arbook.info. <https://arbook.info/>
11. Пікун, В., & Полюхович, Н. (2024). Аналіз платформ для створення ігрового освітнього застосунку з інформатики. У *Інформаційні технології в професійній діяльності: Матеріали XVII Всеукраїнської науково-практичної конференції* (с. 221–224). РВВ РДГУ.
12. Пікун, В., & Полюхович, Н. (2025). Функціональні можливості гейміфікованого мобільного застосунку InformatykaGO для підтримки індивідуальної освітньої траєкторії. У *Підготовка педагогів до професійної діяльності в умовах змішаного навчання: Матеріали IV Всеукраїнської науково-практичної конференції* (с. 200–203). РВВ РДГУ.
13. Про забезпечення функціонування української мови як державної, Закон України № 2704-VIII (2025) (Україна). <https://zakon.rada.gov.ua/laws/show/2704-19#Text>
14. Про освіту, Закон України № 2145-VIII (2025) (Україна). <https://zakon.rada.gov.ua/laws/show/2145-19#Text>
15. Про фахову передвищу освіту, Закон України № 2745-VIII (2025) (Україна). <https://zakon.rada.gov.ua/laws/show/2745-19#Text>
16. Стефанська, І. (2023). *Рік в онлайн: Як змінилася шкільна освіта в Україні*. <https://mind.ua/openmind/20224266-rik-v-onlajni-yak-zminilasya-shkilna-osvita-v-ukrayini>
17. Токарева, А. (2019). *Навчальні цифрові ігри: Від теорії до практики (керівні принципи для викладачів)*. Lambert Academic Publishing.
18. Чернова, Т. Ю., Ольшевський, А. О., & Іванов, В. О. (2025). Гейміфікація як інструмент мотивації до самостійного професійного розвитку майбутніх педагогів. *Педагогічна Академія: Наукові записки*, (18). <https://doi.org/10.5281/zenodo.15501944>
19. Чижикова, І. (2023). Розвиток навчальної автономії студентів в умовах гейміфікації навчального процесу у зво. *Освіта. Інноватика. Практика*, 11(2), 54–59. <https://doi.org/10.31110/2616-650X-vol11i2-008>

20. Шлєіна, Л. І. (2023). *Самостійна робота – одна із складових навчального процесу*. ВНЗ. <https://elar.tsatu.edu.ua/server/api/core/bitstreams/2454d6a1-fe25-4364-9635-34f772597227/content>

21. Ярич, І. (2025). Гейміфікація як інноваційний тренд практичної підготовки майбутніх фахівців. *Інноваційна професійна освіта*, 1(22), 801–807. <https://doi.org/10.32835/2786-619X.2025.1.22.801-807>

22. Allison william. (2025, 28 липня). *How to build a chat app using flutter and firebase*. Друкарня. https://drukarnia.com.ua/articles/how-to-build-a-chat-app-using-flutter-and-firebase-fh_N2

23. *Android mobile app developer tools - android developers*. (б. д.). Android Developers. <https://developer.android.com/>

24. *Apache cassandra | apache cassandra documentation*. (б. д.). Apache Cassandra. <https://cassandra.apache.org/>

25. *Browse fonts - google fonts*. (б. д.). Google Fonts. <https://fonts.google.com/>

26. Bulgakova, O. S., & Zosimov, V. V. (2023). Gamified active learning and 3d technologies for enhancing students' academic achievements. *Itlt*, 96(4), 46–57. <https://doi.org/10.33407/itlt.v96i4.5380>

27. Burton, M. (2015). *Android app development for dummies*. Wiley & Sons, Incorporated, John.

28. *Create an Android App - AppYourself*. (б. д.). AppYourself. <https://appyourself.net/en/create-android-app/>

29. *Create your first android app | android developers*. (б. д.). Android Developers. <https://developer.android.com/codelabs/basic-android-kotlin-compose-first-app#0>

30. *Dart documentation*. (б. д.). Dart programming language. <https://dart.dev/docs>

31. Dissanayake, A., Lakpriya, V., Sanjeevan, M., Fahmi, A., Siriwardana, S., & Samaraweera, M. (2024). Application to enhance the programming

knowledge of children. *Y Proceedings of the international conference on advanced computing (ICAC)*. IEEE. <https://doi.org/10.1109/ICAC64487.2024.10851171>

32. *Figma: The collaborative interface design tool*. (б. д.). Figma. <https://www.figma.com/>

33. *Firebase | google's mobile and web app development platform*. (б. д.). Firebase. <https://firebase.google.com/>

34. *Flutter on mobile*. (2025). Flutter - Build apps for any screen. <https://flutter.dev/multi-platform/mobile>

35. *Gamification and game-based learning*. (б. д.). University of Waterloo. <https://uwaterloo.ca/centre-for-teaching-excellence/catalogs/tip-sheets/gamification-and-game-based-learning>

36. *Google play vs ios app store*. (2025, 1 травня). Mobile and CTV App Intelligence | 42matters. <https://42matters.com/stats#apps-categories-distribution>

37. *Google trends*. (2025, 17 серпня). Google Trends. <https://trends.google.com>

38. IBM. (б. д.). *How do I make an android app?* | IBM. <https://www.ibm.com/think/topics/android-app>

39. Ishaq, K., & Alvi, A. (2023). Personalization, cognition, and gamification-based programming language learning: A state-of-the-art systematic literature review. <https://arxiv.org/pdf/2309.12362>

40. Lampropoulos, G., Keramopoulos, E., Diamantaras, K., & Evangelidis, G. (2023). Integrating augmented reality, gamification, and serious games in computer science education. *Education Sciences*, 13(6), 618. <https://doi.org/10.3390/educsci13060618>

41. Makhovych, I. (2025). The role of competition as a gamification element in enhancing learning motivation among computer science students. *У Науково-технічна конференція молодих вчених «Актуальні проблеми інформаційних технологій» : Матеріали доповідей* (с. 49–51). Київ.

42. *Material design*. (б. д.). Material Design. <https://m3.material.io/>

43. *Mobile operating system market share ukraine*. (2025, червень). StatCounter Global Stats. <https://gs.statcounter.com/os-market-share/mobile/ukraine/#monthly-202406-202506-bar>
44. *MongoDB: The world's leading modern database*. (б. д.). MongoDB. <https://www.mongodb.com/>
45. *MySQL*. (б. д.). MySQL. <https://www.mysql.com/>
46. *React Native · Learn once, write anywhere*. (б. д.). React Native · Learn once, write anywhere. <https://reactnative.dev/>
47. Sirikong, N., & Visutsak, P. (2024). Gamifying python: Enhancing learning outcomes and motivation through a game-based approach. *Y Proceedings of the InCIT*. IEEE. <https://doi.org/10.1109/InCIT63192.2024.10810560>
48. *Software management (practitioners)*. (2006). Wiley-IEEE Computer Society Pr.
49. *Today for iPhone - App Store*. (б. д.). App Store. <https://apps.apple.com/ua/iphone/>
50. Zhang, J.-P. (2008). Hybrid learning and ubiquitous learning. *Y Hybrid learning and education* (с. 250–258). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-85170-7_22
51. Zichermann, G., & Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps*.

ДОДАТКИ

Додаток А

Код файлу main.dart

```
import 'package:flutter/material.dart';
import 'package:mongo_dart/mongo_dart.dart' as mongo;
import 'pages/home_page.dart';
import 'pages/theory_page.dart';
import 'pages/test_page.dart';
import 'pages/profile_page.dart';
import 'package:firebase_core/firebase_core.dart';
import 'welcome_page.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'nickname_page.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(const MyApp());
}

Future<void> checkMongoConnection() async {
  try {
    var db = await mongo.Db.create(
      'mongodb+srv://dbAdmin:*****@cluster0.qo2kv.mongodb.net/?retryWrites
      =true&w=majority&appName=Cluster0');
    await db.open();
    // print(' Підключено до MongoDB!');
    await db.close();
  }
```

```

    } catch (e) {
      // print(' Помилка підключення: $e');
    }
  }
}

```

```

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'InformatykaGO',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.blueAccent),
        useMaterial3: true,
      ),
      home: StreamBuilder<User?>(
        stream: FirebaseAuth.instance.authStateChanges(),
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const Scaffold(
              body: Center(child: CircularProgressIndicator()),
            );
          } else if (snapshot.hasError) {
            return const Scaffold(
              body: Center(child: Text('Помилка при завантаженні користувача')),
            );
          } else if (snapshot.hasData) {
            return const MainScreen();
          } else {

```

```

        return const WelcomePage();
    }
},
),
);
}
}

```

```

class MainScreen extends StatefulWidget {
  const MainScreen({super.key});

  @override
  State<MainScreen> createState() => _MainScreenState();
}

```

```

class _MainScreenState extends State<MainScreen> {
  int _selectedIndex = 0;
  bool _checkingNickname = true;

  final List<Widget> _pages = const [
    HomePage(),
    LecturesPage(),
    TestsPage(),
    ProfilePage(),
  ];

```

```

  @override
  void initState() {
    super.initState();
    _checkNickname();

```

```

}

Future<void> _checkNickname() async {
  final user = FirebaseAuth.instance.currentUser;
  if (user != null) {
    final docSnapshot = await FirebaseFirestore.instance
      .collection('users')
      .doc(user.uid)
      .get();

    final data = docSnapshot.data();

    final hasNickname = data != null &&
      data.containsKey('nickname') &&
      data['nickname'].toString().trim().isNotEmpty;

    if (!docSnapshot.exists || !hasNickname) {
      if (mounted) {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(builder: (_) => NicknamePage(user: user)),
        );
      }
    } else {
      setState(() {
        _checkingNickname = false;
      });
    }
  }
}

```

```
void _onItemTapped(int index) {
  setState(() {
    _selectedIndex = index;
  });
}
```

```
@override
```

```
Widget build(BuildContext context) {
  if (_checkingNickname) {
    return const Scaffold(
      body: Center(child: CircularProgressIndicator()),
    );
  }
}
```

```
return Scaffold(
  body: IndexedStack(
    index: _selectedIndex,
    children: _pages,
  ),
  bottomNavigationBar: NavigationBar(
    height: 80,
    elevation: 1,
    indicatorColor: Colors.blue.shade100,
    backgroundColor: Color(0xFFEDF6FF),
    surfaceTintColor: Colors.white,
    selectedIndex: _selectedIndex,
    onDestinationSelected: _onItemTapped,
    labelBehavior: NavigationDestinationLabelBehavior.alwaysShow,
    destinations: const [
```

```
NavigationDestination(  
  icon: Icon(Icons.home_outlined),  
  selectedIcon: Icon(Icons.home, color: Color(0xFF2C65C8)),  
  label: 'Головна',  
)  
NavigationDestination(  
  icon: Icon(Icons.book_outlined),  
  selectedIcon: Icon(Icons.book, color: Color(0xFF2C65C8)),  
  label: 'Лекції',  
)  
NavigationDestination(  
  icon: Icon(Icons.quiz_outlined),  
  selectedIcon: Icon(Icons.quiz, color: Color(0xFF2C65C8)),  
  label: 'Тести',  
)  
NavigationDestination(  
  icon: Icon(Icons.person_outline),  
  selectedIcon: Icon(Icons.person, color: Color(0xFF2C65C8)),  
  label: 'Профіль',  
)  
],  
)  
);  
}
```

```
}
```

Додаток Б

ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ГЕЙМІФІКОВАНОГО МОБІЛЬНОГО ЗАСТОСУНКУ INFORMATYKAGO ДЛЯ ПІДТРИМКИ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ

<i>Матеріали ІІ Всеукраїнської науково-практичної конференції «Підвищення педагогіки до професійної діяльності в умовах змішаного навчання»</i> Abstract. The functional capabilities of the InformatykaGO mobile application, aimed at supporting learners' individual educational trajectories in a blended learning environment, are examined. Key words: blended learning, mobile application, gamification, Flutter, MongoDB, Android, individual educational trajectory У сучасному освітньому середовищі дедалі більше уваги приділяється інноваційним цифровим інструментам, які сприяють індивідуалізації навчання, підвищенню мотивації здобувачів освіти та організації ефективного освітнього процесу в умовах змішаного навчання. Одним із таких інструментів є мобільні освітні застосунки, що забезпечують гнучкий доступ до навчального матеріалу, дозволяють поєднувати формальну і неформальну освіту та створюють можливості для активного залучення студентів до навчання в зручному для них форматі. З поміж великої кількості мобільних рішень для освіти важливу роль відіграють спеціалізовані застосунки для вивчення інформатики та комп'ютерних дисциплін, які не лише надають доступ до контенту, а й інтегрують гейміфіковані елементи, здатні змінити ставлення здобувачів освіти до навчального предмета (Холдунова, 2023). Застосунок <i>InformatykaGO</i> розроблений як інтерактивний навчальний інструмент, що поєднує теоретичний, практичний і мотиваційний компоненти для вивчення інформатики та комп'ютерних дисциплін. Програма орієнтована на здобувачів фахової передвищої та професійної освіти, які навчаються в умовах змішаного або дистанційного формату. Основна мета застосунку – зробити вивчення інформатики та комп'ютерних дисциплін доступним, зрозумілим і водночас захоплюючим. Для реалізації проєкту використано фреймворк Flutter (Flutter Docs, 6. д.), що дозволяє створювати кросплатформні мобільні додатки, та MongoDB як базу даних для зберігання контенту – теоретичних матеріалів і тестів (MongoDB Docs, 6. д.). Структура <i>InformatykaGO</i> включає кілька основних розділів: теорію, тести, 201	<i>Матеріали ІІ Всеукраїнської науково-практичної конференції «Підвищення педагогіки до професійної діяльності в умовах змішаного навчання»</i> до успішності. Такий підхід дозволяє говорити про <i>InformatykaGO</i> як про перспективний інструмент нової цифрової педагогіки. Список використаних джерел - Холдунова, В. (2023). Гейміфікація як інновація в освіті. <i>Наукові інновації та передові технології</i>, 2(16)). https://doi.org/10.52058/2786-5274-2023-2(16)-407-417 - Документація flutter. (6. д.). Flutter Docs. https://docs.flutter.dev/ - Документація MongoDB. (6. д.). MongoDB. https://www.mongodb.com/docs/ - Програмний сайт застосунку. (6. д.). InformatykaGO Google Sites. https://sites.google.com/view/informatycago References - Khodunova, V. (2023). <i>Heimifikaia yak innovatsiia v osviti</i> [Gamification as an innovation in education]. <i>Naukovі innovatsii ta peredovi tekhnologii</i>, 2(16), 407-417. https://doi.org/10.52058/2786-5274-2023-2(16)-407-417 (in Ukrainian). - Flutter documentation. (n.d.). Flutter. https://docs.flutter.dev/ - MongoDB documentation. (n.d.). MongoDB. https://www.mongodb.com/docs/ - Application website. (n.d.). InformatykaGO Google Sites. https://sites.google.com/view/informatycago (in Ukrainian). РОЛЬ УЧИТЕЛЯ У ВИПРОВАДЖЕННІ ЗМІШАНОГО НАВЧАННЯ: МЕТОДИЧНИЙ АСПЕКТ Ольга ПОПКО, <i>здобувачка другого (магістерського) рівня вищої освіти</i> <i>Рівненський державний гуманітарний університет</i> Наталія ОСТАПЧУК, <i>кандидат педагогічних наук, доцент, професор кафедри</i> <i>цифрових технологій та методик навчання інформатики</i> <i>Рівненський державний гуманітарний університет</i> 203
<i>Матеріали ІІ Всеукраїнської науково-практичної конференції «Підвищення педагогіки до професійної діяльності в умовах змішаного навчання»</i> лише оптимізувати використання навчального часу, але й значно підвищити рівень залученості учнів до навчального процесу. Список використаних джерел - Білоусова, Н.В., & Горішнюк, Т.В. (2019). Застосування технології «Перевернутого навчання» в роботі загальноосвітнього навчального закладу. <i>Молодий вчений</i>, 102-106. - Білоусова, Н.В., & Hordienko, T.V. (2019). Application of the «Flipped Learning» Technology in the Work of General Educational Institutions. <i>Young Teacher</i>, 102-106. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ ГЕЙМІФІКОВАНОГО МОБІЛЬНОГО ЗАСТОСУНКУ INFORMATYKAGO ДЛЯ ПІДТРИМКИ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ Владислав ПІКУН, <i>здобувач другого (магістерського) рівня вищої освіти</i> <i>Рівненський державний гуманітарний університет</i> Наталія ПОЛЮХОВИЧ, <i>кандидат педагогічних наук, доцент, доцент кафедри цифрових технологій та методик навчання інформатики</i> <i>Рівненський державний гуманітарний університет</i> Анотація. Розглянуто функціональні можливості мобільного застосунку InformatykaGO, орієнтованого на підтримку індивідуальної освітньої траєкторії здобувачів освіти в умовах змішаного навчання. Ключові слова: змішане навчання, мобільний застосунок, гейміфікація, Flutter, MongoDB, Android, індивідуальна освітня траєкторія. Vladyslav PIKUN, Natalia POLUKHOVICH. Functional Capabilities of the Gamified Mobile Application InformatykaGO for Supporting an Individual Educational Trajectory 200	<i>Матеріали ІІ Всеукраїнської науково-практичної конференції «Підвищення педагогіки до професійної діяльності в умовах змішаного навчання»</i> а також особистий профіль. Особливістю застосунку є вбудована система гейміфікації, що виступає рушієм навчальної мотивації. Користувачі мають змогу накопичувати бали за виконання завдань та проходити різні рівні складності. Закопичувальні елементи у вигляді бейджів, досягнень і візуального прогресу підсилюють інтерес до навчання, сприяють змаганням із самим собою та формують внутрішню мотивацію до досягнення кращих результатів (InformatykaGO, 6. д.). У контексті змішаного навчання <i>InformatykaGO</i> дозволяє органічно поєднувати самостійну роботу здобувачів із очними заняттями. Педагог може рекомендувати конкретні теми для опрацювання, переглядати результати виконання тестів або практичних завдань, що дозволяє здійснювати індивідуальний супровід навчання та оперативно реагувати на типові помилки. Застосунок зручно використовувати як під час освітнього процесу в аудиторії, так і в позааудиторній діяльності, що робить його гнучким інструментом відповідно до потреб освітнього закладу. Актуальність такого програмного рішення зумовлена необхідністю підвищення якості цифрової освіти, створення умов для розвитку цифрових компетентностей здобувачів освіти, а також підготовки їх до викликів інформаційного суспільства. Гейміфікований підхід, реалізований у <i>InformatykaGO</i>, підтвердив свою ефективність у процесі тестування застосунку серед студентів спеціальності Професія освіти (Цифрові технології): користувачі відзначають зручність, зрозумілість структури, а також високий рівень зацікавленості під час навчання. Таким чином, мобільний застосунок <i>InformatykaGO</i> є ефективним інструментом організації змішаного навчання інформатики у закладах професійної освіти. Він сприяє підвищенню мотивації здобувачів, забезпечує доступ до якісного контенту та підтримує самостійну навчальну діяльність. У подальшому планується інтеграція додаткових гейміфікованих елементів, таких як щоденні навчальні виклики та адаптивне формування траєкторій відповідно 202

Рис. А.1 Тези автора дослідження



Рис. А.2 Сертифікат учасника конференції

Додаток В

АНАЛІЗ ПЛАТФОРМ ДЛЯ СТВОРЕННЯ ІГРОВОГО ОСВІТНЬОГО
ЗАСТОСУНКУ З ІНФОРМАТИКИ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРОФЕСІЙНІЙ ДІЯЛЬНОСТІ 221 АНАЛІЗ ПЛАТФОРМ ДЛЯ СТВОРЕННЯ ІГРОВОГО ОСВІТНЬОГО ЗАСТОСУНКУ З ІНФОРМАТИКИ <i>Пікун Владислав, здобувач другого (магістерського) рівня вищої освіти Науковий керівник: Поліхович Наталія Вікторівна, кандидатка педагогічних наук, доцентка Рівненський державний гуманітарний університет</i> Анотація. Проведено порівняльний аналіз платформ React Native, Android Studio та Flutter з метою вибору оптимального інструмента для створення гейміфікованих освітніх рішень. Ключові слова: React Native, Android Studio, Flutter, гейміфікація, самоосвіта, мобільні застосунки. VLADYSLAV PIKUN, NATALIA POLYUKHOVYCH. ANALYSIS OF PLATFORMS FOR CREATING A GAMIFIED EDUCATIONAL APPLICATION IN COMPUTER SCIENCE. Abstract. A comparative analysis of the platforms React Native, Android Studio, and Flutter has been conducted to select the optimal tool for creating gamified educational solutions. Key words: React Native, Android Studio, Flutter, gamification, self-education, mobile applications. У сучасних умовах самоосвіти здобувачів передової освіти є важливим аспектом освітнього процесу. Інтерактивні та гейміфіковані підходи допомагають підвищити мотивацію до навчання, а мобільні застосунки відіграють ключову роль, надаючи здобувачам доступ до навчальних матеріалів у зручному форматі [1]. Метою цієї роботи є визначення найоптимальнішої платформи для створення мобільного застосунку, що сприятиме організації самоосвіти здобувачів закладів передової освіти з інформатики за допомогою гейміфікованих технологій. Такий підхід забезпечує доступність навчальних матеріалів у зручному та цікавому інтерактивному форматі, стимулює здобувачів до більш активної участі в освітньому процесі та підвищує їх мотивацію [1].	Матеріали XVI Всеукраїнської науково-практичної конференції 222 Серед найпопулярніших інструментів для створення мобільних застосунків виділяють React Native, Android Studio та Flutter, кожен з яких має свої переваги та обмеження. React Native, розроблений компанією Meta (Facebook), використовує JavaScript і базується на React, що дозволяє розробникам створювати кросплатформні застосунки з одним кодовим базисом. Ця платформа привертає увагу завдяки активній спільноті розробників і високій швидкості розробки. Однак, для проєктів, які вимагають складних анімацій чи гейміфікованих елементів, React Native може бути обмеженням у продуктивності та зручності інтеграції нативних функцій, що може ускладнити створення багатифункціональних освітніх застосунків [4]. Офіційне інтегроване середовище Android Studio є ідеальною для розробки нативних застосунків під Android, забезпечуючи доступ до всіх функцій операційної системи та високий рівень продуктивності. Це забезпечує розробникам значну гнучкість у реалізації освітніх рішень, однак створення застосунків для iOS вимагає окремих зусиль і додаткового коду. Такий підхід може стати перешкодою, якщо необхідно підтримувати застосунок на різних платформах, що знижує ефективність використання Android Studio для кросплатформних освітніх проєктів [2]. Flutter, розроблений компанією Google, пропонує інший підхід, який здобув популярність завдяки високій продуктивності та підтримці кросплатформної розробки. Використання мови Dart дозволяє досягати високих показників швидкості, що критично важливо для гейміфікованих освітніх застосунків за можливості додавання анімацій та інших інтерактивних елементів. Flutter також має багатий набір збудованих підґрунтя і компонентів, які спрощують додавання гейміфікованих функцій, надаючи розробникам можливість створювати динамічні інтерфейси без необхідності додаткових бібліотек. Крім того, Flutter підтримує створення застосунків як для Android, так і для iOS з єдиною кодовою базою, що підвищує зручність у розробці та обслуговуванні застосунків [3]. З огляду на проведений аналіз основних платформ, Flutter був обраний для даного проєкту як найбільш ефективний інструмент. Завдяки високій продуктивності, багатифункціональності та можливості легко інтегрувати гейміфіковані елементи, Flutter дозволяє створювати застосунки, які ефективно підтримують навчальні цілі, роблячи процес навчання або самоосвіти більш інтерактивним і мотивуючим для здобувачів. Таким чином, вибір Flutter як основної платформи для реалізації гейміфікованих навчальних програм обумовлений його здатністю забезпечити високу якість виконання,
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРОФЕСІЙНІЙ ДІЯЛЬНОСТІ 223 кросплатформність і зручність у реалізації інтерактивних освітніх елементів, що відповідає вимогам сучасної цифрової освіти. Список використаних джерел - Овсієнко Л. Гейміфікація освітнього процесу студентів і курсантів зно. <i>Наукові інновації та передові технології</i>. 2024. № 8(36). С.1308-1320 - Android studio overview. <i>Android Studio</i>. URL: https://developer.android.com/ (дата звернення: 28.10.2024). - Flutter documentation. <i>Flutter</i>. URL: https://docs.flutter.dev/ (дата звернення: 28.10.2024). - React native documentation. <i>React Native</i>. URL: https://reactnative.dev/docs (дата звернення: 28.10.2024).	

Рис. Б.1 Тези автора дослідження



Рис. Б.2 Сертифікат учасника конференції

Додаток Г

Міністерство освіти і науки України
Департамент освіти і науки Рівненської ОДА
Рівненський державний гуманітарний університет



ПРОГРАМА
XVIII Всеукраїнської
науково-практичної конференції
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
В ПРОФЕСІЙНІЙ ДІЯЛЬНОСТІ»

ПЛЕНАРНЕ ЗАСІДАННЯ

З вітальним словом звернувся:

- **Войтович Ігор Станіславович**, доктор педагогічних наук, професор, професор кафедри цифрових технологій та методики навчання інформатики, проректор з навчально-виховної роботи Рівненського державного гуманітарного університету

- **Антонюк Микола Степанович**, кандидат педагогічних наук, доцент, доцент кафедри цифрових технологій та методики навчання інформатики, в.о. декана факультету математики та інформатики

- **Павлова Наталія Степанівна**, доктор педагогічних наук, доцент, професор, завідувач кафедри цифрових технологій та методики навчання інформатики

Доповіди:

СУЧАСНЕ ІНФОРМАЦІЙНО-ОСВІТНЄ СЕРЕДОВИЩЕ: ЗАСОБИ РОЗРОБЛЕННЯ; ОРГАНІЗАЦІЯ ТА ПІДТРИМКА ДІЯЛЬНОСТІ УЧАСНИКІВ ОСВІТНЬОГО ПРОЦЕСУ

Карташова Любов Андріївна, доктор педагогічних наук, професор, професор кафедри освіти дорослих та цифрових технологій ДЗВО Університет менеджменту освіти Національної академії педагогічних наук України

МОДЕЛЬ ВПРОВАДЖЕННЯ ІННОВАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗВИТКУ ЦИФРОВОЇ КОМПЕТЕНТНОСТІ ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ ЗАКЛАДІВ ПРОФЕСІЙНОЇ ОСВІТИ

Гуменний Олександр Дмитрович, кандидат педагогічних наук, старший науковий співробітник відділу цифрових освітніх ресурсів Інститут професійної освіти НАПН України, м. Київ

ГЕЙМІФІКОВАНИЙ МОБІЛЬНИЙ ЗАСТОСУНОК ІНФОРМАТУКАГО ЯК ІНСТРУМЕНТ ФОРМУВАННЯ НАВИЧОК САМООСВИТИ ЗДОБУВАЧІВ ФАХОВОЇ ПЕРЕДВИЩОЇ ОСВІТИ

Пікун Владислав, здобувач другого (магістерського) рівня вищої освіти спеціальності Професійна освіта (Цифрові технології) Рівненський державний гуманітарний університет

ВЕБПОРТФОЛІО ЯК ІНСТРУМЕНТ ПРОФЕСІЙНОГО РОЗВИТКУ Й ЦИФРОВОЇ ІДЕНТИЧНОСТІ ІТ-ФАХІВЦІВ

Шроль Олександр, здобувач першого (бакалаврського) рівня вищої освіти спеціальності Професійна освіта (Цифрові технології) Рівненський державний гуманітарний університет

10 листопада 2025 року
Рівне

Рис. Г.1 Зміст програми конференції

Додаток Д



Рис. Д.1 Подяка за участь у конференції