

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
Кафедра української мови та прикладної лінгвістики

**СТВОРЕННЯ ІГРОВОГО ДОДАТКА ДЛЯ ВИВЧЕННЯ
ЛЕКСИЧНОГО ЗАПАСУ УКРАЇНСЬКОЇ ТА ІНОЗЕМНИХ МОВ**

**Кваліфікаційна робота
бакалавра**

студента IV курсу

ОПП «Прикладна
(комп'ютерна) лінгвістика та
англійська мова», спеціальності
035 «Філологія»,

спеціалізації 035.10

«Прикладна лінгвістика»,
галузі знань 03 «Гуманітарні
науки»

Романа ШИДЕЯ

Науковий керівник:

к. техн. н., доц. кафедри
інформаційних систем та
технологій

Микола КОСТІКОВ

«Допущено до захисту»

Протокол засідання кафедри

української мови та прикладної лінгвістики

від 28.05.2025 № 15

Завідувач кафедри _____ **Сергій РІЗНИК**

Анотація

У роботі розглянуто перспективи використання електронних засобів навчання та автоматизованих навчальних систем у контексті вивчення лексики української та іноземних мов. Об'єктом дослідження виступає процес опанування лексичного матеріалу за допомогою сучасних цифрових технологій, а предметом – методи реалізації ігрового додатку із застосуванням гейміфікації, багатомовного інтерфейсу та адаптивного налаштування параметрів користувача. Метою дослідження є створення функціонального програмного інструменту, що поєднує освітню ефективність із мотиваційною складовою гри. У процесі виконання дослідження застосовано методи аналізу, індукції, порівняння, моделювання, підбору та анкетування. На основі опитування визначено рівень зацікавленості цільової аудиторії у розроблюваному додатку. У результаті було реалізовано прототип програми з підтримкою користувацьких профілів, збереженням прогресу, вибором мови вивчення, теми інтерфейсу та розміру шрифту. Проведено аналіз проблем, що виникали на етапах підготовки лексичного матеріалу, створення структури бази даних і програмування логіки інтерфейсу. Структура роботи складається з двох розділів: теоретичного огляду існуючих АНС та практичної реалізації власного додатку. У висновках представлено оцінку ефективності реалізованої моделі та окреслено напрямки подальшого вдосконалення.

Ключові слова: гейміфікація, методи навчання, вивчення лексики, електронні засоби навчання, автоматизовані навчальні системи, персоналізація.

Abstract

This paper explores the potential of using electronic learning tools and automated learning systems in the context of studying Ukrainian and foreign language vocabulary. The object of the research is the process of acquiring lexical material through modern digital technologies, while the subject is the methods of implementing a game-based application featuring gamification, a multilingual interface, and adaptive user settings. The purpose of the study is to create a functional software tool that combines educational efficiency with the motivational component of gameplay. The research applies methods of analysis, induction, comparison, modelling, selection, and survey. A questionnaire was used to assess the interest level of the target audience in the developed application. As a result, a prototype was implemented with support for user profiles, progress tracking, language selection, theme, and font size customization. Challenges during the stages of vocabulary preparation, database structuring, and interface logic programming were identified and analyzed. The structure of the thesis consists of two sections: a theoretical review of existing ALS and a practical implementation of a custom-built application. The conclusions present an evaluation of the model's effectiveness and outline directions for further improvement.

Key words: gamification, learning methods, vocabulary learning, electronic learning tools, automated learning systems, personalization.

ЗМІСТ

ЗМІСТ.....	4
ВСТУП.....	6
РОЗДІЛ 1. ОГЛЯД СУЧАСНИХ ЕЛЕКТРОННИХ ЗАСОБІВ НАВЧАННЯ, АВТОМАТИЗОВАНИХ НАВЧАЛЬНИХ СИСТЕМ ТА ЗАГАЛЬНИХ МЕТОДИК ДЛЯ ВИВЧЕННЯ МОВ.....	8
1.1 Переваги та недоліки АНС і ЕЗН.....	8
1.2 Порівняння різних АНС для вивчення мов (функції, відгуки, статистика).....	10
1.3 Ефективність гейміфікації у процесі вивчення іноземної мови....	14
1.4 Аналіз інших методів у процесі вивчення іноземної мови.....	17
Висновки до розділу 1.....	20
РОЗДІЛ 2. СТВОРЕННЯ ВЛАСНОЇ ПРОГРАМИ ДЛЯ ВИВЧЕННЯ ЛЕКСИКИ.....	21
2.1 Розробка концепту.....	21
2.2 Підготовка словників.....	24
2.3 Написання коду додатку.....	31
2.4 Аналіз проблем і помилок.....	42
2.5 Ідеї для вдосконалення.....	45
Висновки до розділу 2.....	46
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ.....	52
Додаток А.....	52
Додаток Б.....	52
Додаток В.....	52

ВСТУП

Із розвитком технологій та все більшим проникненням їх у повсякденне життя пересічної людини, з'являється велика кількість можливостей для використання сучасних розробок для особистого розвитку. Чудовим прикладом цього є зростання популярності, а відповідно, й кількості та якості курсів і додатків для ознайомлення із певною сферою роботи або ж для вивчення іноземних мов. Враховуючи сучасні тенденції, можна заявити, що створення ігрових додатків, які поєднують навчальну мету з елементами гри, мотивуючи користувача до регулярного повторення матеріалу, є одним із перспективних напрямів у сфері вивчення іноземних мов.

Темою дослідження є вивчення та узагальнення наукових позицій стосовно використання електронних засобів навчання (ЕЗН) та автоматизованих навчальних систем (АНС), їхніх недоліків і переваг; огляд наявних популярних АНС для вивчення іноземних мов, а також аналіз ефективності різних методів у процесі вивчення іноземної мови.

Метою дослідження є створення ігрового додатку, що забезпечує ефективне й цікаве вивчення лексики української та іноземних мов із використанням сучасних технологій та з нахилом на запам'ятовування орфографії.

Актуальність теми полягає у зацікавленні аудиторії, особливо молоді, до легкого та цікавого вивчення іноземних мов. У майбутньому до застосунку можна буде додавати більше слів та категорій, в тому числі з інших мов, що стане значно кориснішим для користувачів, які захочуть вивчати лексику та практикуватись у орфографії.

Об'єкт дослідження – процес вивчення лексичного матеріалу мови за допомогою сучасних інформаційних технологій.

Предмет дослідження – засоби та методи реалізації ігрового додатку для вивчення лексики української та іноземних мов.

Завдання дослідження наступні:

- 1) провести аналіз сучасних АНС і ЕЗН, визначити їх переваги й недоліки;
- 2) порівняти функціонал популярних мовних додатків;
- 3) проаналізувати ефективність різних методів вивчення лексики;
- 4) розробити концепцію власного ігрового додатку;
- 5) підготувати лексичну базу;
- 6) реалізувати програмну частину додатку;
- 7) протестувати додаток та оцінити його роботу;
- 8) проаналізувати виявлені помилки й запропонувати шляхи вдосконалення.

Методи, що застосовувались у дослідженні: метод аналізу, метод індукції, метод порівняння, метод моделювання, метод анкетування, метод підбору.

РОЗДІЛ 1. ОГЛЯД СУЧАСНИХ ЕЛЕКТРОННИХ ЗАСОБІВ НАВЧАННЯ, АВТОМАТИЗОВАНИХ НАВЧАЛЬНИХ СИСТЕМ ТА ЗАГАЛЬНИХ МЕТОДИК ДЛЯ ВИВЧЕННЯ МОВ

Для створення ефективного інструменту вивчення лексики важливо насамперед ознайомитися з уже наявними електронними засобами навчання та автоматизованими навчальними системами. У цьому розділі буде проаналізовано їх функціонал, принципи роботи, ефективність, відмінності між ними, а також проведено огляд уже існуючих методик для вивчення іноземних мов. Такий огляд дозволить не лише виявити сильні сторони підходів і краще розуміти їхню специфіку, а й надасть розуміння того, які шляхи в принципі можливі для розвитку навичок іноземної мови.

1.1 Переваги та недоліки АНС і ЕЗН

Для початку розглянемо, що відрізняє ЕЗН від АНС.

ЕЗН – це будь-який засіб навчання, який доступний у електронному форматі (Чернилевський, n.d.). Подібне трактування терміну надає і Положення про порядок організації проведення апробації електронних засобів призначення для загальноосвітніх закладів: “Електронні засоби навчального призначення – засоби навчання, що зберігаються на цифрових або аналогових носіях даних і відтворюються на електронному обладнанні” (Верховна Рада України, 2025).

Стосовно класифікацій та відмінностей всередині ЕЗН існує доволі багато думок. Згаданий вище Д.В. Чернилевський класифікує електронні засоби освітнього спрямування за методичним призначенням (Чернилевський, n.d.), Т.Л. Петровська та М.П. Шишкіна першочерговим критерієм для розподілу ЕЗН вважають елементи штучного інтелекту (Петровська Т.Л., n.d.), (Шишкіна, 2009), тоді як Р.П. Шевчук пропонує розглядати ЕЗН за роллю в навчальному процесі як педагогічні ЕЗН та

ЕЗН загального призначення, що в свою чергу діляться на програмні та програмно залежні (Шевчук, 2008). Додатково збір та аналіз можливих класифікацій ЕЗН проведено в роботі Д.С. Антонюка (Антонюк, 2019).

Якщо підсумувати, до ЕЗН можуть входити оцифровані версії підручників (при чому неважливо, відскановані версії чи оригінальні цифрові видання); відео-, аудіо- та фотоматеріали, які зберігаються в електронному вигляді й також можуть бути доступними через інтернет, онлайн-курси на електронних платформах; окремі тестові завдання.

Із переваг ЕЗН можемо визначити наступні:

- Доступність у будь-який час;
- Наглядність, можливість візуалізації складних понять (відео, анімації, інтерактивні схеми);
- Швидке оновлення та поширення навчального контенту;
- Зменшення потреби в паперових матеріалах;
- Підходять як для самостійного, так і для групового навчання;

Із недоліків:

- Переважно пасивне засвоєння інформації;
- Відсутність адаптації до рівня учня;
- Не забезпечують зворотного зв'язку або контролю знань;
- Не завжди доступні офлайн (якщо зберігаються у хмарі).

АНС, за визначенням В.В. Коломенської та Р.С. Пастухова, – це “комп’ютерні програми, призначені для передачі користувачам знань і розвитку у них практичних навичок з різних навчальних дисциплін” (В.В. Коломенська, Р.С. Пастухов). Вони зазвичай поєднують теоретичний матеріал, практичні завдання та можливість перевірки знань.

Л. Пархуць та С. Ясинська розуміють АНС як “узгоджену сукупність навчальних матеріалів, засобів їх розробки, зберігання, передачі і доступу до них, призначену для цілей навчання і засновану на

використанні сучасних інформаційних технологій” (Ясинська and Пархуць, n.d.).

Тож беручи до уваги всі трактування, робимо висновок, що АНС включають спеціально розроблені курси, які дають змогу не лише переглядати матеріал, а й взаємодіяти з ним, формувати індивідуальний підхід до користувача за рахунок реакцій на його дії. В той час як інтерактивність та залученість, які пропонують ЕЗН – доволі низькі, АНС адаптуються до користувача та часто містять гейміфіковані елементи, що підвищують мотивацію. До популярних АНС для вивчення мов можна віднести такі платформи, як Duolingo, Rosetta Stone, Memrise, Busuu, Drops та LingQ (Busuu, 2025; Drops, 2025; Duolingo, 2025; LingQ, 2025; Memrise, 2025; Rosetta Stone, 2025).

До переваг АНС входять:

- індивідуалізація навчання (адаптація до рівня учня);
- можливість автоматичної перевірки знань і статистики успішності;
- інтерактивність, активне залучення користувача;
- регулярне підкріплення знань через вправи, тести, нагадування та в цілому більший нахил до інтервального повторення;

Тоді як до недоліків:

- в багатьох випадках потреба в стабільному інтернет-з'єднанні;
- функції чи матеріали, які доступні лише з платною підпискою.

В цілому АНС значно переважають ЕЗН за зручністю використання і сильно полегшують навчальний процес, тому деякі їхні недоліки зазвичай не надто впливають на загальну ефективність.

1.2 Порівняння різних АНС для вивчення мов (функції, відгуки, статистика)

Першою в списку є платформа **Duolingo**, одна з найпопулярніших у світі для вивчення іноземних мов. Вона надає широкий вибір курсів для вивчення (більше 40 мов, проте з обмеженням із середнім або низьким максимумом рівня володіння); залучає своїми гейміфікованими

елементами, як-от щоденний страйк, очки досвіду та проходження рівнів на зірки; дає можливість проходити конкретні типи уроків на один тип діяльності (щоправда, це доступно лише користувачам із платною підпискою). Окремим плюсом варто виокремити те, що вивчення мов доступне не лише англомовним користувачам, деякі мови мають доступ до кількох мовних курсів без посередництва англійської. За рейтингами застосунок має 4,5 зірки у Google Play (33 млн. відгуків) та 4,7 у App Store (4,1 млн.). Користувачі зазвичай хвалять кількість безкоштовного контенту та оригінальні нагадування про уроки, втім є й критика щодо зміни розгалуженої системи тем на лінійну, малої кількості уваги до граматики і політики компанії стосовно використання ІІІ. За залученістю MAU (monthly active users) складає більше 120 млн у першому кварталі 2025 року, а DAU (daily active users) зросли з 21,4 млн до 34,1 млн за останній рік (Curry, n.d.).

Далі переходимо до **Rosetta Stone**. Ця платформа заснована ще в 1992 році та є своєрідним ветераном на ринку. Вона може похвалитись хорошою структурою курсів, можливістю записатися на групові чи індивідуальні заняття із наставником, та пропонує метод повного занурення, де значення слів та виразів потрібно відгадувати самому за допомогою картинок з асоціаціями. Однак доступ до курсу з однієї мови на 3 місяці обійдеться в 15\$ на місяць, а якщо користувач хоче отримати доступ надовше і одразу до кількох мов, існує опція купити пожиттєву підписку за 219\$ із відкриттям курсів із усіх 25 мов, які є доступними для вивчення. За рейтингами застосунок має 4,7 зірки у Google Play (близько 400 тис. відгуків) та 4,8 у App Store (226 тис. відгуків). Із позитиву виокремлюють більш стандартний та серйозний підхід до вивчення мови, тоді як серед мінусів називають високу ціну підписки, малу кількість мов у порівнянні з іншими додатками та одноманітність вправ.

Наступним є додаток **Memrise**, який дає користувачам навчатись за допомогою флеш-карток, і також має таблиці лідерів, очки та досягнення, як і Duolingo. Головною родзинкою є можливість створення власних курсів від спільноти, які можуть покривати лексику та теми, що недоступні у програмному курсі. За рейтингами застосунок має 4,5 зірки у Google Play (більше 1,5 млн. відгуків) та 4,8 у App Store (205 тис. відгуків). Позитив у відгуках в основному фокусується на ефективності вправ для вивчення лексики, проте для практики розмовних навичок самого лише додатку не вистачить.

Після цього проаналізуємо **Busuu** – платформу, яка в першу чергу відома за супроводом великої кількості лексики аудіо- та відеозаписами, зробленими носіями мови. Користування безкоштовною версією дозволяє вивчати лише одну мову одночасно, а деякі уроки потребують платної підписки. За рейтингами застосунок має 4,7 зірки у Google Play (більше 940 тис. відгуків) та 4,7 у App Store (більше 90 тис. відгуків). Із хороших сторін платформи виокремлюють нотатки з інформацією про культуру країни та якість контенту, в той час із негативних – надто швидкий темп занять для початківців і недостатня гнучкість у налаштуванні тривалості уроків (Smarter Language, 2023).

Drops зроблена засновниками **Kahoot!**, популярного сайту і додатку для створення та проходження тестів. Програма використовує мінімалістичні ілюстрації, прості ігри та контекстуальне навчання. За рейтингами застосунок має 4,7 зірки у Google Play (більше 300 тис. відгуків) та 4,7 у App Store (майже 70 тис. відгуків). Основна критика припадає на обмеження у 5 хвилин на день у безкоштовній версії, нерівномірність повторень деяких слів, а також оманливий маркетинг (додаток рекламує можливість вивчити мову через нього, тоді як він призначений лише для вивчення лексики) (Richards, 2022). Втім, схоже, що привабливий дизайн, офлайн-режим та широкий вибір мов перекривають зазначені недоліки.

І останньою для аналізу обрано платформу **LingQ**. Вона має статистику прогресу, та рекламує можливість перетворити власноруч обраний контент на уроки. За рейтингами застосунок має 4,8 зірки у Google Play (майже 40 тис. відгуків) та 4,7 у App Store (трохи більше 7 тис. відгуків). Застосунок хвалять за гнучкість, багатство автентичних матеріалів і зручність інтерактивного рідера для закріплення лексики, але критикують за відсутність уваги до граматики (за винятком окремого невеликого посібника для кожної мови), погіршення інтерфейсу в оновленнях та практично відсутність безкоштовної версії (у ній можна додати до словника лише 20 слів та імпортувати 5 текстів) (Vliet, 2020).

Структуроване та підсумоване порівняння АНС можна розглянути на таблиці 2.1.

Таблиця 2.1

	Duolingo	Rosetta Stone	Memrise	Busuu	Drops	LingQ
Кількість мов	43+	25	35	14	50+	23
Гейміфікація	+	—	+	+	+	—
ІІІ функції	+	+	+	+	—	—
Режим офлайн	+(платний)	+	+	+	+(преміум)	+
Модель ціноутворення	Умовно-безкоштовна	Підписка	Умовно-безкоштовна	Умовно-безкоштовна	Умовно-безкоштовна	Підписка
Оцінки на Google Play	4,5	4,7	4,5	4,7	4,7	4,8
Оцінки на App Store	4,7	4,8	4,8	4,7	4,7	4,7
Особливі риси	Система страйку,	Метод повного	Можливість	Зроблені	Контекстуал	Навчання на

	сильна гейміфікація	занурення	створити власні курси	носіям і записи	ьне навчання	власних матеріалів
--	------------------------	-----------	-----------------------	-----------------	--------------	--------------------

Порівняння АНС для вивчення мов

Враховуючи переваги та недоліки, про які вказано в огляді до кожної АНС, можна зробити такі висновки:

- свобода вибору теми для вивчення та контенту, на основі якого можна навчатись, дає можливість персоналізувати навчання та врахувати індивідуальні вподобання та потреби більшості користувачів;
- гейміфікація елементів заохочує користувачів проводити більше часу за навчанням, а статистика дозволяє відстежувати й аналізувати свій прогрес;
- оманливий маркетинг, а також штучні обмеження, спрямовані на спонукання до купівлі платної підписки, залишають негативні враження і можуть не виправдати очікувань.

1.3 Ефективність гейміфікації у процесі вивчення іноземної мови

Безумовно, найбільшу роль у ефективності вивчення будь-якої дисципліни, не лише іноземної мови, відіграє мотивація учня. Не маючи мотивації для зростання, усі інші обставини майже не мають значення, оскільки пасивне засвоєння інформації без бажання її застосовувати не може призвести до задовільних результатів.

Для підвищення мотивації часто використовується метод гейміфікації (або ігрофікації), що полягає у використанні ігрових механік, естетики та ігрового мислення для додаткового залучення до процесів, сприяння навчанню та вирішенню проблем. Цей підхід урізноманітнює рутинні завдання, налаштовує кращу взаємодію учня із матеріалом, цим самим стимулюючи пізнавальну активність (Pasichnyk, 2018; Тріщук, Фіголь and Волик, 2019; Kapp, 2025).

Цінність гейміфікації в тому, що ігрові елементи застосовуються у неігровому середовищі – це водночас заохочує користувача до дій і зберігає об’єкт вивчення, не позбавляючи весь процес практичного застосування. В освіті такий метод особливо корисний, тому що гра має багато спільного із навчанням, хоч це може бути неочевидно на перший погляд. В першу чергу – структура заснована на правилах та послідовному досягненні цілей. Як у грі, так і в навчанні учасник долає перешкоди й виконує завдання різного рівня складності задля досягнення результату. Кожен етап є важливим і впливає на подальший прогрес: у грі — це стратегічні рішення, в освіті – формування знань і навичок (Pasichnyk, 2018).

Для ефективного використання гейміфікації ці етапи повинні бути кроками до чітко визначеної мети (опанування якоїсь теми, досягнення певного рівня володіння мовою тощо). Після цього розподілу потрібно зайнятись проектуванням самої гри, створити правила, вигадати завдання та бонуси, які даватимуться гравцям в разі успішного виконання чи досягнення додаткових цілей. Це все стимулюватиме учнів до творчості, конкурентності та/або командної роботи. Хорошими прикладами ігрових елементів, які можна застосовувати і в навчанні: впровадження системи очок, обмеження на час для проходження уроків, винагороди за досягнення додаткових цілей.

Іноді замість перетворювати навчання на гру можна використати вже існуючу гру для ілюстрування певного матеріалу, наприклад, під час лекції з історії, дизайну, так далі. Нещодавно в заголовки новинних видань потрапив вчитель історії, який використав вільну камеру у відеогрі *Assassin's Creed: Odyssey* для того, щоб краще показати архітектуру того часу і розповісти деталі битви при Фермопілах (Harrold, 2024). Іншим випадком застосування ігор у навчанні є встановлення мови інтерфейсу й/чи озвучки тією мовою, яку вивчає студент. В цьому разі гравцеві легше буде зрозуміти незнайоме слово, ніж у тексті

підручника, адже тексти в іграх часто оточені контекстом: послідовністю реплік, тоном голосу персонажа, візуальним відображенням поточних подій, атмосферою музикою. Але хоч такий варіант і є ефективним, проте він здебільшого залежить від вибору самого учня, адже потрібно враховувати і вподобання за жанром, і віковий рейтинг, і наявність або ж відсутність мови, яка вивчається, серед тих, які присутні у грі.

На користь гейміфікації працює також факт того, що все більше людей, якщо навіть не грають у відеоігри самостійно, то принаймні оточені інформацією про них, знають про їхнє існування. Це ж саме стосується використання електронних гаджетів – зараз важко знайти когось, хто певним обсягом не користувався б смартфоном, ноутбуком чи іншим гаджетом. Зростання поширеності технологій у нашому повсякденні спонукає до опанування технічної грамотності, що в свою чергу робить інтуїтивнішим процес навчання з використанням гейміфікації та цифрових пристроїв.

Іще однією з важливих складових гейміфікації є миттєвий зворотній зв'язок. У традиційному навчанні учень зазвичай повинен зачекати, доки його роботу перевірить та оцінить наставник або вчитель (якщо ми говоримо про письмові завдання). Проте створивши тест на якійсь інтерактивній платформі, такій як Kahoot!, та розіславши його учням, платформа самостійно перевірить, чи було обрано правильну відповідь, а також подасть тест у більш цікавому форматі, ніж роздрукований аркуш паперу. Отримання результату одразу після закінчення тесту зберігає рівень залученості у процес, а таблиця лідерів може дати мотивацію для удосконалення (Саган, 2022). Втім, творчі завдання досі потребують людської перевірки.

Уже згаданий Duolingo є чудовим прикладом того, як гейміфікація може ефективно працювати в освіті. Користувач встановлює собі за мету певну кількість очок досвіду, після здобуття яких він також отримує бонуси – це дає відчуття досягнення. Проходження уроку за день

подовжує страйк користувачу, проте в разі, якщо уроку не зроблено, страйк обнуляється – і це утримує від того, щоб перестати користуватись додатком. Іноді Duolingo пропонує користувачеві додаткові цілі, як продовження страйку ще певну кількість днів, обгін інших користувачів його ліги за кількістю очок досвіду, за що також видають нагороди у вигляді модифікаторів очок досвіду – (це є прикладом орієнтованості на розваги та конкуренцію) (Shortt et al., 2023).

Варто пам'ятати і про недоліки гейміфікації. Надмірне захоплення ігровими елементами може зашкодити подачі та поясненню навчального матеріалу, і тоді важче перейти до серйознішого вивчення теми, з'являється ризик формування легковажного ставлення до дисципліни. Може бути складно включити до однієї гри учнів чи студентів із різним рівнем підготовки. Також при використанні гаджетів збільшується час, який учні проводять за екраном (Chyzhykova and Tokarieva, 2021).

Отже, гейміфікація у вивченні іноземних мов виступає потужним інструментом підвищення мотивації та залучення учнів до навчального процесу. Вона сприяє глибшому засвоєнню матеріалу завдяки елементам гри, миттєвому зворотному зв'язку та емоційному залученню, однак потребує зваженого впровадження, аби уникнути надмірної ігрової складової.

1.4 Аналіз інших методів у процесі вивчення іноземної мови

Окрім процесів для залучення учнів до навчальної діяльності, варто пам'ятати і про методики, які використовуються для ефективнішого вивчення матеріалу.

Як узагальнили у своїй роботі О.В. Іванова та О.Р. Журба, в основі викладання закладено один із двох методів: традиційний та нетрадиційний, який включає в себе активні методи. Вони вже поділяються за спрямованістю на неімітаційні та імітаційні, де останні складаються з неігрових та ігрових. Імітаційні методи навчають професійним умінням та навичкам та за своїм принципом моделюють

певну ситуацію. Відмінність між ігровими та неігровими методами полягає в тому, що у перших учні розігрують певні ролі у змодельованій ситуації, а в других – аналізують ситуацію та вирішують її задачі (Іванова, Журба, 2013).

І. Голубєва також розділяє методики для вивчення іноземних мов на традиційні та новітні. До традиційних авторка зараховує класичну, пряму, граматико- та текстуально-лексико-перекладну, аудіовізуальну і аудіолінгвальну, комунікативну та лінгвосоціокультурну. До інноваційних: комп'ютерну підтримку навчання мов, метод сценарію та симуляцій, рольові та ділові ігри, метод «case study», метод навчальних станцій, метод проєктів, мозковий штурм (Holubieva, 2023). Варто зазначити, що жоден із цих методів не є повністю автономним, вони більше зосереджені на вивченні якогось окремого аспекту мови, і дають найкращий результат при застосуванні одразу кількох із них одночасно.

Дуже ефективним при вивченні мови є ефект занурення – коли учень оточує себе мовою, яку він вивчає, вводячи користування нею у повсякденні речі. Це може бути як перехід на споживання контенту цією мовою, так і зміна мови інтерфейсу окремих програм чи пристрою, так і спілкування цією мовою, або ж усе одразу. Занурення може бути частковим або повним. Зазвичай повне занурення в мовне середовище відчують туристи та мігранти, і хоч воно швидко дає результати – з ним пов'язано кілька ризиків.

Першим є складність або навіть неможливість встановити бодай мінімальний контакт із співрозмовником через лексичні непорозуміння, що часто провокує виникнення психологічних бар'єрів. Такі бар'єри можуть суттєво ускладнити комунікацію з носіями іншої мови та спричинити серйозні наслідки в подальшому (Nalyvaiko, Nuralieva and Syvolotska, 2021).

Не варто виключати культурний шок – якщо учень до цього не мав достатньо інформації про культуру країни, мову якої він вивчає, у

свідомості може виникнути конфлікт між старими та новими культурними нормами від перенасичення новим середовищем. Тому соціокультурному аспекту вивчення мови слід приділяти достатньо уваги, щоб принаймні пом'якшити негативні сторони повного занурення.

Замість занурення учень може спробувати повільніший, проте не менш ефективний спосіб вивчення мови – інтервальне повторення. Ідея полягає в тому, що при вивченні чогось невідомого знання залишається в короткостроковій пам'яті, але щоб відкласти його в довгострокову пам'ять, потрібно повторити його певну кількість разів. Втім, повторення варто проводити не одразу, а протягом деякого часу із перервами. Таким чином, якщо інформація буде періодично виринати на поверхню, мозок з більшою ймовірністю запам'ятає її. Це, в свою чергу, призводить до того, що в ситуації, коли ці знання потрібно використати, їх набагато легше пригадати (Noor et al., 2021; Белінська, 2023; Центр фінансово-економічних наукових досліджень, n.d.). Що важливо, ця методика заохочується практично в усіх АНС для вивчення іноземних мов за рахунок системи страйку – користувача заохочують повертатись у додаток щодня, а отже – не забувати вивчений матеріал.

Як варіант, для допомоги у вивченні іноземної зараз також можна залучати штучний інтелект – він може посприяти персоналізації навчання, аналізувати мову користувача, адаптувати матеріал до його потреб і допомагає усунути прогалини в знаннях. ШІ також дозволяє ефективніше організовувати навчальний процес, оцінювати рівень знань ще до початку навчання та робити його цікавішим і продуктивнішим як для учня, так і для викладача (Зубенко, 2023). Це більше стосується саме систем ШІ, розроблених під навчання мови, хоч до інструментів для вивчення мов із допомогою ШІ також зараховують системи машинного перекладу (як Google Translate чи DeepL) і генеративні мовні системи (як ChatGPT або Gemini) (Rebolledo Font de la Vall and González Araya, n.d.).

Висновки до розділу 1

Огляд сучасних електронних засобів навчання (ЕЗН) і автоматизованих навчальних систем (АНС) показує, що кожен із підходів має свої переваги та обмеження. ЕЗН забезпечують доступність матеріалу та зручність користування, однак часто не адаптуються до рівня учня і передбачають переважно пасивне засвоєння знань. У порівнянні з ними, АНС дозволяють більш гнучкий і персоналізований підхід, активне залучення користувача, а також мають інструменти для перевірки прогресу. Гейміфіковані елементи таких систем як Duolingo чи Memrise підвищують мотивацію, залучення та регулярність занять. Огляд платформ показав, що попри спільні риси, кожна має свої унікальні функції, підходи до навчання та цільову аудиторію. Особливої уваги заслуговує використання автентичних матеріалів, можливість створення користувацького контенту та зворотний зв'язок. Аналіз ефективності гейміфікації доводить її позитивний вплив на мотивацію та активність учнів. Таким чином, поєднання гейміфікації, адаптивності й гнучкого доступу є ключем до створення результативних інструментів для вивчення мов.

РОЗДІЛ 2. СТВОРЕННЯ ВЛАСНОЇ ПРОГРАМИ ДЛЯ ВИВЧЕННЯ ЛЕКСИКИ

У цьому розділі буде показано процес створення програми для вивчення лексики української та іноземних мов, починаючи від розробки концепту та закінчуючи написанням і тестуванням коду. Також проаналізовано помилки, що виникали під час роботи, і вказано покращення, які можна зробити.

2.1 Розробка концепту

Для реалізації будь-якої ідеї потрібно мати чітке уявлення, як саме вона працюватиме. Від початку планувалося створити додаток, який допомагав би вивчати слова через складання цих слів із літер, доступних на ігровому полі. Саму концепцію ігрового поля запозичено із гри Bookworm Adventures Deluxe, випущеною студією PopCap Games у 2007 році, де гравцеві теж потрібно складати слова із доступних літер (Рис. 2.1).



Рис. 2.1 Вигляд ігрового раунду в грі “Bookworm Adventures Deluxe”.

(Джерело: <https://tinyurl.com/32pczjt>)

Зі спільних рис також варто додати існування кількох профілів користувачів, щоб була можливість використовувати навчальний додаток на одному пристрої кільком людям одночасно. З цього випливає, що буде імплементовано відслідковування та збереження прогресу.

Перша відмінність полягає у тому, що в оригіналі не існує чітко заданих слів, які гравець може скласти (точніше, список можливих варіантів дуже великий, трохи менше 300 000), а новий додаток повинен був мати відносно невеликий список слів, який визначався б усередині самої гри в залежності від тематики, яку обрав учень.

Друга відмінність – гра від PopCap заохочувала складання довших слів для отримання бонусів; в той час, як навчальний додаток зосереджувався лише на словах із уроку та не дозволяв би скласти інші слова.

Третя відмінність – для більшої гейміфікації планувалося запровадити таймер на кожен раунд в залежності від кількості та складності слів.

В результаті було розроблено концептуальний вигляд додатку, який можна побачити на Рис. 2.2 - 2.3.

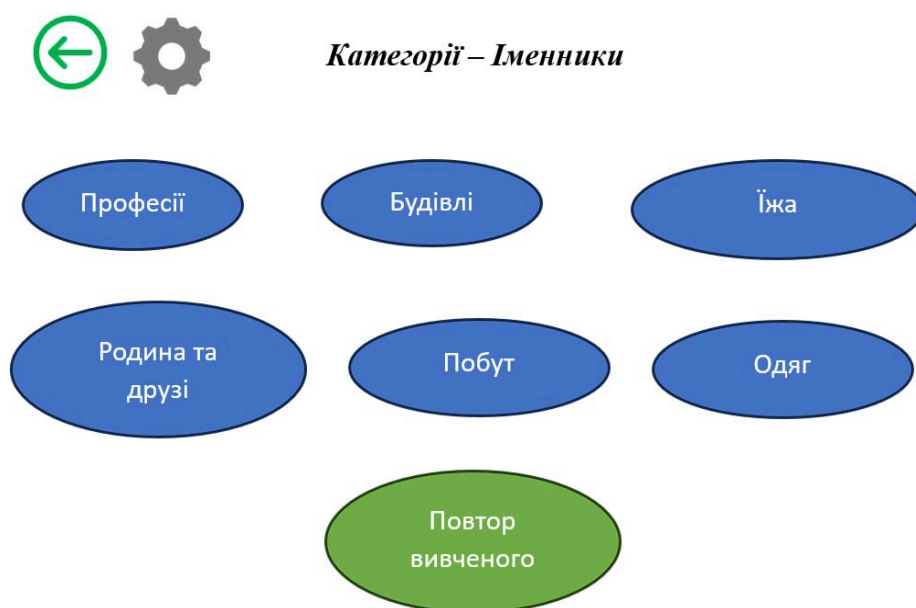


Рис. 2.2 Концепт меню категорій



Раунд I



00:27

potato ☒

картопля

carrot ☒

beet ☒

cabbage ☒

капуста

pea ☒

bean ☒

p	e	a
---	---	---

горох

c	o	r	b
t			e
b		p	n
a	e	r	t

Рис. 2.3 Концепт ігрового раунду в навчальному додатку

Додаток було вирішено розробляти мовою програмування Python3, оскільки вона добре підходить для цього завдання та має багато бібліотек, що спростять процес розробки коду. Із доступних мов для вивчення у першій версії додатку планувалося давати користувачеві вибір між українською та англійською мовами. До того ж, потрібно передбачити додавання інших мов до доступних, такі як французька, іспанська тощо.

Після створення концепту було вирішено провести анкетування за допомогою Google Forms, щоб дослідити цікавість до навчальних АНС із вивчення мов, а також дізнатись думки та вислухати пропозиції стосовно додатку для вивчення лексики через складання слів із літер на час. На Рис. 2.4 показано одну з діаграм із згаданого опитування, результати якого можна повністю переглянути у Додатку А. Цільовою аудиторією опитування переважно були студенти навчально-наукового

інституту філології КНУ ім. Т. Шевченка. Як бачимо, трохи більше третини висловились на підтримку такої ідеї, біля третини не були впевнені та ще біля третини не були зацікавлені у використанні додатку.

Чи були б ви зацікавлені у використанні додатку для вивчення лексики, який заснований на принципі побудови слів із літер на час?

76 відповідей



Рис. 2.4 Діаграма відповідей на питання про зацікавленість у додаткові

Також наприкінці форми деякі респонденти зазначили, що тематично структуровані слова були б хорошим варіантом, хорошим стимулом щось вивчати була б додаткова конкуренція, а також що ідея цікава через те, що користувач буде вчитися правильно писати слова (це може бути особливо корисним при додаванні мов, які містять багато діакритичних знаків у словах або малозрозумілі правила словозміни).

2.2 Підготовка словників

Спочатку для перевірки справжності слів планувалося використати слова із одного з уже існуючих словників – бажано двомовних, щоб одразу протестувати роботу додатку після написання коду. Для цієї цілі було знайдено словники В. Лева та І. Вербяного (Лев and Вербяний, n.d.) та М.І. Балли (Балла М.І., 1996). Однак мова розвивається швидко, і навіть за 5 років деякі тлумачення слів можуть застаріти, вже не кажучи про 20 і більше років. Тому конкретно ці словники було відкинуто, а новіших альтернатив у вільному доступі знайти не вдалося.

Наступним варіантом було використати один із паралельних корпусів на сайті OPUS (OPUS, 2025). Проте проблемою стало те, що ці

корпуси мають розподіл та вирівнювання лише за реченнями та абзацами і хоча технічно можливо виокремити окремі слова, їхній переклад та перевести їх у лемми, це було б надто трудомістким та непрактичним процесом. Через цю причину варіант із корпусом також було відкинуто.

Далі було знайдено проєкт ВЕСУМ (Великий електронний словник української мови) (GitHub, 2025), і хоч цей словник є одномовним, ідея уже зібраного та впорядкованого списку слів здалася непоганою. Втім, одразу після нього вдалося відшукати лексичний мінімум з української мови, який містить біля 1000 слів, словоформ та декілька фраз українською мовою, усі посортвані як алфавітно, так і тематично (Borodin and Turkevych, 2024).

Тепер, коли належне джерело було знайдено, залишилося підготувати його до застосування. Спочатку слова було скопійовано із розділу із сортуванням у відповідності за темами, до файлу `wordlist_ukr.txt`. Потім перекладено англійською і скопійовано переклад до файлу `wordlist_eng.txt` у такому ж порядку. Далі списки було відредаговано за такими критеріями:

1. Видалено дублікати перекладу, які зустрічались у різних частинах списків.
2. Видалено назви тематик.
3. В деяких місцях змінено подачу паралельних форм та парадигм числа й роду.

Після цього створено файл `DB_fill.py`, в якому було написано код, що створює базу даних `word_base.db` типу SQLite, та додає туди таблиці **Interface_labels**, **Ukrainian**, **English**, **Users**, **User_progress**. Спочатку таблиці **Interface_labels**, **Users** та **User_progress** не планувались, тому що написи для інтерфейсу спочатку знаходились у словнику всередині файлу із основним кодом, проте з часом стало зрозуміло, що логічніше та структурованіше буде перенести ці написи у базу даних. Таблиця

Users з'явилася після ідей про кастомізацію додатку під себе (як зміна теми, шрифту і мови інтерфейсу). Замість таблиці **User_progress** очікувалось, що якщо користувач уже зустрів слово у цій мові, інформація про це записувалася б у одній із колонок *enc_1*, *enc_2* або *enc_3*; проте створення окремої таблиці для цього здалося зручнішим варіантом. Щоб правильно спланувати, як працюватиме кожен стовпець у кожній таблиці, необхідно було переглянути документацію SQLite про типи даних (SQLite, 2025). Повну структурну схему бази даних зі зв'язками можна роздивитись на Рис. 2.5.

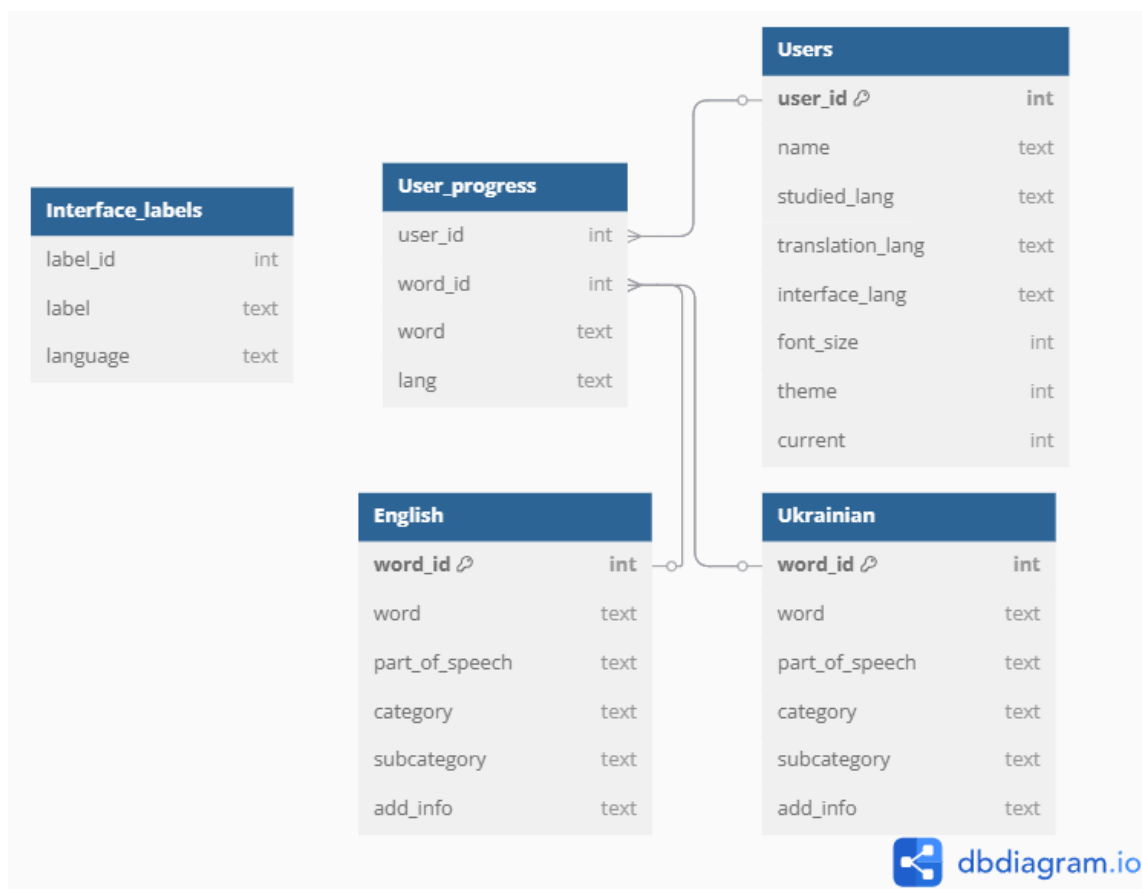


Рис. 2.5 Структурна схема залежностей бази даних *word_base.db*

Щодо структури самих таблиць:

1. Таблиці **English** та **Ukrainian** мають ідентичну побудову – стовпці *word_id*, *word*, *part_of_speech*, *category*, *subcategory*, *add_info*.
 - 1.1. Перший стовпець відповідає за порядковий номер конкретного слова, який потім використовується для того,

щоб відстежувати відомі та невідомі слова для таблиці **User_progress**.

- 1.2. Стовець *word* – запис самого слова (часом разом із паралельною формою або іншим словом, пов'язаним за парадигмою відмінків, числа або роду). Єдиний стовець, який має значення, що відрізняються між таблицями з мовами (оскільки всі інші стовпці мають залишатися з однаковими значеннями для відстеження потрібної інформації).
- 1.3. *part_of_speech* відповідає за частину мови, до якої слово належить (наприклад, NOUN, VERB, ADJ), на кількох комунікативних фразях тег має значення UNKNOWN, оскільки вони складаються із кількох слів. Для правильного трактування тегів частин мови іноді доводилось їх перевіряти за Universal POS tags (Universal Dependencies, 2025).
- 1.4. *category* – стовець, який містить число категорії навпроти кожного слова. Це необхідно для подальшої навігації всередині додатку та для правильної підготовки слів для гри. Число категорії відповідає номеру напису із таблиці **Interface_labels**, що відповідає за назву цієї категорії.
- 1.5. Стовець *subcategory* також має число підкатегорії, проте не завжди, а лише у випадках, коли категорія є надто обширною для того, щоб подавати слова з неї одним цілим. Число працює так само, як і у стовпці *category*.
- 1.6. Останній стовець містить число, яке відповідає за напис, що висвічуватиметься біля слова у раунді, якщо це слово має більше ніж одну форму чи подано одразу кілька слів, пов'язаних парадигмою числа, вінмінків чи роду. Такий формат запису слів присутній в оригінальному лексичному

мінімумові через деякі особливості, які можуть бути неочевидними для учнів, що вивчають українську мову (наприклад, чергування звуків або опущення деяких літер при словозміні).

2. Таблиця **Interface_labels** має лише три стовпці: *label_id*, *label* та *language*.

2.1. *label_id* відповідає за порядковий номер кожного напису, який потрібен для пошуку правильного значення.

2.2. *label* містить усі написи, що можуть зустрітися всередині додатку, тобто назви усіх категорій і підкатегорій, а також назви кнопок чи написів по типу “Вихід” або “Розмір шрифту”.

2.3. *language* – стовпець, у якому зберігається код мови (ukr, eng) навпроти кожного слова і який потрібен для визначення мови інтерфейсу. Кожне значення у *label*, яке має тег ukr у *language*, також продубльовано англійською із відповідним тегом eng та ідентичним *label_id*.

3. Таблиця **Users** має вісім стовпців: *user_id*, *name*, *studied_lang*, *translation_lang*, *interface_lang*, *font_size*, *theme* і *current*.

3.1. *user_id* – порядковий номер користувача.

3.2. *name* – ім’я користувача. За замовчуванням усі три можливі користувачі мають ім’я User1, User2 та User3 відповідно, проте його можна змінити всередині додатку, і тоді нове ім’я буде збережено у базу даних.

3.3. *studied_lang* – стовпець, який має відповідати за списки слів для вивчення обраною мовою. За замовчуванням стоїть ukr.

3.4. *translation_lang* – стовпець, який має відповідати за мову, якою буде відображатися переклад слів для вивчення під час гри. Існує незалежно від мови інтерфейсу для більшої гнучкості налаштувань. За замовчуванням стоїть eng.

- 3.5. *interface_lang* – стовпець, який визначає мову, якою відображаються написи інтерфейсу додатку. Можна обрати в налаштуваннях, і тоді нова мова збережеться для цього користувача. За замовчуванням стоїть *eng*.
- 3.6. *font_size* – стовпець, який зберігає значення розміру шрифту, за замовчуванням стоїть середній розмір, проте його можна змінити в налаштуваннях на малий та великий. Саме ж значення – номер напису із **Interface_labels**.
- 3.7. Стовпець *theme* відповідає за те, яку тему (світлу чи темну) обрав користувач. За замовчуванням стоїть світла, проте це можна змінити в налаштуваннях і тоді зміни буде збережено в базі даних. Оскільки кнопки теми не мають написів, значення завжди дорівнює 0 (світла тема) або 1 (темна тема).
- 3.8. *current* є останнім стовпцем і його завдання – передавати значення 1 поточному користувачеві при зміні профілю, тоді як у решти значення залишається 0. Поточний користувач може бути лише один, проте в разі, якщо ще не обрано жодного (при першому запускові, наприклад), основний код зробить поточним перший профіль у списку.
4. Таблиця **User_progress** містить чотири стовпці: *user_id*, *word_id*, *word* та *lang*.
- 4.1. Перший стовпець заповнюється значеннями, які код бере за номером поточного профілю із таблиці **Users** та зі стовпця з ідентичною назвою (*user_id*). Пізніше при перевірці слів, які знає користувач, це значення показуватиме, який саме профіль їх знає.
- 4.2. *word_id* заповнюється порядковими номерами слів, які зустрів поточний користувач. Значення беруться із стовпця з аналогічною назвою у таблиці зі списком слів (це може бути **English**, **Ukrainian** або інша таблиця), а пошук проводиться

за номером поточного профілю та тегом мови зі стовпця *studied_lang* у таблиці **Users**. При перевірці відомих слів значення у цьому стовпці показуватимуть, які слова знає користувач.

4.3. Стовпець *word* функціонує повністю так само, як і попередній (з урахуванням, що значення беруться тепер із стовпця *word* з інших таблиць), однак він не потрібен для перевірки слів. Його функція лише в тому, щоб покращити вигляд списку у самій базі даних.

4.4. *lang* – стовпець, у якому зберігається код мови для кожного знайомого слова. Значення наповнюється із стовпця *studied_lang* поточного профілю таблиці **Users**, і хоч вони самі по собі не потрібні для функціонування програми (як і значення із попереднього стовпця), проте дають більше інформації для списку в базі даних.

Перша частина файлу *DB_fill.py* містить код, що імпортує бібліотеки *sqlite3* та *stanza* і створює базу даних та таблиці у ній, якщо їх не існує. Перша бібліотека потрібна для роботи з базою даних, друга – для визначення частини мови кожного слова при наповненні таблиць. Далі йдуть списки *interface_ukr* та *interface_eng*, які містять однакову кількість елементів – усі написи, що зустрічаються всередині додатку. Після цього йдуть фрагменти коду, які виконуються лише один раз – це код, який наповнює таблиці. Після одного використання потрібного сегменту коду він виділяється та коментується (до кожного рядка додається символ “#” на початку, щоб код став неактивним), оскільки повторний запуск цього сегменту приведе до появи дублікатів у базі даних. В кінці файлу зберігаються будь-які зроблені зміни до бази даних та закривається доступ до неї. Файл *DB_fill.py*, списки слів і готова база даних доступні за посиланням у Додатку Б.

Варто зазначити, що стовпці *category* та *subcategory* у таблицях **Ukrainian** і **English** заповнювалися вручну, оскільки було вирішено внести деякі зміни у належність певних слів до кількох категорій. Наприклад, якщо в оригінальному лексичному мінімумові була категорія, що містила лише кілька слів, за належних умов ці слова прив'язувались до іншої категорії. Наприклад, слова із теми “Канцелярські товари” у кількості 5 штук було переміщено до теми “Навчання”, а тему “Засоби гігієни” зроблено як підкатегорію до теми “Побут”, і так далі.

Також вручну було створено та записано пояснення для слоту додаткової інформації у таблицях зі словами. Наприклад, “водій / водійка” потребує пояснення №68 “чоловічий та жіночий роди”, оскільки в англійській мові немає фемінітиву для цієї професії і в обох варіантах англійською слово буде перекладатися просто як “driver”. Крім того, усі слова, що потребують додаткової інформації для контексту, також були вручну перезаписані у такому форматі, щоб пізніше при роботі додатку ці слова правильно відображались та зараховувались. Наприклад, “водій / водійка” стало “водій/-ка”, “вдома, удома” стало “(в/у)дома” тощо.

2.3 Написання коду додатку

На перших стадіях написання коду найбільш доцільним вважалось використання уже знайомої бібліотеки `tkinter` із доповненим пакетом `ttk`, проте для покращення загального вигляду було вирішено перейти на бібліотеку `customtkinter`. Також на початку коду імпортуються бібліотеки `sqlite3` (робота з базою даних), `re` (регулярні функції), `random` (випадковий розподіл) та завантажується модуль `Image` із бібліотеки `PIL` (відображення картинок всередині додатку).

Спочатку було написано частину коду, яка створювала вікно додатку із назвою “Словинка” та розмірами 1260x720. Пізніше також

було додано функцію *center_window()*, яка слідкує за тим, щоб вікно завжди відкривалося по центру екрана.

Для керування інтерфейсом було впроваджено систему фреймів, що дозволяє логічно розділяти вікна налаштувань, головного меню, категорій та самої гри. У словнику із назвою *frames* зберігаються всі створені фрейми для подальшого перемикавання між ними. За створення фрейму відповідала функція *create_frame()*, проте пізніше було здійснено перехід на одноразову ініціалізацію усіх фреймів за циклом *for* одразу після створення словника з ними. Для перемикавання між фреймами створено функцію *show_frame()*, яка дозволяє навігацію по додатку.

Для підтримки багатомовного інтерфейсу створено функцію *load_interface_texts()*, яка витягує текстові значення для поточної мови з таблиці **Interface_labels**, зберігає у списку *labels* і передає їх до відповідних елементів інтерфейсу.

Після написання необхідних функцій почалися спроби наповнення меню кнопками та реалізувати перехід між різними фреймами. Кожна кнопка створювалась вручну через *ctk.CTkButton()*, поки не було вирішено злегка автоматизувати цей процес та додати функцію-обгортку *create_labeled_button()*, яка дозволяє відслідковувати фрейм та команду, які закладено в кнопку, а ще додавати окремий текст до основного. По аналогії з цією функцією було написано *create_labeled_label()*, яка працює з написами. У ході створення нової кнопки чи напису, вони додаються до списку віджетів *interface_labels* у форматі об'єктів класу *LabeledWidget*. Для того, щоб правильно налаштувати як створення об'єктів, так і подальше їхнє оновлення, дуже допомогли пояснення на навчальній платформі GeekForGeeks (GeeksforGeeks, 20:29:43+00:00; 10:57:49+00:00)

Першим фреймом, який був повністю укомплектований і залишався практично без змін протягом подальшої розробки, був фрейм *settings*, тобто віконце налаштувань. Було вирішено, що кнопка

налаштувань з'являтиметься на кожному іншому фреймі у вигляді невеликої іконки шестерні, щоб користувач міг в будь-який момент підлаштувати вигляд додатку під себе. Оскільки загальна кількість фреймів доволі значна – для полегшення процесу створено функцію `create_settings_button()`, яка запускається після того, як усі інші кнопки на фреймові вже створено та відображено. Розташування та вигляд функцій у налаштуваннях можна роздивитись на Рис. 2.6.

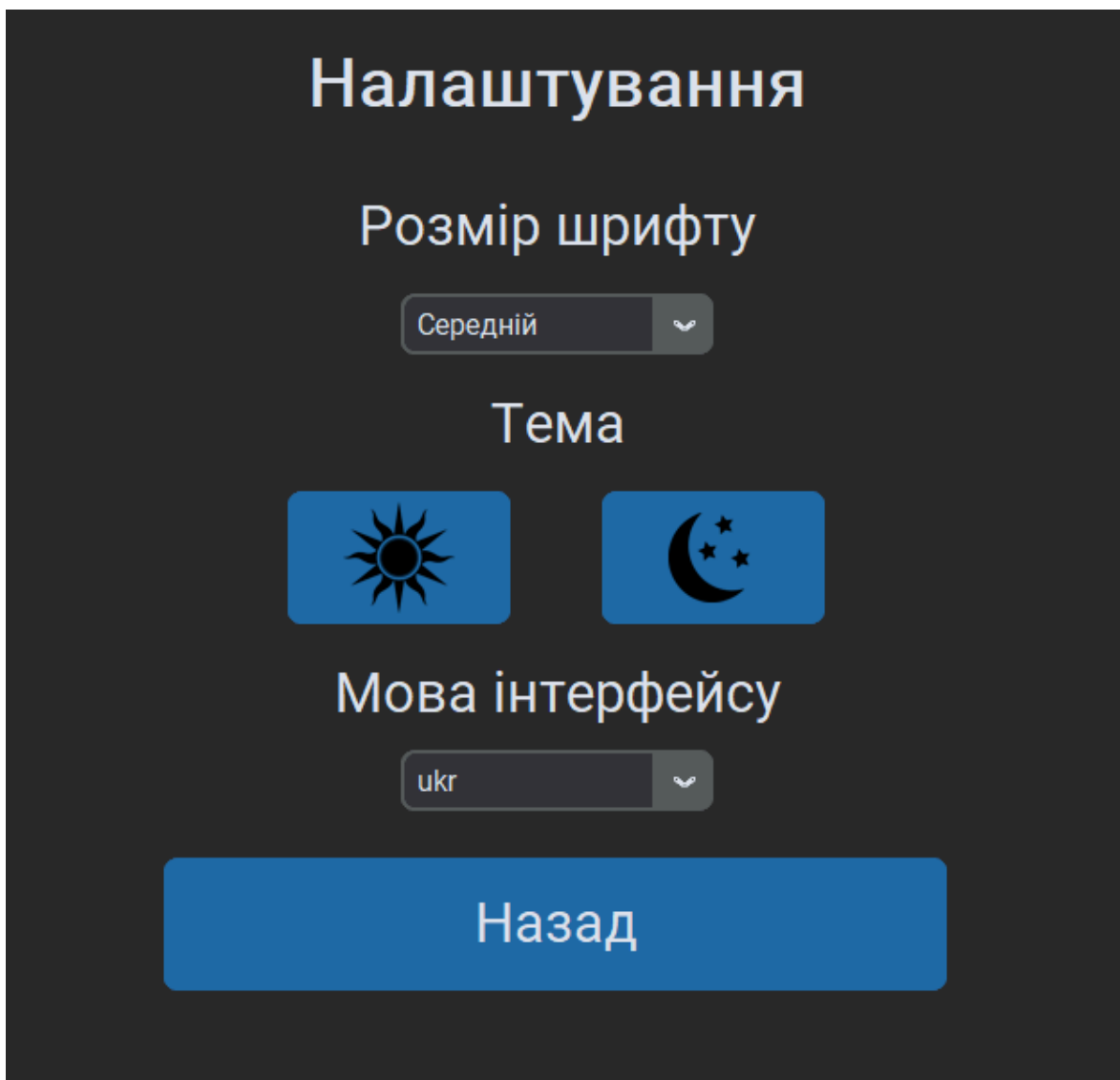


Рис. 2.6 Поточний вигляд функціоналу фрейму налаштувань із увімкненою темною темою

Першою змінною опцією у вікні налаштувань є розмір шрифту, і для правильного функціонування цієї опції існує багато змінних. Наприклад, словник `font_sizes` установлює відповідність між базовим

розміром шрифтом та текстом, який відображатиметься у спадному списку. Функція *apply_font_size()* при спрацюванні оновлює розмір всіх наявних шрифтів, а також збільшує розмір усіх кнопок, що містять текст. Кнопки з текстом мають однакові розміри, щоб вмістити найдовший напис, не переносючи його в другий рядок, і таким чином зберігаючи кращу читабельність і постійність вигляду в порівнянні з іншими кнопками.

Всього існує три шрифти: *BUTTON_FONT*, *TITLE_FONT* та *GAME_BUTTON_FONT*. Кожен із них застосовується у відповідному середовищі, наприклад, *TITLE_FONT* використовується у заголовках фреймів, *BUTTON_FONT* – на кнопках із текстом та кількох звичайних написах (як ті ж самі назви налаштувань по типу “Розмір шрифту” та “Тема”), а *GAME_BUTTON_FONT* – для відображення літер на сітці ігрового поля.

Вибір розміру шрифту надається у спадному списку *font_combo*, який пропонує при натисканні один із трьох варіантів: Малий, Середній та Великий (або Small, Medium і Big, якщо користувач обрав англійську мову інтерфейсу). При натисканні на одну з опцій спрацьовує команда *change_font()*, яка записує вибір користувача у таблицю **Users** у базі даних, а також запускає вже знайому функцію *apply_font_size()*. Причина, чому ці дві функції існують окремо одна від одної, в тому, що при запускові програми не потрібно нічого зчитувати зі спадного списку, код просто завантажує уже наявний збережений вибір, а *change_font()* потрібна саме для роботи зі спадним списком у налаштуваннях.

Наступною опцією для кастомізації вигляду додатку є вибір між світлою та темною темою. Для перемикання між ними існує дві невеликі кнопки, *light_butn* та *dark_butn*, розташовані симетрично від центру, з іконками сонця та місяця й зірок відповідно. Перша кнопка вмикає командою *toggle_theme()* світлу тему, друга – темну, функція також запам’ятовує вибір користувача та записує його у базу даних.

Відмінність між ними у тому, що із ввімненою темною темою фон стає майже чорним, а написи – білими, тоді як зі світлою темою фон світло-сірий, а написи – чорні. Кнопки та текст на них залишаються незмінними.

Далі йде зміна мови інтерфейсу. Подібно до зміни розміру шрифту, ця опція також прив'язана до спадного списку *lang_combo*, який запускає команду *change_language()*. Ця команда зберігає вибір користувача в базу даних та активує функцію *update_texts()*, яка в свою чергу замінює варіанти спадного списку *font_combo* на їхні аналоги іншою мовою, а також оновлює текст кожного іншого віджета зі списку *interface_elements*. Функції *change_language()* та *update_texts()* працюють за схожим сценарієм, як і ті, що потрібні для шрифтів, перша – для спадного списку та збереження вибору, друга – лише для підвантаження нових налаштувань і оновлення віджетів.

Останнім елементом на фреймі налаштувань є кнопка повернення на попередній фрейм, яка присутня і практично на кожному іншому фреймові та повертає користувача на попередній. Якщо попереднім фреймом було якраз вікно налаштувань, воно ігнорується. Це працює за рахунок змінної *previous_frame*, яка оновлюється при переході на інший фрейм, та функції *show_frame()*.

На головному меню, яке можна роздивитись на Рис. 2.7, видно у правому верхньому куті кнопку налаштувань, а по центру – п'ять інших віджетів. Почати варто із напису-привітання зверху. При запускові програми код шукає у таблиці **Users**, чи існує профіль, який має значення 1 у стовпці *current*. Якщо так, тоді повертається текст із стовпця *name* цього профілю та доповнює напис-привітання на головному меню як додатковий текст, що йде після основного (змінна *add_text1*). Якщо такого профілю не існує, тоді значення 1 присвоюється першому користувачеві у списку, він стає поточним, і тоді висвітлюється його ім'я.

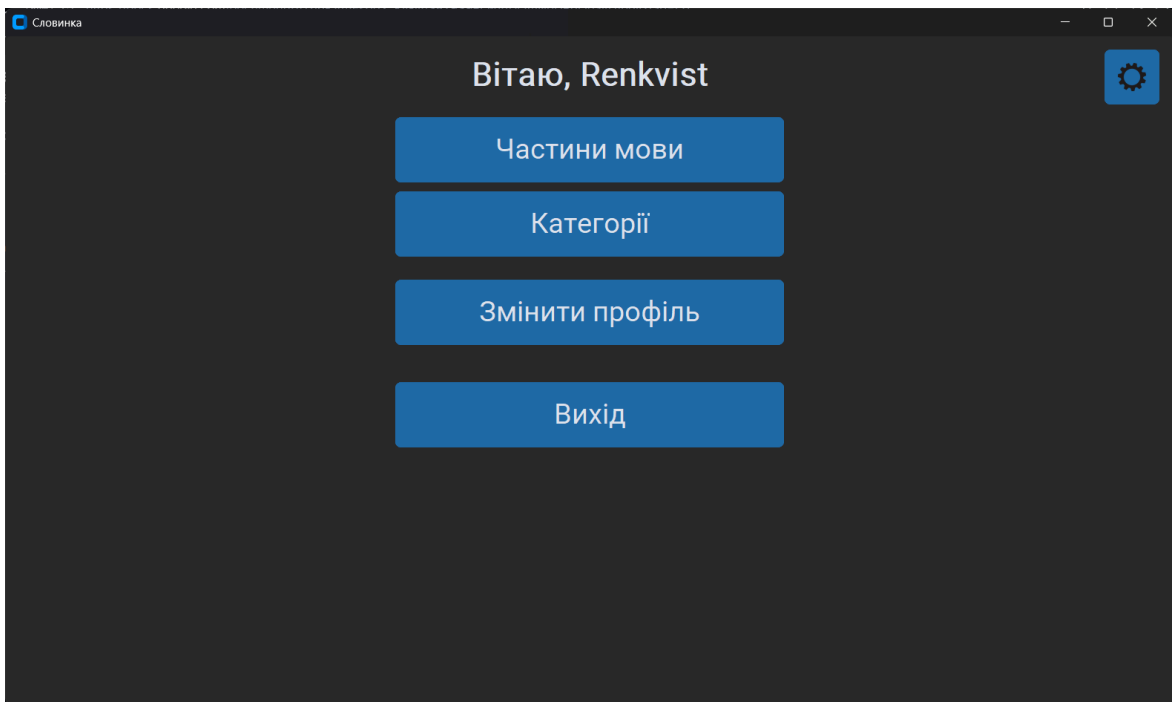


Рис. 2.7 Поточний вигляд фрейму головного меню

Кнопка виходу працює практично як і всі інші кнопки, єдина відмінність в тому, що її команда – *root.quit*, базова команда бібліотеки *customtkinter*, яка закриває вікно *root*, тобто вікно програми.

Змінити профіль можна, натиснувши на відповідну кнопку та перейшовши на фрейм *users*, де користувачеві надається вибір із трьох доступних профілів, а також можливість перейменувати профіль поточного користувача. Кожна із перших трьох кнопок користується функцією *select_user()*, яка в свою чергу запускає функцію *load_user()* та повертає користувача до меню. *load_user()* – це функція, яка визначається на самому початку файлу і при запускові зчитує дані того профілю, кнопку якого натиснув користувач, призначає цей профіль поточним і оновлює всі елементи програми за допомогою уже знайомих функцій *update_texts()* та *apply_font_size()*, а також функцію *update_users()*. Ця функція послідовно витягує імена існуючих профілів з бази даних і так само послідовно встановлює їх як написи для перших трьох кнопок на фреймі *users*. Варто зазначити, що *load_user()* один раз запускається самостійно перед стартом програми, в цьому разі замість

вибору іншого профілю код просто шукає той, якому вже присвоєне значення “1” у стовпці *current*, тобто обирається останній активний профіль перед попереднім закриттям програми. Це робить непотрібним вибір свого профілю щоразу при запускові додатка, якщо зазвичай ним користується лише одна людина на цьому пристрої.

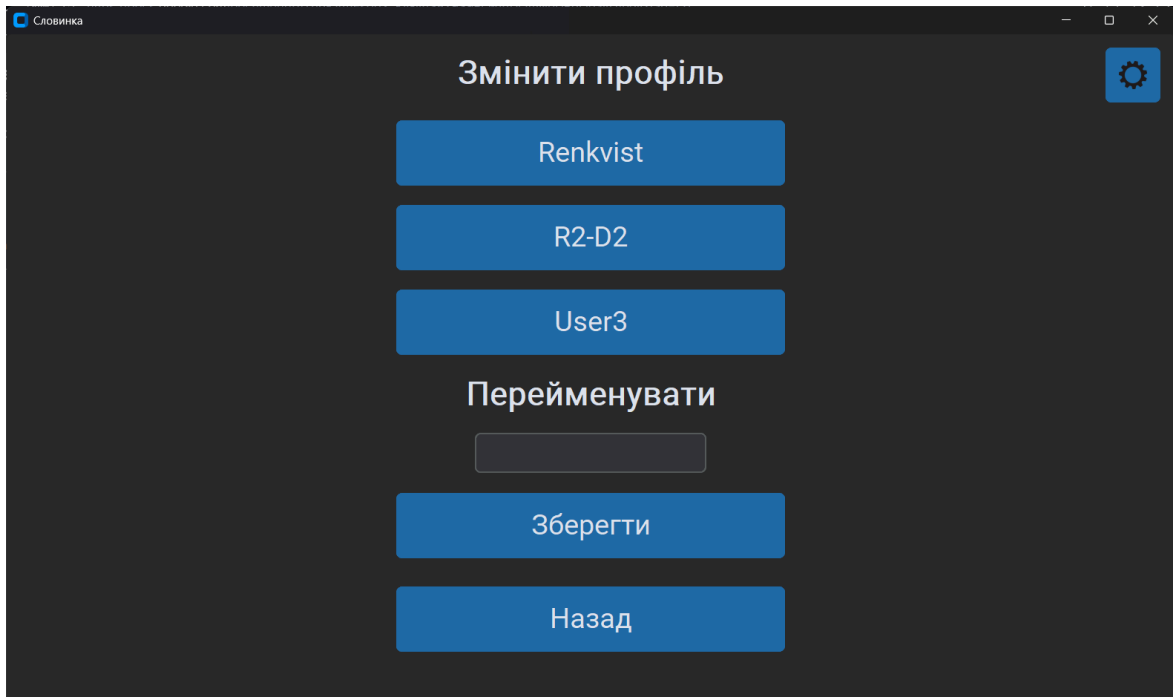


Рис. 2.8 Поточний вигляд фрейму зміни профілю

Опція перейменування доступна через віджет *entry*, який зберігає змінну *user_name_var*, яку заповнює користувач. Після того, як текст введено, потрібно натиснути кнопку нижче, вона активує функцію *rename_user()*, яка зберігає нове ім'я для поточного користувача в базу даних, очищує введений текст, а потім запускає вже знайому функцію *update_users()*, щоб одразу відобразити нове ім'я на відповідній кнопці профілю.

Повертаючись до головного меню (Рис. 2.7), далі розглянемо частини мови. Натиснувши на кнопку з цією назвою, користувач переходить на відповідний фрейм, який можна роздивитись на Рис. 2.9, і де є чотири кнопки із назвами частин мов: Іменники, Дієслова, Прикметники та Прислівники (в подальшому планується додати іще

Числівники та Службові частини мови). Кожна із цих кнопок запускає гру, проте слова для неї підбираються не за категоріями, а за тегом належної частини мови.

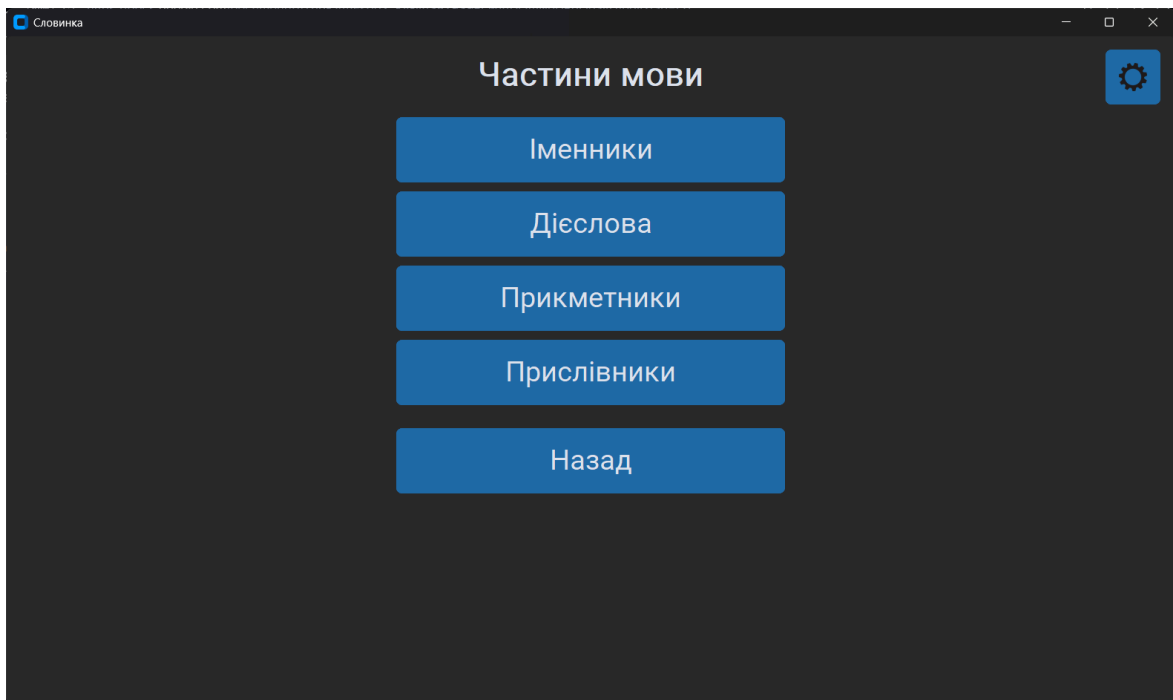


Рис. 2.9 Поточний вигляд фрейму частин мови

І останньою до розгляду на головному меню є кнопка категорій, яка переносить користувача на перший із чотирьох фреймів, адже всього категорій 24. Відповідно, щоб розмістити кожну кнопку із врахуванням максимуму у шість штук на фрейм (не враховуючи кнопку повернення), потрібно було створити аж чотири окремі фрейми. Для навігації між ними створено окремі кнопки-стрілочки, які можна побачити на Рис. 2.10. Оскільки такі кнопки доволі однотипні, було вирішено написати функцію `create_nav_button()`, яка запускається один раз для кожної окремої кнопки. Всього кнопок шість, оскільки на першому на останньому фреймові немає куди йти вліво та вправо відповідно. Також в залежності від поточного фрейму, заголовок відрізнятиметься – на кожному фреймі категорій додатковий текст показує номер фрейму: (1/4), (2/4), і так далі.

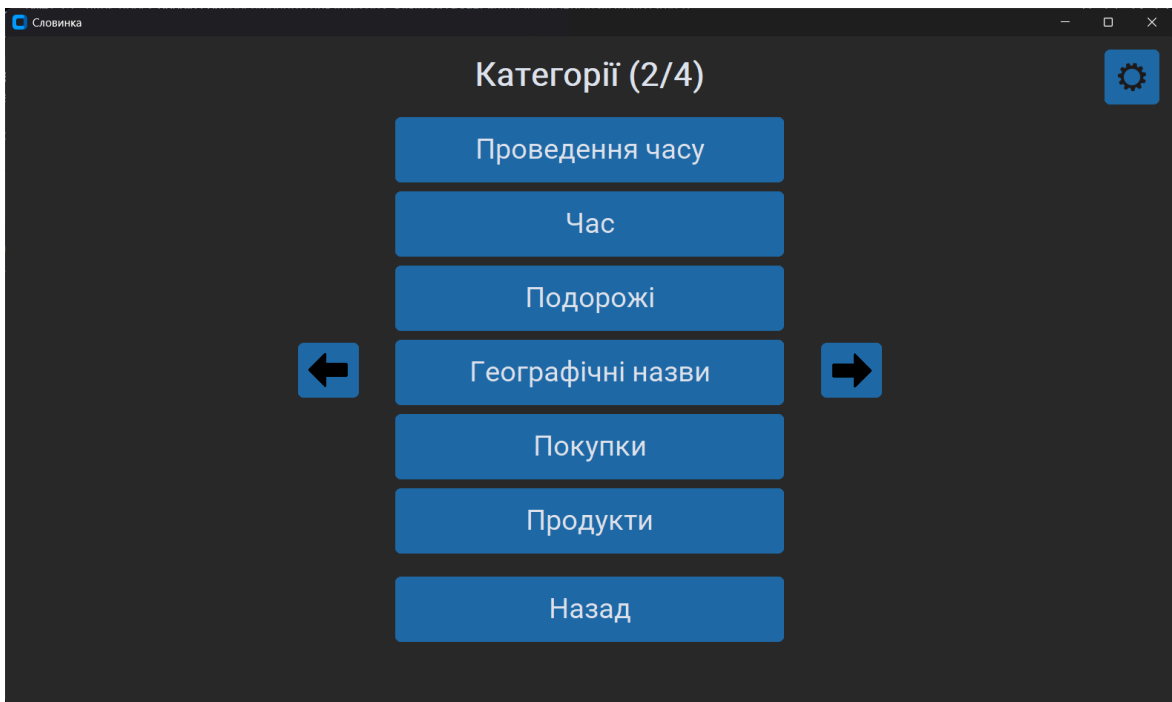


Рис. 2.10 Поточний вигляд другого фрейму категорій

Такі категорії як “Людина”, “Місто й село”, “Побут”, “Проведення часу” та “Природа” не ведуть користувача одразу до гри, а натомість показують власні фрейми із назвами *cat_13*, *cat_16*, *cat_17*, *cat_19* і *cat_29* відповідно. Це потрібно для того, щоб розмістити кнопки підкатегорій, тому що ці п’ять категорій мають забагато слів, які можна тематично згрупувати. Вигляд та функціонал кнопок підкатегорій мало чим відрізняється від решти у цій секції, тому переходимо одразу до ігрового фрейму; переглянути його структуру можна на Рис. 2.11.

Для створення та відображення написів існує функція *create_game_labels()*, яка окрім слів для вивчення та їхніх перекладів, також відповідає за заголовок ігрового фрейму, а також за кнопку початку гри та повернення. Написи зі словами створюються за циклом *for* та додаються у список *game_labels* для відслідковування. Для гри беруться шість (часом менше) слів українською, які розміщуються у 1 та 3 ряду; та їхні переклади англійською, які розміщуються в 2 та 4 ряду. Важливим моментом є те, що для сітки з написів та кожного окремого ряду з написами існує свій невидимий міні-фрейм, як і для двох кнопок

унизу. Це пов'язано з тим, що для розташування елементів використовується метод *.pack()*, і щоб добре відцентрувати ці елементи, краще це робити разом для всієї групи.

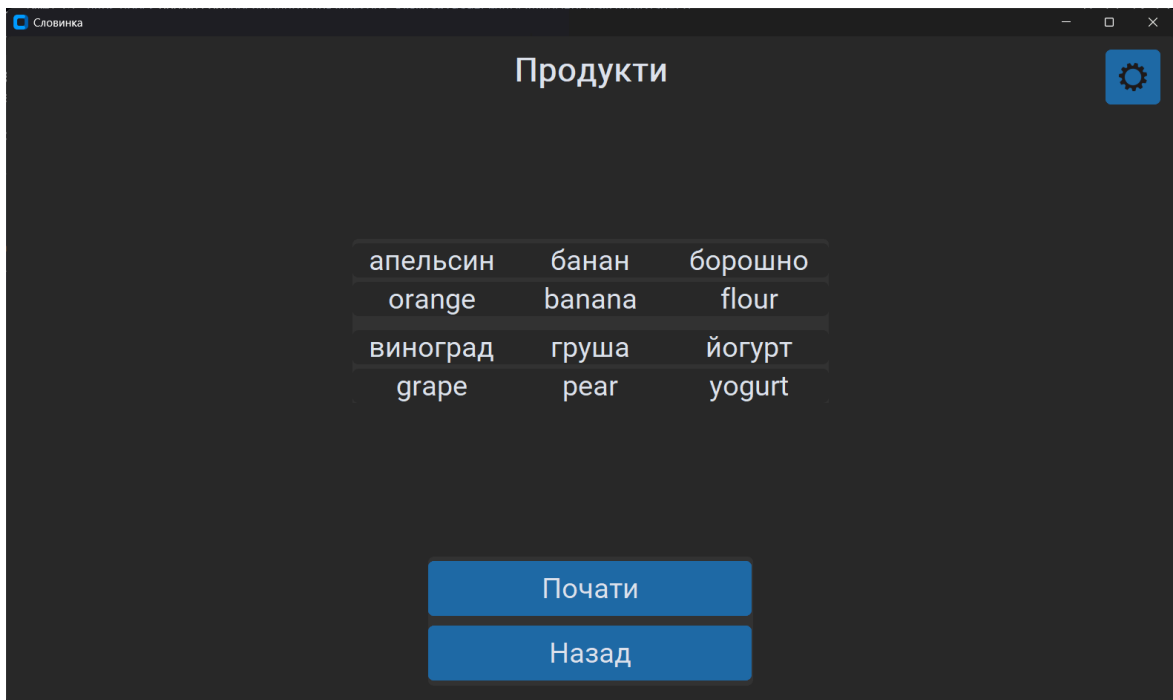


Рис. 2.11 Поточний вигляд ігрового фрейму категорії “Продукти” до початку гри

Після цього йде ініціалізація класу *GameLetterButton*, оскільки як і з класом *LabeledWidget*, його об'єкти потрібно буде оновлювати з уже існуючим текстом, а створити подібний клас – доволі вдале рішення для цього.

Далі створюється функція *generate_letter_buttons()*, при запускові якої приховуються кнопки початку гри та повернення, а натомість створюється ще один невидимий міні-фрейм, в який поміщаються 16 кнопок, згенеровані циклом *for*. На кожній кнопці відображається одна літера, а процес відбору літер для кнопок передбачає можливість для користувача в будь-який момент скласти хоча б одне слово із потрібних шести. Це працює за рахунок вибору одного слова (*chosen_word*), усі літери якого гарантовано потрапляють на поле, а решта літер добираються із інших слів у випадковому порядку. Після того, як

створено та наповнено кнопки у сітці, справа від них з'являється поле *letter_entry*, а під ним – кнопки *check_button* та *reset_button*. Поле має заблокований стан, який розблоковується лише перед введенням у нього літери із сітки, що запобігає ручному вводу літер для користувача. Перша кнопка при створенні має сірий колір і також заблокована, проте вона розблоковується, якщо користувач ввів правильне слово, таким чином сигналізуючи про можливість здати це слово. Наступна кнопка дозволяє очистити поле та повертає літери на сітці до їхнього звичайного стану. Функціонал цих та наступних елементів добре помітно на Рис. 2.12.

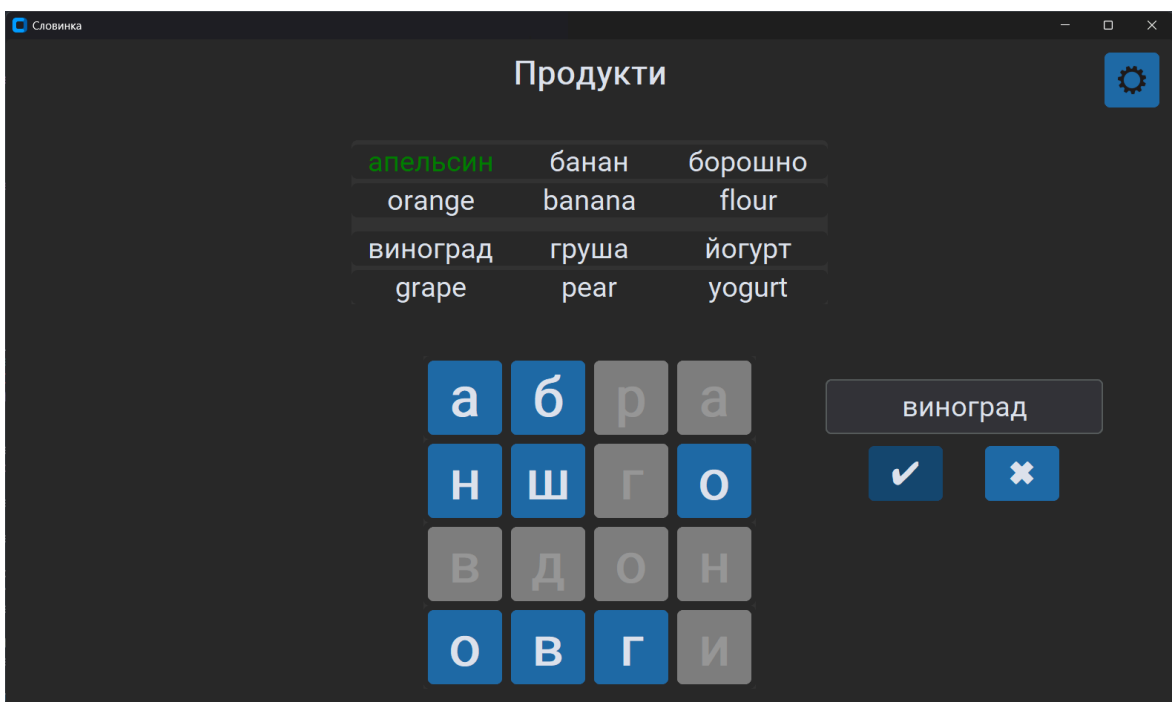


Рис. 2.12 Поточний вигляд ігрового фрейму категорії “Продукти” після успішного складення слова

При натисканні будь-якої кнопки з літерою на сітці ця кнопка стає сірого кольору та не може бути натиснута знову, що відбувається завдяки функції *on_click()* класу *GameLetterButton*. Одночасно з цим натиснута кнопка тимчасово розблоковує поле *letter_entry*, щоб додати туди свою літеру, потім знову заблоковує його, а далі запускає функцію *check_word_validity()*. Ця функція перевіряє, чи є введене на даний момент значення одним із заданих слів. Якщо так – вона розблоковує

кнопку з галочкою (*check_button*) та надає їй звичайний колір решти кнопок.

Як уже було згадано, в разі, якщо кнопка галочки доступна для натискання, користувач може здати своє слово через команду *submit_word()*, яка додає побудоване слово до списку *words_built* та видаляє його зі списку *words_left*. Далі це ж слово знаходиться серед написів над ігровою сіткою та перефарбовується у зелений колір, щоб показати користувачеві, що це слово уже знайдено. У разі, якщо це було останнє слово із вибірки, гра зупиняється. Якщо ж ні – запускається команда *field_update()*. Ця команда багато в чому схожа на *generate_letter_buttons()*, тому що вона теж обирає наступне гарантоване слово та розподіляє літери по кнопках (лише натиснутих, всі інші залишаються як є). На жаль, її не вдалося зробити універсальною та застосовувати як частину функції із створенням сітки, хоча подібну стратегію було реалізовано в багатьох інших випадках (коли функція, що застосовується першою, містить лише незначні доповнення, а потім користується функцією, яка щось оновлює). В будь-якому разі, *field_update()* після розподілу та призначення літер викликає *reset_letters()*, функцію, що видаляє всі символи з текстового поля, повертає натиснуті кнопки сітки до звичайного стану, а також вимикає кнопку галочки. Цю ж команду використовує і кнопка *reset_button*, яка потрібна для очистки тексту й відновлення натиснутих кнопок у разі, якщо користувач не може скласти слово або помилився при вводі.

Повну роботу додатка можна переглянути у відео за посиланням у Додатку В.

2.4 Аналіз проблем і помилок

Впродовж усіх стадій роботи над додатком виникали проблеми, що варіювались за своєю складністю та часом, витраченим на їх подолання.

Першою проблемою, яка виникла ще навіть до початку написання коду, є вибір джерела лексики для гри. Звісно, лексичний мінімум має

достатньо багато слів, якими українці користуються в повсякденному житті, проте якщо додавати до усіх таблиць із мовами переклад цих слів, тобто брати це джерело за основу для всієї лексики, стане зрозуміло, що частина цієї лексики навряд чи підходить для вивчення інших мов. Річ у тім, що деякі слова або належать виключно до українського культурного контексту (як “борщ”), або ж мають сильну відмінність у контексті при вживанні слова та його перекладу іншою мовою. Тому з додаванням нових мов варто або робити мовні пари, що значно ускладнить процес, або обмежуватись відносно бідним набором “однозначної” лексики.

При написанні коду довелося зустрітись із багатьма проблемами, зокрема помилками по типу `NameError`, `IndexError`, `TclError`, а також неробочими кнопками, неправильне розташування елементів і так далі.

`NameError` часто виникала, коли при використанні однієї функції вона не мала доступу до змінної, яка була створена в іншій функції та не була оголошена глобально. Зазвичай вирішувалась за допомогою рядка “`global ...`”, але іноді доводилось змінювати порядок рядків у коді або змінювати структуру функції.

`IndexError` іноді виникав, якщо код намагався отримати елемент, який не існував у словнику чи списку. Найчастіше таке траплялось, коли ще не було оновлено список із текстами інтерфейсу у базі даних, а якась кнопка чи напис вже потребували доступу до цього тексту. Дуже легко виправлялось оновленням бази даних із новими текстами.

`TclError` виникнув один раз при спробі застосувати метод розміщення `.grid()`, коли в цьому фреймі вже були елементи, розміщені за допомогою методу `.pack()`. Вирішення – переробити планування для розміщення елементів, які мали бути розміщені через `.grid()`, створити окремий невидимий міні-фрейм для них, а потім продовжити розміщувати все через `.pack()`.

`KeyError` можна було зустріти, коли тип даних не відповідав тому, який потрібен був для роботи. Наприклад, із бази даних міг зчитуватись

рядок і перед використанням цього значення як індекса для списку воно не було переведено через функцію `int()` у число. Щоб це вирішити, достатньо лише перевести значення у потрібний тип даних (найчастіше `INTEGER`).

Один раз довелося зіткнутись із `ValueError`, коли ще не було створено окремого словника для встановлення відповідності між написів, які відображались у спадному списку, та розміром шрифту. Проте після цього кроку проблема зникла.

Іноді замість помилок у коді були просто непередбачувані результати при тестуванні. Наприклад, на Рис. 2.13 можна помітити, як розташувались елементи на ігровому фреймі перш ніж було добре налаштовано геометрію.

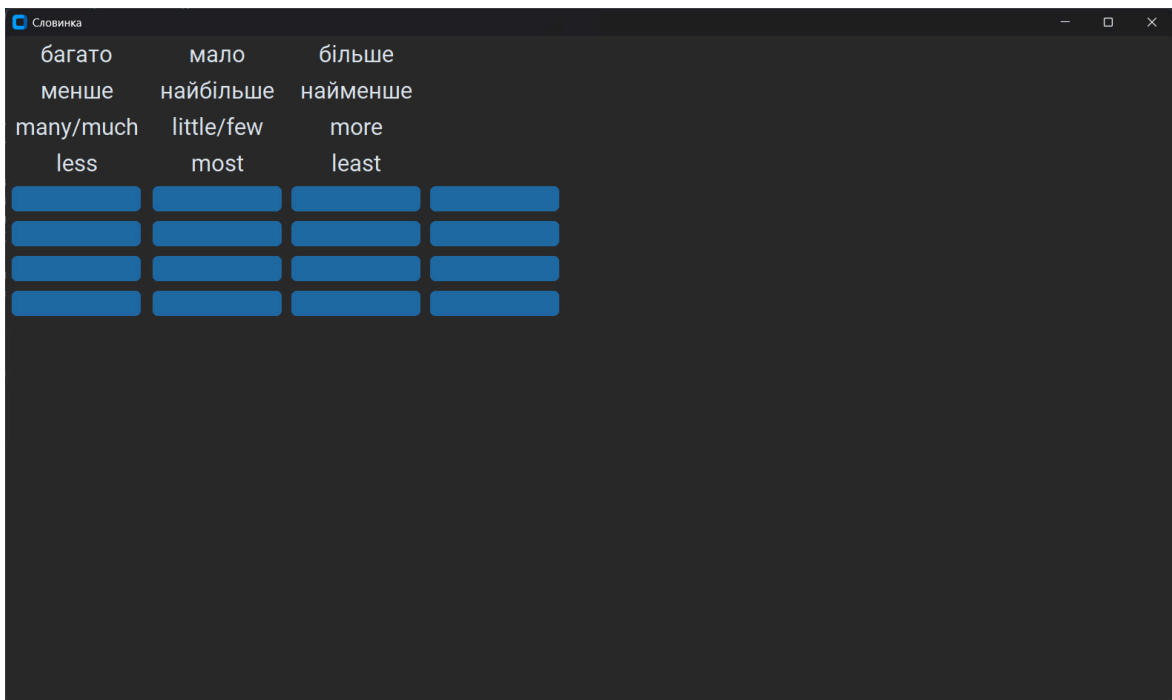


Рис. 2.13 Розташування ігрових елементів до правильного налаштування

При спробах імплементувати функцію `field_update()` у `generate_letter_buttons()` були проблеми із призначенням літер до кнопок. Часом кнопка відображала одну літеру, проте при натисканні додавала до поля іншу. Іноді після здачі слова оновлювалися всі кнопки, навіть ті,

які не були натиснуті, що перемішувало всі літери на сітці. Якийсь час була проблема із правильним реагуванням на *chosen_word*, і тоді не всі потрібні літери для складення обраного слова могли з'явитись, в той час як продовжували з'являтися літери з інших слів. Як наслідок, довелося робити дві окремі функції, які частково повторюються.

2.5 Ідеї для вдосконалення

Хоча програма доволі непогано працює на теперішньому рівні, існує кілька ідей, як можна покращити її роботу. Більшість із цих ідей задумувалися у додаткові від самого початку, проте не змогли бути реалізовані через брак часу або технічні труднощі.

1. **Додавання другого раунду гри.** Він має йти після першого, а потрібні для збирання слова приховувались би до моменту їх введення. Це змушувало б користувача запам'ятовувати слова впродовж першого раунду і дало би більше результатів у вивченні лексики.
2. **Додавання таймера.** Таймер можна помістити на концепті ігрового раунду на Рис. 2.3, і він міг би послужити хорошим мотиватором та елементом гейміфікації. Обмеження на час підштовхувало б користувача діяти швидше, а в разі невдачі – спонукало б перепройти рівень, щоб все ж дійти до кінця. Разом з тим, після завершення рівня зайвий час можна конвертувати в очки або ж записувати найменший витрачений час як найкращий результат для цього рівня. Це додасть мотивації перепроходити рівні, щоб отримати навіть кращі результати.
3. **Додавання таблиці лідерів.** Після попереднього пункту, цей здається доволі природним. Загальна таблиця лідерів, яка показувала б найкращих гравців, функціонувала б схоже до ліги у Duolingo. Варто додати можливість відфільтрувати позиції за рівнем, очками та мовами. Проблема цього пункту в тому, що для більшої ефективності потрібне велике залучення користувачів, а

відповідно – треба, щоб статистика збиралася із усіх користувачів та показувалась усім також. Показ статистики можливо реалізувати через сайт.

4. **Додавання інших мов або мовних пар.** Оскільки додаток багато уваги приділяє правильному вибору літер для складання слів, він допомагає вивчити написання цих слів. Відповідно, значно корисніше буде використовувати його із тими мовами, які мають багато діакритичних знаків у словах або малозрозумілі правила словозміни (як французька чи німецька, наприклад).
5. **Переведення додатку у формат .exe або перенесення його функціоналу на сайт.** Це дуже спростило б використання додатку, оскільки в такому разі користувачам не потрібно буде мати встановлений Python для функціонування коду, що сильно допомогло б розширити аудиторію.

Висновки до розділу 2

У ході розробки власної програми вдалося реалізувати базовий ігровий концепт із динамічною збіркою слів з літер, а також забезпечити багатомовність інтерфейсу та збереження прогресу користувача через SQLite-базу. Підготовка лексичних списків потребувала значних зусиль: довелося відмовитися від застарілих словників і шукати тематично структурований лексичний мінімум, що вимагало ручного редагування та нормалізації форм. Написання коду з CustomTkinter забезпечило приємніший інтерфейс, але у процесі доводилося виправляти багато помилок та несправностей у роботі. Розроблено систему налаштувань (розмір шрифту, тема, мова), що зберігає вибори користувача.

Завдяки анкетуванню вивчені очікування цільової аудиторії, отримано відгуки про потребу в конкуренції та тематичному поділі слів.

ВИСНОВКИ

У Розділі №1 здійснено огляд сучасних ЕЗН і АНС, визначено їхні переваги та недоліки й проведено порівняння ключових платформ за функціоналом, відгуками й статистикою. Встановлено, що адаптивні системи з елементами гейміфікації значно підвищують мотивацію та залученість учнів порівняно з пасивними електронними підручниками. Огляд методик підтвердив важливість інтервального повторення, контекстуалізації матеріалу та миттєвого зворотного зв'язку.

У Розділі 2 розроблено робочий прототип додатку, який поєднав виявлені в Розділі 1 підходи до гейміфікації та мультимовності. Базові цілі, а саме реалізація ігрового процесу, збереження прогресу й налаштування інтерфейсу – були успішно досягнуті.

На всіх етапах розробки (підготовка лексичної бази, написання коду, тестування) виникали труднощі, які в більшості були подолані. На жаль, не всі заплановані функції були реалізовані у поточній версії, проте потенціал для вдосконалення та можливість допрацювання додатку залишаються високими. Подальший розвиток передбачає розширення бази лексики та впровадження додаткових опцій у налаштуваннях.

Загалом, поставлена в дослідженні мета завдання виконана, а виявлені недоліки вказують напрям для удосконалення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Антонюк, Д.С., 2019. Електронні засоби навчання: сутність поняття та їх класифікація. *Фізико-математична освіта*, (3), pp.12–18.

Балла М.І., 1996. *Англо-український словник: у 2 т.* [Електронний ресурс] URL: <http://archive.org/details/enukr1996> [Дата звернення: 3.06.2025].

Белінська, С.М., 2023. Використання інтервального методу в навчанні лексичного складу англійської мови студентів філологічних спеціальностей. [Електронний ресурс] URL: <https://krs.chmnu.edu.ua/jspui/handle/123456789/2777> [Дата звернення: 1.06.2025].

Верховна Рада України, 2025. *Про затвердження Положення про порядок організації та проведення апробації електронних засобів навчального призначення для загальноосвітніх навчальних закладів.* [Електронний ресурс] Офіційний вебпортал парламенту України. URL: <https://zakon.rada.gov.ua/go/z0757-04> [Дата звернення: 24.05.2025].

Зубенко, О.В., 2023. Штучний інтелект і вивчення іноземної мови. [Електронний ресурс] URL: <https://dspace.uzhnu.edu.ua/jspui/handle/lib/63345> [Дата звернення: 2.06.2025].

Іванова, О.В., Журба, О.Р., 2013. ЕФЕКТИВНІ МЕТОДИ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ. Науковий вісник Національного університету біоресурсів і природокористування України. Серія : Філологічні науки. - 2013. - Вип. 186(2). - С. 193-198. [Електронний ресурс] URL: <https://academy-journals.in.ua/index.php/visnyk/article/download/274/249> [Дата звернення: 26.05.2025]

Коломенська, В.В., Пастухов, Р.С., 2006. АВТОМАТИЗОВАНА НАВЧАЛЬНА СИСТЕМА ПО РОЗРОБЦІ WEB-СТОРІНОК. *DSpace at KDPU: Теорія та методика навчання математики, фізики, інформатики. Випуск VI (Т. 3: Теорія та методика навчання інформатики) ст. 78-82.* [Електронний ресурс] URL: <https://elibrary.kdpu.edu.ua/handle/0564/670> [Дата звернення: 24.05.2025].

Лев, В. and Вербяний, І., n.d. Англійсько-Український Словник.

Пасічник О., 2018. Гейміфікація процесу навчання іноземної мови студентів ВНЗ. [Електронний ресурс] URL: <https://elar.khmnu.edu.ua/handle/123456789/6400> [Дата звернення: 23.05.2025].

Петровська Т.Л., n.d. ЕЛЕКТРОННІ ЗАСОБИ НАВЧАННЯ З ФІЗИКИ, ЇХ ВИДИ ТА ПРИЗНАЧЕННЯ – Кафедра інформатики і інформаційно-комунікаційних технологій. URL: https://informatika.udpu.edu.ua/?page_id=1328 [Дата звернення: 24.05.2025].

Саган, О.В., 2022. ГЕЙМІФІКАЦІЯ ЯК СУЧАСНИЙ ОСВІТНІЙ ТРЕНД. *Збірник наукових праць «Педагогічні науки»*, (100), pp.12–18. <https://doi.org/10.32999/ksu2413-1865/2022-100-2> .

Тріщук, О.В., Фіголь, Н.М. and Волик, Н.С., 2019. Гейміфікація в освітньому процесі. *Технологія і техніка друкарства*, (3(65)), pp.72–79. [https://doi.org/10.20535/2077-7264.3\(65\).2019.202000](https://doi.org/10.20535/2077-7264.3(65).2019.202000) .

Центр фінансово-економічних наукових досліджень, n.d. СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ НАУКИ, ОСВІТИ, ТЕХНОЛОГІЙ І СУСПІЛЬСТВА. *Центр фінансово-економічних наукових досліджень*. URL: <https://www.economics.in.ua/2023/06/16-2023.html> [Дата звернення: 2.06.2025].

Чернилевський, Д.В., n.d. ЕЛЕКТРОННІ ЗАСОБИ НАВЧАННЯ.

Шевчук, Р., 2008. Удосконалення комплексу вимог до педагогічних програмних засобів. *Правова інформатика*, (2), pp.72–78.

Шишкіна, М.П., 2009. Класифікація програмних засобів навчального призначення. *Наукові записки. – Серія: Педагогічні науки*, 2(82), pp.286–292.

Ясинська, С.Ю. and Пархуць, Л., n.d. Information systems in education: automated training systems. [Електронний ресурс] URL: https://core.ac.uk/outputs/147037367/?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1 [Дата звернення: 24.05.2025].

Borodin, K. and Turkevych, O., 2024. 1000 і 1 слово. [Електронний ресурс] <https://doi.org/10.18452/28236> .

Busuu, 2025. *Busuu Homepage*. [Електронний ресурс] Busuu. URL: <https://www.busuu.com> [Дата звернення: 23.05.2025].

Chyzykova, I. and Tokarieva, A., 2021. USE OF MODERN GAMIFIED TEACHING MEANS IN THE CONTEXT OF TEACHING FOREIGN LANGUAGES TO STUDENTS OF TECHNICAL AND ECONOMIC SPECIALTIES. *Pedagogy of the formation of a creative person in higher and secondary schools*, 2(77), pp.147–150. <https://doi.org/10.32840/1992-5786.2021.77-2.27> .

Curry, D., n.d. Duolingo Revenue and Usage Statistics (2025). *Business of Apps*. URL: <https://www.businessofapps.com/data/duolingo-statistics/> [Дата звернення: 24 May 2025].

Drops, 2025. *Drops - Free language learning for 50+ Languages*. [Електронний ресурс] Drops. URL: <https://www.languagedrops.com/> [Дата звернення: 24.05.2025].

Duolingo, 2025. *Learn a language for free*. [Електронний ресурс] Duolingo. URL: <https://www.duolingo.com/> [Дата звернення: 24.05.2025].

GeeksforGeeks, 10:57:49+00:00. *Changing Class Members in Python*. [Електронний ресурс] GeeksforGeeks. URL: <https://www.geeksforgeeks.org/g-fact-42-changing-class-members-python/> [Дата звернення: 6.06.2025].

GeeksforGeeks, 20:29:43+00:00. *self in Python class*. [Електронний ресурс] GeeksforGeeks. URL: <https://www.geeksforgeeks.org/self-in-python-class/> [Дата звернення: 6.06.2025].

GitHub, 2025. *brown-uk/dict_uk*. URL: https://github.com/brown-uk/dict_uk [Дата звернення: 3.06.2025].

Harrold, K., 2024. *Assassin's Creed Odyssey used by teacher to give history lessons*. [Електронний ресурс] GAMINGbible. URL: <https://www.gamingbible.com/news/assassins-creed-odyssey-used-teacher-give-history-lessons-929463-20240214> [Дата звернення: 28.05.2025].

Holubieva, I., 2023. COMPARATIVE ANALYSIS OF THE USE OF TRADITIONAL AND MODERN METHODS OF LEARNING FOREIGN LANGUAGES. *Humanities science current issues*, 1(64), pp.322–328. <https://doi.org/10.24919/2308-4863/64-1-48>.

Kapp, K., 2025. *The Gamification of Learning and Instruction | Summary, Quotes, FAQ, Audio*. [Електронний ресурс] URL: <https://sobrief.com/books/the-gamification-of-learning-and-instruction> [Дата звернення: 26.05.2025].

LingQ, 2025. *Вивчайте нову мову з улюбленого контенту*. [Електронний ресурс] LingQ. URL: <https://lingq.com> [Дата звернення: 24.05.2025].

Memrise, 2025. *Learn a language. Memrise is authentic, useful & personalised*. [Електронний ресурс] URL: <https://www.memrise.com> [Дата звернення: 24.05.2025].

Nalyvaiko, O., Nuralieva, L. and Syvolotska, L., 2021. Immersion in a foreign environment as an effective way of development of communicative

competence in the process of learning foreign languages. *Scientific notes of the pedagogical department*, (49), pp.42–51.

<https://doi.org/10.26565/2074-8167-2021-49-05> .

Noor, N.M., Yunus, K., Yusoff, A.M.H., Nasir, N.A.M. and Yaacob, N.H., 2021. Spaced learning: A review on the use of spaced learning in language teaching and learning. *Journal of Language and Linguistic Studies*, 17(2), pp.1023–1031. <https://doi.org/10.3316/informit.216509458456602> .

OPUS, 2025. *OPUS - Corpora*. [Электронный ресурс] URL:

<https://opus.nlpl.eu/> [Дата звернення: 3.06.2025].

Rebolledo Font de la Vall, R. and González Araya, F., n.d. (PDF) Exploring the Benefits and Challenges of AI-Language Learning Tools. *ResearchGate*. [Электронный ресурс] <https://doi.org/10.18535/ijsshi/v10i01.02> .

Richards, O., 2022. *Drops Language Review 2022: Is the Drops App Worth it?* [Электронный ресурс] URL: <https://storylearning.com/blog/drops-review> [Дата звернення: 25.05.2025].

Rosetta Stone, 2025. *Official Rosetta Stone® - Language Learning - Learn a Language*. [Электронный ресурс] URL: <https://eu.rosettastone.com/> [Дата звернення: 24.05.2025].

Shortt, M., Tilak ,Shantanu, Kuznetcova ,Irina, Martens ,Bethany and and Akinkuolie, B., 2023. Gamification in mobile-assisted language learning: a systematic review of Duolingo literature from public release of 2012 to early 2020. *Computer Assisted Language Learning*, 36(3), pp.517–554. <https://doi.org/10.1080/09588221.2021.1933540> .

Smarter Language, 2023. *The Smarter Language Review of Busuu - SmarterLanguage*. URL: <https://www.smarterlanguage.com/review-of-busuu/> [Дата звернення: 25.05.2025].

SQLite, 2025. *Datatypes In SQLite*. [Электронный ресурс] URL: <https://www.sqlite.org/datatype3.html> [Дата звернення: 6.06.2025].

Universal Dependencies. 2025. *Universal POS tags*. [Электронный ресурс] URL: <https://universaldependencies.org/u/pos/> [Дата звернення: 6.06.2025].

Vliet, N. van, 2020. The Only LingQ Review You Need. *Smart Language Learner*. URL: <https://www.smartlanguagelearner.com/lingq-review/> [Дата звернення: 26.05.2025].

ДОДАТКИ

Додаток А

Результати соціопитування щодо АНС:

https://docs.google.com/forms/d/1OpZO-suHw8DF1nP8YCx6hJZqhFh51Mud2V_x1fWtJO8/edit#responses

Додаток Б

Код програми разом із списками слів та базою даних:

<https://github.com/Renkivist/Slovynka.git>

Додаток В

Демонстрація роботи програми (відео):

https://drive.google.com/drive/folders/1vGpD2L_7Si1OezQFgQtio9YtAqRJmhcx?usp=sharing