

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
кафедра української мови та прикладної лінгвістики

**ПРОЄКТУВАННЯ ЕЛЕКТРОННОГО ЗАСОБУ НАВЧАННЯ
МОРФОЛОГІЇ НІМЕЦЬКОЇ МОВИ**

Кваліфікаційна робота бакалавра
студента 4 курсу
освітньої програми
*«Прикладна (комп'ютерна) лінгвістика
та англійська мова»*
спеціальності – 035.10 Філологія (прикладна
лінгвістика)
галузі знань – 03 гуманітарні науки
Олексія БОНДАРА
Науковий керівник:
кандидат техн. н., доц. Микола КОСТИКОВ

«Допущено до захисту»

Протокол засідання

кафедри української мови та прикладної лінгвістики

протокол № 15 від « 06 » 06 2024 року

завідувач кафедри _____ (підпис)

к.філол.н., доц. Сергій РІЗНИК

АНОТАЦІЯ

У високошвидкісному сучасному світі значення використання передових технологій в освітньому процесі постійно зростає. Одним з найбільш актуальних напрямків є створення та впровадження ефективних електронних інструментів для вивчення іноземних мов. Важливим елементом цього процесу є створення інструменту, який спрямований на вивчення морфології німецької мови.

Головною метою цієї кваліфікаційної роботи є створення та оцінювання ефективності електронного інструменту, який сприяє покращенню процесу вивчення морфології німецької мови. Для досягнення цієї мети було визначено завдання, які включають розробку програмного продукту, дослідження теоретичних аспектів морфології німецької мови та аналіз результатів застосування електронного інструменту.

Ця робота включає в себе огляд теоретичних аспектів морфології німецької мови, аналіз сучасних методів вивчення мов, розгляд процесу проєктування та реалізації електронного інструменту, а також оцінювання його ефективності. Результати цієї роботи можуть бути використані як основа для подальших досліджень та розвитку інноваційних інструментів у сфері вивчення іноземних мов.

Ключові слова: сучасний світ, інноваційні технології, освіта, електронні засоби, вивчення іноземних мов, морфологія німецької мови, кваліфікаційна робота, програмний продукт, теоретичні аспекти, ефективність.

ABSTRACT

In the high-speed modern world, the importance of using advanced technologies in the educational process is constantly increasing. One of the most relevant directions is the creation and implementation of effective electronic tools for learning foreign languages. An important element of this process is the creation of a tool aimed at studying the morphology of the German language.

The main goal of this qualification work is to create and evaluate the effectiveness of an electronic tool that contributes to the improvement of the process of studying the morphology of the German language. To achieve this goal, tasks were defined, which include the development of a software product, the study of theoretical aspects of the morphology of the German language, and the analysis of the results of the application of the electronic tool.

This work includes an overview of the theoretical aspects of the morphology of the German language, an analysis of modern methods of language learning, a consideration of the process of designing and implementing an electronic tool, as well as evaluating its effectiveness. The results of this work can be used as a basis for further research and development of innovative tools in the field of foreign language learning.

Keywords: modern world, innovative technologies, education, electronic means, learning foreign languages, morphology of the German language, qualification work, software product, theoretical aspects, efficiency.

ЗМІСТ

АНОТАЦІЯ	4
ВСТУП	5
РОЗДІЛ 1	6
ПОНЯТТЯ ЕЛЕКТРОННИХ ЗАСОБІВ НАВЧАННЯ ТА	6
ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ ВИВЧЕННЯ НІМЕЦЬКОЇ МОВИ	6
1.1. Електронні засоби навчання:	6
поняття, класифікація та методи розробки	6
1.2. Огляд та порівняння електронних засобів навчання німецької мови	8
1.3. Проблеми розглянутих електронних засобів навчання німецької мови	10
РОЗДІЛ 2	13
ПОСТАНОВКА ЗАДАЧІ НА ПРОЄКТУВАННЯ	13
ЕЛЕКТРОННОГО ЗАСОБУ	13
2.1. Визначення цільової аудиторії та функціональних вимог	13
2.2. Архітектура засобу	15
2.3. Інтерфейс користувача	16
РОЗДІЛ 3	18
ПРОЄКТУВАННЯ ТА РОЗРОБКА ЕЛЕКТРОННОГО ЗАСОБУ	18
3.1. Проектування бази даних для	18
зберігання всіх форм дієслів німецької мови	18
3.2. Розробка додатку	22
3.2. Оцінка ефективності та аналіз результатів	41
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТКИ	50

ВСТУП

В сучасному світі, що стрімко розвивається, використання інноваційних технологій у сфері освіти набуває все більшого значення. Однією з актуальних тенденцій є розробка та впровадження ефективних електронних засобів для вивчення іноземних мов. Серед них особливо важливе місце займає розробка засобу, спрямованого на навчання морфології німецької мови.

Зростання інтересу до вивчення іноземних мов та недоліки традиційних методів навчання створюють необхідність у використанні сучасних підходів та засобів для поліпшення процесу навчання. В цьому контексті, розробка електронного засобу для вивчення морфології німецької мови стає актуальним завданням, спрямованим на поліпшення якості освіти та забезпечення ефективного вивчення іноземних мов.

Метою даної кваліфікаційної роботи є розробка та оцінка ефективності електронного засобу, спрямованого на покращення навчання морфології німецької мови. Для досягнення цієї мети визначено завдання з розробки програмного продукту, вивчення теоретичних аспектів морфології німецької мови та аналізу результатів використання електронного засобу.

Окремі положення роботи висвітлено у [2].

Дана робота розкриває огляд теоретичних аспектів морфології німецької мови, аналізує сучасні методи вивчення мов, розглядає процес проектування та реалізації електронного засобу, а також проводить оцінку його ефективності. Результати цієї роботи можуть служити основою для подальших досліджень та розвитку інноваційних засобів у галузі вивчення іноземних мов.

РОЗДІЛ 1

ПОНЯТТЯ ЕЛЕКТРОННИХ ЗАСОБІВ НАВЧАННЯ ТА ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ ВИВЧЕННЯ НІМЕЦЬКОЇ МОВИ

1.1. Електронні засоби навчання: поняття, класифікація та методи розробки

Електронні засоби навчання – це інструменти та технології, що використовуються для підтримки та покращення навчального процесу через цифрові платформи [3]. Вони можуть включати різноманітні форми, такі як онлайн-курси, відеоуроки, інтерактивні вправи, симуляції, електронні підручники та інші навчальні ресурси.

ЕЗН забезпечують студентам доступ до навчальних матеріалів у будь-який час і в будь-якому місці, що робить навчання більш гнучким та індивідуалізованим. Вони також можуть включати елементи гейміфікації, що підвищують мотивацію та залучення учнів до навчального процесу.

Електронні засоби навчання можна класифікувати за різними критеріями:

1. За типом взаємодії:
 - Асинхронні: дозволяють учням вчитися у своєму темпі (наприклад, онлайн-курси, записи лекцій).
 - Синхронні: вимагають одночасної участі учнів та викладачів (наприклад, вебінари, відеоконференції).
2. За формою подачі інформації:
 - Текстові матеріали: електронні підручники, статті, довідники.
 - Мультимедійні матеріали: відеоуроки, аудіозаписи, анімації.
 - Інтерактивні матеріали: симуляції, тренажери, інтерактивні вправи.
3. За рівнем інтерактивності:
 - Пасивні: студенти отримують інформацію без активної взаємодії (наприклад, перегляд відео).

- Активні: студенти беруть участь у процесі навчання (наприклад, виконання завдань, участь у дискусіях).

4. За предметною областю:

- Загальноосвітні: математика, історія, біологія тощо.
- Спеціалізовані: навчальні курси для конкретних професій або дисциплін (наприклад, програмування, інженерія).

Процес розробки ЕЗН включає кілька основних етапів:

1. Аналіз потреб:

- Визначення цілей навчання та вимог аудиторії.
- Аналіз існуючих навчальних програм та матеріалів.
- Ідентифікація прогалин у знаннях та навичках студентів.

2. Проектування:

- Створення структури навчального курсу.
- Визначення методів та інструментів подачі матеріалу.
- Розробка інтерактивних елементів та завдань.

3. Розробка:

- Створення контенту (текстів, відео, аудіо, графіки).
- Програмування інтерактивних елементів.
- Інтеграція контенту в навчальну платформу.

4. Тестування:

- Перевірка функціональності та якості матеріалів.
- Збір зворотного зв'язку від тестової групи користувачів.
- Внесення необхідних коригувань.

5. Впровадження:

- Запуск курсу на навчальній платформі.
- Інформування студентів та викладачів про новий ресурс.
- Підтримка та технічна допомога користувачам.

6. Оцінка та вдосконалення:

- Моніторинг результатів навчання та зворотного зв'язку.
- Аналіз ефективності курсу та його впливу на навчальні досягнення.

- Внесення покращень на основі отриманих даних.

Електронні засоби навчання відіграють важливу роль у сучасній освіті, забезпечуючи доступність, інтерактивність та ефективність навчання. Вони можуть бути класифіковані за різними критеріями, включаючи тип взаємодії, форму подачі інформації, рівень інтерактивності та предметну область. Процес розробки ЕЗН включає аналіз потреб, проектування, розробку, тестування, впровадження та оцінку, що дозволяє створювати якісні та корисні навчальні ресурси. Використання сучасних методів та технологій у розробці ЕЗН сприяє підвищенню якості освіти та досягненню кращих результатів у навчанні.

1.2. Огляд та порівняння електронних засобів навчання німецької мови

Сучасні технології відкривають нові можливості для вивчення іноземних мов, зокрема німецької. Електронні засоби навчання надають широкий спектр інструментів, що роблять процес навчання більш інтерактивним та доступним. У цьому підрозділі розглянемо основні електронні засоби навчання німецької мови, такі як Duolingo, Babbel, Rosetta Stone та Memrise, а також порівняємо їх функції, переваги та недоліки.

1. Duolingo

Duolingo - популярна платформа для вивчення мов, яка використовує гейміфікацію для підвищення мотивації учнів. Основні функції Duolingo включають інтерактивні уроки, що включають завдання на переклад, вимову та сприйняття на слух, можливість встановлення цілей навчання та відстеження прогресу та систему нагород та досягнень для підтримки інтересу до навчання [39].

2. Babbel

Babbel пропонує курси, розроблені професійними лінгвістами, з акцентом на практичне використання мови. Основні функції Babbel включають уроки, що покривають різні аспекти мови: граматику, лексику, вимову, персоналізовані

вправи та зворотний зв'язок, аудіо та відеоматеріали з носіями мови для покращення навичок сприйняття на слух [36].

3. Rosetta Stone

Rosetta Stone відома своєю методикою повного занурення в мовне середовище без використання перекладів. Основні функції Rosetta Stone включають інтерактивні уроки з використанням зображень, текстів та аудіо для контекстного вивчення мови, розпізнавання мови для покращення вимови, мобільний додаток, що дозволяє вчитися будь-де і будь-коли.

4. Memrise

Memrise фокусується на вивченні лексики через мнемотехнічні прийоми та повторення. Основні функції Memrise включають курси, створені як професіоналами, так і користувачами, використання відео з носіями мови для автентичного досвіду навчання та систему повторень для закріплення знань. [40]

Таблиця 1. Порівняння найпопулярніших додатків для вивчення німецької мови

	Duolingo	Babbel	Rosetta Stone	Memrise
Інтерактивні уроки	✓	✓	✓	✓
Гейміфікація	✓	✗	✗	✓
Персоналізовані вправи	частково	✓	✓	✓
Повне занурення	✗	✗	✓	✗
Відео з носіями мови	✗	✗	✗	✓

Розпізнавання мови	✗	✗	✓	✗
Мобільний додаток	✓	✓	✓	✓
Наявність української мови	✗	✗	✗	✗

Кожен з електронних засобів навчання німецької мови має свої особливості та підходи, що можуть відповідати різним потребам та стилям навчання. Duolingo та Memrise підходять для тих, хто шукає гейміфіковані та інтерактивні методи навчання, тоді як Babbel та Rosetta Stone пропонують більш структурований та глибокий підхід до вивчення мови. Вибір платформи залежить від індивідуальних потреб, мети навчання та особистих уподобань користувача. Проте, основним недоліком, який пов'язує всі аналізовані додатки є відсутність української мови інтерфейсу, що є ще одним вагомим аргументом для реалізації додатку, який задовільнить більшість потреб українських студентів, які вивчають граматику німецької мови.

1.3. Проблеми розглянутих електронних засобів навчання німецької мови

Хоча електронні засоби навчання, такі як Duolingo, Babbel, Rosetta Stone та Memrise, мають багато переваг і пропонують численні можливості для вивчення німецької мови, вони також мають певні обмеження та проблеми. Однією з основних проблем є відсутність підтримки української мови, що створює додаткові труднощі для українських студентів, що, в свою чергу, підкріплює основну тезу наукової роботи про необхідність проектування електронного засобу навчання морфології німецької мови.

Duolingo пропонує безліч мовних курсів для вивчення, проте українська мова відсутня як мова інтерфейсу для курсів німецької. Це означає, що українським користувачам доводиться використовувати інші мови (частіше за все англійську) для вивчення німецької, що може створювати додаткові бар'єри для тих, хто не володіє англійською достатньо добре. Відсутність української мови обмежує доступність та зручність користування для українських студентів, знижуючи ефективність навчального процесу.

Babbel також не підтримує українську мову як мову інтерфейсу або навчання для німецької. Це може бути серйозною перешкодою для українських користувачів, особливо тих, хто відчуває труднощі з англійською чи іншими мовами, які підтримуються платформою. Багато учнів можуть втрачати мотивацію через необхідність навчатися через посередника (англійську мову), що знижує ефективність навчання та викликає додатковий стрес.

Методика **Rosetta Stone**, що базується на повному зануренні, взагалі не використовує переклад і спирається на візуальні та аудіальні підказки. Однак відсутність української мови як базової мови інтерфейсу може все ж ускладнювати процес навчання для українських користувачів. Крім того, цей підхід може не підходити всім учням, особливо тим, хто звик до традиційних методів навчання з перекладом та поясненням.

Memrise, хоча й пропонує курси, створені користувачами, все ще має обмежену підтримку української мови для навчання німецької. Це може призводити до проблем з якістю курсів та їхньою ефективністю для українських користувачів. Відсутність структурованої підтримки української мови може створювати бар'єри для учнів, які не мають достатніх знань англійської або іншої посередницької мови.

Відсутність підтримки української мови в розглянутих електронних засобах навчання німецької мови підкреслює важливість розробки спеціалізованого електронного засобу для українських студентів. Таке рішення могло б значно покращити доступність та ефективність навчання німецької мови, особливо для тих, хто має обмежені знання інших мов.

Проектування електронного засобу навчання морфології німецької мови з підтримкою української мови могло б включати:

1. Український інтерфейс: Забезпечення зручного інтерфейсу українською мовою для легкого доступу до всіх функцій платформи.
2. Морфологічні вправи: Розробка інтерактивних вправ з акцентом на морфологію німецької мови, що відповідають потребам українських студентів.
3. Пояснення та переклад: Надання детальних пояснень та перекладів українською мовою для кращого розуміння та засвоєння матеріалу.
4. Локалізація контенту: Створення локалізованого контенту, що враховує культурні та мовні особливості українських користувачів.

Отже, проектування електронного засобу навчання морфології німецької мови з підтримкою української мови може значно підвищити доступність і ефективність навчання для українських студентів, сприяючи більш глибокому та швидшому засвоєнню матеріалу.

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ НА ПРОЄКТУВАННЯ ЕЛЕКТРОННОГО ЗАСОБУ

2.1. Визначення цільової аудиторії та функціональних вимог

Процес розробки електронного засобу навчання вимагає чіткого розуміння цільової аудиторії та їхніх потреб. Визначення цільової аудиторії та функціональних вимог є ключовими етапами, що дозволяють створити ефективний та зручний інструмент для вивчення мови.

Основною цільовою аудиторією для електронного засобу навчання морфології німецької мови є студенти та учні середніх шкіл:

- Вікова категорія: 14-22 роки;
- Мета: підготовка до іспитів, поглиблене вивчення німецької мови;
- Рівень володіння мовою: від початкового до середнього (A1-B2).

Основні потреби цільової аудиторії включають:

- Доступність матеріалів українською мовою: Забезпечення можливості вивчати німецьку мову через зрозумілий інтерфейс та матеріали рідною мовою;
- Інтерактивність: Використання інтерактивних завдань для підвищення мотивації та залученості учнів;
- Адаптивне навчання: Можливість адаптації курсу під індивідуальні потреби та рівень знань користувача;
- Мультимедійні матеріали: Використання відео, аудіо та візуальних матеріалів для покращення сприйняття та засвоєння інформації;
- Зворотній зв'язок: Наявність механізмів для отримання зворотного зв'язку та оцінки прогресу.

Основні функціональні вимоги, які варто реалізувати в електронному засобі навчання морфології німецької мови:

1. Український інтерфейс

Електронний засіб навчання повинен мати повний переклад інтерфейсу українською мовою для забезпечення зручності використання українськими учнями. Це включає меню, інструкції, підказки та всі навчальні матеріали.

2. Адаптивне навчання

Платформа повинна мати функцію адаптивного навчання, яка дозволяє обирати контент відповідно до граматичної теми та індивідуальних потреб кожного користувача. Це включає можливість пропускати відомі теми та зосереджуватись на складних для учня аспектах.

3. Інтерактивні вправи

Система повинна пропонувати різноманітні інтерактивні вправи, що включають завдання на переклад, вибір правильних варіантів, конструювання речень, диктанти та інші види завдань, які допоможуть закріпити морфологічні знання.

4. Мультимедійні ресурси

Необхідно забезпечити наявність зображень для легкого засвоєння інформації. Інтерактивні медіа, як таблиці, допоможуть учням краще розуміти контекст використання морфологічних конструкцій.

5. Можливість роботи в автономному режимі

Платформа повинна підтримувати можливість завантаження матеріалів для використання в автономному режимі, що дозволить учням займатися навіть за відсутності доступу до Інтернету.

Таблиця 2. Порівняння з попередньо розглянутими засобами

	Duolingo	Babbel	Rosetta Stone	Memrise	Проектований засіб
Український інтерфейс	✗	✗	✗	✗	✓
Адаптивне навчання	✓	✓	✓	✓	✓
Інтерактивні вправи	✓	✓	✓	✓	✓

Мультимедійні ресурси	обмежено	✓	✓	✓	✓
Гейміфікація	✓	✗	✗	✓	✗
Автономний режим	обмежено	✓	✓	✓	✓
Аналітика та звіти	обмежено	✓	✓	✓	✗

Українські учні потребують платформи, що враховує їхні специфічні потреби, такі як підтримка української мови, адаптивне навчання, інтерактивні вправи та мультимедійні ресурси. Можливість роботи в автономному режимі додатково підвищать мотивацію та ефективність навчання. Таким чином, проектування електронного засобу навчання, що відповідає цим вимогам, може значно покращити процес вивчення німецької мови для українських студентів.

2.2. Архітектура засобу

Проектування ефективного електронного засобу навчання морфології німецької мови вимагає розробки відповідної архітектури та інтерфейсу користувача. В цьому підрозділі детально розглянуто архітектуру засобу, вибір технологій та принципи побудови інтерфейсу користувача для ПК з використанням бібліотеки Tkinter. Такий підхід забезпечить зручність та доступність для українських учнів, враховуючи специфіку навчального процесу та особливості користувачів.

Архітектура електронного засобу навчання складається з кількох ключових компонентів:

- Клієнтська частина: Інтерфейс користувача (UI), розроблений за допомогою Tkinter, який забезпечує взаємодію учня з програмою.
- Логіка додатку: Основний функціонал, включаючи адаптивні вправи, обробку результатів, надання зворотного зв'язку та аналіз помилок.

- База даних: Зберігання навчальних матеріалів, результатів учнів та статистики.

Клієнтська частина реалізується за допомогою бібліотеки Tkinter, яка дозволяє створювати прості та ефективні графічні інтерфейси для Python-додатків. Tkinter є частиною стандартної бібліотеки Python, що забезпечує легкість у використанні та інтеграції з іншими компонентами додатку.

2.3. Інтерфейс користувача

Інтерфейс користувача повинен бути інтуїтивно зрозумілим та зручним для використання, враховуючи різний рівень комп'ютерної грамотності учнів.

Основні принципи проєктування включають:

- Простота та зрозумілість: Інтерфейс повинен бути простим, з мінімальною кількістю елементів, які відволікають, з чіткими інструкціями та підказками.
- Консистентність: Всі елементи інтерфейсу повинні мати однаковий стиль і поведінку, що сприятиме легшому запам'ятовуванню та використанню.
- Адаптивність: Інтерфейс має адаптуватися до потреб учня, пропонуючи відповідні теми для вивчення та типи завдань.
- Мультимедійність: Використання різних медіа-форматів (текст, зображення) для покращення сприйняття та засвоєння матеріалу.

Основні компоненти інтерфейсу:

1. Меню логіну / реєстрації

Меню логіну / реєстрації повинно містити дві опції входу в додаток для користувача: вхід для зареєстрованих користувачів та реєстрація для нових користувачів, використовуючи електронну адресу та пароль. Це дозволить користувачам увійти в програму.

2. Головне меню

Головне меню повинно містити основні розділи програми: теоретичний матеріал, тестування, прогрес та налаштування. Це дозволить користувачам легко орієнтуватися в програмі.

3. Вікно з теоретичним матеріалом

Вікно з теоретичним матеріалом дозволяє користувачам вибрати відповідну граматичну тему для вивчення (Prasens, Prateritum, Perfekt, Plusquamperfekt, Futur I, Futur II). Це забезпечує адаптивність навчання та можливість індивідуального підходу до кожного учня.

4. Вікно з тестуванням

Вікно з тестуванням включає різні типи завдань (тести з можливістю вибору одного варіанту відповіді та декількох, відкриті запитання, переклад речень). Інтерактивні вправи забезпечують ефективне засвоєння матеріалу.

5. Вікно з прогресом користувача

Вікно з прогресом користувача містить в собі прогрес виконання тестувань з наявних тем користувачем, у відсотковому значенні. Це дозволяє користувачу відстежувати свій прогрес у засвоєнні матеріалу та надає можливість зрозуміти які теми були засвоєні користувачем, а які потребують доопрацювання.

6. Вікно налаштувань

Вікно налаштувань дозволяє користувачам налаштувати параметри програми відповідно до своїх уподобань, такі як вибір мови інтерфейсу, зміна кольорової теми додатку та вихід з додатку за потреби.

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РОЗРОБКА ЕЛЕКТРОННОГО ЗАСОБУ

3.1. Проєктування бази даних для зберігання всіх форм дієслів німецької мови

Для виконання цієї задачі було обрано бібліотеку для Python – sqlite3. Це легка, вбудована система управління базами даних, яка не вимагає окремого серверного програмного забезпечення. Вона зручна для невеликих до середніх обсягів даних, забезпечує високу продуктивність та підтримує стандарт SQL. SQLite є особливо корисною для додатків, де потрібна проста і надійна база даних без складних налаштувань.

Для організації бази даних, яка буде містити всі форми дієслів німецької мови, необхідно врахувати багато аспектів, таких як час, особа та вид дієслова. Було створено реляційну базу даних, яка буде зберігати ці дані в декількох таблицях, що дозволить ефективно організувати та отримувати необхідну інформацію.

База даних складатиметься з таких основних таблиць:

1. **verbs**: Основна таблиця для дієслів;
2. **tenses**: Таблиця для часів;
3. **persons**: Таблиця для особових форм;
4. **verb_forms**: Таблиця для зберігання конкретних форм дієслів.

Розробка бази даних для зберігання всіх форм дієслів німецької мови передбачає створення декількох взаємопов'язаних таблиць, які забезпечують ефективне зберігання та пошук необхідної інформації. Використання реляційної бази даних дозволяє організувати дані таким чином, щоб забезпечити зручний доступ до всіх форм дієслів з урахуванням роду, числа, часу та особи. Це сприятиме більш ефективному використанню даних у процесі навчання та розробці навчальних програм.

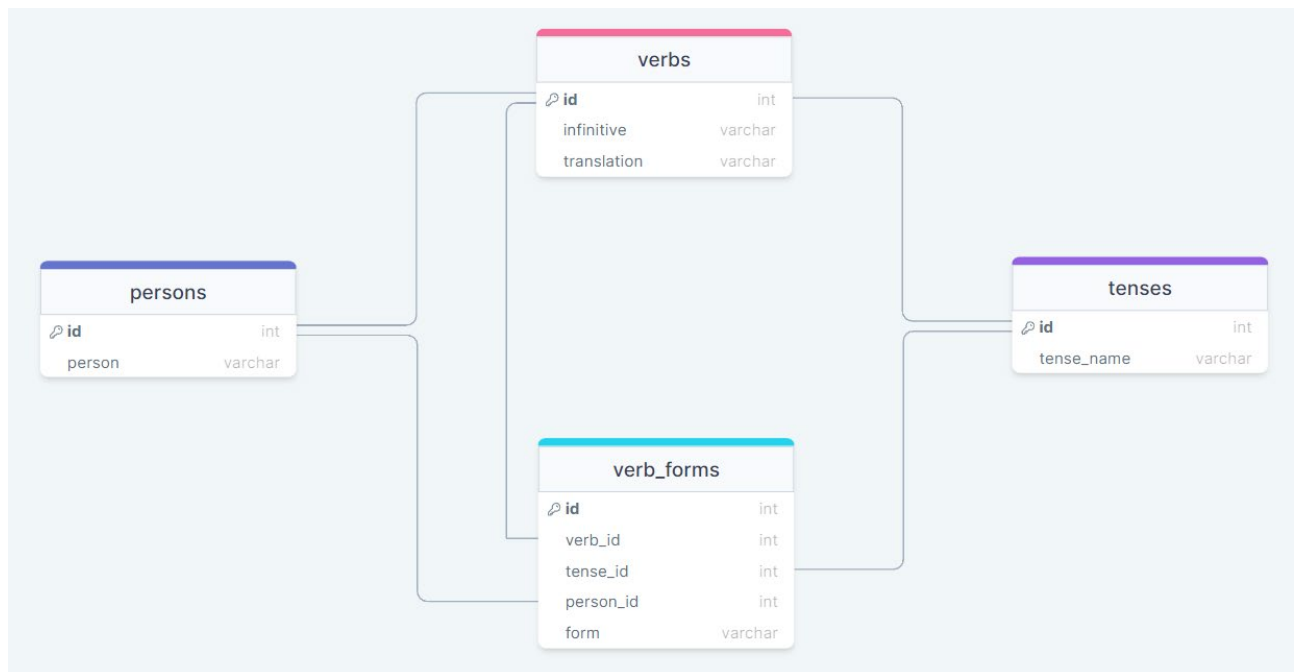


Рис. 1. Структура бази даних

Для реалізації бази даних використовуються наступні кроки:

- 1) З'єднання з базою даних SQLite встановлюється за допомогою модуля `sqlite3` в Python. Це дозволяє виконувати SQL-запити безпосередньо з Python-коду.

```
import sqlite3
connection = sqlite3.connect('german_verbs.db')
cursor = connection.cursor()
```

- 2) Створення таблиць виконується за допомогою SQL-запитів, де вказуються структури таблиць, типи даних колонок та їх обмеження.

```
cursor.execute('''
CREATE TABLE IF NOT EXISTS verbs (
    id INTEGER PRIMARY KEY,
    infinitive TEXT NOT NULL,
    translation TEXT NOT NULL
)
''')
```

```

cursor.execute('''
CREATE TABLE IF NOT EXISTS tenses (
    id INTEGER PRIMARY KEY,
    tense_name TEXT NOT NULL
)
''')

cursor.execute('''
CREATE TABLE IF NOT EXISTS persons (
    id INTEGER PRIMARY KEY,
    person TEXT NOT NULL
)
''')

cursor.execute('''
CREATE TABLE IF NOT EXISTS verb_forms (
    id INTEGER PRIMARY KEY,
    verb_id INTEGER,
    tense_id INTEGER,
    person_id INTEGER,
    form TEXT NOT NULL,
    FOREIGN KEY (verb_id) REFERENCES verbs (id),
    FOREIGN KEY (tense_id) REFERENCES tenses
(id),
    FOREIGN KEY (person_id) REFERENCES persons
(id)
)
''')

```

3) Заповнення таблиць необхідним змістом за допомогою SQL-запитів.

Наприклад, нижче продемонстровано метод заповнення таблиці “verbs”:

```
verbs = [  
    (1, 'sein', 'бути'),  
    (2, 'haben', 'мати'),  
    (3, 'werden', 'ставати'),  
    (4, 'können', 'могти'),  
    (5, 'müssen', 'мусити'),  
    (6, 'wollen', 'хотіти'),  
    (7, 'machen', 'робити'),  
    (8, 'geben', 'давати'),  
    (9, 'kommen', 'приходити'),  
    (10, 'gehen', 'ходити'),  
    (11, 'wissen', 'знати'),  
    (12, 'sehen', 'бачити'),  
    (13, 'finden', 'шукати'),  
    (14, 'bleiben', 'залишатись'),  
    (15, 'nehmen', 'брати')  
]  
  
cursor.executemany('INSERT INTO verbs (id,  
infinitive, translation) VALUES (?, ?, ?)', verbs)
```

4) Закриття бази даних після успішного заповнення її таблиць даними:

```
connection.commit()  
connection.close()
```

Після того, як всі маніпуляції були виконані, база даних набула наступного вигляду: *Див. додаток В*

3.2. Розробка додатку

Розробка додатку для вивчення німецької мови з використанням сучасних засобів програмування є важливим етапом для забезпечення функціональності та зручності користувача. У цьому підрозділі розглянемо ключові аспекти розробки додатку, включаючи вибір інструментів, структуру коду та інтеграцію різних функцій.

Для розробки додатку було обрано мову програмування Python [11], зокрема її бібліотеки tkinter [13] та customtkinter [14] для створення графічного інтерфейсу користувача (GUI). Використання customtkinter дозволяє створювати більш сучасний і привабливий інтерфейс у порівнянні зі стандартним tkinter. Також було використано бібліотеку sqlite3 [12] для роботи з базою даних, яка зберігає інформацію про користувачів та їхній прогрес.

Розробка додатку розпочалася зі створення базової структури проекту.

Основними компонентами додатку є:

- Вікно реєстрації та входу;
- Головне вікно з вкладками: "Теорія", "Тестування", "Прогрес" та "Налаштування";
- База даних для зберігання інформації.

Нижче наведено фрагмент коду для створення вікна реєстрації та входу, використовуючи customtkinter:

```
# Ініціалізація основного вікна
def create_main_window():
    ctk.set_appearance_mode("Light") # Темна/світла
    тема базуючись на системних налаштуваннях
    ctk.set_default_color_theme("blue") # Тема кольорів

    root = ctk.CTk() # Використовуємо CTk замість Tk
    root.title("Deutschify")
```

```

root.geometry("400x500")

# Створення основного фрейму
frame = ctk.CTkFrame(master=root)
frame.pack(pady=20, padx=60, fill="both",
expand=True)

# Вкладки для входу та реєстрації
tabview = ctk.CTkTabview(master=frame)
tabview.pack(fill="both", expand=True)

login_tab = tabview.add("Увійти")
register_tab = tabview.add("Реєстрація")

create_login_ui(login_tab, root)
create_register_ui(register_tab)

return root

# Функція для створення інтерфейсу входу
def create_login_ui(parent, root):
    login_label = ctk.CTkLabel(master=parent,
text="Вхід", font=ctk.CTkFont(size=24, weight="bold"))
    login_label.pack(pady=12, padx=10)

    email_entry = ctk.CTkEntry(master=parent,
placeholder_text="Електронна адреса")
    email_entry.pack(pady=12, padx=10)

```

```

        password_entry = ctk.CTkEntry(master=parent,
placeholder_text="Пароль", show="*")
        password_entry.pack(pady=12, padx=10)

        login_button = ctk.CTkButton(master=parent,
text="Увійти",
                                command=lambda:
handle_login(email_entry, password_entry, root))
        login_button.pack(pady=12, padx=10)

# Функція для створення інтерфейсу реєстрації
def create_register_ui(parent):
    register_label = ctk.CTkLabel(master=parent,
text="Реєстрація", font=ctk.CTkFont(size=24,
weight="bold"))
    register_label.pack(pady=12, padx=10)

    reg_email_entry = ctk.CTkEntry(master=parent,
placeholder_text="Електронна адреса")
    reg_email_entry.pack(pady=12, padx=10)

    reg_password_entry = ctk.CTkEntry(master=parent,
placeholder_text="Пароль", show="*")
    reg_password_entry.pack(pady=12, padx=10)

    reg_confirm_password_entry =
ctk.CTkEntry(master=parent,
placeholder_text="Підтвердити пароль", show="*")
    reg_confirm_password_entry.pack(pady=12, padx=10)

```

```

        register_button = ctk.CTkButton(master=parent,
text="Зареєструватись", command=lambda: print("Register
clicked"))

        register_button.pack(pady=12, padx=10)

# Функція для обробки входу
def handle_login(email_entry, password_entry,
login_window):
    email = email_entry.get()
    password = password_entry.get()

    if email == "user@gmail.com" and password ==
"123123":
        messagebox.showinfo("Deutschify", "Вітаємо у
Deutschify!")
        login_window.destroy()
        open_home_window()
    else:
        messagebox.showerror("Deutschify", "Неправильна
електронна адреса або пароль.")

```

Однією з ключових функцій додатку є вкладка "Теорія", яка містить інформацію про граматичні правила. Для прикладу, було розроблено інтерфейс, що дозволяє користувачу переглядати різні теми теоретичного матеріалу. Нижче наведено фрагмент коду для створення вкладки "Теорія":

```

# Функція для створення інтерфейсу теорії
def create_theory_ui(parent):
    topics = ["Тема 1. Das Präsens", "Тема 2. Das

```

Präteritum", "Тема 3. Perfekt", "Тема 4.

Plusquamperfekt",

"Тема 5. Futur I", "Тема 6. Futur II"]

content = {

"Тема 1. Das Präsens": ""

Теперішній час (Das Präsens) – це час, який вказує на події, що відбуваються в даний момент часу

або взагалі сталися. Він також

використовується для вираження загальних фактів та регулярних

подій. У німецькій мові теперішній час утворюється за допомогою закінчення дієслова, яке змінюється

в залежності від особи та числа.

Наприклад:

- ich spiele (я граю)
- du spielst (ти граєш)
- er/sie/es spielt (він/вона/воно грає)
- wir spielen (ми граємо)
- ihr spielt (ви граєте)
- sie/Sie spielen (вони грають)

Також варто відзначити, що деякі німецькі дієслова мають неправильні форми у теперішньому часі.

КОН'ЮГАЦІЯ ТЕПЕРІШНЬОГО ЧАСУ: РЕГУЛЯРНІ ДІЄСЛОВА

""",

"Тема 2. Das Präteritum": ""

Минулий час (Das Präteritum) – це час, який використовується для опису дій або подій, що відбулися в минулому.

Він часто використовується в письмовій мові, особливо в літературі, розповідях та описах.

Наприклад:

- ich spielte (я грав)
- du spieltest (ти грав)
- er/sie/es spielte (він/вона/воно грав)
- wir spielten (ми грали)
- ihr spieltet (ви грали)
- sie/Sie spielten (вони грали)

Для неправильних дієслів форми Präteritum можуть значно відрізнятися від стандартних. Ось деякі приклади:

- ich war (я був)
- du warst (ти був)
- er/sie/es war (він/вона/воно був)
- wir waren (ми були)
- ihr wart (ви були)
- sie/Sie waren (вони були)

- ich hatte (я мав)
- du hattest (ти мав)

- er/sie/es hatte (він/вона/воно мав)
- wir hatten (ми мали)
- ihr hattet (ви мали)
- sie/Sie hatten (вони мали)

Präteritum часто використовується в офіційних письмових текстах, новинах, історіях та романах.

КОН'ЮГАЦІЯ – РЕГУЛЯРНІ ДІЄСЛОВА

""",

"Тема 3. Perfekt": ""

Завершений час (Perfekt) – це час, який використовується для опису дій або подій, що відбулися в минулому,

але мають зв'язок з теперішнім часом. Він часто використовується в розмовній мові.

Наприклад:

- ich habe gespielt (я грав)
- du hast gespielt (ти грав)
- er/sie/es hat gespielt (він/вона/воно грав)
- wir haben gespielt (ми грали)
- ihr habt gespielt (ви грали)
- sie/Sie haben gespielt (вони грали)

Зауважте, що для утворення Perfekt використовується допоміжне дієслово (haben або sein) та

причастя минулого часу основного дієслова.

УТВОРЕННЯ PERFECT

""",

"Тема 4. Plusquamperfekt": """

Передминулий час (Plusquamperfekt) – це час, який використовується для опису дій або подій, що відбулися до іншої події в минулому.

Наприклад:

- ich hatte gespielt (я грав)
- du hattest gespielt (ти грав)
- er/sie/es hatte gespielt (він/вона/воно грав)
- wir hatten gespielt (ми грали)
- ihr hattet gespielt (ви грали)
- sie/Sie hatten gespielt (вони грали)

Зауважте, що для утворення Plusquamperfekt використовується допоміжне дієслово (haben або sein) в Präteritum та причастя минулого часу основного дієслова.

УТВОРЕННЯ PLUSQUAMPERFEKT

""",

"Тема 5. Futur I": """

Майбутній час I (Futur I) – це час, який використовується для опису дій або подій, що відбудуться в майбутньому, або для вираження припущень.

Наприклад:

- ich werde spielen (я буду грати)
- du wirst spielen (ти будеш грати)
- er/sie/es wird spielen (він/вона/воно буде грати)
- wir werden spielen (ми будемо грати)
- ihr werdet spielen (ви будете грати)
- sie/Sie werden spielen (вони будуть грати)

Зауважте, що для утворення Futur I використовується допоміжне дієслово werden та інфінітив основного дієслова.

УТВОРЕННЯ FUTUR I

""",

"Тема 6. Futur II": ""

Майбутній час II (Futur II) – це час, який використовується для опису дій або подій, які будуть завершені в майбутньому.

Наприклад:

- ich werde gespielt haben (я зігратиму)
- du wirst gespielt haben (ти зігратимеш)
- er/sie/es wird gespielt haben (він/вона/воно зігратиме)
- wir werden gespielt haben (ми зігратимемо)

- ihr werdet gespielt haben (ви зіграєте)
- sie/Sie werden gespielt haben (вони зіграють)

Зауважте, що для утворення Futur II використовується допоміжне дієслово werden, причастя минулого час

УТВОРЕННЯ FUTUR II

```
"""  
}  
  
frame = ctk.CTkFrame(master=parent)  
frame.pack(fill="both", expand=True)  
  
# Список кнопок для вибору тем  
button_frame = ctk.CTkFrame(master=frame)  
button_frame.pack(side="left", fill="y")  
  
content_frame = ctk.CTkFrame(master=frame)  
content_frame.pack(side="right", fill="both",  
expand=True)  
  
content_text = tk.Text(master=content_frame,  
wrap="word", font=("Helvetica", 12), state="disabled")  
content_text.pack(pady=20, padx=20, fill="both",  
expand=True)  
  
for topic in topics:
```

```

        button = ctk.CTkButton(master=button_frame,
text=topic,

                                command=lambda t=topic:
display_content(t, content, content_text))
        button.pack(pady=5, padx=5, fill="x")

def display_content(topic, content, text_widget):
    text_widget.config(state="normal")
    text_widget.delete(1.0, "end")

    # Додавання тексту в текстовий віджет
    text_widget.insert("insert", content[topic])

    if topic == "Тема 1. Das Präsens":
        # Завантаження малюнку
        image = PhotoImage(file="prasens.png")

        # Додавання малюнку в кінець текстового віджету
        text_widget.image_create(END, image=image)

        # Зберігання змінної для додавання малюнків
        text_widget.image = image

    if topic == "Тема 2. Das Präteritum":
        # Завантаження малюнку
        image = PhotoImage(file="prateritum.png")

        # Додавання малюнку в кінець текстового віджету
        text_widget.image_create(END, image=image)

```

```
# Зберігання змінної для додавання малюнків
text_widget.image = image

if topic == "Тема 3. Perfekt":
    # Завантаження малюнку
    image = PhotoImage(file="perfekt.png")

    # Додавання малюнку в кінець текстового віджету
    text_widget.image_create(END, image=image)

    # Зберігання змінної для додавання малюнків
    text_widget.image = image

if topic == "Тема 4. Plusquamperfekt":
    # Завантаження малюнку
    image = PhotoImage(file="plusquamperfekt.png")

    # Додавання малюнку в кінець текстового віджету
    text_widget.image_create(END, image=image)

    # Зберігання змінної для додавання малюнків
    text_widget.image = image

if topic == "Тема 5. Futur I":
    # Завантаження малюнку
    image = PhotoImage(file="futur1.png")

    # Додавання малюнку в кінець текстового віджету
    text_widget.image_create(END, image=image)
```

```

        # Зберігання змінної для додавання малюнків
        text_widget.image = image

    if topic == "Тема 6. Futur II":
        # Завантаження малюнку
        image = PhotoImage(file="futur2.png")

        # Додавання малюнку в кінець текстового віджету
        text_widget.image_create(END, image=image)

        # Зберігання змінної для додавання малюнків
        text_widget.image = image

    text_widget.config(state="disabled")

```

Не менш важливою є функція яка імплементує тестові завдання в даний додаток. Тестування складається зі завдань з вибором однієї/кількох відповідей та відкриті питання з довільним вводом від користувача. Нижче надано фрагмент коду який стосується створення вікна з тестуванням:

```

# Функція для створення інтерфейсу тестування

def create_practice_ui(parent):
    label = ctk.CTkLabel(parent, text="Оберіть тему для
тестування", font=ctk.CTkFont(size=20, weight="bold"))
    label.pack(pady=20)

    test_topics = ["Das Präsens", "Das Präteritum",
"Perfekt", "Plusquamperfekt", "Futur I", "Futur II"]

```

```

    for topic in test_topics:
        button = ctk.CTkButton(parent, text=topic,
command=lambda t=topic: open_test_window(t))
        button.pack(pady=10)

# Функція для зміни мови інтерфейсу
def change_language(value):
    language_var.set(value)
def create_settings_ui(parent):
    label = ctk.CTkLabel(parent, text="Налаштування",
font=ctk.CTkFont(size=20, weight="bold"))
    label.pack(pady=20)

    # Налаштування теми
    theme_frame = ctk.CTkFrame(parent)
    theme_frame.pack(pady=10, padx=20, anchor="w")

    theme_label = ctk.CTkLabel(theme_frame,
text="Тема:", font=ctk.CTkFont(size=14))
    theme_label.pack(side="left")

    def toggle_theme():
        current_theme = theme_var.get()
        if current_theme == "світла":
            # Змініть кольори для світлої теми
            root.configure(bg="white")
            # Інші віджети можуть також мати більше
світлих кольорів
        elif current_theme == "темна":

```

```

        # Змінить кольори для темної теми
        root.configure(bg="black")

        # Інші віджети можуть також мати більше
темних кольорів

        theme_var = ctk.BooleanVar(value=True) # Початкова
тема - світла

        theme_switch = ctk.CTkSwitch(theme_frame,
variable=theme_var, command=toggle_theme)
        theme_switch.pack(side="left")

        # Кнопка "Вийти з акаунту"
        def logout():
            if messagebox.askyesno("Підтвердження", "Ви
впевнені, що хочете вийти з акаунту?"):
                parent.destroy() # Закрити головне вікно

        logout_button = ctk.CTkButton(parent, text="Вийти з
акаунту", command=logout)
        logout_button.pack(pady=20)

def open_test_window(topic=""):
    global test_questions
    test_window = ctk.CTkToplevel()
    test_window.geometry("800x600")
    test_window.title(f"Тестування: {topic}")

    # Створення прокручуваної області для питань
    canvas = ctk.CTkCanvas(test_window)

```

```

        scrollbar = ctk.CTkScrollbar(test_window,
orientation="vertical", command=canvas.yview)
        scrollable_frame = ctk.CTkFrame(canvas)

        scrollable_frame.bind(
            "<Configure>",
            lambda e: canvas.configure(
                scrollregion=canvas.bbox("all")
            )
        )

        canvas.create_window((0, 0),
window=scrollable_frame, anchor="nw")
        canvas.configure(yscrollcommand=scrollbar.set)

        canvas.pack(side="left", fill="both", expand=True)
        scrollbar.pack(side="right", fill="y")

        # Завантаження питань для вибраної теми
        questions = test_questions.get(topic, [])
        responses = []

        def submit_test():
            score = 0
            for idx, response in enumerate(responses):
                q_index, user_answer, q_type = response
                correct_answer = questions[q_index][2]

                if q_type == "single":
                    if int(user_answer.get()) ==

```

```

correct_answer:
    score += 1
    elif q_type == "multiple":
        correct_answers = set(correct_answer)
        user_answers = set([i for i, var in
enumerate(user_answer) if var.get() == 1])
        if user_answers == correct_answers:
            score += 1
    elif q_type == "text":
        if user_answer.get().strip().lower() ==
correct_answer.lower():
            score += 1
    elif q_type == "translation":
        if user_answer.get().strip().lower() ==
correct_answer.lower():
            score += 1

percentage = (score / len(questions)) * 100
result_message = f"Ви відповіли правильно на
{score} з {len(questions)} питань. Ваш результат:
{percentage:.2f}%."

if percentage >= 100:
    result_message += "Відмінно!"
elif percentage >= 80:
    result_message += "Добре!"
elif percentage >= 60:
    result_message += "Задовільно."
else:
    result_message += "Не здано."

```

```

        result_label = ctk.CTkLabel(test_window,
text=result_message, font=ctk.CTkFont(size=18,
weight="bold"))
        result_label.pack(pady=20)

        if percentage >= 60:
            return_button = ctk.CTkButton(test_window,
text="Повернутись в основне вікно",
command=test_window.destroy)
            return_button.pack(pady=10)
        else:
            retry_button = ctk.CTkButton(test_window,
text="Повторити спробу",
command=lambda:
[test_window.destroy(), open_test_window(topic)])
            retry_button.pack(pady=10)
            review_button = ctk.CTkButton(test_window,
text="Повторити матеріал",
command=lambda: [test_window.destroy(),
tabview.select("Теоретичний матеріал")])
            review_button.pack(pady=10)

        for i, (question, options, answer, q_type) in
enumerate(questions):
            q_frame = ctk.CTkFrame(scrollable_frame)
            q_frame.pack(fill="both", pady=10, padx=10)

```

```

        q_label = ctk.CTkLabel(q_frame, text=f"{i + 1}.
{question}", font=ctk.CTkFont(size=16))
        q_label.pack(anchor="w", pady=5)

        if q_type == "single":
            answer_var = ctk.StringVar()
            for opt in options:
                rb = ctk.CTkRadioButton(q_frame,
text=opt, variable=answer_var, value=options.index(opt))
                rb.pack(anchor="w", padx=20)
                responses.append((i, answer_var, q_type))

        elif q_type == "multiple":
            answer_vars = []
            for opt in options:
                cb_var = ctk.IntVar()
                cb = ctk.CTkCheckBox(q_frame, text=opt,
variable=cb_var)
                cb.pack(anchor="w", padx=20)
                answer_vars.append(cb_var)
                responses.append((i, answer_vars, q_type))

        elif q_type == "text":
            answer_entry = ctk.CTkEntry(q_frame)
            answer_entry.pack(anchor="w", padx=20)
            responses.append((i, answer_entry, q_type))

        elif q_type == "translation":
            answer_entry = ctk.CTkEntry(q_frame)
            answer_entry.pack(anchor="w", padx=20)

```

```

        responses.append((i, answer_entry, q_type))

    submit_button = ctk.CTkButton(scrollable_frame,
text="Завершити тестування", command=submit_test)
    submit_button.pack(pady=20)

```

3.2. Оцінка ефективності та аналіз результатів

Для забезпечення високої якості програмного продукту, який був розроблений, необхідно провести оцінку ефективності. Це дозволяє перевірити, чи відповідає додаток усім вимогам, які до нього висувалися на початкових етапах розробки.

Оцінка ефективності додатку включає аналіз різних аспектів, таких як продуктивність, зручність користування, відповідність функціональним вимогам та надійність. Оцінка проводиться по 5-бальній шкалі, де 1 – це погано реалізований аспект, а 5 – найкраще втілений.

Основні критерії оцінки ефективності додатку для вивчення німецької мови:

Таблиця 3. Оцінка ефективності додатку

КРИТЕРІЙ ОЦІНЮВАННЯ	1	2	3	4	5
1. Продуктивність:					
Час завантаження додатку та окремих його компонентів					✓
Швидкість обробки даних у базі даних SQLite					✓
Відгук користувацького інтерфейсу на дії користувача				✓	
2. Зручність користування:					

Інтуїтивність та простота навігації в додатку					✓
Відповідність інтерфейсу сучасним стандартам					✓
Зручність введення даних та отримання зворотного зв'язку				✓	
3. Відповідність функціональним вимогам:					
Наявність та коректне функціонування всіх заявлених функцій (реєстрація, вхід, перегляд теорії, проходження тестування, відстеження прогресу тощо)				✓	
Відсутність критичних помилок, що можуть вплинути на користувацький досвід				✓	
4. Надійність:					
Стійкість до неправильних дій користувачів				✓	
Захист даних користувачів та їхній коректний зберігання					✓

Оцінка ефективності додатку вказує на те, що розроблений додаток демонструє високу продуктивність, зручність користування, відповідність функціональним вимогам та надійність. Аналіз результатів дозволяє зробити наступні висновки:

1. Продуктивність

Час завантаження додатку та окремих його компонентів, а також швидкість обробки даних у базі даних SQLite отримали найвищі оцінки (5 з 5), що свідчить про ефективну оптимізацію цих аспектів. Однак, відгук користувацького інтерфейсу на дії користувача отримав оцінку 4, що вказує на можливі місця для покращення, особливо в умовах високого навантаження.

2. Зручність користування

Інтуїтивність та простота навігації в додатку, а також відповідність інтерфейсу сучасним стандартам UX/UI отримали оцінку 5, що демонструє відмінну роботу

у цих напрямках. Зручність введення даних та отримання зворотного зв'язку оцінена на 4, що вказує на наявність деяких незначних недоліків, які варто врахувати при подальших покращеннях.

3. Відповідність функціональним вимогам

Наявність та коректне функціонування всіх заявлених функцій отримали оцінку 4, що свідчить про загальну відповідність вимогам, проте є деякі незначні аспекти, які потребують доопрацювання. Відсутність критичних помилок також оцінена на 4, що вказує на високий рівень стабільності, хоча і не ідеальний.

4. Надійність

Стійкість до неправильних дій користувачів отримала оцінку 4, що вказує на високу, але не абсолютну, стійкість системи. Захист даних користувачів та їхній коректний зберігання отримали найвищу оцінку 5, що свідчить про високий рівень безпеки та конфіденційності.

Загалом, додаток демонструє високі показники ефективності, зручності користування, відповідності функціональним вимогам та надійності. Незважаючи на відмінні результати, існують певні аспекти, які можуть бути покращені у майбутніх версіях, такі як відгук інтерфейсу та зручність введення даних. Підсумовуючи, можна стверджувати, що додаток повністю відповідає очікуванням та вимогам, встановленим на початкових етапах розробки, і може бути успішно використаний для навчання німецької мови.

ВИСНОВКИ

Розробка додатку для вивчення німецької мови була складним і багатоетапним процесом, що вимагав інтеграції різних технологій та рішень. Основні інструменти для створення додатку включали Python, бібліотеки tkinter та customtkinter для графічного інтерфейсу користувача, а також SQLite для управління базою даних. Основна мета полягала в створенні зручного та функціонального додатку, який допомагав би користувачам у вивченні німецької мови, забезпечуючи інтерактивне середовище для вивчення теорії та практики.

На початковому етапі були визначені основні функціональні вимоги до додатку, такі як можливість реєстрації та входу користувачів, перегляд теоретичних матеріалів, виконання практичних завдань та відстеження прогресу навчання. Для забезпечення зручного та інтуїтивного користувацького інтерфейсу використовувалися бібліотеки tkinter та customtkinter. Головне вікно додатку було спроектоване з вкладками для теорії, практики, прогресу навчання та налаштувань профілю.

Ключові функції додатку, такі як обробка користувацьких даних, відображення теоретичних матеріалів та практичних завдань, були реалізовані за допомогою Python та SQLite. Особлива увага приділялася забезпеченню коректного функціонування кожної з цих функцій. Після реалізації основних функцій додатку було проведено детальне тестування для виявлення та виправлення помилок, що дозволило перевірити стабільність та надійність додатку, його відповідність вимогам та зручність користування.

Оцінка ефективності додатку показала, що розроблений додаток демонструє високу продуктивність, зручність користування, відповідність функціональним вимогам та надійність. Загалом, додаток виявився ефективним, зручним у використанні та відповідає функціональним вимогам. Незважаючи на відмінні результати, існують певні аспекти, які можуть бути покращені у майбутніх версіях, такі як відгук інтерфейсу та зручність введення даних.

Підсумовуючи, можна стверджувати, що додаток повністю відповідає очікуванням та вимогам, встановленим на початкових етапах розробки, і може бути успішно використаний для навчання німецької мови.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Асоянц П. Г. Основи методики створення і застосування комп'ютерних програм у навчанні іноземних мов [Текст] / Асоянц П. Г., Сердюков П. І., Чекаль Г. С. — К : КДППМ, 1993. — 108 с.
2. Бондар О. О., Костіков М. П. Проектування електронного засобу навчання морфології німецької мови // Матер. X Міжнар. наук.-техн. Internet-конф. «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 24 листоп. 2023 р. — К.: НУХТ, 2023. — С. 175.
3. Буртовий, С. В. Електронні засоби навчання – від теорії до практики. Методичний посібник. Кіровоград: КЗ «КОІППО імені Василя Сухомлинського», 2014. 48 с.
4. Жалдак М. І. Педагогічний потенціал комп'ютерно-орієнтованих систем навчання математики [Текст] / М. І. Жалдак // Комп'ютерно-орієнтовані системи навчання : зб. наук. праць. — К. : НПУ ім. М. П. Драгоманова. — Вип. 7. — 2003. — С. 13–16.
5. Застосування інформаційних технологій та мережі Інтернет на уроках іноземних мов [Текст] / Чернів. облдержадмін., Голов. упр. освіти і науки, Чернів. обл. ін-т післядиплом. пед. освіти ; [підгот.: С. М. Куриш та ін.]. — Чернівці ; Вижниця : Черемош, 2011. — 144 с. : рис., табл. — (Освіта Чернівеччини).
6. Інформаційні технології та електронні засоби навчального призначення – ознака сучасного уроку. Конспекти уроків / [Укл.: Цимбал І.І., Дьяченко Б.А., Сорочан Т.М. та ін.], Луганськ : Знання, 2006. — 504 с.
7. Коваль Т. І. Підготовка викладачів вищої школи: інформаційні технології у педагогічній діяльності [Текст] : навч.-метод. посіб. / Т. І. Коваль. — К. : Вид. центр НЛУ, 2009. — 380 с.

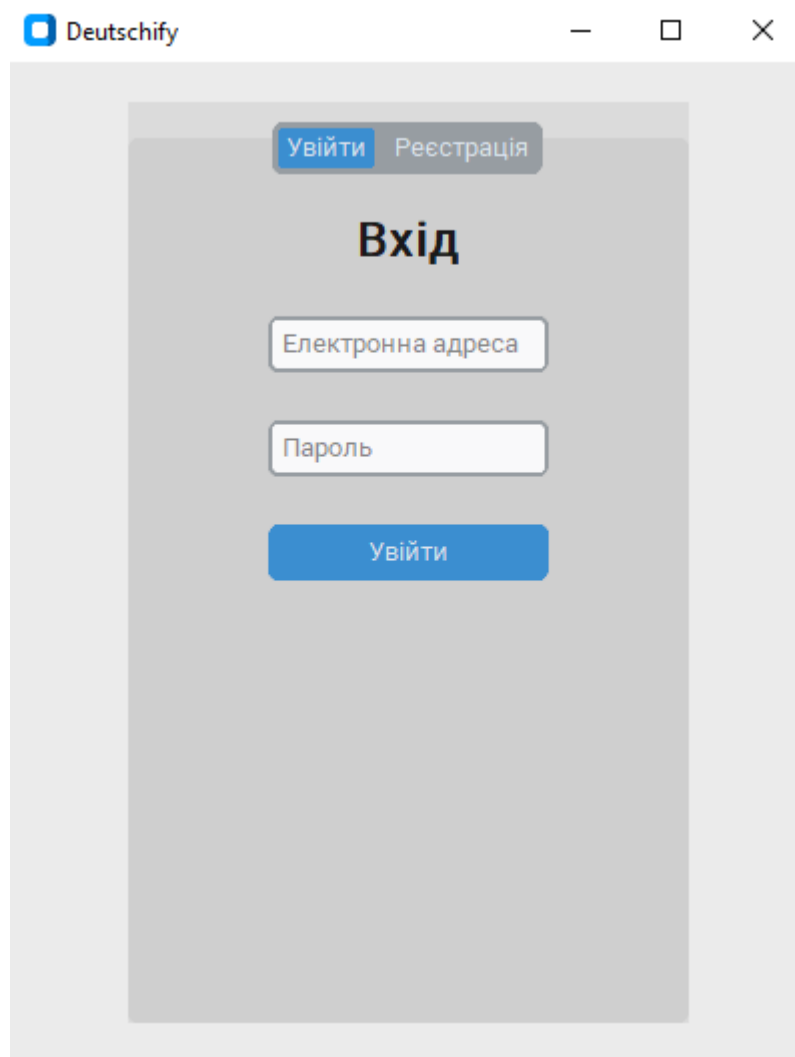
8. Меньяйленко О. С. Автоматизовані педагогічні навчальні системи [Текст] : моногр. / О. С. Меньяйленко. — Луганськ : Альма-матер, 2003. — 272 с.
9. Основи нових інформаційних технологій навчання [Текст] : посіб. для вчителів / Машбиць Ю. І., Гокунь О. О., Жалдак М. І. [та ін.] ; Інститут психології ім. Г. С. Костюка АПН України, Інститут змісту і методів навчання. — К. : [б.в.], 1997. — 260 с.
- 10.Офіційна документація Pillow. URL: <https://pillow.readthedocs.io/en/stable/>.
- 11.Офіційна документація Python. URL: <https://docs.python.org/3>.
- 12.Офіційна документація sqlite3. URL: <https://docs.python.org/3/library/sqlite3.html>.
- 13.Офіційна документація tkinter. URL: <https://docs.python.org/3/library/tkinter.html>.
- 14.Офіційна документація CustomTkinter. URL: <https://github.com/TomSchimansky/CustomTkinter/wiki>.
- 15.Скрипка Г. В. Використання електронних засобів навчального та загального призначення в Кіровоградській області / Г. В. Скрипка // Науково-методичний посібник. Технологія фахової майстерності: ІКТ компетентність в освітніх процесах, 2007. — 116 с. — С. 50-54.
16. Bule J. Models for Adaptive Computer-Based Learning Management [Текст] : doctoral thesis summary / Ekaterina Bule ; Riga Technical University. — Riga : RTU, 2011. — 41 p.
17. Moore, J. L., Dickson-Deane, C., Galyen, K. E-Learning, online learning, and distance learning environments: Are they the same? The Internet and Higher Education, 2011. Vol. 14, No. 2, pp. 129-135.
18. Clark, R. C., Mayer, R. E. E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning. John Wiley & Sons, 2016. 512 p.
19. Anand, S., Ghantasala, A. V. Performance optimization techniques for a big data recommendation system using heterogeneous hybridization of prediction

- models. URL:
<https://www.sciencedirect.com/science/article/pii/S1877050921007729>.
20. Chelihi, B., Mezaour, A. Investigating the impact of e-learning technology on academic performance. URL: <https://link.springer.com/article/10.1007/s10639-022-11377-x>.
21. Черныш, Н. А., Бурлаченко, О. М. Применение телекоммуникационных технологий в образовании: анализ и перспективы. URL:
<https://www.ajird.journalspark.org/index.php/ajird/article/view/975>.
22. Kharchenko, O. Empirical study of professional competences development in translation studies. URL: https://www.researchgate.net/profile/Kostiantyn-Mizin-2/publication/376642581_Berufsbild_des_Translators_Globalisierungsursache_n_der_Anderung/links/6581a19f6f6e450f19883c8c/Berufsbild-des-Translators-Globalisierungsursachen-der-Aenderung.pdf#page=28.
23. Han, Y., Lueg, C. P. Exploring user satisfaction with mobile language learning applications. URL:
<https://www.tandfonline.com/doi/abs/10.1080/09588221.2018.1456465>.
24. Lubanovic, B. Introducing Python: Modern Computing in Simple Packages. O'Reilly Media, 2019. 364 p.
25. Sweigart, A. Automate the Boring Stuff with Python. No Starch Press, 2019. 504 p.
26. Horton, W. E-Learning by Design. John Wiley & Sons, 2011. 400 p.
27. Bates, A. W. Teaching in a Digital Age: Guidelines for Designing Teaching and Learning. Tony Bates Associates Ltd, 2019. 518 p.
28. Python Software Foundation. (n.d.). Tkinter – Python Interface to Tcl/Tk. URL: <https://docs.python.org/3/library/tkinter.html>.
29. Python Software Foundation. (n.d.). Official Python Documentation. URL: <https://docs.python.org/3>.
30. Tom Schimansky. (n.d.). CustomTkinter. URL:
<https://github.com/TomSchimansky/CustomTkinter>.

31. Moore, A. D. Python GUI Programming with Tkinter. Packt Publishing, 2022. 568 p.
32. Lutz, M. Programming Python. O'Reilly Media, 2010. 1632 p.
33. Levy M. CALL Dimensions: Options and Issues in Computer-Assisted Language Learning [Текст] / Mike Levy, Glenn Stockwell. — Mahwah NJ : Lawrence Erlbaum, 2006. — 328 p.
34. Hagel, G. Advanced SQL Queries and Techniques for High Performance. O'Reilly Media, 2019. 400 p.
35. Warschauer M. Computer Assisted Language Learning: an Introduction [Текст] / Mark Warschauer // Multimedia language teaching. — Tokyo : Logos International, 1996. — P. 3–20.
36. Babbel. (n.d.). A Mobile Language Learning App. URL:
https://www.researchgate.net/profile/Musa-Nushi/publication/322937460_Babbel_A_Mobile_Language_Learning_App/links/5b470ef145851519b4b0a1b0/Babbel-A-Mobile-Language-Learning-App.pdf.
37. Nuft, U. (n.d.). Food safety and hygiene in catering services. URL:
<https://dspace.nuft.edu.ua/server/api/core/bitstreams/62e49dd6-c6db-44f6-adc5-f505a2c86ca4/content>.
38. KPI, ELA. (n.d.). Technical aspects of electronic education. URL:
<https://ela.kpi.ua/server/api/core/bitstreams/d28e8e62-a40f-4e39-8173-4361b097260a/content>.
39. Duolingo. (n.d.). Language Learning Platform. URL:
<https://www.duolingo.com/>.
40. Rosetta Stone. (n.d.). Language Learning Software. URL:
<https://www.rosettastone.com/>.

ДОДАТКИ

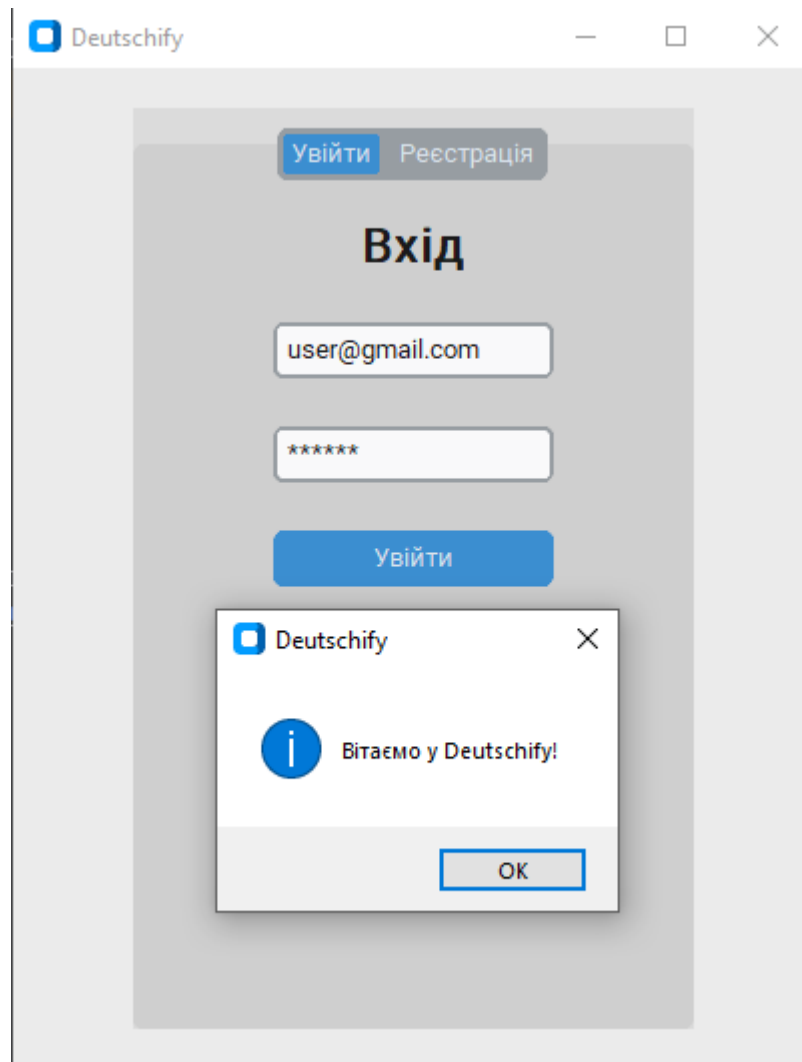
Додаток А. Скріншоти результатів роботи програми



Увійти **Реєстрація**

Реєстрація

Зареєструватись



Deutschify

Теоретичний матеріал

Тестування

Налаштування

Тема 1. Das Präsens

Тема 2. Das Präteritum

Тема 3. Perfekt

Тема 4. Plusquamperfekt

Тема 5. Futur I

Тема 6. Futur II

Теперішній час (Das Präsens) - це час, який вказує на події, що відбуваються в даний момент часу або взагалі сталися. Він також використовується для вираження загальних фактів та регулярних подій. У німецькій мові теперішній час утворюється за допомогою закінчення дієслова, яке змінюється в залежності від особи та числа.

Наприклад:

- ich spiele (я граю)
- du spielst (ти граєш)
- er/sie/es spielt (він/вона/воно грає)
- wir spielen (ми граємо)
- ihr spielt (ви граєте)
- sie/Sie spielen (вони грають)

Також варто відзначити, що деякі німецькі дієслова мають неправильні форми у теперішньому часі.

КОНЮГАЦІЯ ТЕПЕРІШНЬОГО ЧАСУ: РЕГУЛЯРНІ ДІЄСЛОВА

	Особа Person	Закінчення Endung	Приклад Beispiel
Однина Singular	1. Person	ich	-e ich spiele
	2. Person	du	-st du spielst
	3. Person	er/sie/es/man	-t er spielt
Множина Plural	1. Person	wir	-en wir spielen
	2. Person	ihr	-t ihr spielt
	3. Person / Formal	sie/Sie	-en sie spielen

Оберіть тему для тестування

Das Präsens

Das Präteritum

Perfekt

Plusquamperfekt

Futur I

Futur II

Тестування: Das Präsens

3. Оберіть дієслова HE у формі Präsens

- ☐ spiele
☒ spielte
☒ gespielt
☐ spielen
☒ gesehen

4. Оберіть дієслова HE у формі Präsens, 2 особи множини

- ☐ spielt
☐ spieltet
☒ spielten
☒ gespielt

5. Вставте пропущене слово: Er ___ zum Fußballtraining

- ☒ geht
☐ ging
☐ gegangen
☐ gehts

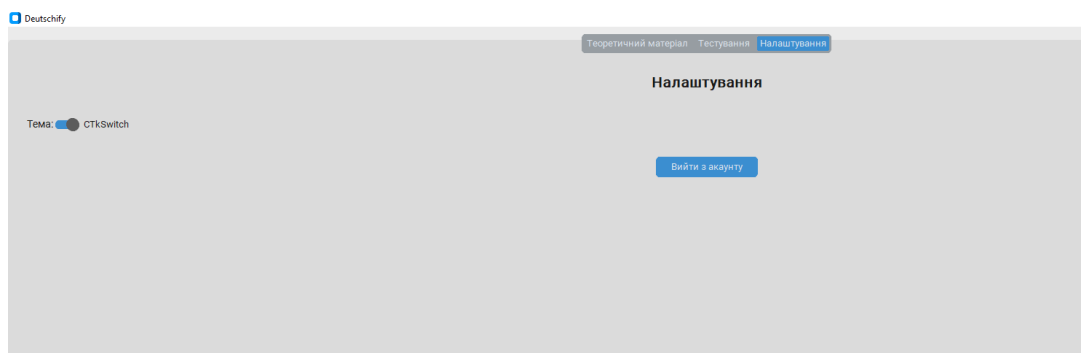
6. Вставте пропущене слово: Das ___ Oleg!

- ☐ ist
☒ war
☐ wird
☐ sein

Ви відповіли правильно на 2 з 10 питань. Ваш результат: 20.00%. Не здано.

Повторити спробу

Повторити матеріал



Додаток Б. Повний код програми

Повний код програми можна проглянути за [наступним посиланням](#).

Додаток В. Вигляд бази даних з німецькими дієсловами

		id	verb_id	tense_id	person_id	form
Фи...		Фильтр	Фильтр	Фильтр	Фильтр	
		1	1	1	1	bin
		2	1	1	2	bist
		3	1	1	3	ist
		4	1	1	4	sind
		5	1	1	5	seid
		6	1	1	6	sind
		7	2	1	1	habe
		8	2	1	2	hast
		9	2	1	3	hat
		10	2	1	4	haben
		11	2	1	5	habt
		12	2	1	6	haben
		13	3	1	1	werde
		14	3	1	2	wirst
		15	3	1	3	wird
		16	3	1	4	werden
		17	3	1	5	werdet
		18	3	1	6	werden
		19	4	1	1	make
		20	4	1	2	machst

Додаток Г. Повна база даних з німецькими дієсловами

Повну базу даних з німецькими дієсловами можна проглянути за [наступним посиланням](#).