

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення комп'ютерних
технологій

Наталія СТЕФУРАК / _____ /

підпис

« ____ » _____ 2025 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 122 «Комп'ютерні науки»

на тему: «Мобільний тренажер з основ гри на баяні»

Студент групи КН-41 Роман ЩОТЧУК _____
(підпис)

Керівник роботи Олександра ЧУБЕЙ _____
(підпис)

Консультанти:
з техніко-економічного Любов МЕЛЕНЧУК _____
обґрунтування (підпис)

нормоконтролер Наталя КУЛЬЧИНСЬКА _____
(підпис)

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення комп'ютерних
технологій

Наталія СТЕФУРАК / _____ /

підпис

« ___ » _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Щотчуку Роману Романовичу

(прізвище, ім'я та по-батькові студента)

1. Тема роботи Мобільний тренажер з основ гри на баяні
затверджена наказом по коледжу від “25” Листопада 2024 р., №253а-н
2. Термін здачі студентом завершеної роботи “25” Червня 2025 р.
3. Вихідні дані до роботи: результати аналізів мобільних додатків для навчання гри на баяні.
4. Перелік питань, які повинні бути розроблені:
 - а) основна частина Аналіз предметної області та постановка завдань.
Проектування системи. Реалізація та тестування системи.
 - б) техніко-економічне обґрунтування Аналіз ринку. Розрахунок витрат на проектування. Обґрунтування необхідності розробки.
5. Перелік графічного матеріалу Діаграма варіантів використання додатку,
Контекстна діаграма додатку, Функціональна структура додатку

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
З техніко-економічного обґрунтування	Любов МЕЛЕНЧУК	20.04.2025	20.05.2025

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	Пошук теми роботи, вивчення вимог для кваліфікаційної роботи	16.11.2024	25.11.2024
2	Аналіз існуючих рішень та визначення їхніх переваг і недоліків	25.11.2024	29.11.2024
3	Вивчення технологій та інструментів реалізації.	02.12.2024	03.12.2024
4	Формування архітектури застосунку, визначення функціональних вимог.	07.12.2024	11.12.2024
5	Проектування структури застосунку, бази даних та інтерфейсу користувача	30.12.2024	21.01.2025
6	Тестування застосунку та усунення помилок	25.04.2025	04.05.2025
7	Аналіз ринку збуту, визначення користувачів і оцінка економічної доцільності	03.05.2025	13.05.2025
8	Оформлення пояснювальної записки.	15.05.2025	17.06.2025
9	Попередній захист кваліфікаційної роботи	13.06.2025	13.06.2025
10	Підготовка до захисту кваліфікаційної роботи	14.06.2025	24.06.2025
11	Захист кваліфікаційної роботи	25.06.2025	25.06.2025

Дата видачі “___” _____ 202_ р. Керівник _____ / Чубей О. О.

Завдання прийняв до виконання _____ /Щотчук Р. Р

Реферат

Мобільний тренажер з основ гри на баяні. Кваліфікаційна робота. Щотчук Роман Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 «Комп'ютерні науки», 2025.

Об'єкт дослідження – мобільні додатки для музичної освіти та засоби їхньої кросплатформенної реалізації.

Метою роботи є створення мобільного додатку "Bayan Trainer", що функціонує на різних платформах, з використанням мови програмування Dart та фреймворку Flutter.

Мобільний додаток повинен забезпечити доступ до навчальних матеріалів (теорія, ноти, відеоуроки), надати інтерактивний тренажер з імітацією клавіатур баяна, а також реалізувати систему персоналізації через профілі користувачів. Ключовими функціями є відстеження індивідуального прогресу, налаштування інтерфейсу та можливість додавання користувачами власних пісень з нотами та текстами.

Для реалізації поставленої задачі було використано сучасні технології кросплатформенної розробки, зокрема фреймворк Flutter, мову Dart, середовище розробки Android Studio, а також ключові пакети для керування станом (Provider), збереження даних (SharedPreferences), відтворення аудіо (audioplayers) та роботи з файлами користувача (file_picker, image_picker).

Результатом розробки став завершений програмний продукт, який є гнучким, масштабованим та готовим до використання та подальшого розширення функціоналу.

Ключові слова: **МОБІЛЬНИЙ ДОДАТОК, БАЯН, ТРЕНАЖЕР, FLUTTER, DART, PROVIDER, НАВЧАННЯ, МУЗИЧНА ОСВІТА, КРОСПЛАТФОРМЕНІСТЬ.**

Abstract

Mobile Trainer for Learning the Basics of Bayan Playing. Qualification Work. Shchotchuk, Roman. Viacheslav Chornovil Halych Professional College, Department of Computer Technologies. Specialty 122 "Computer Sciences", 2025.

The object of study is mobile applications for music education and the means of their cross-platform implementation.

The goal of the work is to create the "Bayan Trainer" mobile application, which functions on various platforms, using the Dart programming language and the Flutter framework.

The mobile application must provide access to educational materials (theory, sheet music, video lessons), offer an interactive trainer with a simulation of bayan keyboards, and implement a personalization system through user profiles. Key functions include tracking individual progress, interface customization, and the ability for users to add their own songs with sheet music and lyrics.

To implement the set task, modern cross-platform development technologies were used, particularly the Flutter framework, the Dart language, the Android Studio development environment, as well as key packages for state management (Provider), data persistence (SharedPreferences), audio playback (audioplayers), and user file operations (file_picker, image_picker).

The result of the development is a completed software product that is flexible, scalable, and ready for use and future functional expansion.

Keywords: MOBILE APPLICATION, BAYAN, TRAINER, FLUTTER, DART, PROVIDER, LEARNING, MUSIC EDUCATION, CROSS-PLATFORM.

ЗМІСТ

Скорочення та умовні позначки	6
Вступ.....	7
1 Аналіз Предметної Області Та Постановка Завдань.....	8
1.1 Опис предметної області.....	8
1.2 Аналіз наявних рішень	9
1.3 Аналіз вимог до програмного засобу та постановка завдання.....	15
2 Проєктування Системи	16
2.1 Формування цілей і основних завдань мобільного додатку.....	16
2.2 Функціональна структура мобільного додатку	18
2.3 Проєктування інтерфейсу мобільного додатку	20
3 Реалізація Та Тестування Системи.....	25
3.1 Розробка інтерфейсу додатку	25
3.2 Розробка функціоналу додатку.....	32
3.3 Тестування додатку.....	36
4 Техніко-Економічне Обґрунтування.....	43
4.1 Аналіз ринку	43
4.2 Розрахунок витрат на проєктування	44
4.3 Обґрунтування необхідності розробки.....	47
Висновки	49
Перелік джерел посилання	50
Додатки.....	51

					КР.КН 25.608.23.000 ПЗ						
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Мобільний тренажер з основ гри на баяні						
Розроб.		Щотчук Р.Р							<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Перевір.		Чубей О.О							5	5	65
Рценз		Сиротюк О.Б							ГФК. ВКТ. КН-41		
Н.контр.		Кульчинська Н.З									
Зав. від.		Стефурак Н.А									

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

UI – User Interface.

UX – User Experience.

SDK – Software Development Kit.

API – Application Programming Interface.

IDE – Integrated Development Environment.

JSON – JavaScript Object Notation.

CRUD – Create, Read, Update, Delete.

AOT – Ahead-of-Time.

JIT – Just-in-Time.

SQL – Structured Query Language.

ПЗ – Програмне забезпечення.

ОС – Операційна система.

ЄСВ – Єдиний соціальний внесок.

ПДВ – Податок на додану вартість.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Сучасний світ характеризується стрімким розвитком цифрових технологій, які трансформують різні сфери життя, включаючи освіту. Музична освіта, зокрема навчання гри на інструментах, традиційно вимагає значної кількості часу, фінансових вкладень та особистого контакту з викладачем. Проте, поширення мобільних пристроїв відкриває нові можливості для демократизації та підвищення ефективності навчального процесу. Гра на баяні, як складне мистецтво, що потребує координації, технічної вправності та розвинутого музичного слуху, досі не має широкого спектру якісних мобільних додатків для початкового навчання. Ця обставина обґрунтовує актуальність створення інноваційного мобільного тренажера, що здатен заповнити існуючу прогалину.

Метою даної роботи є розробка мобільного додатку «Мобільний тренажер з основ гри на баяні», який забезпечить інтерактивний та доступний інструмент для самостійного навчання. Завданнями роботи є аналіз предметної області, вивчення існуючих аналогів, розробка архітектури додатку, проектування інтерфейсу користувача, безпосередня програмна реалізація функціоналу тренажера та обґрунтування його економічної доцільності й суспільної значущості. Розроблений додаток має на меті не лише полегшити процес навчання, але й сприяти популяризації баяна як важливого народного інструменту, а також продемонструвати можливості впровадження сучасних технологій у мистецьку сферу.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ

1.1 Опис предметної області

Гра на баяні є складним мистецтвом, що потребує координації, технічної вправності та розвинутого музичного слуху. Навчання гри на баян традиційно відбувається у музичних школах, консерваторіях або під керівництвом приватного викладача. Проте, розвиток цифрових технологій відкриває можливості для створення мобільних додатків, які можуть значно полегшити процес навчання та зробити його більш доступним. Сучасні мобільні додатки для навчання гри на музичних інструментах зазвичай містять інтерактивні уроки, вправи, відеоматеріали, а також функції контролю виконання. Однак, вибір якісного мобільного застосунку саме для навчання гри на баяні є досить обмеженим, що й обґрунтовує необхідність розробки нового тренажера.

Історія баяну:

Баян є одним із найпопулярніших народних інструментів, що походить від гармоніки. Назва «баян» закріпилася за інструментом у 1907 році, коли майстри об'єднали кілька технічних удосконалень у єдину модель.

Протягом ХХ століття баян активно використовувався в академічній музиці, народному виконанні, а також в оркестрових ансамблях. Він став невід'ємною частиною російської, української та білоруської музичної культури. Сучасні баяни мають розширений діапазон і складнішу систему реєстрів, що дозволяє виконувати музику різних жанрів, від класики до джазу.

Цікаві факти про баян:

- У 2010 році баян увійшов до списку нематеріальної культурної спадщини ЮНЕСКО як важливий елемент народної музики.
- Баяністи можуть грати одночасно двома руками різні мелодії завдяки складній конструкції клавіатури.
- У деяких оркестрах баян замінює навіть орган завдяки своїй унікальній тембральній палітрі.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

— Перші баяни мали лише кілька рядів кнопок, тоді як сучасні можуть містити понад 100 клавіш.

У сучасному світі баян зберігає свою популярність і залишається актуальним інструментом як у професійній, так і аматорській музиці. Незважаючи на конкуренцію з боку електронних інструментів, баян використовується в різних жанрах – від класичної музики до джазу та експериментальних композицій.

Значну роль у популяризації баяна відіграють сучасні виконавці, які демонструють унікальні можливості цього інструмента. Завдяки вдосконаленню конструкції баяна та його звукових характеристик, він успішно інтегрується в сучасні оркестри, ансамблі та навіть сольні виступи.

Крім того, баян активно використовується у фольклорних ансамблях, допомагаючи зберігати культурну спадщину різних народів. Також розвиток цифрових технологій сприяє створенню електронних версій баяна, що розширює його можливості для експериментальної та студійної роботи.

Отже, баян продовжує бути важливим музичним інструментом, який не лише зберігає традиції, але й адаптується до сучасних умов, залишаючись актуальним у світі музики.

1.2 Аналіз наявних рішень

На сучасному ринку мобільних додатків представлено багато рішень для навчання гри на музичних інструментах. Деякі з них зосереджені на імітації гри, інші ж пропонують комплексний підхід з навчальними матеріалами та вправами. Проте, спеціалізованих застосунків, які б повністю відповідали потребам баяністів, майже немає. Важливо проаналізувати найпопулярніші з них, визначити їхні переваги та недоліки, щоб зрозуміти, яких саме функцій не вистачає для повноцінного навчання гри на баяні.

Accordion Chromatic Cassoto:

Для оцінки наявних рішень у сфері мобільних додатків для навчання гри на баяні слід розглянути кілька популярних варіантів. Наприклад, який

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

заслуговує на увагу, є Accordion Chromatic Cassoto. Це мобільний додаток, який максимально точно імітує звучання хроматичного баяна, враховуючи особливості акустики та механіки натискання клавіш. Він створений для музикантів, які хочуть тренувати свої навички або експериментувати з музикою, використовуючи цифровий інструмент. Основною перевагою програми є реалістична симуляція гри, що дозволяє користувачам отримати відчуття гри на справжньому баяні. Крім того, додаток надає можливість налаштовувати тембр звучання, змінювати регістри та використовувати різні конфігурації клавіш, що є значною перевагою для досвідчених музикантів.



Рисунок 1.1 – Частина інтерфейсу Accordion Chromatic Cassoto

Проте, Accordion Chromatic Cassoto має і свої недоліки. Одним із головних є відсутність навчальних матеріалів та інтерактивних вправ, що робить його непридатним для новачків, які лише починають знайомитися з інструментом. Також варто зазначити, що використання сенсорного екрану замість фізичних клавіш може не повністю передавати відчуття гри на реальному баяні, що може впливати на процес навчання

Piano by Yousician:

Piano by Yousician є одним із найпопулярніших додатків для навчання гри на клавішних інструментах, включаючи фортепіано та синтезатор. Його головною особливістю є інтерактивна система навчання, яка аналізує гру

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

користувача в реальному часі, даючи миттєвий зворотний зв'язок. Додаток пропонує структуровані курси, які розбиті на рівні складності — від початківців до досвідчених музикантів.

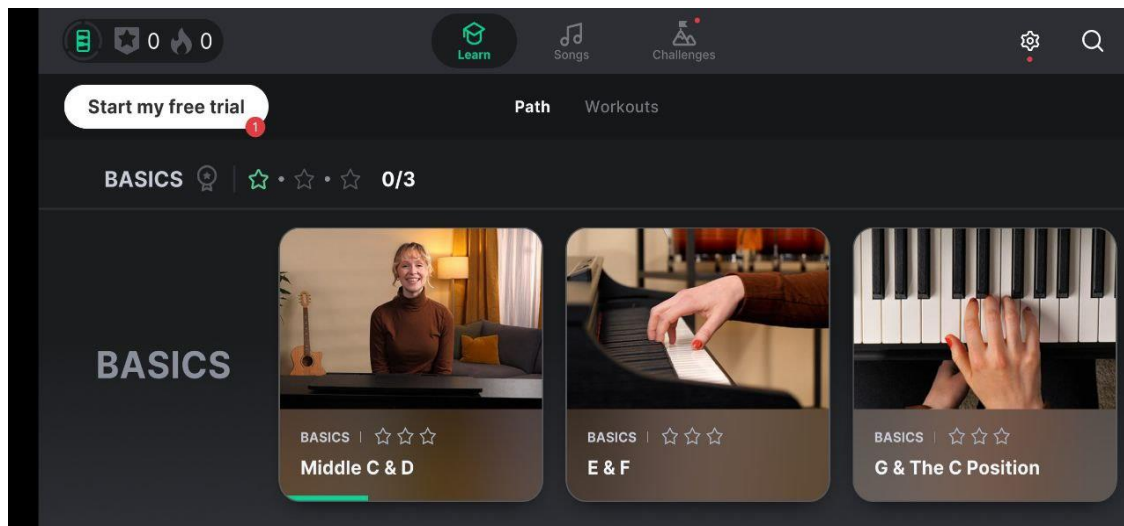


Рисунок 1.2 – Головний інтерфейс Piano by Yousician

Методика навчання Yousician базується на поступовому освоєнні музичної теорії, ритму, нотного запису та технічних навичок гри. Уроки побудовані таким чином, що користувач спочатку вивчає новий матеріал, потім грає вправи, а після цього отримує аналіз своїх помилок та поради щодо їх виправлення.

Додаток використовує гейміфікацію: нарахування балів, досягнення та рівні мотивують користувачів продовжувати навчання.

Однією з основних переваг Yousician є можливість використання мікрофона для розпізнавання звуків, що дозволяє користувачеві грати на реальному інструменті, а не лише на віртуальному клавішному полі. Крім того, у додатку є велика бібліотека пісень і вправ, що дозволяє тренувати навички на популярних музичних композиціях.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

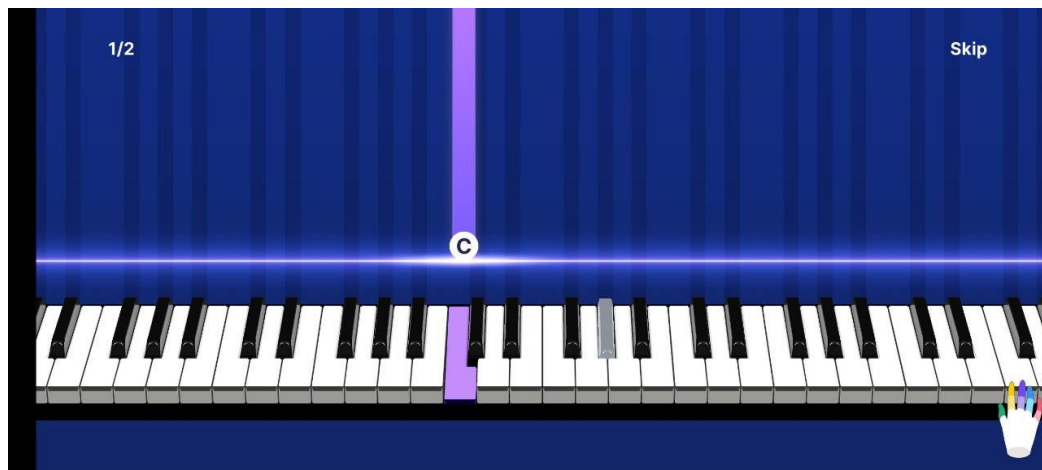


Рисунок 1.3 – Приклад виконання вправи

Додаток також містить велику кількість відеоуроків, у яких професійні викладачі пояснюють основи гри, розбирають складні моменти та дають рекомендації щодо поліпшення техніки. Відеоуроки розділені на категорії за рівнем складності та тематикою, що дозволяє користувачам поступово просуватися від базових знань до складніших музичних творів. Завдяки наочному демонструванню та поясненням, користувачі можуть легше засвоювати нові технічні прийоми та музичну теорію.

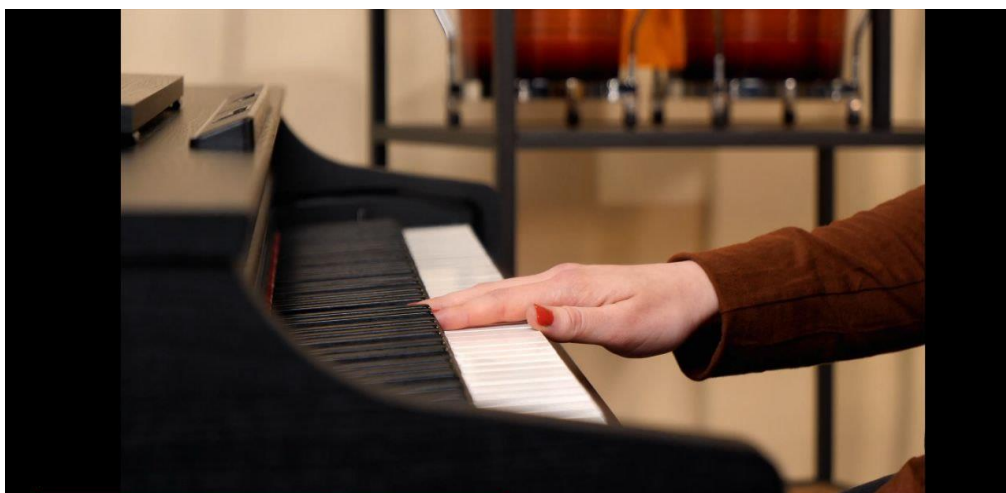


Рисунок 1.4 – Кадр з відеоуроку

Проте, варто відзначити, що Piano by Yousician більше орієнтований на клавішні інструменти, такі як фортепіано та синтезатор, а не баян. Він не враховує специфічних особливостей гри на баяні, таких як використання міхів

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

і особливості лівої клавіатури. Також доступ до повного функціоналу додатка можливий лише за підпискою, що може стати перешкодою для деяких користувачів.

Accordion Piano:

Accordion Piano – ще один популярний додаток у Play Market, що дозволяє користувачам вивчати гру на баяні та акордеоні. Він містить різноманітні режими навчання, включаючи інтерактивні вправи, ноти та акомпанемент. Додаток підтримує різні конфігурації клавіатури та регістри, що допомагає відтворювати автентичне звучання інструмента.



Рисунок 1.5 – Головний інтерфейс

На даній таблиці можна побачити основні переваги та недоліки даного застосунку.

Таблиця 1.1 – Переваги та недоліки додатків

Переваги	Недоліки
Простий та зрозумілий інтерфейс	Наявність реклами у безкоштовній версії
Вбудовані уроки для початківців	Обмежений функціонал без преміум-доступу
Можливість запису виконання	Відсутність просунутих навчальних матеріалів для професіоналів
Налаштування темпу та стилю гри	Не завжди коректно розпізнає натискання на сенсорному екрані

Основною перевагою Accordion Piano є наявність вбудованих уроків, що робить його зручним для початківців. Він також підтримує запис виконання, що дозволяє користувачам аналізувати свою гру та виправляти помилки. Однак недоліком є реклама в безкоштовній версії та необхідність придбання преміум-доступу для розширеного функціоналу.

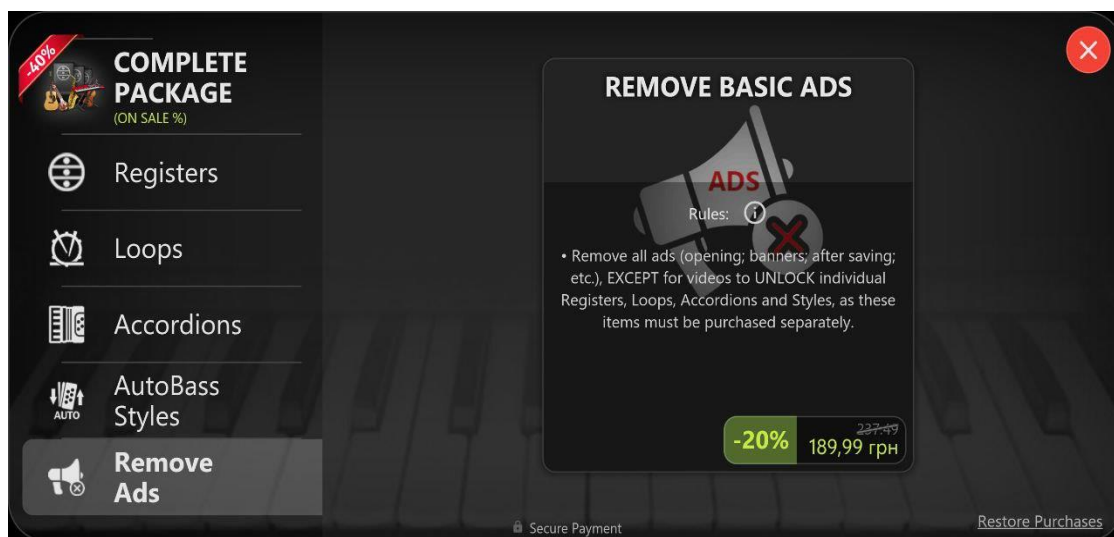


Рисунок 1.6 – Преміум доступ

Accordion Piano – це сучасний мобільний додаток, який надає можливість навчання гри на баяні та акордеоні в інтерактивному форматі. Завдяки інтуїтивно зрозумілому інтерфейсу, користувачі можуть швидко освоїти базові навички гри. Вбудовані уроки допомагають початківцям засвоїти основи, а наявність функції запису дозволяє аналізувати власний прогрес.

Попри численні переваги, такі як можливість налаштування темпу гри, стилю акомпанементу та використання інтерактивних вправ, додаток має і певні обмеження. Наприклад, наявність реклами у безкоштовній версії може заважати навчальному процесу, а відсутність розширених курсів для професійних музикантів робить його менш привабливим для досвідчених виконавців. Крім того, точність розпізнавання сенсорного натискання інколи може бути недостатньо високою, що впливає на загальний досвід гри.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Загалом, Accordion Piano є хорошим інструментом для новачків та любителів, які бажають покращити свої навички гри на баяні. Це доступне та корисне рішення для тих, хто шукає зручний спосіб навчання, проте для професійного використання можуть знадобитися більш спеціалізовані програми або традиційні методи навчання.

1.3 Аналіз вимог до програмного засобу та постановка завдання

Проектування мобільного додатку Bayan Trainer вимагає чіткого формулювання функціональних і нефункціональних вимог, що дозволить забезпечити ефективність та зручність його використання для навчання гри на баяні. Основною цільовою аудиторією додатку є початківці, учні музичних шкіл та всі охочі опанувати базові навички гри на цьому інструменті в інтерактивному форматі.

До функціональних вимог належать: можливість реєстрації та авторизації користувача, доступ до навчальних модулів з теоретичними матеріалами та інтерактивними вправами, відтворення звуків баяна, надання візуальних підказок під час гри, автоматична перевірка правильності виконання вправ та формування зворотного зв'язку. Додаток повинен зберігати прогрес користувача, відображати статистику навчання та надавати рекомендації для подальшого вдосконалення навичок.

Нефункціональні вимоги охоплюють адаптивність інтерфейсу до різних розмірів екранів мобільних пристроїв, високу швидкодію, стабільність роботи, зручність використання, а також безпеку персональних даних. Інтерфейс має бути інтуїтивно зрозумілим, візуально привабливим і відповідати тематиці музичної освіти.

Завданням розробки є створення мобільного застосунку, що об'єднає теоретичну базу, практичні тренування та систему оцінювання результатів, що сприятиме ефективному дистанційному навчанню гри на баяні. Особливу увагу планується приділити музичній точності, інтерактивності та адаптивності системи до різного рівня підготовки користувача.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Формування цілей і основних завдань мобільного додатку

Мобільний додаток "Мобільний тренажер з основ гри на баяні" розробляється з метою створення ефективного та доступного інструменту для навчання основам гри на баяні. Цей додаток має забезпечити користувачів можливістю отримувати теоретичні знання, практикуватися у грі та перевіряти свої навички у зручному форматі, незалежно від місця та часу. Головною метою є створення інтерактивного навчального середовища, яке дозволить поступово освоювати музичні основи, розвивати технічні навички та закріплювати отримані знання за допомогою інтерактивних вправ. Додаток також надаватиме можливість користувачам отримувати зворотний зв'язок щодо правильності виконання вправ, що сприятиме підвищенню якості навчання.

Для досягнення цієї мети мобільний додаток має виконувати низку завдань. Насамперед, він повинен містити модулі теоретичного навчання, які охоплюватимуть основи нотної грамоти, базові акорди та техніку гри на баяні. Також необхідно реалізувати інтерактивні вправи, які допоможуть користувачам закріпити отримані знання на практиці. Однією з важливих функцій додатку є перевірка правильності виконання вправ, що дозволить автоматично аналізувати успішність користувача та надавати рекомендації щодо покращення результатів. Для забезпечення комфортної взаємодії додаток повинен мати інтуїтивно зрозумілий інтерфейс, що дозволить швидко орієнтуватися в його функціях. Крім того, система має забезпечувати збереження прогресу користувача, дозволяючи йому відстежувати власні досягнення. Важливим компонентом стане також інтеграція з аудіо- та відеоматеріалами, що зробить навчальний процес більш наочним та ефективним.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Контекстна діаграма мобільного додатку відображає його взаємодію з основними зовнішніми суб'єктами. Основними суб'єктами є користувач, який взаємодіє з додатком, проходить навчання та виконує вправи, сервер бази даних, що зберігає інформацію про користувачів, їхній прогрес та матеріали курсу, а також адміністратор, який додає та оновлює навчальні матеріали, здійснює технічну підтримку та стежить за коректністю функціонування системи. Завдяки такій структурі взаємодії забезпечується безперебійне функціонування додатку та ефективне управління навчальним контентом.



Рисунок 2.1 – Контекстна діаграма додатку

На рисунку 2.2 зображено діаграму варіантів використання мобільного додатку *Bayan Trainer*, яка відображає основні функціональні можливості системи з точки зору користувача. Основним актором є користувач, який взаємодіє з додатком через чотири ключові сценарії використання (use cases).

Серед доступних функцій:

Реєстрація користувача — процес створення облікового запису з метою персоналізації досвіду навчання та збереження прогресу.

Перегляд вправ — можливість ознайомлення з доступними навчальними матеріалами та вправами, згрупованими за модулями.

Виконання вправи — активна взаємодія користувача з навчальним матеріалом, яка передбачає практику гри на віртуальному баяні.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Перегляд прогресу — функція аналізу результатів навчання, що дозволяє відслідковувати досягнення та оцінювати динаміку покращення навичок.

Ця діаграма чітко демонструє межі системи та основні сценарії, що реалізуються в межах мобільного додатку. Вона є важливою складовою при моделюванні вимог до програмного забезпечення та служить основою для подальшого проектування структури інтерфейсу та логіки роботи додатку.

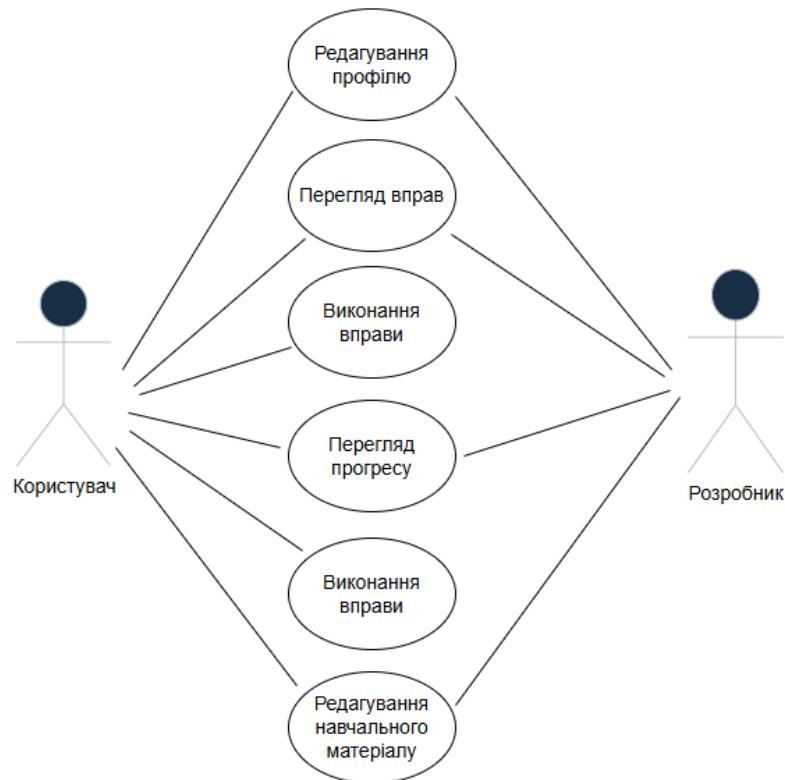


Рисунок 2.2 – Діаграма варіантів використання додатку

2.2 Функціональна структура мобільного додатку

Мобільний додаток *Bayan Trainer* має модульну функціональну структуру, яка забезпечує зручну та ефективну взаємодію користувача з навчальними матеріалами, вправами і відстеженням особистого прогресу. Вся система умовно поділяється на дві категорії користувачів: Користувач (учень) і Розробник (адміністратор).

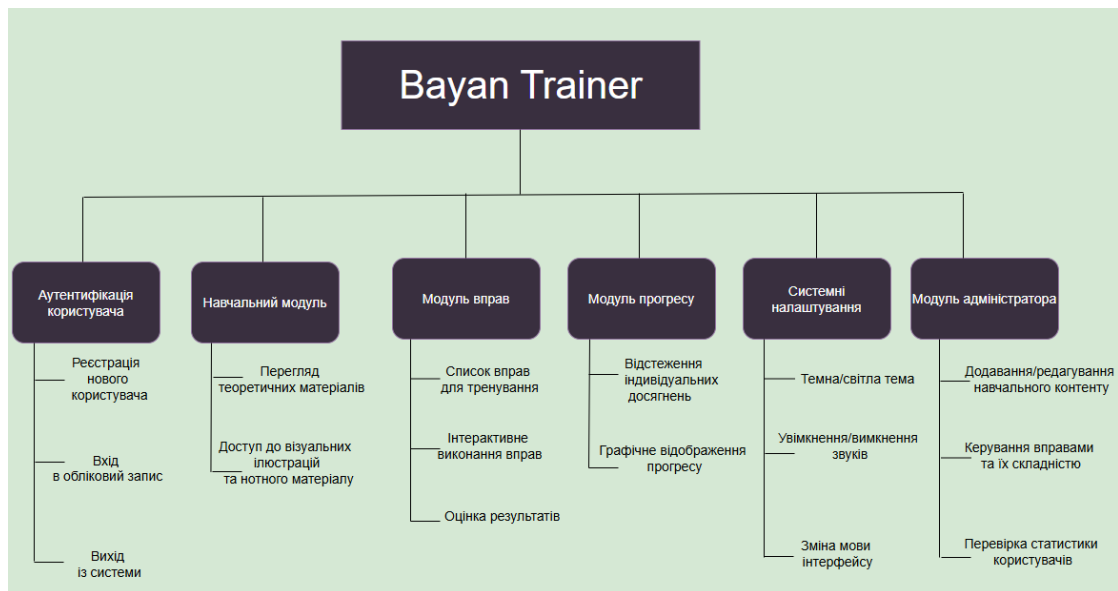


Рисунок 2.3 – Функціональна структура додатку

Серед ключових елементів функціональності додатку виділяється модуль аутентифікації, що дозволяє реєструвати нового користувача, здійснювати вхід до облікового запису, а також вихід із системи. Навчальний модуль дає можливість переглядати теоретичні матеріали, такі як будова баяна, класифікація клавіш та техніка гри, а також включає доступ до візуальних ілюстрацій та нотного матеріалу.

У модулі вправ користувач може переглядати список доступних тренувальних завдань, виконувати їх в інтерактивному режимі та отримувати оцінку результатів. Модуль прогресу дозволяє відстежувати індивідуальні досягнення користувача, зокрема у вигляді діаграм, що візуалізують навчальні результати.

Системні налаштування надають змогу змінювати мову інтерфейсу, керувати звуковими ефектами та обирати між темною і світлою темою оформлення. Для розробника передбачено окремий адміністративний модуль, що включає функції додавання та редагування навчального контенту, керування вправами та їх рівнем складності, а також перегляду статистики користувачів, що забезпечує підтримку актуальності та ефективності освітнього процесу.

2.3 Проектування інтерфейсу мобільного додатку

Проектування інтерфейсу користувача (UI) та користувацького досвіду (UX) є абсолютно фундаментальним етапом у розробці будь-якого мобільного додатку, включаючи наш "Bayan Trainer". Створення продуманого інтерфейсу не просто додає візуальної привабливості; воно робить додаток інтуїтивно зрозумілим, зручним у використанні та, зрештою, ефективним у досягненні його цілей. Для нашої програми ми прагнемо гармонійно поєднати функціональність навчання з приємним та безперешкодним користувацьким досвідом.

Все починається з глибокого розуміння нашої цільової аудиторії та їхніх потреб. Ми маємо чітко уявити, хто саме користуватиметься "Bayan Trainer": це можуть бути абсолютні початківці, учні музичних шкіл, дорослі, які навчаються для власного задоволення, або навіть досвідчені баяністи, які шукають нові методи практики.

Важливо визначити їхні основні цілі – чи то вивчення нових пісень, вдосконалення техніки, запам'ятовування мелодій, відстеження власного прогресу, чи просто доступ до нот та аудіоматеріалів. Також необхідно передбачити потенційні виклики, з якими вони можуть зіткнутися, як-от труднощі з читанням нот, відсутність інструмента під рукою, потреба в персоналізованих вправах або ж ризик швидкого нудьгування від монотонних занять. Усвідомлення цих аспектів формує ядро наших подальших дизайнерських рішень.

Наступним кроком є дослідження та аналіз конкурентів. Ми повинні ретельно вивчити існуючі музичні навчальні додатки – це можуть бути програми для гітари, піаніно, або навіть інші програми для баяна чи акордеона, якщо такі є на ринку.

Мета цього етапу полягає у виявленні успішних елементів, що добре працюють та є інтуїтивно зрозумілими для користувачів. Водночас, ми повинні ідентифікувати недоліки та "больові точки" інших додатків – те, чого

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

їм бракує або що викликає розчарування у користувачів. Це дасть нам безцінні ідеї для інтеграції унікальних функцій та покращень у "Bayan Trainer".

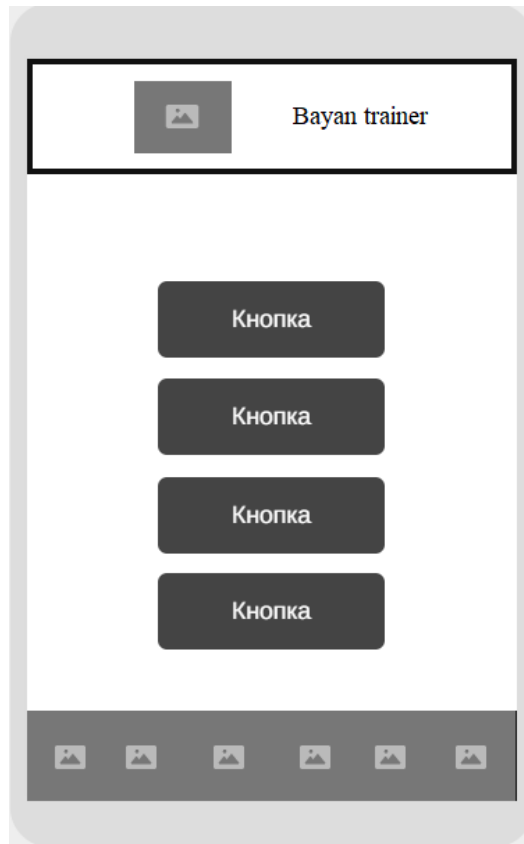


Рисунок 2.4 – Макет головної сторінки додатку

Після цього ми переходимо до створення користувацьких сценаріїв та діаграм потоків (Flow-Chart). Користувацький сценарій – це детальний опис того, як типовий користувач буде взаємодіяти з нашим додатком для досягнення конкретної мети.

Наприклад, ми можемо уявити, як Іван, наш уявний користувач, відкриває додаток, щоб вивчити нову пісню. Він переходить до розділу "Пісні", обирає "Веселу Мелодію", переглядає ноти, слухає аудіозапис, а потім переходить до розділу "Тренажер", щоб практикуватися.

Діаграма потоку, або Flow-Chart, доповнює сценарій, візуалізуючи шлях користувача через увесь додаток. Це допомагає візуально спланувати кожен екран, кожну взаємодію та всі можливі шляхи, забезпечуючи логічну та послідовну навігацію.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

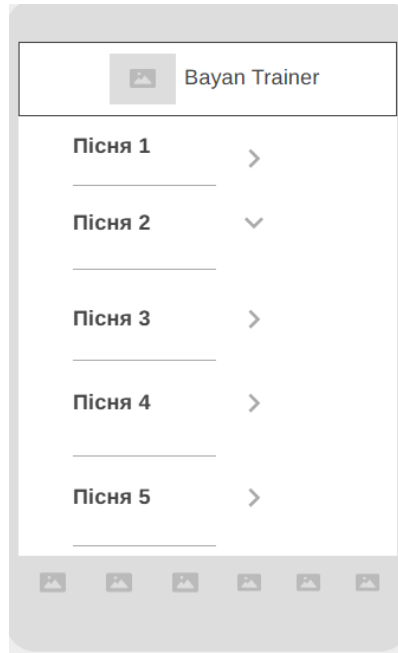


Рисунок 2.5 – Макет сторінки музичних творів

Далі слідує етап Wireframing (створення каркасів). Це процес розробки чорнових, низькодеталізованих візуальних представлень кожного екрану додатку. На цьому етапі ми зосереджуємося виключно на розміщенні елементів, структурі та навігації, повністю абстрагуючись від естетики.

Для цього можна використовувати прості інструменти, як-от папір та олівець, або ж спеціалізоване програмне забезпечення, наприклад, Figma, Sketch чи Adobe XD. На каркасах ми позначаємо заголовки сторінок ("Головна", "Пісні", "Календар", "Тренажер"), основні елементи керування (кнопки, поля вводу, списки), навігаційні елементи (нижнє меню, кнопки "Назад") та основні блоки контенту, де будуть розміщені зображення, текст тощо.

Наприклад, для екрану "Пісні" каркас може включати AppBar з назвою, основну область для списку пісень з іконками, назвами та авторами, а внизу – стандартний BottomNavigationBar.

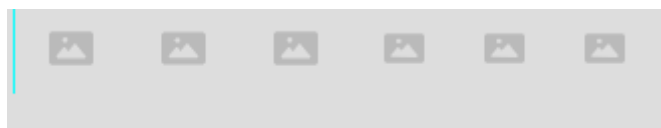


Рисунок 2.6 – Макет навігаційної панелі

Після затвердження каркасів ми переходимо до Mockups та Prototyping (макетування та створення прототипів). Мокапи – це високодеталізовані, статичні зображення інтерфейсу, які вже показують, як додаток виглядатиме у реальному житті, включаючи кольори, типографіку, іконки та зображення.

Для "Bayan Trainer" ми можемо визначити нашу кольорову палітру: яскравий бірюзовий (#00ADB5) як основний акцентний колір, контрастний червоний (#E94560) для кнопок дії та виділення, та більш темні відтінки сірого (#222831, #121212, #393E46, #16213E) для фону, AppBar та елементів списку. Типографіка має бути чіткою та розбірливою, а іконографія – поєднувати стандартні іконки Material Design з кількома кастомними, що асоціюються з баяном або музикою.

Прототипування, на відміну від статичних мокапів, створює інтерактивні версії макетів, які імітують поведінку додатку. Це дозволяє нам тестувати навігацію та взаємодію ще до написання коду, виявляючи потенційні проблеми UX на ранніх стадіях та отримуючи цінний зворотний зв'язок від потенційних користувачів.



Рисунок 2.7 – Кольорова гама додатку

Завершивши візуальну частину, ми переходимо до специфікацій дизайну та guidelines. Це надзвичайно важливо для команди розробників. Ми

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

створюємо документ або використовуємо функціонал інструментів дизайну (наприклад, "Inspect" у Figma), щоб надати точні значення кольорів (HEX, RGB), розмірів шрифтів, відступів та радіусів скруглення. Особливу увагу приділяємо адаптивності: як інтерфейс виглядатиме на різних розмірах екранів, від смартфонів до планшетів.

Ключовим принципом тут є послідовний дизайн: усі елементи, кнопки, типографіка та кольори повинні використовуватися одноманітно по всьому додатку, створюючи відчуття цілісності та професіоналізму.

Підсумовуючи, ключові принципи UX/UI для "Bayan Trainer" включають простоту та зрозумілість (уникнення перевантажених екранів, фокусування на одній основній задачі), інтуїтивну навігацію (користувачі завжди повинні розуміти, де вони знаходяться), візуальну ієрархію (важливі елементи помітніші), доступність (контраст кольорів, розмір шрифту) та загальну привабливість (чистий, сучасний та приємний дизайн).

Застосування цього комплексного підходу забезпечить міцну основу для створення зручного та ефективного мобільного додатку «Bayan Trainer».

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Розробка інтерфейсу додатку

Функціональна структура мобільного додатку «Мобільний тренажер з основ гри на баяні» визначає архітектуру програмного забезпечення з точки зору його основних функціональних модулів. Кожен модуль відповідає за надання певного набору можливостей користувачеві та забезпечує логічне розмежування функціоналу. Така структура сприяє ефективності розробки, полегшує подальшу підтримку та розширення системи, а також забезпечує інтуїтивно зрозумілу взаємодію з користувачем.

Центральною точкою доступу до всіх функціональних розділів є Модуль «Головна сторінка» (рис 3.1). Він слугує основним навігаційним хабом, надаючи швидкий огляд та можливість переходу до ключових секцій додатку за допомогою інтуїтивних кнопок або іконок. Тут користувач може отримати загальну інформацію та перейти до подальших дій, що робить цей модуль ефективним стартовим майданчиком для будь-якої активності в додатку.

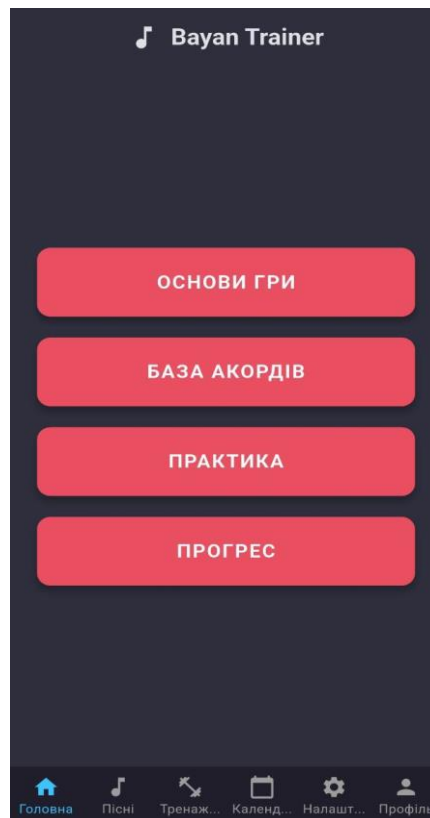


Рисунок 3.1 – Головна сторінка додатку

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Навчальний аспект реалізовано через Модуль «Основи Гри» (рис 3.2), який надає повний спектр теоретичних та довідкових матеріалів, необхідних для початкового освоєння баяна.

Він містить детальні текстові описи, доповнені візуальними ілюстраціями, що охоплюють будову баяна, призначення кожної з його клавіатур (мелодійної та басової/акордової), а також роль міхів у звуковидобуванні.

Окрім цього, модуль містить інформацію про правильну постановку рук та корпусу під час гри (рис 3.3), базові відомості з нотної грамоти, адаптовані до специфіки баяна, та рекомендації щодо догляду за інструментом. Всі зображення (схеми, фотографії) мають можливість детального перегляду.

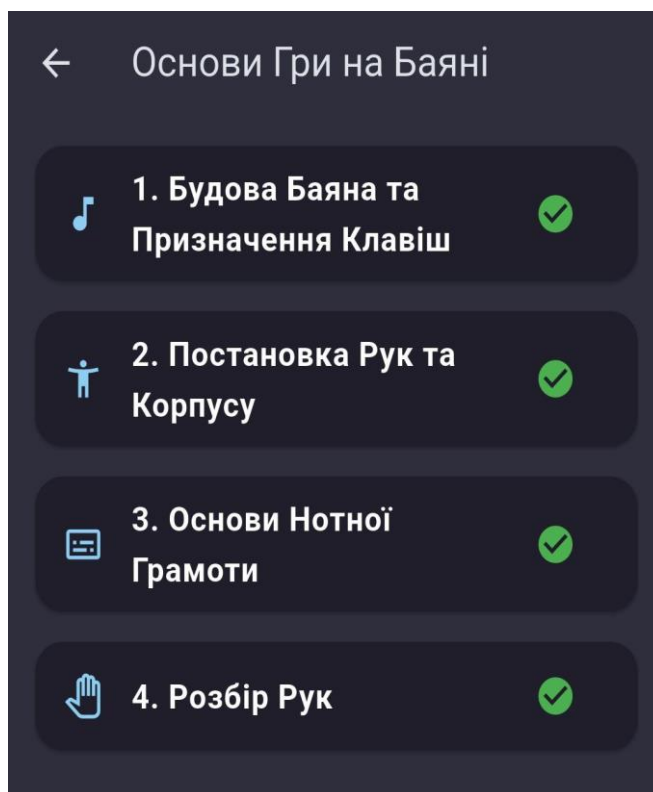


Рисунок 3.2 – Розділ основи гри

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

3. Міх

Розташований між двома корпусами баяна, міх дозволяє створювати тиск повітря, необхідний для звучання язичків. Від характеру руху міхів (розтягування чи стискання) залежить динаміка та інтонація гри.



Рисунок 3.3 – Будова баяна

Для практичного відпрацювання навичок ключову роль відіграє Модуль "Тренажер" (рис 3.4, рис 3.5). Цей інтерактивний компонент надає користувачеві віртуальну імітацію правої та лівої клавіатур баяна, що дозволяє відтворювати реалістичні звуки інструменту при натисканні на відповідні віртуальні елементи.

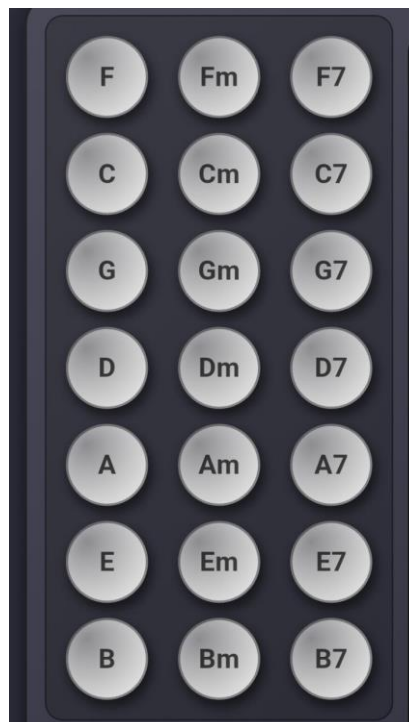


Рисунок 3.4 – Інтерфейс тренажера баяна

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Тренажер підтримує різні режими тренування, включаючи вільну гру, відпрацювання гам та арпеджіо, а також виконання заданих послідовностей нот та акордів з інтерактивними візуальними та звуковими підказками. Можливість налаштування темпу за допомогою вбудованого метронома дозволяє адаптувати вправи під індивідуальні потреби учня.

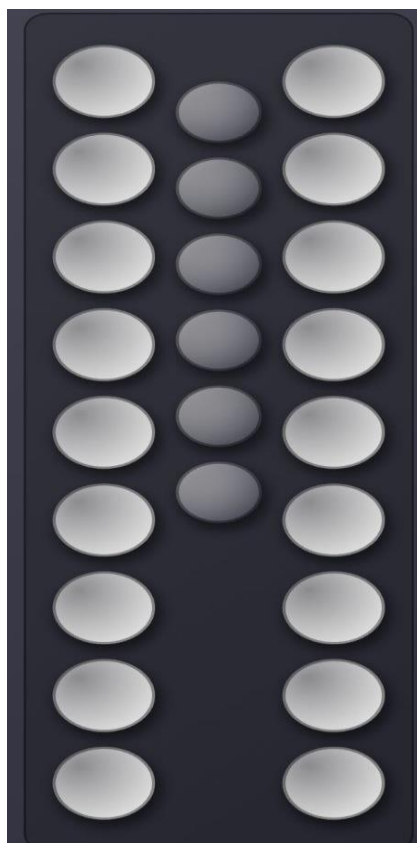


Рисунок 3.5 – Інтерфейс тренажеру баяну

Розширення репертуару та ознайомлення з музичними творами забезпечує Модуль "Пісні / Музичні Твори" (рис 3.6). Він містить бібліотеку початкових музичних композицій та вправ, представлених у вигляді списку з назвою та автором. Користувач може обирати будь-який твір для детального ознайомлення, прослуховувати його аудіозапис за допомогою вбудованого програвача та переглядати відповідний нотний матеріал, а також має можливість додавати свої пісні.

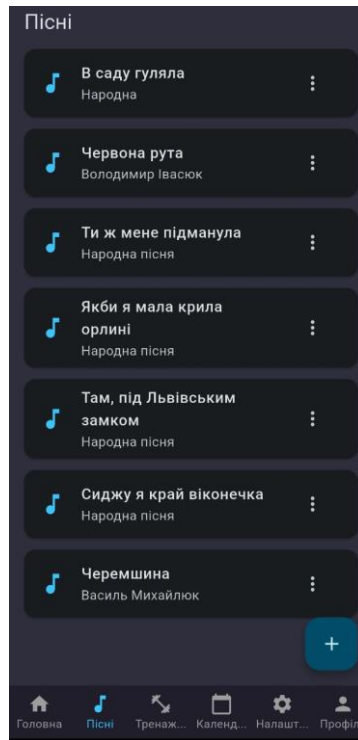


Рисунок 3.6 – Музичні твори

Для зручного вивчення гармонії передбачено Модуль "База Акордів" (рис 3.7). Цей довідник містить каталог різноманітних акордів (мажорних, мінорних, септакордів тощо), дозволяючи користувачеві не тільки переглядати візуальні схеми акордів на віртуальній клавіатурі.



Рисунок 3.7 – База акордів

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Для контролю за процесом навчання впроваджено Модуль "Календар (рис 3.8) та Прогрес (рис 3.9)". Цей модуль дозволяє відстежувати навчальний процес, надаючи календар для відзначення днів занять та їх тривалості, зберігаючи детальну історію тренувань (включаючи виконані вправи, час та результати), а також візуалізуючи статистику прогресу за допомогою графіків та показників успішності.

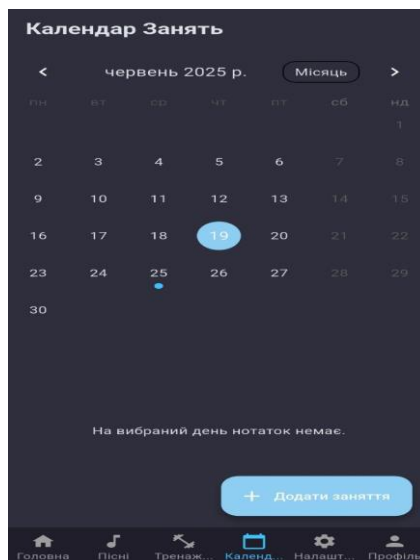


Рисунок 3.8 – Календар занять

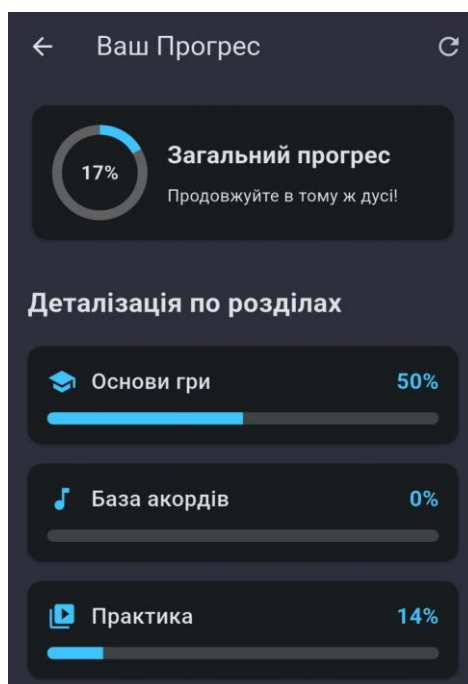


Рисунок 3.9 – Сторінка прогресу

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Персоналізація досвіду взаємодії з додатком забезпечується через Модуль "Профіль Користувача" (рис 3.10), який дозволяє користувачеві керувати своїми персональними даними, переглядати та редагувати інформацію, а також, за наявності функціоналу, змінювати свій аватар. Додаткові налаштування функціонування додатку здійснюються через Модуль "Налаштування" (Settings Module), що надає можливості зміни теми інтерфейсу (світла/темна), регулювання гучності звуків баяна та інших аудіоелементів, а також вибору мови інтерфейсу.

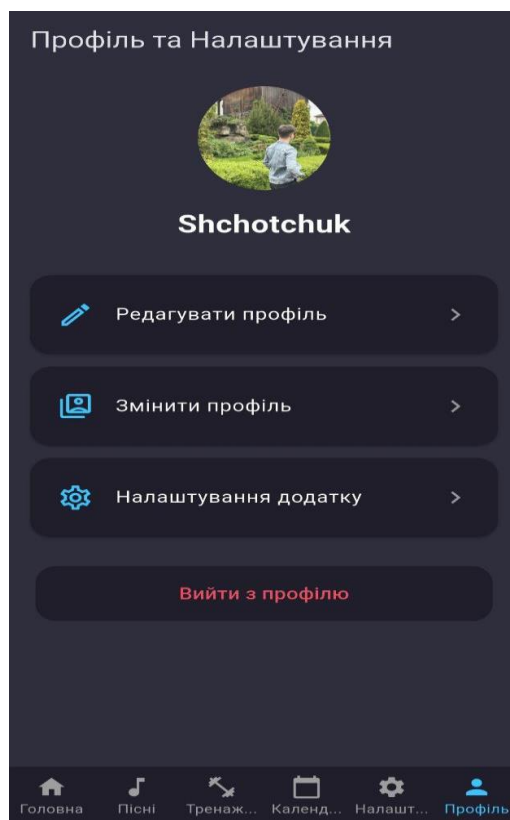


Рисунок 3.10 – Профіль користувача

Кожен з цих модулів розробляється як незалежний компонент, проте всі вони тісно взаємодіють між собою через єдиний користувацький інтерфейс та внутрішню логіку програми. Така модульна структура забезпечує високу гнучкість, спрощує масштабування та ефективність розробки мобільного додатку "Мобільний тренажер з основ гри на баяні", роблячи його зручним та потужним інструментом для самонавчання.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

3.2 Розробка функціоналу додатку

Внутрішня архітектура та реалізація програмного коду мобільного додатку "Мобільний тренажер з основ гри на баяні" базується на принципах модульності та чіткого розділення відповідальності між компонентами. Розробка велася з використанням мови Dart та фреймворку Flutter, що дозволило застосувати сучасні архітектурні патерни для забезпечення гнучкості, масштабованості та легкості підтримки системи.

Основою будь-якого функціонального додатку є його моделі даних, які представляють сутності, що зберігаються та обробляються системою. Для "Мобільного тренажера з основ гри на баяні" визначені ключові моделі, що структурують інформацію. Зокрема, клас Note розроблений для представлення окремої музичної ноти. Кожна нота містить такі атрибути, як назва (наприклад, "C", "D#"), номер октави, шлях до відповідного аудіофайлу в ресурсах додатка (soundAssetPath), а також унікальний ідентифікатор віртуальної клавіші або кнопки баяна (keyId), якій ця нота відповідає. Цей клас є незмінним після створення, що спрощує керування даними про окремі звуки.

Приклад структури класу Note на мові Dart:

```
class Note {  
    final String name;  
    final int octave;  
    final String soundAssetPath;  
    final int keyId;  
    Note({required this.name, required this.octave, required  
this.soundAssetPath, required this.keyId});  
    // Можуть бути додаткові фабричні конструктори або методи  
для серіалізації/десеріалізації.  
}
```

Модель Exercise інкапсулює всю інформацію про окрему навчальну вправу або музичний уривок. Вона містить унікальний ідентифікатор вправи (id), її назву (title), детальний опис (description), а також ключовий елемент – список об'єктів Note (sequence), що представляє послідовність нот для виконання вправ на тренажері. Додатково, для кожної вправи можуть бути вказані шляхи до зображення нотного матеріалу (sheetMusicImagePath) та

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

аудіофайлу повного виконання (`audioPlaybackPath`), а також її рівень складності (`difficultyLevel`).

Приклад структури класу `Exercise` на мові Dart:

```
import 'note.dart'; // Припустимо, Note знаходиться в
окремому файлі
class Exercise {
  final int id;
  final String title;
  final String description;
  final List<Note> sequence; // Послідовність нот для
тренажера
  final String? sheetMusicImagePath; // Шлях до зображення
нот
  final String? audioPlaybackPath; // Шлях до аудіо
виконання
  final int difficultyLevel;
  Exercise({
    required this.id,
    required this.title,
    required this.description,
    required this.sequence,
    this.sheetMusicImagePath,
    this.audioPlaybackPath,
    required this.difficultyLevel,
  });
  // ... toJson/fromJson методи для роботи з даними
```

Прогрес користувача відстежується за допомогою моделі `UserProgress`. Цей клас зберігає дані про проходження користувачем конкретної вправи, включаючи ідентифікатор вправи (`exerciseId`), дату завершення (`completionDate`), оцінку точності виконання (`accuracyScore`), час, витрачений на вправу (`timeSpentSeconds`), та булеве значення, що вказує, чи вправа була повністю завершена (`isCompleted`). Об'єкти `UserProgress` є фундаментом для функціоналу відстеження прогресу та календаря занять.

Приклад структури класу `UserProgress` на мові Dart:

```
class UserProgress {
  final int exerciseId;
  final DateTime completionDate;
  final double accuracyScore;
  final int timeSpentSeconds;
  final bool isCompleted;
  UserProgress({
    required this.exerciseId,
    required this.completionDate,
```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        required this.accuracyScore,
        required this.timeSpentSeconds,
        required this.isCompleted,
    });
    // Методи для серіалізації в JSON та десеріалізації з JSON
    // використовуються для збереження даних у локальному
сховищі
    }

```

Реалізація основного функціоналу додатку здійснюється шляхом поєднання віджетів Flutter з відповідною бізнес-логікою, яка взаємодіє з цими моделями даних та сторонніми плагінами. Ключовим для аудіо відтворення є модуль відтворення звуків баяна, який інтегрує Flutter-плагін audioplayers. Кожному елементу віртуальної клавіатури або об'єкту Note присвоюється унікальний шлях до аудіофайлу (наприклад, assets/sounds/bayan_keys/C4.mp3). При натисканні на віртуальну клавішу, метод play() об'єкта AudioPlayer викликається з вказаним шляхом, забезпечуючи реалістичне звучання. Для уникнення затримок та забезпечення одночасного відтворення кількох нот, налаштовується режим PlayerMode.lowLatency.

Фрагмент коду, що демонструє використання audioplayers:

```

import 'package:audioplayers/audioplayers.dart';
class SoundPlayer {
    final AudioPlayer _player = AudioPlayer(); //
Ініціалізація гравця
    Future<void> playNote(String assetPath) async {
        // Відтворення звуку з вказаного шляху до асету
        await _player.play(AssetSource(assetPath), mode:
PlayerMode.lowLatency);
    }
    void dispose() {
        _player.dispose(); // Звільнення ресурсів гравця
    }
}

```

Для ефективного керування даними та оновленням інтерфейсу, особливо при зміні стану даних, застосовувався патерн Provider. Класи, які виступають у ролі ViewModel-подібних об'єктів, розширюють ChangeNotifier та надають дані віджетам, сповіщаючи їх про будь-які зміни. Це гарантує автоматичне оновлення інтерфейсу користувача при зміні базових даних, забезпечуючи плавну та реактивну взаємодію.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Приклад використання ChangeNotifier для керування станом:

```
import 'package:flutter/material.dart';
import '../models/exercise.dart';
// import '../data/exercise_repository.dart'; // Припустимо,
// є репозиторій для отримання даних

class ExerciseViewModel extends ChangeNotifier {
  // final ExerciseRepository _repository =
  ExerciseRepository(); // Приклад використання репозиторію
  List<Exercise> _exercises = [];

  List<Exercise> get exercises => _exercises; // Геттер для
  доступу до списку вправ

  Future<void> loadExercises() async {
    // _exercises = await _repository.getExercises(); //
    Завантаження даних
    // Приклад статичних даних для демонстрації
    _exercises = [
      Exercise(id: 1, title: "Гамма До-мажор", description:
"Опис...", sequence: [], difficultyLevel: 1),
      // ... інші вправи
    ];
    notifyListeners(); // Сповіщає віджети, що
    спостерігають, про зміни
  }
}
```

Локальне збереження налаштувань користувача (тема, мова) та даних прогресу реалізовано за допомогою плагіна `shared_preferences`. Цей механізм дозволяє просто зберігати та отримувати пари "ключ-значення". Для складніших структур даних, таких як списки об'єктів `UserProgress`, вони спочатку серіалізуються у JSON-рядки, а потім зберігаються, забезпечуючи збереження стану додатку між сесіями та персоналізацію досвіду користувача.

Приклад використання shared_preferences:

```
import 'package:shared_preferences/shared_preferences.dart';
class AppSettingsService {
  static const _themeKey = 'app_theme';
  Future<void> saveTheme(String themeName) async {
    final prefs = await SharedPreferences.getInstance();
    await prefs.setString(_themeKey, themeName);
  }
  Future<String?> getTheme() async {
    final prefs = await SharedPreferences.getInstance();
    return prefs.getString(_themeKey);
  }
}
```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

Інтеграція модулів-розширень, таких як `image_picker` та `image_cropper`, дозволила додати функціонал завантаження фотографій користувача (наприклад, для аватара) або власних нотних матеріалів безпосередньо з галереї або камери пристрою. Логіка використання цих плагінів включає виклик їхніх методів для вибору та обрізки зображення, а потім його збереження у файловій системі пристрою.

Візуальна частина функціоналу, тобто сам інтерфейс користувача, створена за допомогою багатого набору віджетів Flutter. Головна сторінка використовує Scaffold з BottomNavigationBar для легкої навігації та GridView або Column для відображення кнопок швидкого доступу до основних модулів. Модуль "Тренажер" складається з кастомних віджетів, які візуально імітують клавіатуру баяна, обробляючи натискання за допомогою GestureDetector. Модуль "Основи Гри" використовує ListView.builder для прокручуваного контенту та ExpansionTile для структуризації інформації, а Image.asset застосовується для вбудованих зображень схем та ілюстрацій. Модулі "Пісні / Музичні Твори" та "База Акордів" також активно використовують Image.asset для відображення нотних матеріалів та схем акордів. Нарешті, модуль "Календар та Прогрес" інтегрує плагін table_calendar для відображення календаря занять. Загалом, архітектура програмного коду додатка ґрунтується на принципах чистої архітектури та реактивного керування станом. Це забезпечує чітке розділення між даними, бізнес-логікою та представленням, що дозволяє легко додавати новий функціонал, ефективно підтримувати існуючий та забезпечувати високу продуктивність та зручність використання мобільного тренажера.

3.3 Тестування додатку

Тестування є критично важливою частиною процесу розробки програмного забезпечення, що забезпечує якість, стабільність та коректну роботу мобільного додатку "Bayan Trainer". Воно допомагає виявляти та виправляти помилки, забезпечує відповідність вимогам, підвищує якість коду

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

та покращує загальну надійність та продуктивність програми. Комплексний підхід до тестування дозволяє виявити та усунути дефекти на ранніх етапах розробки, гарантуючи, що продукт відповідає очікуванням користувачів та поставленим бізнес-цілям. Процес тестування "Bayan Trainer" буде охоплювати кілька рівнів та типів, щоб забезпечити його всебічну функціональність та зручність. Існує кілька ключових типів тестування, які будуть застосовані. Це включає модульне тестування, інтеграційне тестування, системне тестування та перевірку відповідності вимогам. Тестування може проводитися вручну або бути автоматизованим за допомогою різних інструментів та фреймворків, що є критично важливим для забезпечення високої якості програмного забезпечення та задоволення потреб користувачів.

Модульне тестування (Unit Testing) зосереджується на перевірці окремих функціональних блоків або компонентів програми. Метою є ізолювання та тестування кожного модуля на предмет правильності його поведінки, що дозволяє виявити будь-які дефекти до їх поширення на інші частини системи. Для "Bayan Trainer" це включатиме перевірку логіки, що відповідає за відтворення звуків нот, роботу внутрішніх алгоритмів тренажера (наприклад, оцінка точності відтворення), коректність форматування часу або дати, а також логіку роботи з даними моделей, таких як Song та Lesson.

Інтеграційне тестування (Integration Testing) перевіряє взаємодію та правильну інтеграцію різних модулів та компонентів системи. Його мета – виявити проблеми, пов'язані з інтерфейсами та взаємодією між модулями, які не були виявлені під час модульного тестування. У контексті "Bayan Trainer", це може бути перевірка того, як "Тренажер" взаємодіє з модулем відтворення

звуку, або як збереження налаштувань за допомогою `shared_preferences` впливає на зовнішній вигляд та поведінку додатку. Також це тестування взаємодії між різними екранами, наприклад, перехід від списку пісень до екрану деталей пісні та коректне завантаження її контенту.

Системне тестування (System Testing) оцінює повну інтегровану систему для перевірки відповідності вимогам, правильної інтеграції компонентів і

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

забезпечення загальної функціональності системи. Для "Bayan Trainer" це означає перевірку загальної функціональності програми як цілісного продукту: від процесу запуску до коректної роботи всіх основних розділів – "Головна", "Пісні", "Тренажер", "Календар", "Налаштування" та "Профіль".

Вимоги до тестування визначають критерії та умови, за яких система вважається успішно протестованою. Вони охоплюють різні аспекти, такі як функціональність, продуктивність, безпека, доступність та сумісність. Для "Bayan Trainer" ці вимоги включатимуть чітке визначення того, як повинні працювати ключові функції, наприклад, розпізнавання нот у тренажері, точність відтворення аудіо, коректність відображення нотних аркушів та стабільність роботи на різних пристроях. Навігація та доступ до розділів. Перевіримо, чи коректно працює основне нижнє меню (BottomNavigationBar), дозволяючи легко перемикатися між "Головна", "Пісні", "Тренажер", "Календар", "Налаштування" та "Профіль". Кожен розділ повинен завантажуватися без затримок та відображати відповідний контент.

Функціонал "Основи Гри" (рис 3.11). Перевіримо перехід до екрану "Основи Гри" з головної сторінки та коректне відображення його вмісту, що забезпечує доступ до базових уроків гри на баяні.

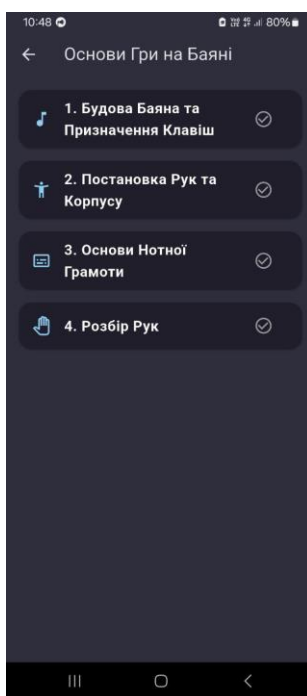


Рисунок 3.11 – Розділ основи гри відкривається успішно

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

Перегляд та відтворення пісень. На екрані "Пісні" перевіriamo відображення списку доступних пісень, їх назв та авторів. Після вибору пісні, додаток повинен коректно переходити на сторінку деталей пісні, де має бути можливість переглядати ноти (якщо доступні) та відтворювати аудіозапис. Ми перевіriamo функції відтворення, паузи та зупинки аудіо. Результати цього тестування зображено на рисунках 3.12 та 3.13.



Рисунок 3.12 – Перегляд пісні

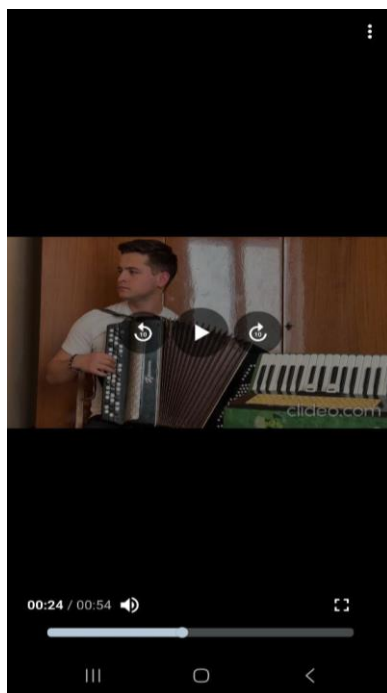


Рисунок 3.13 – Кадр з відео уроку пісні

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Використання "Тренажера". Хоча функціонал тренажера ще в розробці, ми перевіримо його запуск та відображення базового інтерфейсу. У майбутньому це буде включати перевірку розпізнавання зіграних нот та надання зворотного зв'язку користувачу.

Взаємодія з "Календарем". На сторінці "Календар" перевіримо коректність відображення календаря та можливість вибору дат.

Важливо перевірити, чи відображаються події (якщо такі будуть додані) для вибраних днів, а також чи функціонує навігація по місяцях та роках.

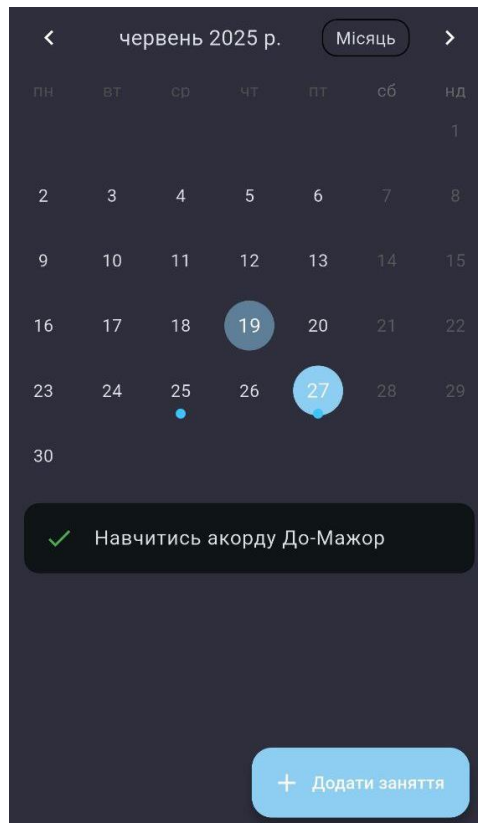


Рисунок 3.14 – Успішне тестування сторінки календар

Налаштування додатку. На екрані "Налаштування" перевіримо функціональність зміни теми (світла/темна) та переконаємося, що обрані налаштування зберігаються після перезапуску додатку за допомогою `shared_preferences`.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

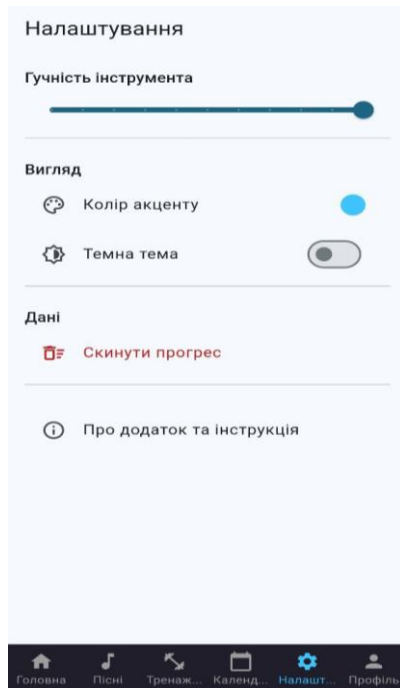


Рисунок 3.15 – Успішне тестування налаштувань

Екран "Прогрес". Перевіряємо чи працює функція збереження прогресу, за допомогою створених галочок для відмітки виконання завдання чи перегляду матеріалу. Зробимо це на прикладі розділу «База Акордів». Результати зображено на рисунку 3.16 та 3.17.



Рисунок 3.16 – Відмічення усіх переглянутих акордів

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

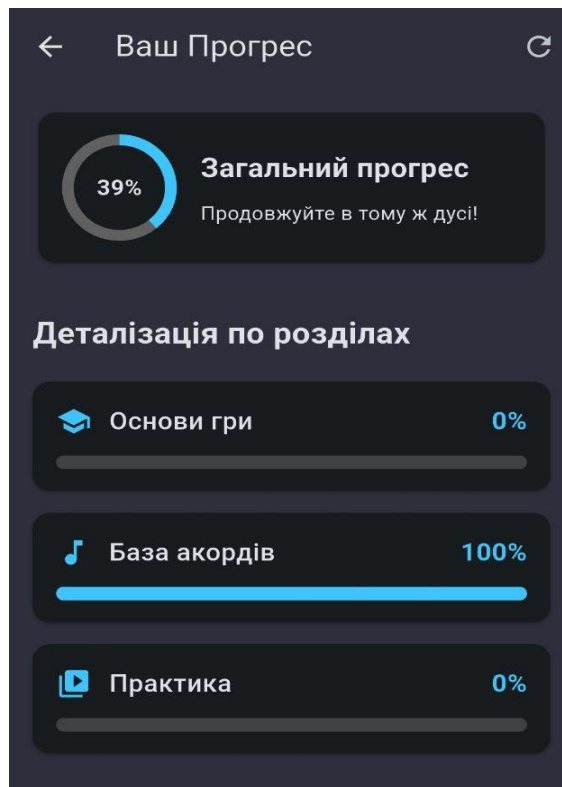


Рисунок 3.17 – Результат у прогресі

Під час тестування були перевірені різні функціональні можливості додатку, включаючи навігацію, доступ до основних розділів, відтворення музичних композицій та відображення їх нот, а також збереження користувацьких налаштувань. Усі перевірені функції працювали належним чином, відповідно до очікувань та вимог, демонструючи базову стабільність та функціональність "Bayan Trainer". Загалом, результати тестування показали, що додаток функціонує коректно та відповідає встановленим вимогам на поточному етапі розробки.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

4.1 Аналіз ринку

Мобільний тренажер з основ гри на баяні є інноваційним навчальним програмним продуктом, який розроблено як мобільний застосунок для платформ Android. Його функціональні можливості охоплюють навчальні курси, інтерактивні вправи, відеоуроки, базу акордів, а також модулі для тренування ритміки та техніки гри. Програму спроектовано з урахуванням потреб як новачків, так і досвідчених користувачів, що забезпечується адаптивністю контенту, наявністю системи відстеження прогресу та інтуїтивно зрозумілим інтерфейсом українською та англійською мовами. Завдяки малій вазі застосунку (приблизно 100 МБ) та можливості автономної роботи після завантаження контенту, він є доступним для широкого кола користувачів.

Розроблений продукт є абсолютно новим на ринку, оскільки спеціалізованих мобільних тренажерів для навчання гри на баяні практично не існує. Водночас на ринку присутні універсальні музичні додатки, які не охоплюють специфіку гри на народних інструментах. Тому запропонований застосунок займає вузькоспеціалізовану нішу й вирізняється своєю унікальністю.

Потенційними замовниками є учні та викладачі музичних шкіл, студенти мистецьких закладів, самоуки, фольклорні гурти, а також професійні музиканти, які прагнуть вдосконалити техніку гри на баяні. Основними ринками збуту вважаються Україна, де баян має широку популярність, а також країни Центральної та Східної Європи, зокрема Польща, Білорусь, Молдова та Балтійські держави. Крім того, великий інтерес очікується з боку діаспори в США, Канаді та Німеччині.

Попит на продукт прогнозується як стабільно позитивний, з огляду на зростання популярності дистанційного навчання та мобільних освітніх платформ. У перший рік після запуску планується залучити 10–15 тисяч

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

активних користувачів. Просування продукту здійснюватиметься переважно онлайн, через магазини додатків Google Play та App Store, із використанням моделі freemium. Передбачено запуск рекламних кампаній у соціальних мережах, співпрацю з музичними школами та залучення лідерів думок у музичній сфері.

Організація сервісного обслуговування включає надання технічної підтримки через email і чат, а також систематичне оновлення застосунку. Допродажне обслуговування охоплює ознайомлювальні матеріали, демонстраційні відео та інструкції. У разі успішного просування, очікується реалізація щонайменше 3000–5000 преміум-передплат протягом перших двох років.

Непрямими конкурентами продукту виступають великі платформи, такі як Yousician або Simply Piano. Вони мають сучасний дизайн, широкі функціональні можливості та популярність серед користувачів, проте не підтримують навчання гри на баяні. Вартість підписки на подібні платформи становить від 5 до 15 доларів на місяць. У порівнянні з ними мобільний тренажер з основ гри на баяні має перевагу завдяки спеціалізації, нижчій вартості, підтримці української мови та простому інтерфейсу.

Очікуваний життєвий цикл продукту становить близько 5–7 років. Протягом цього періоду планується регулярне оновлення функціоналу, розширення навчального контенту та додавання нових мов інтерфейсу. Серед основних конкурентних переваг застосунку — якість навчального матеріалу, технічна стабільність, гарантія кваліфікованої підтримки та лояльна цінова політика, яка включає можливість знижок і бонусів для постійних користувачів. У сукупності ці фактори свідчать про доцільність виведення продукту на ринок і високий потенціал його комерційного успіху.

4.2 Розрахунок витрат на проектування

Розробка мобільного додатку «Мобільний тренажер з основ гри на баяні» потребує ретельного планування ресурсів, як людських, так і

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

матеріально-технічних. Для забезпечення якісного, стабільного та функціонального програмного продукту необхідно проаналізувати всі етапи створення додатку та відповідні їм витрати. Основними складовими загальної вартості є: заробітна плата команди розробників, обов'язкові соціальні внески, вартість обладнання та програмного забезпечення, витрати на тестування й розгортання, накладні витрати, резерви на непередбачені витрати, а також податки.

Однією з найбільших статей витрат є заробітна плата фахівців, залучених до розробки додатку. До команди входять: провідний Android-розробник, два Android-розробники, веб-розробник, UI/UX-дизайнер, аналітик вимог та тестувальник. Загальна сума витрат на оплату праці протягом 6 місяців становить 450 000 грн. Згідно з чинним законодавством, додатково нараховується єдиний соціальний внесок у розмірі 22%, що становить ще 99 000 грн.

Для забезпечення ефективної роботи команди необхідно закупити ноутбуки, тестові пристрої, периферійне обладнання, а також ліцензійне програмне забезпечення, середовища розробки, інструменти проєктування та тестування. Сумарна вартість цієї статті витрат становить 90 000 грн.

Невід'ємною частиною проєктування є тестування застосунку та його розгортання в офіційних магазинах додатків, зокрема Google Play Store. Ці витрати становлять 15% від загального фонду оплати праці, що дорівнює 67 500 грн. Додаткові прямі витрати включають витрати на зв'язок, інтернет, оренду офісу, канцелярські товари, навчання персоналу та інші супутні витрати. Вони оцінюються як 25% від зарплати, тобто 112 500 грн.

Усього прямих витрат, включаючи зарплати, ЄСВ, технічне забезпечення, тестування та супутні витрати, становить 819 000 грн. До цієї суми додаються накладні витрати — адміністративні та офісні витрати, пов'язані з організацією роботи над проєктом. Вони оцінені в 286 650 грн.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Щоб убезпечити реалізацію проєкту від непередбачених ситуацій (наприклад, затримок, зміни в технічному завданні, додаткових витрат на сервіси чи ліцензії), передбачено планові накопичення у розмірі 276 412 грн.

Таким чином, кошторисна вартість розробки додатку «Мобільний тренажер з основ гри на баяні» становить:

450000 (зарплата)+99000 (ЄСВ)+90000 (обладнання)+67500 (тестування)+112500 (інші прямі витрати)+286650 (накладні витрати)+276412 (резерв)= 1 382 062.

Відповідно до податкового законодавства, до цієї суми додається податок на додану вартість (ПДВ) у розмірі 20%, тобто 276 412 грн.

Отже, загальна договірна ціна розробки мобільного додатку становить:

$1\ 382\ 062 + 276\ 412 = 1\ 658\ 474$ грн
 $1\ 382\ 062 + 276\ 412 = 1\ 658\ 474$ грн

Ця сума є остаточною і включає повний спектр витрат, необхідних для успішної реалізації проєкту: проєктування, розробку, тестування, розгортання, забезпечення стабільної роботи та адаптацію до потреб користувачів.

Таблиця 4.1 – Загальна сума витрат

№	Стаття витрат	Сума (грн)	Примітки
1	Заробітна плата команди	450 000	Розробники, дизайнер, тестувальник,
2	Єдиний соціальний внесок (22%)	99 000	Від зарплати
3	Обладнання та програмне забезпечення	90 000	Ноутбуки, телефони, ліцензії,
4	Тестування та розгортання (15% від ЗП)	67 500	Google Play, QA, автоматизація
5	Інші прямі витрати (25% від ЗП)	112 500	Оренда, зв'язок, канцелярія
6	Накладні витрати	286 650	Адміністративні витрати, комунальні послуги
7	Планові накопичення	276 412	Резерв на непередбачені витрати

Продовження таблиці 4.1:

8	Кошторисна вартість проекту	1 382 062	Загальна сума без ПДВ
9	ПДВ (20%)	276 412	Податок на додану вартість
10	Загальна договірنا ціна розробки	1 658 474	Повна сума інвестицій з урахуванням податків

4.3 Обґрунтування необхідності розробки

Сучасні тенденції в освіті та культурі свідчать про зростаючу потребу в цифрових засобах навчання, які є доступними, мобільними та адаптованими до індивідуальних потреб користувача. Однією з актуальних проблем у музичній освіті є обмежений доступ до якісного навчання гри на народних інструментах, зокрема баяні, особливо в сільських регіонах або в умовах відсутності музичних шкіл. Саме тому створення мобільного додатку «Мобільний тренажер з основ гри на баяні» є повністю обґрунтованим як з економічної, так і з соціально-культурної точки зору.

Додаток дозволить користувачам у зручній та інтерактивній формі навчатися грі на баяні без потреби в постійних заняттях із викладачем. Це забезпечує економію коштів, оскільки користувачі отримають доступ до повного курсу навчання за значно нижчою вартістю, ніж при відвідуванні індивідуальних занять. Крім того, застосунок забезпечить економію часу завдяки автоматизованому плануванню уроків, самоперевірці результатів та адаптації складності до рівня користувача.

Окрему увагу слід приділити персоналізованому підходу, який закладений у функціонал додатку: навчальні модулі побудовані таким чином, щоб користувач міг самостійно обирати темп, складність і стиль навчання. Це позитивно впливає на мотивацію, залучення до навчального процесу та дозволяє досягати кращих результатів у коротші терміни.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Поліпшення фізичного і психологічного стану також є важливим аспектом. Гра на музичному інструменті сприяє розвитку моторики, концентрації, знижує стрес і розвиває креативність. Таким чином, користування додатком позитивно впливає на якість життя.

З технічної точки зору, реалізація такого програмного забезпечення дозволяє скоротити навантаження на викладачів у закладах освіти, а також розширити охоплення музичної освіти серед населення, незалежно від географічного положення чи фінансових можливостей.

Також важливим аргументом є перспективи монетизації. Завдяки наявності платної преміум-версії, додаток здатен генерувати стабільний дохід. Розширені функції можуть включати доступ до відеоуроків, онлайн-консультацій, інтеграцію з електронним щоденником прогресу тощо. Також реалізація функцій «Поділитися з другом», система заохочень за запрошення нових користувачів, зв'язок із соціальними мережами створюють умови для вірусного поширення продукту.

Таким чином, розробка додатку «Мобільний тренажер з основ гри на баяні» є не лише економічно доцільною, але й суспільно значущою, оскільки вона дозволяє:

- зробити музичну освіту доступною та ефективною;
- зберігати й популяризувати народні інструменти;
- впроваджувати сучасні технології у мистецьку сферу;
- забезпечити джерело доходу від цифрового продукту з довготривалим життєвим циклом.

Проект має високий потенціал не лише з точки зору освітньої цінності, але й як комерційно успішне рішення, здатне зайняти свою нішу на ринку мобільних навчальних застосунків.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було успішно розроблено мобільний додаток "Мобільний тренажер з основ гри на баяні", що відповідає поставленим завданням та цілям. Проведений аналіз предметної області підтвердив актуальність створення такого рішення, враховуючи обмежену кількість якісних інструментів для самостійного навчання гри на баяні. Розроблений додаток забезпечує інтерактивний підхід до засвоєння базових навичок гри, пропонуючи користувачам структуровані уроки, практичні вправи та можливість відстеження власного прогресу.

З технічної точки зору, реалізація додатку дозволяє ефективно використовувати функціонал сучасних мобільних платформ, надаючи зручний та інтуїтивно зрозумілий інтерфейс для широкого кола користувачів. Соціальна значущість проекту полягає у забезпеченні доступності музичної освіти незалежно від географічного розташування чи фінансових можливостей, а також у сприянні збереженню та популяризації українського народного інструменту – баяна. Економічна доцільність розробки обґрунтована потенціалом монетизації через преміум-функції, що забезпечує довготривалий життєвий цикл продукту та дозволяє генерувати стабільний дохід.

Таким чином, мобільний тренажер з основ гри на баяні є не лише інноваційним освітнім інструментом, але й має значний потенціал для комерційного успіху та подальшого розвитку. Подальші перспективи включають розширення бібліотеки уроків та пісень, впровадження нових інтерактивних функцій, можливість онлайн-консультацій з викладачами, а також інтеграцію з соціальними мережами для заохочення користувачів та вірусного поширення продукту.

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Офіційна документація фреймворку Flutter. Flutter.dev : вебсайт. URL: <https://flutter.dev> (дата звернення: 15.02.2025).
2. Офіційна документація мови програмування Dart. Dart.dev : вебсайт. URL: <https://dart.dev> (дата звернення: 18.02.2025).
3. Репозиторій пакетів для Flutter та Dart. Pub.dev : вебсайт. URL: <https://pub.dev> (дата звернення: 21.02.2025).
4. Освітня музична платформа Yousician. Yousician : вебсайт. URL: <https://yousician.com> (дата звернення: 25.02.2025).
5. Мобільний додаток Accordion Piano. Google Play : вебсайт. URL: <https://play.google.com/store/apps/details?id=com.bilkon.accordion> (дата звернення: 01.03.2025).
6. Інструмент для проєктування інтерфейсів Figma. Figma : вебсайт. URL: <https://www.figma.com> (дата звернення: 18.03.2025).
7. Набір звукових семплів фортепіано у форматі MP3. GitHub : вебсайт. URL: <https://github.com/fuhton/piano-mp3> (дата звернення: 21.03.2025).
8. Відеоматеріали для розділу "Практика". GitHub : вебсайт. URL: <https://github.com/Shchotchuk11/bayan-videos> (дата звернення: 26.03.2025).
9. Документація пакета provider для керування станом. Pub.dev : вебсайт. URL: <https://pub.dev/packages/provider> (дата звернення: 04.04.2025).
10. Документація пакета audioplayers для відтворення звуку. Pub.dev : вебсайт. URL: <https://pub.dev/packages/audioplayers> (дата звернення: 12.04.2025).

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50


```

MaterialPageRoute(builder: (_) => const ChordsPage())),
    ),
    const SizedBox(height: 20),
    _buildMenuButton(
      text: 'ПРАКТИКА',
      onPressed: () => Navigator.push(context,
MaterialPageRoute(builder: (_) => const PracticePage())),
    ),
    const SizedBox(height: 20),
    _buildMenuButton(
      text: 'ПРОГРЕС',
      onPressed: () => Navigator.push(context,
MaterialPageRoute(builder: (_) => const ProgressPage())),
    ),
  ],
),
),
);
}

// Віджет для створення кнопок точно як на вашому дизайні
Widget _buildMenuButton({required String text, required
VoidCallback onPressed}) {
  return ElevatedButton(
    onPressed: onPressed,
    style: ElevatedButton.styleFrom(
      backgroundColor: const Color(0xFFE94E61), // Ваш
червоний колір
      padding: const EdgeInsets.symmetric(vertical: 20),
      shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(12)),
      elevation: 5,
    ),
    child: Text(
      text,
      style: const TextStyle(
        color: Colors.white,
        fontSize: 18,
        fontWeight: FontWeight.bold,
        letterSpacing: 1.2,
      ),
    ),
  );
}
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

Додаток Б

Лістинг програмного коду сторінки відстеження прогресу:

```
import 'package:b_trainer/basics_page.dart';
import 'package:b_trainer/practice_page.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:b_trainer/progress_service.dart';
import 'package:b_trainer/settings_provider.dart';

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

  @override
  State<ProgressPage> createState() => _ProgressPageState();
}

class _ProgressPageState extends State<ProgressPage> {
  final ProgressService _progressService = ProgressService();

  double _basicsProgress = 0.0;
  double _chordsProgress = 0.0;
  double _practiceProgress = 0.0;
  double _overallProgress = 0.0;

  final List<String> _chordsItemIds = ['chords_c', 'chords_d',
  'chords_e', 'chords_f', 'chords_g', 'chords_a', 'chords_b'];

  @override
  void initState() {
    super.initState();
    _updateProgress();
  }

  Future<void> _updateProgress() async {
    final basicsItemIds = allBasicLessons.map((lesson) =>
    lesson.id).toList();
    final practiceItemIds = allVideoLessons.map((lesson) =>
    lesson.id).toList();

    final basicsP = await
    _progressService.getCategoryProgress(basicsItemIds);
    final practiceP = await
    _progressService.getCategoryProgress(practiceItemIds);
    final chordsP = await
    _progressService.getCategoryProgress(_chordsItemIds);

    final totalItems = basicsItemIds.length +
    practiceItemIds.length + _chordsItemIds.length;
    final completedItems = (basicsP * basicsItemIds.length) +
    (practiceP * practiceItemIds.length) + (chordsP *
    _chordsItemIds.length);
```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

    if (mounted) {
      setState(() {
        _basicsProgress = basicsP;
        _practiceProgress = practiceP;
        _chordsProgress = chordsP;
        _overallProgress = totalItems > 0 ? completedItems /
totalItems : 0.0;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    final accentColor =
Provider.of<SettingsProvider>(context).accentColor;

    return Scaffold(
      appBar: AppBar(
        title: const Text('Ваш Прогрес'),
        actions: [
          IconButton(
            icon: const Icon(Icons.refresh),
            onPressed: _updateProgress,
            tooltip: 'Оновити прогрес',
          ),
        ],
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Card(
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: Row(
                  children: [
                    SizedBox(
                      width: 70, height: 70,
                      child: Stack(
                        fit: StackFit.expand,
                        children: [
                          CircularProgressIndicator(
                            value: _overallProgress,
                            strokeWidth: 8,
                            backgroundColor: Colors.grey[700],
                            valueColor:
AlwaysStoppedAnimation<Color>(accentColor),
                          ),
                          Center(child:
Text('${(_overallProgress * 100).toStringAsFixed(0)}%', style:
const TextStyle(fontSize: 16, fontWeight: FontWeight.bold))),

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

        ],
      ),
    ),
    const SizedBox(width: 20),
    Expanded(
      child: Column(
        crossAxisAlignment:
CrossAxisAlignment.start,
        children: [
          const Text('Загальний прогрес', style:
TextStyle(fontSize: 20, fontWeight: FontWeight.bold)),
          const SizedBox(height: 8),
          Text('Продовжуйте в тому ж дусі!',
style: TextStyle(color:
Theme.of(context).textTheme.bodySmall?.color)),
        ],
      ),
    ),
  ],
),
),
const SizedBox(height: 32),
const Text('Деталізація по розділах', style:
TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),
const SizedBox(height: 20),

    _buildProgressTile(title: 'Основи гри', icon:
Icons.school, progress: _basicsProgress, accentColor:
accentColor),
    _buildProgressTile(title: 'База акордів', icon:
Icons.music_note, progress: _chordsProgress, accentColor:
accentColor),
    _buildProgressTile(title: 'Практика', icon:
Icons.video_library, progress: _practiceProgress, accentColor:
accentColor),
  ],
),
);
}

```

```

Widget _buildProgressTile({required String title, required
IconData icon, required double progress, required Color
accentColor}) {
  return Card(
    margin: const EdgeInsets.only(bottom: 16),
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Row(

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

        children: [
          Icon(icon, color: accentColor, size: 24),
          const SizedBox(width: 12),
          Text(title, style: const TextStyle(fontSize: 18,
fontWeight: FontWeight.w500)),
          const Spacer(),
          Text('${(progress * 100).toStringAsFixed(0)}%',
style: TextStyle(color: accentColor, fontSize: 18, fontWeight:
FontWeight.bold)),
        ],
      ),
      const SizedBox(height: 12),
      ClipRRect(
        borderRadius: BorderRadius.circular(10),
        child: LinearProgressIndicator(
          value: progress,
          minHeight: 10,
          backgroundColor: Colors.grey.withOpacity(0.3),
          valueColor:
AlwaysStoppedAnimation<Color>(accentColor),
        ),
      ),
    ],
  ),
);
}
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Додаток В

Лістинг програмного коду віртуального тренажера

```
import 'dart:io';
import 'package:b_trainer/settings_provider.dart';
import 'package:flutter/material.dart';
import 'package:audioplayers/audioplayers.dart';
import 'package:file_picker/file_picker.dart';
import 'package:path_provider/path_provider.dart';
import 'package:path/path.dart' as p;
import 'package:provider/provider.dart';
import 'package:shared_preferences/shared_preferences.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Віртуальний Баян'),
      ),
      body: SingleChildScrollView(
        // >>> ОСЬ ТУТ ВИПРАВЛЕННЯ: Додано горизонтальну
прокрутку <<<
        scrollDirection: Axis.horizontal,
        child: Center(
          child: Padding(
            padding: const EdgeInsets.symmetric(vertical: 20.0,
horizontal: 10.0),
            child: Container(
              padding: const EdgeInsets.all(12.0),
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  begin: Alignment.topLeft,
                  end: Alignment.bottomRight,
                  colors: [
                    const Color(0xFF4A4A58),
                    const Color(0xFF2E2E3C),
                  ],
                ),
              borderRadius: BorderRadius.circular(20),
              border: Border.all(color:
Colors.black.withOpacity(0.5), width: 2),
              boxShadow: const [
                BoxShadow(
                  color: Colors.black54,
                  blurRadius: 15,
                  offset: Offset(0, 5),
                )
              ],
            ),
          ),
        ),
      ),
    ),
  ),
);
```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

```

        child: Row(
          mainAxisAlignment: MainAxisAlignment.min,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            _buildLeftHandChords(),
            _buildBellows(),
            _buildRightHandKeyboard(),
          ],
        ),
      ),
    ),
  ),
);
}

Widget _buildLeftHandChords() {
  const majorChords = ['F', 'C', 'G', 'D', 'A', 'E', 'B'];
  const minorChords = ['Fm', 'Cm', 'Gm', 'Dm', 'Am', 'Em',
'Bm'];
  const seventhChords = ['F7', 'C7', 'G7', 'D7', 'A7', 'E7',
'B7'];

  return Container(
    padding: const EdgeInsets.all(8),
    decoration: BoxDecoration(
      color: Colors.black.withOpacity(0.2),
      borderRadius: BorderRadius.circular(12),
      border: Border.all(color:
Colors.black.withOpacity(0.4)),
    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        BayanKeyColumn(notes: majorChords, isChord: true),
        const SizedBox(width: 10),
        BayanKeyColumn(notes: minorChords, isChord: true),
        const SizedBox(width: 10),
        BayanKeyColumn(notes: seventhChords, isChord: true),
      ],
    ),
  );
}

Widget _buildRightHandKeyboard() {
  return Container(
    padding: const EdgeInsets.symmetric(horizontal: 8,
vertical: 16),
    decoration: BoxDecoration(
      color: Colors.black.withOpacity(0.2),
      borderRadius: BorderRadius.circular(12),
      border: Border.all(color:

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

Colors.black.withOpacity(0.4)),
    ),
    child: const Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        BayanKeyColumn(notes: ['B3', 'C4', 'D4', 'E4', 'F4',
'G4', 'A4', 'B4', 'C5']),
        Padding(
          padding: EdgeInsets.only(top: 25.0),
          child: BayanKeyColumn(notes: ['A#3', 'C#4', 'D#4',
'F#4', 'G#4', 'A#4'], isBlackKey: true),
        ),
        BayanKeyColumn(notes: ['B3', 'C4', 'D4', 'E4', 'F4',
'G4', 'A4', 'B4', 'C5']),
      ],
    ),
  );
}

```

```

Widget _buildBellows() {
  return Container(
    width: 40,
    margin: const EdgeInsets.symmetric(horizontal: 10,
vertical: 16),
    decoration: BoxDecoration(
      color: Colors.black,
      borderRadius: BorderRadius.circular(8),
      gradient: LinearGradient(
        begin: Alignment.topCenter,
        end: Alignment.bottomCenter,
        colors: [
          Colors.grey[800]!,
          Colors.grey[900]!,
          Colors.grey[800]!,
          Colors.grey[900]!,
          Colors.grey[800]!,
        ],
        stops: const [0.0, 0.25, 0.5, 0.75, 1.0],
      ),
      border: Border.all(color: Colors.black, width: 2),
    ),
  );
}
}

```

// --- ДОПОМІЖНІ ВІДЖЕТИ ---

```

class BayanKeyColumn extends StatelessWidget {
  final List<String> notes;
  final bool isBlackKey;
  final bool isChord;
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

const BayanKeyColumn({
  super.key,
  required this.notes,
  this.isBlackKey = false,
  this.isChord = false,
});

@override
Widget build(BuildContext context) {
  return Column(
    mainAxisAlignment: MainAxisAlignment.min,
    children: notes.map((note) {
      return BayanButton(
        noteName: note,
        isBlackKey: isBlackKey,
        isChord: isChord);
    }).toList(),
  );
}

class BayanButton extends StatefulWidget {
  final String noteName;
  final bool isBlackKey;
  final bool isChord;

  const BayanButton({
    super.key,
    required this.noteName,
    this.isBlackKey = false,
    this.isChord = false,
  });

  @override
  State<BayanButton> createState() => _BayanButtonState();
}

class _BayanButtonState extends State<BayanButton> {
  String? _customSoundPath;
  bool _isPressed = false;

  static const Map<String, List<String>> _chordMap = {
    'F': ['f3', 'a3', 'c4'], 'C': ['c4', 'e4', 'g4'], 'G':
    ['g3', 'b3', 'd4'], 'D': ['d4', 'fs4', 'a4'], 'A': ['a3', 'cs4',
    'e4'], 'E': ['e4', 'gs4', 'b4'], 'B': ['b3', 'ds4', 'fs4'],
    'Fm': ['f3', 'gs3', 'c4'], 'Cm': ['c4', 'ds4', 'g4'], 'Gm':
    ['g3', 'as3', 'd4'], 'Dm': ['d4', 'f4', 'a4'], 'Am': ['a3',
    'c4', 'e4'], 'Em': ['e4', 'g4', 'b4'], 'Bm': ['b3', 'd4',
    'fs4'],
    'F7': ['f3', 'a3', 'c4', 'ds4'], 'C7': ['c4', 'e4', 'g4',
    'as4'], 'G7': ['g3', 'b3', 'd4', 'f4'], 'D7': ['d4', 'fs4',
    'a4', 'c5'], 'A7': ['a3', 'cs4', 'e4', 'g4'], 'E7': ['e4',
    'gs4', 'b4', 'd5'], 'B7': ['b3', 'ds4', 'fs4', 'a4'],
  };
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

```

};

@Override
void initState() {
    super.initState();
    _loadCustomSound();
}

Future<void> _loadCustomSound() async {
    final prefs = await SharedPreferences.getInstance();
    setState(() {
        _customSoundPath =
prefs.getString('sound_${widget.noteName}');
    });
}

Future<void> _playSound() async {
    final settings = Provider.of<SettingsProvider>(context,
listen: false);
    final volume = settings.instrumentVolume;

    try {
        if (_customSoundPath != null &&
File(_customSoundPath!).existsSync()) {
            final player =
AudioPlayer()..setReleaseMode(ReleaseMode.release);
            await player.setVolume(volume);
            await player.play(DeviceFileSource(_customSoundPath!));
            return;
        }

        if (widget.isChord) {
            final List<String>? notesInChord =
_chordMap[widget.noteName];
            if (notesInChord != null) {
                for (final note in notesInChord) {
                    final soundFileName = '${note.replaceAll('#',
's')}}.mp3';
                    final player =
AudioPlayer()..setReleaseMode(ReleaseMode.release);
                    await player.setVolume(volume);
                    player.play(AssetSource('sounds/$soundFileName'));
                }
            }
        } else {
            final soundFileName =
'${widget.noteName.toLowerCase().replaceAll('#', 's')}}.mp3';
            final player =
AudioPlayer()..setReleaseMode(ReleaseMode.release);
            await player.setVolume(volume);
            player.play(AssetSource('sounds/$soundFileName'));
        }
    } catch (e) {

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

```

        print("Помилка відтворення для ${widget.noteName}: $e");
    }
}

Future<void> _assignSound() async {
    FilePickerResult? result = await
FilePicker.platform.pickFiles(
    type: FileType.audio,
    allowedExtensions: ['mp3', 'wav', 'm4a'],
    );

    if (result != null && result.files.single.path != null) {
        File pickedFile = File(result.files.single.path!);
        final appDir = await getApplicationDocumentsDirectory();
        final fileName = p.basename(pickedFile.path);
        final savedFile = await
pickedFile.copy('${appDir.path}/${fileName}');
        final prefs = await SharedPreferences.getInstance();
        await prefs.setString('sound_${widget.noteName}',
savedFile.path);

        await _loadCustomSound();

        if (mounted) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text('Звук для "${widget.noteName}"
оновлено!'),
                    backgroundColor: Colors.green,
                ),
            );
        }
    }
}

@override
void dispose() {
    super.dispose();
}

@override
Widget build(BuildContext context) {
    final double buttonSize = widget.isChord ? 55 :
(widget.isBlackKey ? 42 : 50);
    Color buttonColor;
    Color textColor;

    if (widget.isBlackKey) {
        buttonColor = const Color(0xFF2E2E3E);
        textColor = Colors.white70;
    } else {
        buttonColor = Colors.grey[200]!;
        textColor = Colors.black;
    }
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

```

}

if (widget.isChord) {
  textColor = Colors.black.withOpacity(0.7);
}

bool hasDot = widget.isChord && widget.noteName == 'C';

return Padding(
  padding: const EdgeInsets.all(5.0),
  child: GestureDetector(
    onTapDown: (_) {
      setState(() => _isPressed = true);
      _playSound();
    },
    onTapUp: (_) => setState(() => _isPressed = false),
    onTapCancel: () => setState(() => _isPressed = false),
    onLongPress: _assignSound,
    child: AnimatedContainer(
      duration: const Duration(milliseconds: 100),
      transform: _isPressed ?
(Matrix4.identity()..translate(1.0, 1.0)..scale(0.95)) :
Matrix4.identity(),
      width: buttonSize,
      height: buttonSize,
      decoration: BoxDecoration(
        gradient: RadialGradient(
          colors: _isPressed
            ? [buttonColor.withOpacity(0.8), buttonColor]
            : [Colors.white.withOpacity(0.5),
buttonColor],
        center: const Alignment(-0.5, -0.5),
        radius: 1.0,
      ),
      shape: BoxShape.circle,
      border: Border.all(color:
Colors.black.withOpacity(0.4), width: 1.5),
      boxShadow: _isPressed ? [] : [
        BoxShadow(
          color: Colors.black.withOpacity(0.5),
          offset: const Offset(2, 4),
          blurRadius: 5,
          spreadRadius: 1,
        ),
      ],
    ),
    child: Center(
      child: widget.isChord
        ? Text(
          widget.noteName,
          style: TextStyle(color: textColor, fontWeight:
FontWeight.bold, fontSize: 18),
        )

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

```

        : (hasDot
        ? Container(
width: 12, height: 12,
decoration: BoxDecoration(
  color: Colors.grey[400],
  shape: BoxShape.circle,
  border: Border.all(color:
Colors.black.withOpacity(0.2)),
        ),
      ),
    ),
    : null),
  ),
),
);
}
}

```

					КР.КН 25.608.23.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64