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

РОЗРОБЛЕННЯ ЕЛЕКТРОННОГО ЗАСОБУ НАВЧАННЯ МОРФОЛОГІЇ ІМЕННИКІВ ПОЛЬСЬКОЇ МОВИ

Кваліфікаційна робота бакалавра
студентки IV курсу
ОПП «Прикладна (комп'ютерна) лінгвістика та англійська мова»,
спеціальності 035 «Філологія»,
спеціалізації 035.10 «Прикладна лінгвістика»,
галузі знань 03 «Гуманітарні науки»
Вікторії МАССУР

Науковий керівник:
к.техн.н., доц. кафедри
інформаційних систем та технологій
Микола КОСТІКОВ

«Допущено до захисту»
Протокол засідання кафедри
української мови та прикладної лінгвістики
№15 від 28.05.2025 року
Завідувач кафедри _____ **Сергій РІЗНИК**

КИЇВ-2025

АНОТАЦІЯ

Дослідження присвячено розробці електронного засобу навчання (ЕЗН) "RzeczFlex", орієнтованого на опанування теми відмінювання іменників польської мови. Актуальність дослідження зумовлена зростанням попиту на вивчення польської серед українськомовних користувачів, що обумовлено освітніми, професійними та соціальними факторами. Об’єктом дослідження є словозміна іменників в польській мові, предметом — способи реалізації інтерактивного навчання граматиці через програмні засоби.

Метою роботи є створення десктопного додатка, який поєднує теоретичний та практичний компоненти навчання та забезпечує ефективне засвоєння відмінкових форм найчастотніших польських іменників. Для досягнення мети поставлено завдання: систематизувати граматичний матеріал з теми дослідження, проаналізувати існуючі ЕЗН, розробити функціональну структуру додатка та реалізувати заплановані функції.

Робота складається з чотирьох розділів. У першому розділі подано огляд електронних засобів навчання, визначено їх класифікацію та проведено аналіз популярних додатків для вивчення польської мови. Другий розділ присвячено детальному аналізу граматичного матеріалу з відмінювання іменників польської мови. У третьому розділі описано проєктування додатка, визначено цільову аудиторію, архітектуру та функціональні можливості програми. Четвертий розділ охоплює практичну реалізацію додатка "RzeczFlex", включаючи створення баз даних, розроблення інтерфейсу, модулів тестування та оцінювання ефективності створеного програмного продукту. Дослідження включає лінгвістичний аналіз, програмування та роботу з базами даних. Результатом дослідження є створений навчальний застосунок "RzeczFlex".

Ключові слова: електронний засіб навчання, польська мова, іменник, морфологія, відмінювання, програмний продукт.

ABSTRACT

The research is devoted to the development of the electronic educational tool (EET) "RzeczFlex", aimed at mastering the topic of Polish noun declension. The relevance of the study is determined by the growing demand for learning Polish among Ukrainian-speaking users, which is caused by educational, professional, and social factors. The object of the study is noun inflection in the Polish language, and the subject is the ways of implementing interactive grammar learning through software tools.

The purpose of the work is to create a desktop application that combines theoretical and practical components of learning and ensures effective acquisition of the case forms of the most frequent Polish nouns. To achieve the purpose, the following tasks have been set: to systematize the grammatical material on the research topic, analyze existing electronic educational tools, develop the functional structure of the application, and implement the planned features.

The work consists of four sections. The first section provides an overview of electronic educational tools, defines their classification, and analyzes popular applications for learning Polish. The second section is devoted to a detailed analysis of grammatical material on the declension of Polish nouns. The third section describes the design of the application, defines the target audience, architecture, and functional capabilities of the program. The fourth section covers the practical implementation of the "RzeczFlex" application, including database creation, interface development, testing modules, and assessment of the effectiveness of the developed software product. The research includes linguistic analysis, programming, and working with databases. The result of the research is the created educational application "RzeczFlex".

Keywords: electronic educational tool, Polish language, noun, morphology, declension, software product.

ЗМІСТ

ВСТУП.....6
РОЗДІЛ 1. ПОНЯТТЯ ЕЛЕКТРОННОГО ЗАСОБУ НАВЧАННЯ ТА ОГЛЯД ІСНУЮЧИХ ЕЗН ДЛЯ ВИВЧЕННЯ СЛОВОЗМІНИ ІМЕННИКІВ ПОЛЬСЬКОЇ МОВИ.....8
1.1. Електронні засоби навчання: поняття та класифікація.....8
1.2. Вимоги до ЕЗН.....10
1.3. Огляд електронних засобів навчання польської мови та рівень покриття ними теми словозміни іменників польської мови.....12
Висновки до розділу 1:.....15
РОЗДІЛ 2. ВІДМІНЮВАННЯ ІМЕННИКІВ ПОЛЬСЬКОЇ МОВИ.....16
2.1 Іменник у польській мові та його граматичні категорії.....16
2.2. Правила відмінювання іменників польської мови.....17
2.2.1. Називний відмінок.....17
2.2.2. Родовий відмінок.....19
2.2.3. Давальний відмінок.....22
2.3.4. Знахідний відмінок.....23
2.3.5. Орудний відмінок.....24
2.3.6. Місцевий відмінок.....25
2.3.7. Кличний відмінок.....26
Висновки до розділу 2:.....27
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ.....28
3.1. Цільова авдиторія додатка.....28
3.2. Функціональні можливості.....29
3.3. Архітектура та логіка додатка.....30
Висновки до розділу 3.....34
РОЗДІЛ 4. РОЗРОБЛЕННЯ ДОДАТКА.....35
4.1. Створення та робота з базами даних.....36
4.2. Інтерфейс користувача.....39
4.3. Реєстрація та авторизація.....41
4.4. Реалізація та наповнення розділу "Теоретичний матеріал".....44
4.5. Реалізація та наповнення розділу "Тестування".....47
4.5.1 Реалізація тестування називного відмінка.....49
4.5.2 Реалізація загального шаблону тестування.....50
4.5.3. Реалізація комплексного тесту.....54
4.6. Перевірка відповідей та фіксування результатів.....55

4.7. Оцінка ефективності розробленого додатка..... 58
Висновки до розділу 4.....61
ВИСНОВКИ.....63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 65
ДОДАТКИ.....69
Додаток 1..... 69
Додаток 2..... 69

ВСТУП

Сучасні процеси глобалізації та євроінтеграції України зумовлюють зростання інтересу до вивчення іноземних мов, серед яких польська посідає особливе місце. Це зумовлено не лише географічною близькістю Польщі, але й тісними взаєминами між країнами в освітній, економічній та суспільній сферах. Знання польської мови сьогодні відкриває широкі можливості для інтеграції українців у польське суспільство, що стало особливо актуальним у зв'язку з повномасштабною війною в Україні та масовим переселенням українських громадян до Польщі.

В таких умовах важливим елементом формування мовної компетентності стає ґрунтовне опанування граматичних аспектів польської мови, зокрема морфології іменників і системи їх відмінювання, що забезпечує правильність і чіткість у формулюванні думок та соціальну адаптацію мовця. Попри близькість польської та української мов і схожість систем відмінювання, існують низка граматичних аспектів, відмінності у закінченнях та особливості формотворення, що потребують ретельного опрацювання для досягнення високого рівня мовної компетенції.

Водночас традиційні підручники та методики викладання не завжди повною мірою відповідають сучасним вимогам інтерактивності, адаптивності та індивідуалізації навчального процесу. Тому особливо важливим є створення нових цифрових освітніх засобів, що здатні запропонувати користувачам ефективні способи опанування мови.

Таким чином, **актуальність** обраної теми зумовлена потребою створення нових та ефективних інструментів для вивчення граматики польської мови українськомовним користувачам в умовах стрімкого зростання інтересу до її вивчення. **Практичне значення** роботи полягає у створенні програмного засобу, який істотно спрощує засвоєння теми словозміни польських іменників, що є складною через високу флективність мови.

Мета роботи полягає у створенні десктопного електронного засобу навчання, який дозволить опанувати та закріпити граматичні особливості та парадигми відмінювання найчастотніших іменників польської мови. Для досягнення поставленої мети визначено наступні **завдання**:

- здійснити аналіз наявних цифрових інструментів навчання граматики польської мови;
- систематизувати та структурно подати теоретичний матеріал із морфології іменника польської мови;
- розширити базу даних словоформ найчастотніших польських іменників;
- розробити функціональну структуру додатка;
- реалізувати інтерфейс користувача, систему тестування знань та механізм збереження і аналізу статистики.

Об'єктом дослідження є система відмінювання іменників в польській мові. **Предметом** — спосіб реалізації навчання відмінювання польських іменників за допомогою програмних засобів.

Методологічну основу дослідження становить поєднання сучасних підходів цифрової дидактики, принципів інтерактивного й адаптивного навчання, а також методів прикладної лінгвістики. Для досягнення поставлених завдань у роботі використано методи аналізу наукової й навчальної літератури, порівняльного огляду наявних електронних освітніх засобів, проєктування та програмування навчального застосунку, а також аналізу його функціонування в умовах практичного тестування.

РОЗДІЛ 1. ПОНЯТТЯ ЕЛЕКТРОННОГО ЗАСОБУ НАВЧАННЯ ТА ОГЛЯД ІСНУЮЧИХ ЕЗН ДЛЯ ВИВЧЕННЯ СЛОВОЗМІНИ ІМЕННИКІВ ПОЛЬСЬКОЇ МОВИ

1.1. Електронні засоби навчання: поняття та класифікація

Електронні засоби навчання (ЕЗН) — це програмно-апаратні засоби, створені для підтримки та організації навчального процесу за допомогою інформаційно-комунікаційних технологій [1].

Традиційні та електронні засоби навчання мають спільну мету — забезпечення ефективного засвоєння знань, однак різняться формами подання інформації, способом взаємодії з учнем та організацією навчального процесу. Традиційні засоби, тобто підручники, зошити, друковані посібники тощо, орієнтовані переважно на статичне подання матеріалу й потребують безпосередньої участі викладача. До ЕЗН відносяться інтерактивні підручники, тренажери, навчальні ігри, системи тестування, електронні курси та інші цифрові продукти, що забезпечують інтерактивність, мультимедійність і можливість самостійної або дистанційної роботи. Вони дозволяють адаптувати навчальний процес до індивідуальних потреб здобувачів освіти, автоматизувати оцінювання та зворотний зв’язок, інтегрувати різні типи контенту (текст, відео, анімацію), що значно підвищує мотивацію й залученість. Водночас ефективне використання ЕЗН вимагає технічної бази, цифрової грамотності та продуманого педагогічного супроводу [2] [10].

Таким чином, метою використання ЕЗН є підвищення ефективності навчання, індивідуалізація освітнього процесу, забезпечення доступу до навчальних матеріалів незалежно від часу і місця студента та розвиток цифрових компетентностей учнів.

Детальну класифікацію електронних засобів навчання подає Д. В. Чернилевський, визначаючи їх як програмні засоби навчального призначення, у яких відображена певна предметна галузь, реалізовано технологію її вивчення, а

також створені умови для здійснення різних видів навчальної діяльності [3]. На основі методичного призначення вчений виокремлює кілька категорій електронних засобів навчання:

1. Тренажери — програмні рішення, що дають змогу учневі або студенту набувати досвід через повторне виконання навчальних дій, самостійне відпрацювання алгоритмів і рішень;
2. Контрольно-діагностичні засоби — застосовуються для здійснення поточного або підсумкового контролю знань, а також для самоперевірки;
3. Інформаційно-довідкові ресурси — підтримують навички орієнтації у великому обсязі навчальної інформації, сприяють її систематизації та швидкому пошуку;
4. Імітаційні системи — моделюють окремі аспекти реального світу на основі заданих параметрів, що дозволяє користувачу аналізувати певні закономірності або функціональні зв’язки;
5. Моделювальні засоби — забезпечують можливість побудови умовної або віртуальної моделі об’єкта, явища чи процесу, що є предметом вивчення;
6. Візуалізаційні (демонстраційні) засоби — слугують для ілюстративного представлення навчального матеріалу, допомагаючи краще зрозуміти складні процеси та структури за допомогою графічних чи мультимедійних елементів;
7. Освітні ігрові продукти — поєднують навчальний зміст із елементами гри, стимулюючи інтерес до предмета і залучаючи здобувача освіти до активної пізнавальної діяльності;
8. Розвивальні програмні засоби — орієнтовані на організацію самостійної чи позааудиторної діяльності, спрямованої на розвиток когнітивних функцій (уваги, пам’яті, реакції, логічного мислення тощо);
9. Засоби навчального призначення — спрямовані на подання узагальненої системи знань, формування практичних умінь і навичок, а також

забезпечення зворотного зв’язку для перевірки рівня засвоєння матеріалу [3].

Хоча існує багато інших класифікацій, на наш погляд, саме ця дає змогу більш гнучко адаптувати електронні ресурси до конкретних освітніх потреб, враховуючи цілі, завдання та зміст навчального процесу.

Згідно цієї класифікації, наш проєкт входитиме до категорій тренажери й інформаційно-довідкові ресурси, оскільки вичерпно покриватиме тему словозміни іменників польської мови та надаватиме змогу закріпити опрацьований матеріал у форматі тестування.

1.2. Вимоги до ЕЗН

Якісний ЕЗН це не лише технічно справний програмний продукт, а повноцінний педагогічний інструмент, здатний забезпечити ефективний, гнучкий і результативний освітній процес. Успішність його застосування значною мірою залежить від дотримання цілого комплексу вимог, які охоплюють педагогічний, дидактичний, технічний та психологічний аспекти.

Звісно, педагогічна доцільність є однією з ключових умов створення ефективного електронного засобу. Вміст такого засобу повинен відповідати освітнім стандартам та навчальним планам з конкретної дисципліни, якщо це освітній продукт для державної навчальної установи, або загальній логіці викладу матеріалу з певної теми чи галузі, залежно від мети створення ЕЗН [5][24].

Контент має бути підібраний з урахуванням вікових, пізнавальних і психологічних особливостей цільової авдиторії. Структура ЕЗН повинна передбачати послідовне викладення навчального матеріалу: від пояснення теоретичних положень —- до їх закріплення через практичні завдання, самоперевірку та підсумковий контроль знань [5] [12][30].

Дидактична ефективність електронного ресурсу визначається його здатністю сприяти досягненню цілей навчання. Зміст повинен бути логічно

структурованим, чітко сформульованим і адаптованим до сприйняття. Важливо, щоб ЕЗН забезпечував можливість формування навичок аналізу, узагальнення та критичного мислення. Ефективний засіб має також забезпечувати зворотний зв’язок. Це можуть бути автоматизовані тести, інтерактивні вправи, підказки, пояснення до відповідей, можливість перегляду результатів та розбір помилок [4][29].

З технічного погляду, електронний засіб навчання має бути стабільним, зручним та безпечним. Його робота не повинна супроводжуватися частими помилками чи збоєм функціонування. Якщо це серйозний комерційний продукт, важливо, щоб він був оптимізований для роботи на різних пристроях та підтримував основні операційні системи. Для відкритих продуктів захист персональних даних користувача є обов’язковою умовою [10][31].

Інтерфейс має бути інтуїтивно зрозумілим, доступним навіть для користувачів із базовим рівнем цифрової грамотності. Візуальне оформлення повинно бути стриманим, з достатнім контрастом між текстом і фоном, чітким шрифтом, зручною навігацією та логічною структурою меню. Також варто уникати надмірної кількості анімацій чи інших ефектів, оскільки вони можуть відволікати або викликати втому [5] [10][24].

Якщо додаток зорієнтований на покриття багатьох аспектів мови, варто зробити його мультимедійним, тобто поєднати різні засоби подання інформації (текст, зображення, відео, аудіо), що сприятиме кращому запам'ятовуванню. Утім, використання мультимедійного контенту повинно бути вмотивованим і педагогічно виправданим, бо надмірна кількість візуальних чи аудіо елементів може ускладнити сприйняття інформації [4][10][24].

Адаптивність ЕЗН до потреб користувача є важливою ознакою його якості. Сучасний засіб має враховувати індивідуальний рівень знань та темп роботи кожного учня [5].

Іншою важливою вимогою є наявність інструментів контролю та моніторингу навчальних досягнень. Якісний ЕЗН має передбачати можливість

тестування, відстеження результатів, аналізу типових помилок та огляд прогресу. Це дозволяє як викладачеві, так і самому учневі оцінити рівень засвоєння матеріалу та виявити прогалини, над якими варто працювати [4] [13].

Отже, якісний електронний засіб навчання — це гармонійне поєднання методичної продуманості, технічної надійності, зручного інтерфейсу, педагогічної цінності та відповідності сучасним вимогам цифрового освітнього продукту.

У межах нашої роботи було реалізовано низку ключових вимог до якісного електронного засобу навчання. Зокрема, забезпечено педагогічно обґрунтовану подачу матеріалу на основі граматичних джерел та частотного словника польських іменників; структура застосунку вибудувана логічно та послідовно — від подання теоретичних положень до системи перевірки знань. Дидактична ефективність досягається завдяки використанню різних типів тестових завдань. Програмний засіб відзначається стабільністю роботи, невибагливістю до технічних ресурсів і зручним, інтуїтивно зрозумілим інтерфейсом. Крім того, додаток містить інструменти для контролю та моніторингу навчальних досягнень, що сприяє формуванню повноцінного освітнього середовища.

1.3. Огляд електронних засобів навчання польської мови та рівень покриття ними теми словозміни іменників польської мови

У сучасному світі електронні засоби навчання стали невід’ємною частиною процесу здобуття знань. Зростаюча популярність польської мови, особливо серед українців, сприяє активному зростанню інтересу до її вивчення, що, своєю чергою, стимулює появу польської у великій кількості електронних додатків та платформ для вивчення мов. Проте важливо відзначити, що більшість існуючих ЕЗН обмежуються базовим викладом граматики, приділяючи недостатню увагу детальному опрацюванню словозміни, зокрема темі відмінювання іменників. Розглянемо кілька популярних додатків для

вивчення мови, щоб дослідити, якою мірою в них розкривається тема нашого дослідження (табл. 1.1).

1. Duolingo

Duolingo — одна з найпоширеніших мовних платформ, у якій представлено курс польської мови. Програма пропонує інтерактивні уроки, що охоплюють основні мовні аспекти, однак головний акцент зроблено на вивченні лексики та типових фраз. Морфологія, як частина граматики, подається фрагментарно у рамках вправ різного рівня складності. Основна ідея платформи полягає в інтуїтивному засвоєнні мови через практику, без акценту на правила, тому у повній та необхідній мірі тема словозміни не розкрита [6].

2. Babbel

На відміну від Duolingo, додаток Babbel пропонує більш системний підхід до вивчення польської мови. Курси структуровані за тематиками, що охоплюють граматику, лексику та розмовну мову. У програмі представлено морфологічні теми, зокрема відмінювання іменників, дієслів і прикметників. Babbel подає граматичний матеріал у вигляді коротких пояснень і практичних завдань, що дозволяє вивчати теми, як-от орудний відмінок, окремими блоками. Проте, як і в Duolingo, головний акцент залишається на комунікативних навичках, тому морфологія розглядається в обмеженому обсязі. За відгуками на форумі Reddit [17], багато користувачів вважають Babbel найбільш комплексним додатком для вивчення польської мови, зокрема граматичних тем [7].

3. Mondly

Платформа Mondly орієнтована переважно на розвиток базових комунікативних навичок. Хоча в ній присутні граматичні модулі, зокрема теми, пов’язані з відмінюванням польських іменників, їх обсяг і глибина вкрай обмежені. Такий рівень опрацювання не дозволяє початківцям повноцінно осягнути принципи словозміни іменників [8].

4. Busuu

Додаток Busuu демонструє більш структурований підхід до викладу морфології польської мови порівняно з Mondly. Уроки містять вправи на зміну форм іменників відповідно до контексту, що дозволяє користувачеві практикувати відмінювання. Також у курс включено завдання для закріплення матеріалу щодо дієслів і прикметників. Водночас, попри наявність граматичних пояснень, вони не є достатньо глибокими, що може ускладнити засвоєння морфологічних структур. Загалом, Busuu забезпечує більш збалансований підхід до навчання польської мови, але все ж обмежується у висвітленні теоретичних аспектів [9].

	Duolingo	Babbel	Mondly	Busuu
Зручність інтерфейсу	+	-	+	-
Наявність граматичних тем	Мінімально	+	Дуже обмежено	+
Покриття теми словозміни	-	Поверхневе	Мінімальне	Недостатньо детальне
Основний акцент	Лексика	Комунікація, граматика	Комунікація	Комунікація, граматика, лексика
Наявність вправ для практики словозміни	-	+	Дуже обмежено	+

Таблиця 1.1. Порівняння популярних додатків для вивчення польської мови

Окрім уже згаданих платформ, існує низка менш відомих додатків, які, втім, здебільшого зосереджені на окремих мовних компонентах, переважно лексичних [25].

Таким чином, тема словозміни в польській мові, яка є однією з найскладніших для опанування, особливо для студентів, які не володіють жодною слов'янською мовою, залишається недостатньо висвітленою в проаналізованих додатках. Навіть для українськомовних студентів, попри спорідненість мов, польська морфологія містить низку специфічних відмінностей, що потребують ретельного опрацювання та глибокого розуміння [22]. З огляду на високий ступінь флективності польської мови та складну систему відмінювання, ефективне її засвоєння неможливе без системного й послідовного вивчення принципів словозміни. Тому наш програмний продукт орієнтований на комплексне представлення граматичних особливостей відмінювання іменників польської мови.

Висновки до розділу 1:

Отже, у цьому розділі ми розглянули поняття та класифікацію електронних засобів навчання, окреслили сучасні вимоги до якісних ЕЗН і здійснили огляд найпоширеніших електронних платформ та додатків для вивчення польської мови. Проаналізувавши, як саме подається тема словозміни іменників у таких засобах, як Duolingo, Babbel, Mondly, Busuu, ми визначили, що жоден із них не забезпечує достатньої деталізації цієї складної теми. Окремі ресурси містять вправи для тренування відмінкових форм, проте їм бракує теоретичних пояснень і цілісного системного підходу. Більшість популярних цифрових ресурсів акцентують увагу переважно на лексиці, навичках усного мовлення або окремих граматичних правилах, тоді як відмінювання іменників або подається поверхнево, або взагалі залишається поза увагою.

Таким чином, проведений аналіз засвідчив важливість і актуальність розроблення спеціалізованого електронного засобу навчання, який би відповідав сучасним вимогам до ЕЗН і логічно й послідовно поєднував теоретичний матеріал із практикою та контролем знань.

РОЗДІЛ 2. ВІДМІНЮВАННЯ ІМЕННИКІВ ПОЛЬСЬКОЇ МОВИ

Для розроблення електронного засобу навчання, спрямованого на опанування словозміни іменників польської мови, необхідно спочатку детально проаналізувати теоретичні основи цієї теми. Зібраний і систематизований матеріал стане базою для наповнення відповідних розділів теоретичної частини електронного засобу навчання.

2.1 Іменник у польській мові та його граматичні категорії

Іменник (пол. rzeczownik) — це одна з головних частин мови в польській, що позначає предмети, явища, осіб, тварин, а також статуси, професії тощо [19]. У польській мові іменники належать до змінюваних частин мови (пол. odmienne części mowy). Вони змінюються за відмінками (пол. przypadki), що називається відмінюванням (пол. deklinacja). Важливо не плутати із дієвідмінюванням (пол. koniugacja), що стосується дієслів [20].

І польська, і українська мови мають спільну систему відмінків та подібні запитання до них. Це значно полегшує вивчення мови для українськомовних студентів [23]. У польській налічується сім відмінків (пол. siedem przypadków):

1. Mianownik (укр. називний відмінок) – kto? co? (хто? що?)
2. Dopełniacz (укр. родовий відмінок) – kogo? czego? (кого? чого?)
3. Celownik (укр. давальний відмінок) – komu? czemu? (кому? чому?)
4. Biernik (укр. знахідний відмінок) – kogo? co? (кого? що?)
5. Narzędnik (укр. орудний відмінок) – kim? czym? (ким? чим?)
6. Miejscownik (укр. місцевий відмінок) – o kim? o czym? (на кому? на чому?)
7. Wołacz (укр. кличний відмінок) – hej! [17][18]

Іменники змінюються за:

1. **Родом** (пол. rodzaj): чоловічий (пол. męski), жіночий (пол. żeński) та середній (пол. nijaki).

Також іменники поділяються на категорії істот (пол. istoty) та неістот (пол. nieistoty). До категорії істот належать іменники, що позначають людей та живих

істот. Натомість до категорії неістот відносяться всі інші іменники — ті, що називають предмети, явища, поняття тощо.

Цей поділ особливо важливий для іменників чоловічого роду (пол. rodzaj męski), адже у родовому та знахідному відмінках однини вони можуть мати різні закінчення залежно від того, чи є істотою, чи ні.

2. **Числом** (пол. liczba): однина (пол. liczba pojedyncza) та множина (пол. liczba mnoga).

У польській мові множина має дві окремі форми, залежно від того, хто або що є підметом:

2.1 Чоловічо-особова форма (пол. forma męskoosobowa) — для груп, що включають щонайменше одного чоловіка.

2.2 Нечоловічо-особова форма (пол. forma niemęskoosobowa), до якої відносяться групи жінок, предметів, тварин тощо.

Як і в українській мові, деякі іменники мають лише множину (наприклад, drzwi, okulary). Також для польської характерне чергування звуків в основі (корені) слова, яке ми розглянемо далі [18][20][23].

2.2. Правила відмінювання іменників польської мови

У цьому розділі ми докладно розглянемо правила відмінювання іменників польської мови з метою формування цілісного уявлення про особливості їхньої словозміни, а також для створення теоретичної бази, яка стане основою для наповнення та побудови функціоналу розробленого електронного засобу навчання. Аналіз і систематизація ключових закономірностей, винятків і типових труднощів відмінювання є необхідною передумовою для подальшого практичного застосування матеріалу у програмі.

2.2.1. Називний відмінок

Іменники ***у формі однини*** називного відмінку мають наступні характерні закінчення:

Іменники чоловічого роду зазвичай закінчуються на приголосну. Рідше зустрічається закінчення -a.

Для жіночого роду характерними є закінчення -a або -i, але деякі слова закінчуються на приголосну (наприклад, noc).

Іменники середнього роду закінчуються на -o, -e, або -um. Також деякі іменники закінчуються на -ę (наприклад, imię, ciele̜).

Складнішим є відмінювання ***у множині*** називного відмінку:

Чоловічий рід

Характерним для іменників чоловічого роду є закінчення -owie для іменників на позначення родинних зв'язків, імен, назв професій, національностей та титулів (наприклад, profesor – profesorowie).

Після твердих приголосних, крім -k, -g, -r, додається -i (наприклад, blondyn – blondyni).

Іменники, основи яких закінчуються в однині на -k, -g(a), -r, -iec, приймають закінчення -y і чергуються наступним чином:

1. -g- змінюється на -dz-;
2. -k- змінюється на -c-;
3. -r- змінюється на -rz-;
4. -iec змінюється на -c-.

Наприклад: kolega – koledzy, chłopiec – chłopcy.

Якщо в однині іменник чоловічого роду закінчується на -a, тоді -a опускається, попередній приголосний пом'якшується і відбувається чергування:

1. -t- пом'якшується до -ci;
2. -st- пом'якшується до -ści;
3. -z- пом'якшується до -zi (ź).

Також основи в однині на -d, -ch, приймають закінчення -i у множині (наприклад, Szwed – Szwedzi, Włoch – Włosi). Тут варто звернути увагу на чергування:

1. -d- змінюється на -dz-;

2. -ch- змінюється на -s-.

Чоловічо-особові іменники, які закінчуються в однині на шиплячий та м’який приголосний (-c, -cz, -sz, -ść та -l, -rz) приймають закінчення -e (наприклад: gracz – gracze, lekarz – lekarze).

Винятки**:** brat – bracia; ksiądz – księża; człowiek – ludzie.

Жіночий рід

Іменники жіночого роду в множині мають закінчення, аналогічні закінченням нечоловічо-особових іменників.

Якщо в однині іменник має закінчення -a в однині, то у множині змінюється на -i після твердих приголосних k та g (наприклад, apteka – apteki). Після інших твердих приголосних використовується закінчення -y (наприклад, kobieta – kobiety).

Основи на м'який приголосний або шиплячі (-ń, -sz, -cz, -rz, -ż, -l, -j, -c, -dz, -ś, -ć, -ź) отримують закінчення -e (наприклад, pani – panie, ulica – ulice, pokój – pokoje).

Середній рід

Більшість іменників середнього роду в називному відмінку множини отримують закінчення -a (наприклад, drzewo – drzewa).

Виняток: dziecko – dzieci (діти).

Іменники середнього роду, що закінчуються на -ę в однині, змінюють основу у множині. При закінченні на -mię основа змінюється на -miona (наприклад, imię – imiona).

В інших випадках -ę перетворюється на -ęta (наприклад, dziewczę – dziewczęta) [18][20].

2.2.2. Родовий відмінок

Родовий відмінок у польській мові, подібно до української, використовується для вираження різних типів означальних відношень. Зазвичай

іменник у родовому відмінку слідує за іншим іменником або групою іменників, до яких він належить або які уточнює.

Родовий відмінок вимагають прийменники obok (укр. поруч / біля), bez (укр. без), od (укр. від), do (укр. до), z (укр. з / із), dla (укр. для), naprzeciwko (укр. навпроти). Також деякі дієслова, такі як szukać (укр. шукати), słuchać (укр. слухати), потребують після себе родового відмінка.

Іменники ***в однині*** формуються наступним чином:

Чоловічий рід

Всі іменники на позначення назв істот, тобто такі, які позначають людей або тварин, закінчуються на -a (наприклад, kot - kota; lekarz - lekarza).

Також закінчення -a мають іменники, які означають предмет і закінчуються на м'який приголосний (крім тих, що мають у закінченні -j) або на -yk, -ik (наприклад, pień – pnia, koszyk – koszyka).

Місяці, позначення мір і ваг, інструменти, більшість назв польських міст, мають закінчення -a (наприклад, kilogram – kilograma, Kraków – Krakowa; nóż – noża).

Закінчуються на -u запозичені іменники (наприклад, hotel – hotelu, klub – klubu), назви транспортних засобів (наприклад, tramwaj – tramwaju, rower – roweru), днів тижня (наприклад, czwartek – czwartku, poniedziałek – poniedziałku), незлічувані іменники (наприклад, miód – miodu, piasek – piasku) та інші іменники-неістоти, що не відносяться до категорій попереднього пункту, означають предмети чи абстрактні поняття, а також що закінчуються на -j (наприклад, teatr - teatru, Londyn - Londynu, pokój – pokoju).

Виняток: maj – maja.

З вище наведених прикладів бачимо, що родовому відмінку також характерне чергування голосних, наприклад:

1. -e- випадає;
2. -ó- чергується на -o-;.

Жіночий рід

Для жіночого роду закінчення -y вживається після твердого приголосного (наприклад, cisza – ciszy).

Після приголосних -l-, -k-, -g- вживається закінчення -i (наприклад, książka – książki).

Середній рід

Якщо іменник середнього роду закінчується на -o или -e в називному відмінку, то в родовому відмінку він має закінчення -a (наприклад, oko – oka).

Іменники із закінченням -ę в називному відмінку, в родовому мають закінчення -ia (наприклад, imię – imienia, kurczę – kurczęcia).

Розглянемо закінчення ***множини*** родового відмінка.

Чоловічий рід

Закінчення чоловічого роду залежать від останнього приголосного кореня слова. Більшість іменників закінчуються на -ów (наприклад, sąsiad – sąsiadów). Після м'якого приголосного вживається -i (наприклад, gość – gosci).

Винятки: kraj – krajów.

Іменники, корінь яких закінчується на -cz-, -ż- (-rz-), -sz-, -dz-, -l-, і деякі іменники, що закінчуються на -c, мають закінчення -y (наприклад, pisarz – pisarzy, tysiąc – tysięcy).

Жіночий рід

Іменники жіночого роду, основа яких закінчується на твердий або м'який приголосний та які мають закінчення -a чи -i, зазвичай втрачають закінчення (наприклад, noga – nóg). При збігові двох приголосних між ними з'являється випадне -e- (наприклад, bułka – bułek).

Іменники жіночого роду, які мають нульове закінчення у називному відмінку однини, у родовому множини набувають закінчення -i (наприклад, pamięć – pamięci).

Середній рід

Іменники середнього роду на -e, -o у родовому відмінку множини також втрачають закінчення, а передостання голосна в цих словах може опускатися

або змінюватись на іншу (наприклад, zdjęcie – zdjęć, jabłko – jabłek, okno – okien, święto – świąt).

Іменники середнього роду, що закінчуються на -um, в родовому відмінку мають закінчення -ów (наприклад, muzeum – muzeów).

Іменники на -ę в родовому відмінку множини мають такі форми: imię – imion, cielę – cieląt, zwierzę –zwierząt [18][20].

2.2.3. Давальний відмінок

Давального відмінка потребують прийменники ku (укр. до), dzięki (укр. завдяки), przeciw (укр. проти). Також з давальним відмінком сполучаються дієслова dać (укр. дати), dziękować (укр. дякувати), pomagać (укр. допомагати) тощо.

Чоловічий рід

Іменники чоловічого роду ***в однині*** у давальному відмінку зазвичай мають закінчення -owi, яке додається до форми називного відмінка (наприклад, syn – synowi).

Однак деякі іменники, переважно односкладові, мають закінчення -u (наприклад, brat – bratu).

Іменники чоловічого роду, що в називному відмінку закінчуються на -a, відмінюються за зразком жіночого роду, що буде описаний далі.

Чоловічий та жіночий роди однини

Іменники чоловічого й жіночого роду на -a з основою на твердий приголосний у давальному відмінку ***однини*** мають закінчення -e, при цьому приголосний у кінці основи зазнає пом’якшення (наприклад, kolega – koledze).

Іменники жіночого роду на -a та -i з основою на м’який приголосний у давальному відмінку однини отримують закінчення -i (наприклад, pani – pani, pamięć – pamięci).

Якщо ж основа закінчується на стверділий приголосний, вони мають закінчення -y (наприклад, róża – róży).

Іменники в давальному відмінку однини мають закінчення -u (наприклад, oko – oku). Іменники на -ę мають закінчення -iu (наприклад, imię – imieniu).

Усі іменники в давальному відмінку ***множини*** закінчуються на -om.

Для давального відмінку також характерне чергування звуків в кореневих частинах:

1. ch – sz (наприклад, mucha – musze)
2. de – dzie (наприклад, woda – wodzie)
3. g – dz (наприклад, kolega – koledze)
4. k – c (наприклад, matka – matce)
5. r – rz (наприклад, siostra – siostrze)
6. zd – ździe (наприклад, gwiazda – gwieździe)
7. dz – g (наприклад, koledzy – kolegom)
8. ć – t (наприклад, klienci – klientom) [18][20]

2.3.4. Знахідний відмінок

Форми знахідного відмінка використовуються після дієслів mieć (укр. мати), lubić (укр. любити), jeść (укр. їсти), kupować (укр. купувати), czytać (укр. читати), znać (укр. знати), oglądać (укр. дивитися), prosić (укр. просити).

Для іменників ***в однині*** діють наступні правила:

Іменників-неістоти чоловічого роду, іменники середнього роду і всі іменники жіночого роду із закінченням на приголосний мають однакові форми називного і знахідного відмінків (наприклад, dom – dom).

Чоловічий рід

Всі іменники чоловічого роду на позначення істот в однині мають закінчення -a (наприклад, chłop – chłopa).

Жіночий рід

Якщо іменник жіночого роду в однині називного відмінка закінчується на голосний, то в знахідному він має закінчення -ę (наприклад, ulica – ulicę). Виняток: pani – panią.

Іменники ***множини*** змінюються за такими правилами:

Закінчення знахідного відмінка множини у іменників чоловічого роду на позначення тварин та предметів, жіночого та середнього роду співпадають із закінченнями називного відмінка множини.

Чоловічий рід

Іменники-назви істот чоловічого роду (*лише на позначення людей*) у знахідному відмінку множини здебільшого мають закінчення -ów (наприклад, aktor – aktorów), що співпадає із закінченням у родовому відмінку. Ті з них, що в називному відмінку множини закінчуються на -e, -ie, -ia, можуть мати закінчення -y або -i (наприклад, lekarze – lekarzy) [18][20].

2.3.5. Орудний відмінок

З орудним відмінком використовуються прийменники z (укр. з), nad (укр. над), pod (укр. під), między (укр. між). Також дієслова dojechać (укр. доїхати), pojechać (укр. поїхати), jechać (укр. їхати) вимагають постановки залежних іменників в орудному відмінку. Переважно ці іменники відносяться до засобів пересування (транспорту).

Правила відмінювання ***в однині*** орудного відмінка:

Чоловічий та середній роди

Іменники чоловічого роду з закінченням -a в називному відмінку однини переважно закінчуються на -ą в орудному.

Деякі іменники чоловічого роду та іменники середнього роду мають закінчення -em. Якщо основа закінчується на -k або -g, то приголосний пом’якшується з додаванням літери -iem.

Для жіночого роду *однини* закінчення змінюється на -ą.

У більшості випадків іменники в орудному відмінку ***множини*** мають закінчення -ami (наприклад, panami).

Однак існує невелика група іменників, основа яких закінчується на м’який або стверділий приголосний — у таких випадках вживається закінчення -mi (наприклад, dziećmi) [18][20].

2.3.6. Місцевий відмінок

Місцевий відмінок використовується після прийменників na (укр. на), w (укр. в/у), o (укр. про), przy (укр. біля/при).

Правила відмінювання ***в однині*** місцевого відмінка:

Чоловічий та середній роди

Якщо основа закінчується на тверду приголосну, то закінчення -e (наприклад, okno – na oknie, sklep – w sklepie). Пом’якшується приголосний за допомогою -i-, тобто утворенням закінчення -ie.

Якщо основа леми закінчується на м’який приголосний, -c-, -dz-, -cz-, -dż-, -l-, -rz-, -sż-, -ż- або на -k-, -g-, -ch-, то в місцевому відмінку закінчення -u. Винятки: pan – panu , syn – synu, dom – domu.

Деякі іменники мають у корені чергування -a- з -e- (наприклад, miasto – mieście).

Жіночий рід

Коли основа слова закінчується на тверду приголосну, то закінчення -e. Для місцевого відмінку також типове чергування звуків:

1. ch – sz (наприклад, mucha – o musze)
2. d – dzie (наприклад, woda – w wodzie)
3. g – dz (наприклад, noga – o nodze)
4. k – c (наприклад, matka – o matce)
5. ł – l (наприклад, stół – na stole)
6. r – rz (наприклад, rower – na rowerze)
7. sł – śl (наприклад, krzesło – na krześle)
8. st – ście (наприклад, miasto – w mieście)
9. t – cie (наприклад, uniwersytet – na uniwersytecie)

10.zd – ździe (наприклад, gwiazda – o gwieździe)

Якщо основа закінчується на м’який приголосний, то слово отримує закінчення -і (наприклад, pani – pani).

Слова, основи яких закінчується на -sz-, -cz-, -dz-, -dż-, -ż-, -rz-, -c- мають закінчення -у (наприклад, noc – nocy).

У множині місцевого відмінка, незалежно від роду, всі іменники закінчуються на -ach (наприклад, pan – panach; drzewo – drzewach).

Винятки: Niemcy – w Niemcech, Węgry – na Węgrzech, Włochy – we Włoszech, oko – oczach, imię – imionach, cielę – cielętach [18][20].

2.3.7. Кличний відмінок

Характерною рисою є наявність кличного відмінка у польській мові, що використовується для звертання. Кличний відмінок має форми лише для іменників чоловічого та жіночого роду в однині. У всіх інших випадках, тобто для однини середнього роду та множини, кличний відмінок має таку ж форму як і в називному відмінку. Також у неформальному спілкуванні часто замість кличного відмінка вживається форма називного, особливо для імен. Якщо при звертанні ім’я вживається зі словами pan чи pani, то воно ставиться в кличному відмінку (наприклад, panie Pawle). З прізвищами вживається тільки називний відмінок однини (наприклад, pani Massur).

Правила відмінювання ***в однині*** кличного відмінка:

Чоловічий рід

Іменники чоловічого роду із закінченням на -а змінюють закінчення -о (наприклад, tato).

Якщо основа на твердий приголосний (-b-, -p-, -n-, -m-, -w-, -f-, -s-, -z-), іменник має закінчення -ie (наприклад, Filip – Filipie).

Також можливі чергування:

1. t – cie (наприклад, student - studencie)
2. d – dzie (наприклад, sąsiad - sąsiedzie)

3. r – rze (наприклад, dyrektor - dyrektorze)
4. ł – le (наприклад, poseł - pośle)

Якщо основа на м'який приголосний, іменники з закінченням на -k, -g, -ch та пестливі жіночі імена мають закінчення -u (Tomasz – Tomaszu, Marek – Mareku, Basia – Basiu). При закінченнях -ek- буква -e- випадає.

Якщо слово закінчується на -(i)ec, то закінчення змінюється на -cze (наприклад, chłopcze).

Винятки: Bóg – Boże; gołąb – gołębiu; ksiądz – księż.

Жіночий рід

Якщо іменник жіночого роду в називному відмінку закінчується на -ja,-la, -cia, dzia, -nia, -sia, -zia, то в кличному закінчення змінюється на -u, часто для зменшувально-пестливих слів.

Іменники на -і або на м’який приголосний мають у кличному відмінку однини закінчення -і. Іменники з твердими приголосними закінчуються на -у.

Також, як і в українській, використовується закінчення -о (наприклад, mama – mamo) [18][20].

Висновки до розділу 2:

У другому розділі ми докладно розглянули основні теоретичні положення словозміни іменників у польській мові, проаналізували їх граматичні категорії та окреслили специфіку кожного відмінка й типові особливості зміни закінчень. Було приділено увагу характерним чергуванням голосних та приголосних звуків у корені слова та виняткам у відмінюванні.

Можна помітити, що система відмінків у польській мові загалом схожа з українською — обидві мови мають сім відмінків та подібні питання до кожного з них, що значною мірою спрощує початкове опанування граматики для українськомовних студентів. Проте детальний огляд виявив низку відмінностей і специфічних моментів у темі словозміни іменників, що потребують особливої уваги під час опанування мови. Певною мірою ці відмінності зумовлені не лише

морфологічними рисами, а й різницею алфавітів: польська мова використовує латиницю з розвиненою системою діакритичних символів (ą, ę, ł, ś, ź, ż, ć, ń, ó), які відсутні в українському письмі. Це впливає як на особливості творення відмінкових форм, так і на правила правопису та вимови, створюючи додаткові труднощі для тих, хто вивчає польську мову на основі української [22][23].

Опрацювання всіх цих аспектів стало важливою теоретичною базою для подальшого формування структури навчального матеріалу, що наповнює електронний засіб навчання.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА АРХІТЕКТУРА СИСТЕМИ

3.1. Цільова авдиторія додатка

Програмний продукт призначений для українськомовних користувачів, які вивчають польську мову як іноземну. У зв’язку з актуальними соціальними обставинами та поглибленням міжкультурних зв’язків між Україною та Польщею, потреба у вивченні польської мови охоплює різні вікові категорії — від підлітків до осіб похилого віку. Тож до цільової авдиторії належать як учні шкіл і студенти, так і дорослі, зокрема особи, що планують виїзд до Польщі.

Програма буде корисною для користувачів із початковим та середнім рівнем володіння мовою (A1–B1 за шкалою CEFR) [21], які потребують систематизації знань з польської граматики.

3.2. Функціональні можливості

Для забезпечення повноцінної роботи додатка та врахування потреб користувачів у ньому реалізовано такі функціональні компоненти:

1. **Реєстрація та вхід користувача**. Передбачено створення персонального аккаунта з валідацією електронної пошти та безпечним зберіганням паролю (через хешування SHA-256). Авторизація дозволяє відстежувати результати навчання конкретного користувача [26][28][32].
2. **Теоретичний блок**. У додатку міститься структурована теоретична частина, яка охоплює іменник як основну частину мови у польській та всі сім відмінків польської мови. Наповнення розділу було здійснено із використанням теоретичного матеріалу, представленого у Розділі 2, із мінімальними правками по оформленню. Теорія подається у зручному форматі з можливістю навігації між розділами [30][36].
3. **Модуль тестування**. Реалізовано кілька типів завдань, які охоплюють різні форми контролю [29]:

- вибір правильної форми іменника серед запропонованих варіантів (Multiple Choice);
- тести формату "Так / Ні";
- вільне введення правильної форми вручну;
- комплексний тест на відповідність, де користувач повинен визначити число та відмінок заданої форми.

4. **Зворотний зв’язок після тесту**. Після завершення тестування користувач отримує результат тестування — правильні й неправильні форми з відповідними позначками, а також підсумкову оцінку. Пізніше він може переглянути загальну успішність за кожним розділом тестування у відповідному розділі загального меню [27][31].

Реалізовані можливості відповідають навчальним цілям і потребам користувачів, дозволяючи не лише отримувати нові знання, а й ефективно їх закріплювати та оцінювати індивідуальний прогрес у вивченні польської мови.

3.3. Архітектура та логіка додатка

Програмний засіб реалізовано як десктопний додаток із графічним інтерфейсом. Архітектура побудована за принципами поділу логіки на функціональні блоки, що дозволяє забезпечити чітку організацію коду. Детальніший розгляд реалізації застосунку буде представлено у наступному розділі.

Інтерфейс користувача реалізовано як систему переходів між вікнами, що створюються як окремі фрейми: стартове вікно вхід/реєстрація, вікно входу, вікно реєстрації, головне меню, перегляд теоретичних матеріалів, тестування, статистика успішності користувача та розділ про додаток. Перехід між фреймами реалізовано через функції приховування й показу відповідних елементів, що забезпечує плавну зміну контенту без потреби створення нових

вікон для кожного наступного переходу.

Окремий функціональний блок реалізує **реєстрацію та вхід користувачів**. Кожен користувач має можливість створити власний обліковий запис, у якому зберігатиметься його особиста статистика проходження тестів. Реєстрація реалізована з валідацією правильності введеного email-адресу (з використанням регулярного виразу) та перевіркою унікальності. Пароль перед збереженням у базі хешується з використанням алгоритму **SHA-256**, що гарантує безпечне зберігання даних [32][33]. У разі успішної реєстрації чи входу, збереження облікових даних здійснюється у локальній базі даних **users.db**, а в пам'яті програми зберігається змінна **id**, що ідентифікує поточного користувача. Ця змінна використовується для прив'язки результатів до конкретного облікового запису. Значення **id** у таблиці **users** відповідає полю **user_id** у таблиці **stats**, що дозволяє відстежувати індивідуальні статистичні показники кожного користувача.

Зберігання та обробка даних організовані через дві окремі локальні бази даних **SQLite** [14]. Перша база **users.db** містить облікові записи користувачів (електронна пошта та хешований пароль) і таблицю статистики **stats**, у якій зберігається кількість правильних і загальних відповідей по кожному з відмінків. Друга база **pol.sqlite3** містить таблицю з польськими іменниками та їхніми формами в усіх відмінках і числах. Вона використовується як база для автоматичної генерації тестових завдань та аналізу правильних відповідей.

Окрему увагу приділено модулю тестування, який забезпечує гнучку систему перевірки знань користувача. У додатку реалізовано декілька форматів запитань, що вже були зазначені у попередньому підрозділі.

Під час реалізації логіки тестів було враховано низку граматичних і технічних особливостей, що забезпечують коректність перевірки:

1. Урахування іменників pluralia tantum (іменників, які вживаються лише у формі множини). Для таких слів система автоматично позначає, що форма однини відсутня.
2. Формування варіантів відповідей здійснюється з дотриманням умов унікальності, тобто серед варіантів не допускається дублювання однакових форм, навіть якщо в базі даних такі форми зустрічаються кілька разів. Якщо неможливо зібрати повноцінний набір з унікальних альтернатив, система автоматично скорочує кількість варіантів до мінімально допустимого.
3. Контекстне обмеження варіантів: у випадках, коли декілька відмінків мають однакову форму (граматична омонімія), відповідна форма подається лише один раз — або як правильна, або як відволікаючий варіант, але не одночасно. Це дозволяє уникнути ситуацій, коли дві однакові відповіді можуть бути і правильною, і хибною водночас.
4. Вільне введення відповіді підтримує використання спеціальних польських діакритичних символів (ą, ć, ę, ł, ń, ó, ś, ź, ż), для чого в тесті реалізовано зручну клавіатуру з кнопками для вставлення таких знаків.
5. Розроблено систему оцінювання: кожна правильна відповідь дає один бал, а загальна кількість балів фіксується для кожного користувача окремо. Результати використовуються для виведення персональної статистики з розподілом по відмінках. Підсумковий аналіз відповідей реалізовано у вигляді вікна з докладним зворотним зв’язком, в якому користувач бачить свій вибір, правильну відповідь та позначку (✔ або ✖), що дозволяє самостійно проаналізувати помилки [34][35][36].

Таким чином, архітектура додатка побудована як автономна система з чіткою логічною структурою, різними типами тестування для закріплення інформації, можливістю збереження персональних результатів і гнучким механізмом роботи з навчальними матеріалами.

3.4 Інтерфейс додатка

Інтерфейс користувача розроблено з орієнтацією на простоту, доступність та інтуїтивну зрозумілість, що є особливо важливим для освітніх програм, призначених для широкого кола користувачів різного віку. Структура інтерфейсу:

1. **Стартове вікно**, що містить привітання та дві основні кнопки: "Вхід" та "Реєстрація". Вікно оформлене у світлому стилі з великим заголовком.
2. **Вікна реєстрації та входу** реалізовані окремими фреймами з відповідними полями для введення електронної пошти та паролю. Передбачено перевірку коректності введення (валідація email) та повідомлення про помилки (некоректний формат, порожні поля, невідповідність паролів тощо).
3. **Головне меню користувача** — після входу користувач бачить вікно з кнопками переходу до основних модулів: "Теоретичні матеріали", "Тестування", "Моя успішність", "Про додаток". Привітання містить email користувача, що входить у систему.
4. **Модуль теорії** — реалізований у вигляді двопанельного вікна: ліворуч розташовано навігаційне меню з переліком тем (відмінки), праворуч — текстовий блок із автоматичним форматуванням вмісту (заголовки, списки, таблиці, пояснення). Для зручності перегляду великого обсягу інформації текстовий блок оснащено вертикальною прокруткою [36] .
5. **Модуль тестування** — кожен тип тесту відкривається у власному вікні. Для запитань передбачено марковану нумерацію, відповіді подаються як варіанти вибору, поля вводу або випадаючі списки. Кнопки спеціальних символів дозволяють вставляти діакритичні знаки.
6. **Вікно результатів** — після проходження тесту відкривається окреме вікно з детальним звітом. Відображаються відповіді користувача, правильні форми, позначки правильності та кількість набраних балів.

7. **Перегляд статистики** — у зручному форматі виводиться інформація про відсоток правильних відповідей по кожному з відмінків. Дані автоматично підтягуються з бази даних користувача [37].

Отже, інтерфейс програми розроблено відповідно до принципів зручності, логіки, зорової чистоти та доступності — без зайвих елементів і з фокусом на навчальний процес. Це дозволяє користувачам з будь-яким рівнем цифрової грамотності легко освоїти роботу з додатком та зосередитися на ефективному опануванні матеріалу.

Висновки до розділу 3

У третьому розділі комплексно представлено ключові аспекти проєктування та архітектури електронного засобу навчання для вивчення словозміни іменників польської мови. Ми окреслили цільову аудиторію, до якої належать українськомовні користувачі різного віку та рівня мовної підготовки.

Детально проаналізовано функціональні можливості програми, серед яких варто виокремити реєстрацію та авторизацію користувачів, наявність структурованого теоретичного модуля, гнучку систему тестування з кількома форматами завдань, реалізацію зворотного зв’язку та механізм накопичення статистики успішності. Архітектура програми складається з окремих блоків для роботи з користувачами, модуля тестування та системи збереження результатів у базі даних, що гарантує чітку організацію управління даними.

Особливу увагу у розділі приділено поясненню модулю тестування, де реалізовано різноманітні типи завдань і алгоритми, які враховують специфіку польської мови, забезпечують унікальність варіантів відповідей і підтримують введення польських діакритичних символів. Також докладно описано дизайн і структуру інтерфейсу, який відповідає принципам доступності, простоти й логічної організації.

Таким чином, цей розділ наочно демонструє цілісний підхід до

розроблення освітнього програмного засобу — від визначення цільової аудиторії та постановки функціональних вимог до реалізації технічної архітектури і організації ефективного користувацького досвіду. Перейдемо до розгляду реалізації представлених функцій.

РОЗДІЛ 4. РОЗРОБЛЕННЯ ДОДАТКА

Розроблення додатка здійснювалася мовою програмування **Python** [13] з використанням бібліотеки **Tkinter** [14] для побудови графічного інтерфейсу та **SQLite3** [15] для збереження та обробки даних у базах даних. Назву додатка "RzeczFlex" утворено блендингом (від англ. blend – суміш, комбінація) польського слова *rzeczownik* (іменник) та англійського *flexion* (закінчення) для символізування фокусу програми на відмінюванні іменників польської мови.

У цьому розділі подано узагальнений опис реалізації програмного засобу з демонстрацією ключових фрагментів коду та прикладами роботи програми. Посилання на повний вихідний код та використану базу даних зазначено у Додатку 1.

4.1. Створення та робота з базами даних

Під час запуску програми автоматично відбувається підключення до двох баз даних:

1. **pol.sqlite3** — лінгвістична база, що містить повну парадигму відмінювання найчастотніших польських іменників.
2. **users.db** — база для реєстрації користувачів та збереження їхньої статистики тестування.

Основою для створення бази даних **pol.sqlite3** стала база, попередньо розроблена моїм науковим керівником М. П. Костіковим у межах його дисертаційного дослідження [39]. Вона містила всі форми 356 найчастотніших польських іменників відповідно до частотного словника Єжи Казойча, укладеного у 2011 році [40]. Цей словник було сформовано на основі корпусу сучасних польськомовних текстів різних типів — публіцистичних, художніх, наукових та навчальних. Отже, база даних відображає лексику, яка є найбільш уживаною в реальному мовному середовищі.

Під час підготовки програмного засобу я здійснила редакцію наданої бази даних, зосередившись лише на тих колонках і структурних елементах, які були безпосередньо релевантні для запланованої реалізації навчального функціоналу. Із кількох початкових таблиць залишено лише одну — **tnoun**, яка містить усі необхідні граматичні форми іменників. Інші допоміжні чи технічні таблиці, які мали значення у первинному проєкті, було видалено як надлишкові для задач цього додатка. У фінальній версії таблиці **tnoun** залишено такі поля:

1. id — унікальний порядковий номер кожного слова;
2. sgN, sgG, sgD, sgA, sgI, sgL, sgV — форми в однині для всіх семи відмінків;
3. plN, plG, plD, plA, plI, plL, plV — відповідні форми у множині.

Для розширення бази даних я продовжила добирати найчастотніші іменники зі згаданого частотного словника, за тим самим принципом частотності. Було використано електронні ресурси, які надають повні парадигми відмінювання польських іменників [41]. Отримані форми вносились у таблицю **tnoun** з використанням SQL-запитів у середовищі DB Browser for SQLite. Наприклад, для масового додавання словоформ було використано SQL-запит такого типу:

```
INSERT INTO tnoun (sgN, sgG, sgD, sgA, sgI, sgL, sgV, plN, plG, plD, plA, plI, plL, plV)
VALUES
('liczba','liczby','liczbie','liczbę','liczbą','liczbie','liczbo','liczby','liczb','liczbom','liczby','liczbami','liczbach','liczby'),
('bogini','bogini','bogini','boginię','boginią','bogini','bogini','boginie','bogiń','boginiom','boginie','boginiami','boginiach','boginie'),
```

Отже, база даних **pol.sqlite3** є основним джерелом, на основі якого автоматично формуються тестові завдання та перевіряється правильність відповідей користувачів.

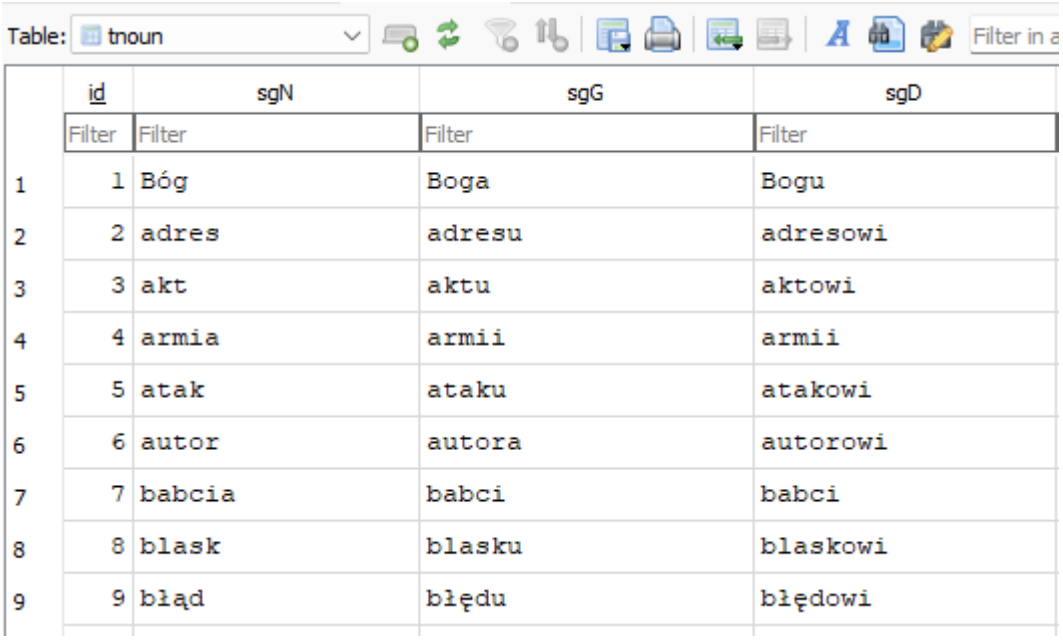
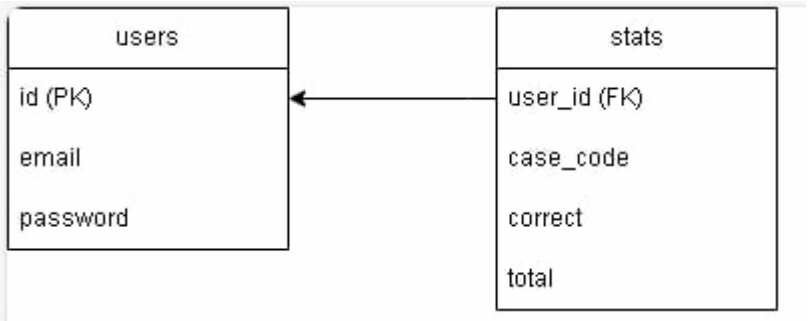


Table: tnoun

	id	sgN	sgG	sgD
	Filter	Filter	Filter	Filter
1	1	Bóg	Boga	Bogu
2	2	adres	adresu	adresowi
3	3	akt	aktu	aktowi
4	4	armia	armii	armii
5	5	atak	ataku	atakowi
6	6	autor	autora	autorowi
7	7	babcia	babci	babci
8	8	blask	blasku	blaskowi
9	9	błąd	błędu	błędowi

*Малюнок 4.1. Приклад записів парадигм відмінювання слів у БД **pol.sqlite3***

Для реалізації функціоналу реєстрації, авторизації та збереження персональних результатів у додатку була створена окрема база даних під назвою **users.db**. Вона відповідає за керування обліковими записами користувачів і фіксацію їхнього індивідуального прогресу у вивченні польської мови. База **users.db** має компактну й логічну структуру та складається з двох основних таблиць: **users** та **stats**.



*Малюнок 4.2. Структура БД **users.db***

Таблиця **users** містить базову інформацію про кожного користувача — його унікальний ідентифікатор (id), адресу електронної пошти (email) та пароль.

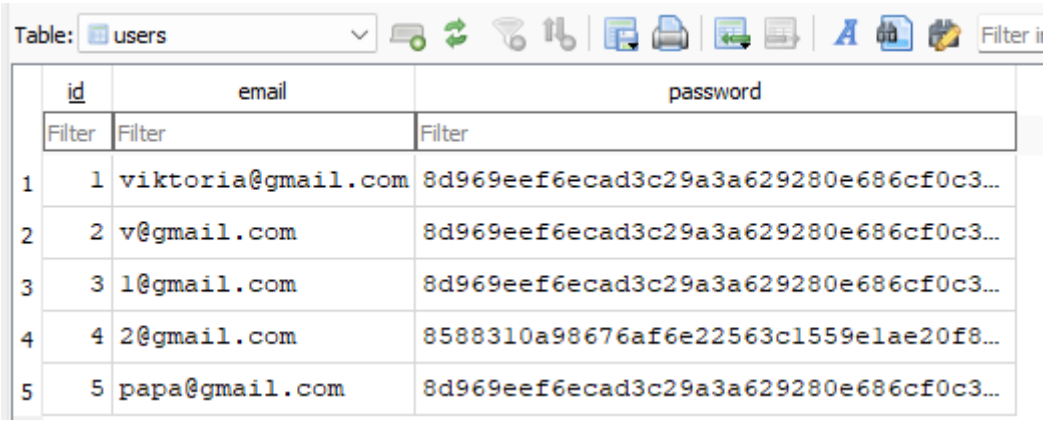


Table: users

	id	email	password
	Filter	Filter	Filter
1	1	viktoria@gmail.com	8d969eef6ecad3c29a3a629280e686cf0c3…
2	2	v@gmail.com	8d969eef6ecad3c29a3a629280e686cf0c3…
3	3	l@gmail.com	8d969eef6ecad3c29a3a629280e686cf0c3…
4	4	2@gmail.com	8588310a98676af6e22563c1559e1ae20f8…
5	5	papa@gmail.com	8d969eef6ecad3c29a3a629280e686cf0c3…

*Малюнок 4.3. Приклад запису інформації у таблиці **users***

Друга таблиця **stats** призначена для зберігання статистики проходження тестів. Кожен запис у цій таблиці містить посилання на користувача (user_id), код відмінка (case_code), кількість правильних відповідей (correct) та загальну кількість спроб (total). Код відмінка відповідає скороченню назви відповідного відмінка польської мови (наприклад, N — називний, G — родовий тощо). Таким чином, для кожного користувача в базі фіксується його результат за кожним розділом окремо. База **users.db** створюється автоматично під час першого запуску додатка, якщо її ще не існує. Для цього у програмному коді передбачено виконання SQL-інструкції CREATE TABLE IF NOT EXISTS, яка створює таблиці **users** та **stats**.

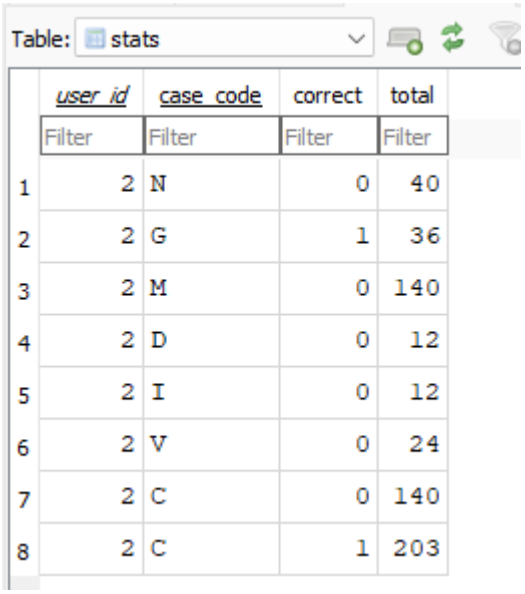


Table: stats

	user_id	case_code	correct	total
	Filter	Filter	Filter	Filter
1	2	N	0	40
2	2	G	1	36
3	2	M	0	140
4	2	D	0	12
5	2	I	0	12
6	2	V	0	24
7	2	C	0	140
8	2	C	1	203

*Малюнок 4.4. Приклад запису інформації у таблиці **stats***

4.2. Інтерфейс користувача

Як вже зазначалося, інтерфейс реалізовано через бібліотеку **Tkinter** за мультифреймовою моделлю, де кожне логічне вікно (реєстрація, вхід, меню, теорія, тестування, статистика) реалізовано у вигляді окремого фрейму **tk.Frame**, який за потреби виводиться, ховається або очищується. Це дозволяє уникнути перевантаження головного вікна та забезпечити гнучкий механізм перемикання між екранами без створення нових вікон. Усі фрейми динамічно відображаються або приховуються відповідно до дій користувача. Головне вікно створюється за допомогою:

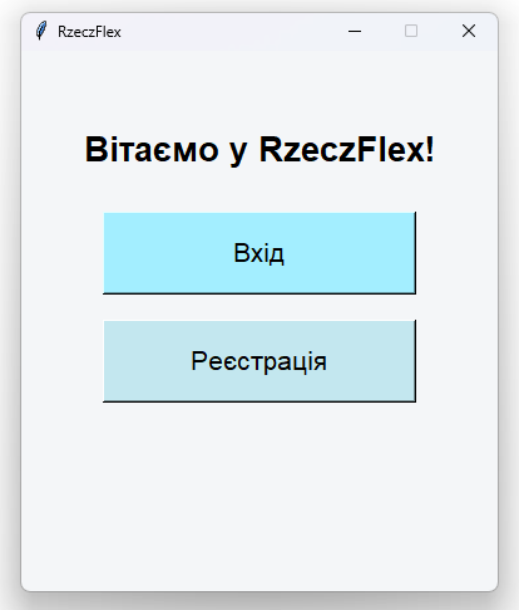
```
root = tk.Tk()
root.title("RzeczFlex")
root.configure(bg="#f4f6f8")
```

Фрейми оголошуються наперед і додаються до головного вікна:

```
selection_frame = tk.Frame(root, bg="#f4f6f8")
login_frame = tk.Frame(root, bg="#f4f6f8")
```

```
register_frame = tk.Frame(root, bg="#f4f6f8")
```

Всі елементи мають сучасне оформлення: використано шрифт Montserrat, світлу кольорову гаму (#f4f6f8), великі кнопки та адаптивне вирівнювання тексту. Вікна автоматично центровано на екрані **configure_window**. Через функції **clear_frame()** та **pack_forget()** реалізовано перемикання між формами без відкриття нових вікон [37].



Малюнок 4.5. Початкове вікно після запуску програми

Більше прикладів реалізації інтерфейсу буде наведено у наступних підрозділах для збереження логіки викладу та демонстрації результатів роботи коду. Скриншоти повного циклу роботи програми представлений у Додатку 2.

4.3. Реєстрація та авторизація

Процес реєстрації реалізовано через окремий фрейм **register_frame**, а сама реєстрація реалізована функцією **register_user(email, password)**. Коли користувач натискає кнопку "Зареєструватися", спрацьовує функція **register_action()**, яка перевіряє:

1. Чи всі поля заповнені:

```
if not email or not password or not confirm_password:
    messagebox.showerror(..., "Всі поля мають бути заповнені", ...)
```

2. Чи коректна адреса електронної пошти:

```
if not validate_email(email):
    messagebox.showerror(..., "Введено некоректну електронну пошту", ...)
```

3. Чи паролі збігаються:

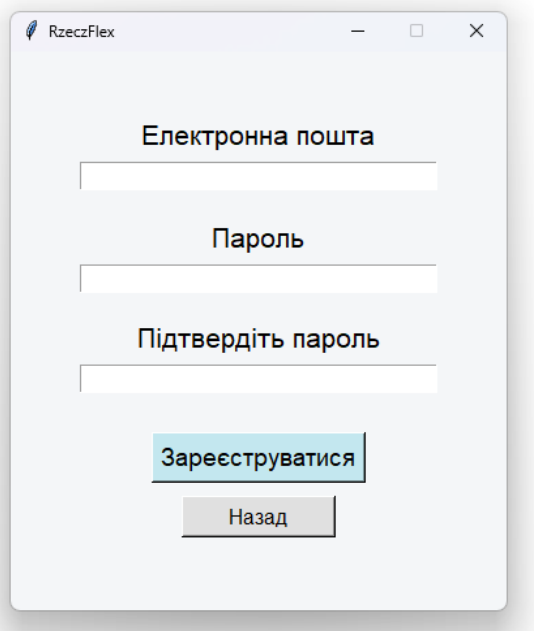
```
if password != confirm_password:
    messagebox.showerror(..., "Паролі не співпадають", ...)
```

4. Чи має пароль достатню довжину:

```
if len(password) < 6:
    messagebox.showerror(..., "Пароль має бути не менше 6 символів", ...)
```

Після всіх перевірок викликається основна функція:

```
def register_user(email, password):
    try:
        hashed = hash_password(password)
        users_cursor.execute(
            "INSERT INTO users (email, password) VALUES (?, ?)", (email, hashed))
        users_conn.commit()
        return True
    except sqlite3.IntegrityError:
        return False
```

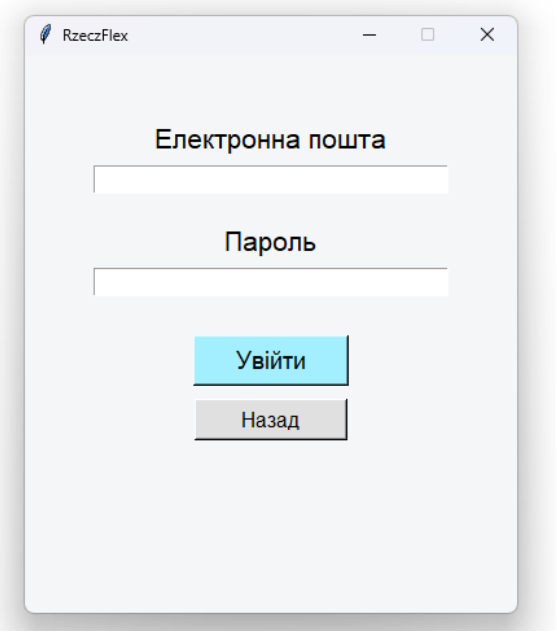


Малюнок 4.6. Вікно реєстрації

Таким чином при реєстрації перевіряється унікальність та відповідність email адреси, хешується пароль з використанням SHA-256 (hash_password) [32][33] і зберігається у таблиці **users**. Всі введені дані перевіряються на коректність, а також дублювання акаунтів (перевірка унікальності email).

При натисканні кнопки "Вхід" відкривається **login_frame**, який також містить поля введення email та пароля. Всі перевірки аналогічні до вищезазначених. Форма входу **open_login_window** дозволяє авторизуватися через перевірку хешованого паролю у функції **login_user(email, password)**:

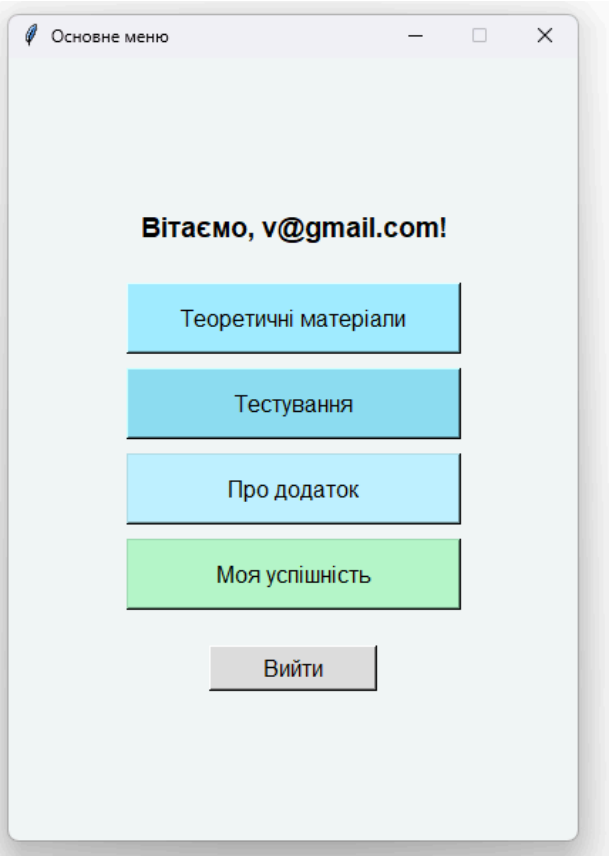
```
#авторизація
def login_user(email, password):
    hashed = hash_password(password)
     users_cursor.execute("SELECT * FROM users WHERE email=? AND password=?", (email, hashed))
    return users_cursor.fetchone().
```



Малюнок 4.7. Вікно входу

Після успішного входу, глобальна змінна **current_user_email** зберігає адресу активного користувача, що дозволяє ідентифікувати його у базі та прив’язувати статистику до облікового запису. Далі після авторизації відкривається головне меню **open_main_window**, у якому передбачено навігацію до:

1. Розділу теоретичного матеріалу (**show_theoretical_materials**);
2. Розділу тестування (**show_testing**);
3. Розділу перегляду статистики (**show_user_stats**);
4. Розділу опису програми (**show_about_app**);
5. Кнопки виходу з акаунта для повернення на початковий фрейм "Вхід / Реєстрація"



Малюнок 4.8. Основне меню

4.4. Реалізація та наповнення розділу "Теоретичний матеріал"

Розділ "Теоретичні матеріали" у програмі "RzeczFlex" призначений для надання користувачам структурованої інформації про іменники в польській мові та їх відмінювання. При переході до теорії виконується функція **show_theoretical_materials()**, яка очищує поточне вікно й ініціалізує новий екран з двома основними частинами:

```
content_frame = tk.Frame(main_win, bg="#f0f8ff")
nav_frame = tk.Frame(content_frame, width=200, bg="#E5F5F8")
text_frame = tk.Frame(content_frame, bg="#f0f8ff")
```

де:

- nav_frame — панель розділів (меню),
- text_frame — зона виводу теоретичного вмісту.

Правий блок має вбудовану прокрутку через Canvas та Scrollbar, що дозволяє переміщатися в межах вікна:

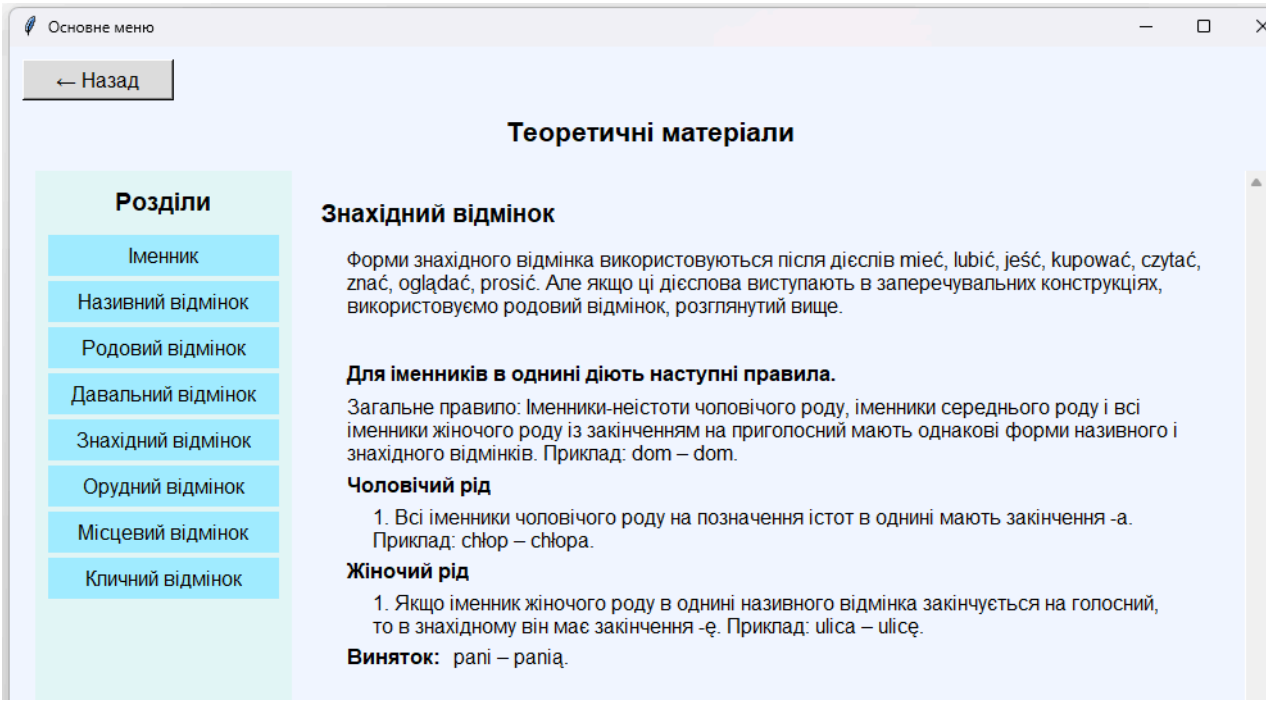
```
canvas = tk.Canvas(text_frame, bg="#f0f8ff", highlightthickness=0)
vsb = tk.Scrollbar(text_frame, orient="vertical", command=canvas.yview)
canvas.configure(yscrollcommand=vsb.set)
```

Весь вміст розміщується у внутрішньому фреймі **inner**, прив’язаному до Canvas. Теоретичні матеріали зберігаються у словнику **theory_content**. Для кожного розділу (напр., "Іменник", "Родовий відмінок") визначено заголовок і список блоків, які можуть бути:

- Рядками тексту (звичайний абзац),
- Кортежами типу ("numbered", [...]) або ("bullet", [...]) — списки,
- Словниками — парні рядки з жирним заголовком і звичайним описом,
- Таблицями — списки списків,
- Спеціальними маркерами — наприклад, {"відмінки": {...}} для таблиці з питаннями до відмінків.

Таке структурування теоретичних матеріалів у словнику **theory_content** було реалізоване з метою уніфікації подачі вмісту. Замість того щоб вручну прописувати стилі, форматування і структуру кожного окремого текстового блоку у коді, створено єдиний формат подання даних, який охоплює всі основні типи інформації: заголовки, звичайний текст, нумерований текст, маркований текст, таблицю тощо. Цей підхід дозволяє кожен теоретичний розділ описати у вигляді компактної, читабельної структури у **theory_content**, а потім автоматично обробити його через одну універсальну функцію виводу. Таким чином, кожен блок розпізнається за своїм типом і відображається у відповідному стилі без необхідності додатково налаштовувати форматування.

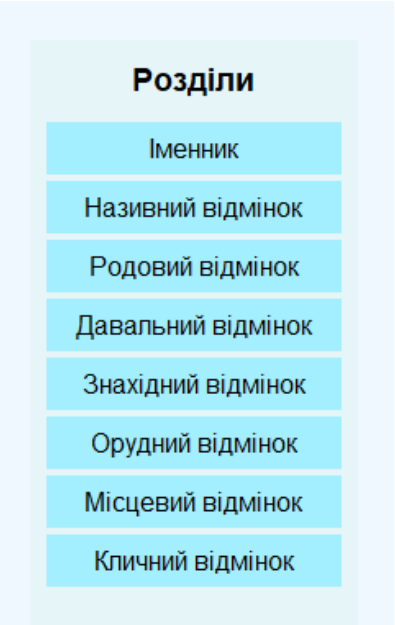
Крім того, така система забезпечує гнучкість і масштабованість, оскільки у разі оновлення чи доповнення теоретичних матеріалів немає потреби змінювати логіку відображення — достатньо додати новий розділ у словник у заданому форматі.



Малюнок 4.9. Повне вікно демонстрації теоретичних матеріалів

Зліва розміщується панель розділів (відмінків і загальної інформації про іменник). Кнопки створюються динамічно:

```
for case_name in cases:
    btn = tk.Button(nav_frame, text=case_name, command=lambda c=case_name: show_theory(c))
    btn.pack(fill="x", pady=2)
```



Малюнок 4.10. Навігаційна панель

При натисканні на кнопку викликається функція **show_theory()**, яка перезавантажує правий фрейм з відповідним розділом. Для відображення таблиці відмінків (див. Додаток 2) у розділі "Іменник" реалізовано окрему логіку з обрахунком ширини колонок та формуванням сітки:

```
table_frame.grid_columnconfigure(c_idx, minsize=col_widths[c_idx], weight=1)
tk.Label(table_frame, text=str(cell_val), ...).grid(row=r_idx, column=c_idx, ...)
```

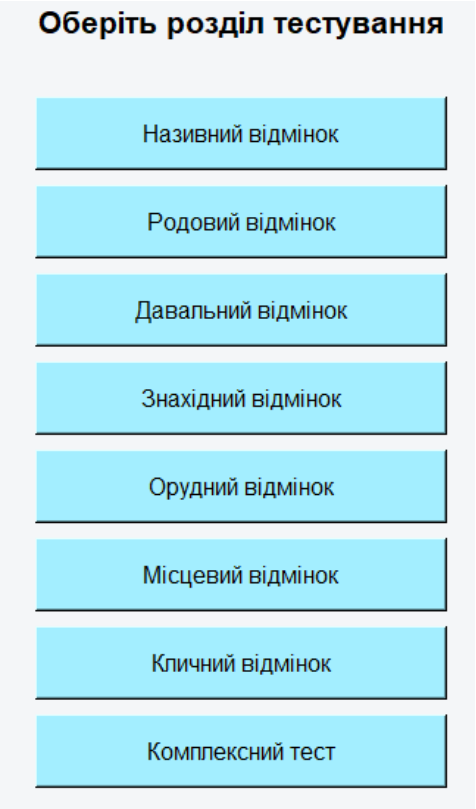
4.5. Реалізація та наповнення розділу "Тестування"

Розділ "Тестування" є центральним функціональним блоком програми, що реалізує систему перевірки знань користувача з теми відмінювання польських іменників. Його структура дозволяє охопити всі граматичні відмінки у двох числах (однина та множина) і забезпечити різноманітність у формах подання завдань.

Користувач із головного меню переходить до модуля тестування через

функцію **show_testing()**, яка відображає перелік відмінків. Кожна кнопка викликає окрему функцію, наприклад:

```
sections_tests = [
    ("Називний відмінок", show_nominative_test, "#A3EEFF"),
    ("Родовий відмінок", show_genitive_test, "#A3EEFF"),
    ...]
```



Малюнок 4.11. Вікно тестування

Усі тести у програмі мають єдину логіку: відображення інструкції тестування, динамічне формування завдань, перевірка відповідей, виведення результатів і збереження статистики. Водночас, залежно від типу відмінка, реалізовано **три окремі підходи до тестування**:

- для **називного відмінка** застосовується спрощена форма тестування з єдиним типом завдань множинний вибір;

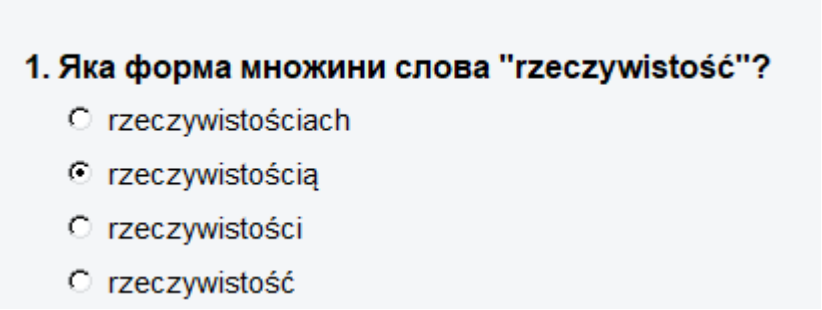
- для **усіх інших відмінків** використовується уніфікована функція генерації, яка охоплює три типи завдань;
- для **комплексного тесту** реалізовано окрему логіку, що перевіряє знання повної парадигми відмінювання одного слова[34][35].

Розглянемо кожен підхід детальніше.

4.5.1 Реалізація тестування називного відмінка

Для **називного відмінка** реалізовано окрему функцію **show_nominative_test()**, оскільки цей відмінок є базовою, початковою формою іменника, яка збігається зі словниковою формою, та не потребує складного аналізу. Завдання у цьому тесті спрощені, тому реалізовано лише тести на множинний вибір.

Логіка побудови тесту полягає в тому, що користувачу подається словоформа або в однині, або в множині, і необхідно обрати правильну відповідну форму множини для слова, поданого в однині, або форму однини для слова, поданого в множині. Наприклад, якщо в запитанні подано форму множини, користувач обирає правильну відповідну форму однини з-поміж інших форм того самого слова, які автоматично підбираються з його парадигми відмінювання. Це дозволяє закріпити базові граматичні зв’язки між формами числа в межах одного відмінка.



Малюнок 4.12. Приклад тесту у розділі "Називний відмінок"

Окрему увагу в реалізації приділено словам **pluralia tantum**, тобто таким, які вживаються лише у множині й не мають форм однини. У таких випадках система автоматично підставляє варіант відповіді **"pluralia tantum"**:

```
correct_answer = sgN_orig if (sgN_orig and sgN_orig.strip()) else 'pluralia tantum'
```

Крім того, реалізовано механізм **очищення варіантів від дублікатів** — це важливо для уникнення ситуацій, коли в списку відповідей з'являються дві або більше однакові форми через граматичну омонімію. Варіанти форм очищуються за допомогою **set**, після чого перетворюються назад у список для відображення:

```
current_options = list(set(current_options))
```

4.5.2 Реалізація загального шаблону тестування

Для всіх інших відмінків, крім називного, реалізовано єдину уніфіковану логіку тестування, яка базується на функції **generate_case_questions(case_code)**, що дозволяє не дублювати код для кожного відмінка, а автоматизувати процес генерації типових завдань. Функція приймає як аргумент код відмінка — скорочене буквене позначення ('G' — родовий, 'D' — давальний, 'A' — знахідний тощо). Функція динамічно формує 12 завдань трьох типів:

- 4 завдання на вибір правильної форми (Multiple Choice);
- 4 завдання у форматі "Так / Ні";
- 4 завдання з ручним введенням відповіді.

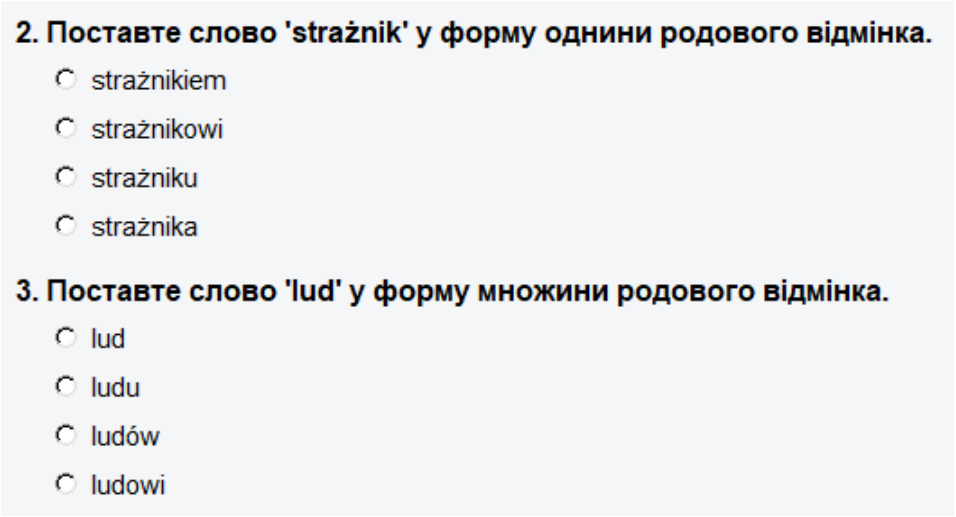
Вибір випадкового слова з бази реалізовано через SQL-запит до таблиці **tnoun**, яка містить усі 14 відмінкових форм іменника в однині та множині:

```
    pol_cursor.execute("SELECT sgN, sgG, sgD, sgA, sgI, sgL, sgV, plN, plG, plD, plA, plI,
plL, plV FROM tnoun ORDER BY RANDOM() LIMIT 1")
    row = pol_cursor.fetchone()
```

Ця команда дозволяє отримати одну повну парадигму іменника, обрану випадково, що забезпечує варіативність тестування, оскільки кожне нове питання ґрунтується на новому слові, випадковим чином обраному з бази даних.

У тестах на множинний вибір користувачеві ставиться запитання, наприклад: *"Яка форма множини давального відмінка відповідає слову "kolega"?"* І надаються 4 варіанти відповіді, серед яких лише один є правильним. Варіанти відповідей генеруються з різних форм цього ж слова, які зберігаються у базі даних. Наприклад:

```
correct = row[plD_index]  # правильна форма множини давального
options = make_opts(row, correct, exclude_idx=plD_index)
```



Малюнок 4.13. Приклад тесту на множинний вибір

make_opts() — це спеціальна функція, яка бере правильну відповідь і підбирає 3 хибні варіанти з інших форм того ж слова, виключаючи дублікати:

```
def make_opts(forms, correct, exclude_idx=None):
    opts = [f for i, f in enumerate(forms) if i != exclude_idx and f and f != correct]
    opts = list(set(opts))  # запобігання дублюванню
    if correct not in opts:
        opts.append(correct)
    random.shuffle(opts)
```

```
    return opts[:4]
```

Завдання формату "Так / Ні" запроваджені для того, щоб перевірити не лише знання конкретних словоформ, а й уміння критично оцінювати їхню відповідність заданим граматичним характеристикам. Користувачу ставиться запитання типу: *"Чи є форма "studentom" формою множини давального відмінка?"* Щоб уникнути ситуації, коли серед кількох завдань усі відповіді можуть бути лише "ні" (тобто неправильні), у коді штучно закладається принаймні 2 правильні відповіді на 4 питання цього формату. Це реалізовано через генерацію 2 питань, де форма справді відповідає заданому відмінку, і ще 2, де форма є відмінною:

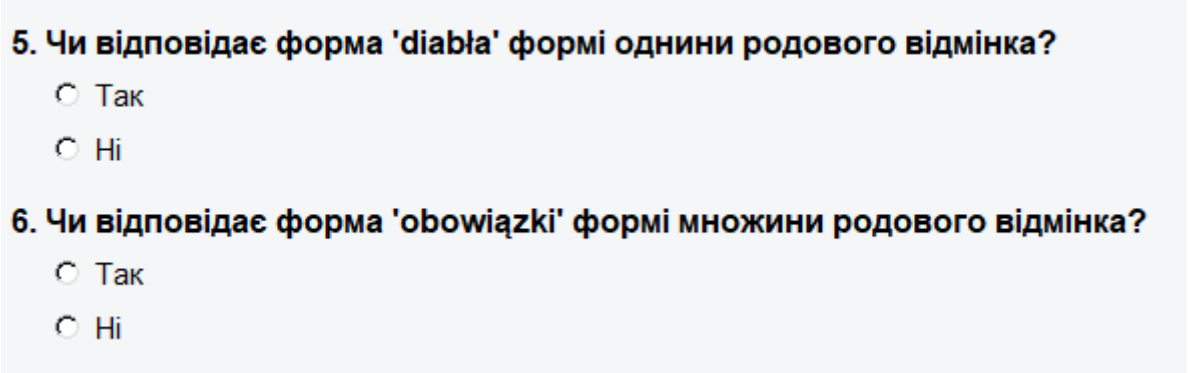
```
used = set()
yes_count = 0
no_count = 0
while yes_count < 2 or no_count < 2:
  idx = random.randint(0, 13)  # індекс форми в парадигмі
  form = row[idx]
  if not form or form in used:
    continue
  used.add(form)

    is_correct = (idx == target_idx)  # чи форма справді відповідає заданому відмінку і числу
  if is_correct and yes_count >= 2:
    continue  # обмежує кількість справжніх "Так"
  if not is_correct and no_count >= 2:
    continue  # обмежує кількість "Ні"

  answer = "yes" if is_correct else "no"
  if answer == "yes":
    yes_count += 1
  else:
    no_count += 1

  questions.append({
    "type": "yesno",
```

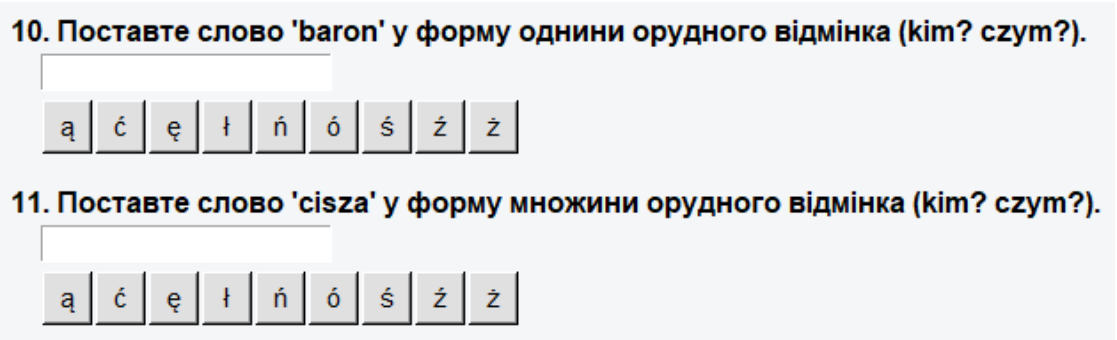
```
    "question": f"Чи є форма '{form}' формою {case_name} у {number}?",
    "correct": answer
})
```



Малюнок 4.14. Приклад тесту формату "Так / Ні"

У завданні з вільним введенням відповіді користувачу потрібно самостійно ввести правильну форму, без підказок. Ставиться питання типу: *"Введіть форму родового відмінка множини для слова "kot" (kogo? czego?)"*. Цей формат слугує для розвитку орфографічної пам’яті та точного знання парадигми. У полі вводу користувач може ввести будь-яке значення, а програма порівнює його з правильним. Для підтримки польських діакритичних знаків реалізовано спеціальну віртуальну клавіатуру:

```
special_chars = ['ą','ć','ę','ł','ń','ó','ś','ź','ż']
for char in special_chars:
    btn = tk.Button(..., text=char, command=lambda c=char: entry.insert(tk.INSERT, c))
```



Малюнок 4.15. Приклад тесту на вільне введення відповіді

У кожній функції створення тесту за загальним шаблоном відбувається виклик головної генераційної функції **render_test()**, яка відповідає за візуальне

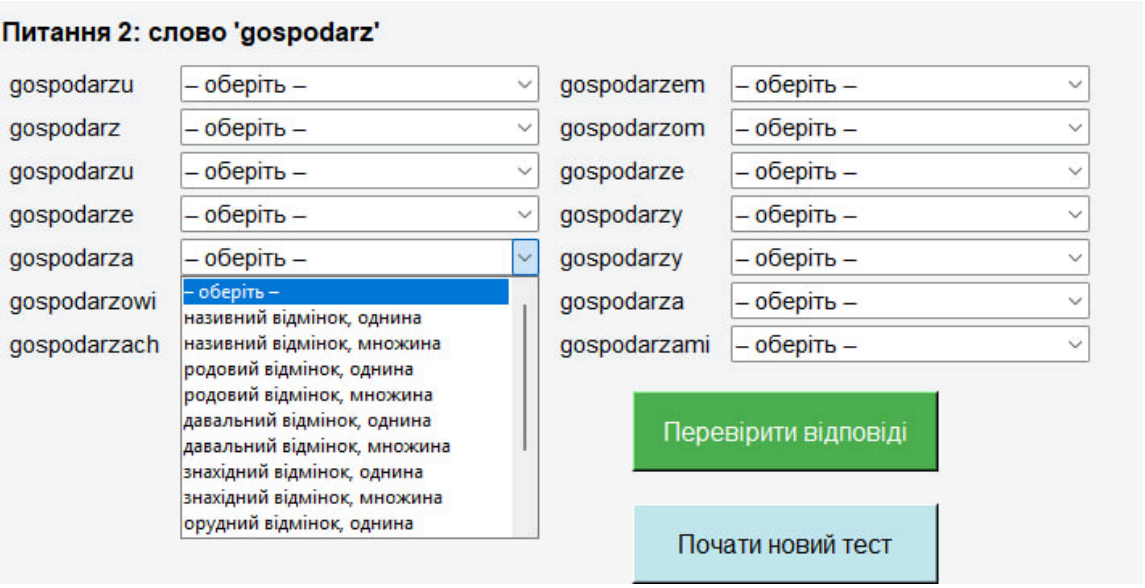
відображення завдань, інтерактивну взаємодію з користувачем та перевірку відповідей. Вона відображає усі типи запитань, обробляє введені варіанти, формує текстовий зворотний зв’язок та зберігає результати в базі даних.

4.5.3. Реалізація комплексного тесту

Комплексний тест на відповідність завдання реалізовано окремо від решти тестів, оскільки він має спеціальну логіку та спрямований на зіставлення словоформи з її граматичною характеристикою. Функція, яка відповідає за реалізацію цього тесту, — **show_matching_test()**. Користувачу пропонується набір усіх 14 форм одного іменника (7 відмінків × 2 числа) у довільному порядку. Навпроти кожної форми розміщено випадаючий список, з якого потрібно обрати правильну комбінацію "відмінок + число" (наприклад, "родовий відмінок, множина"):

```
for form_code, form_text in form_data:
    if not form_text or form_text.strip() == "":
        continue

    label = tk.Label(inner, text=form_text, ...)
    var = tk.StringVar(value="")
            dropdown = ttk.Combobox(inner, textvariable=var, values=options, state="readonly")
```



Малюнок 4.16. Приклад комплексного тесту на відповідність

4.6. Перевірка відповідей та фіксування результатів

Процес перевірки правильності відповідей, підрахунку балів та збереження результатів є завершальним етапом кожного тестування. Після проходження тесту користувач натискає кнопку "Перевірити відповіді", що запускає процедуру звірки — система перебирає всі завдання, зчитує обрані користувачем відповіді, порівнює їх з правильними варіантами та підраховує загальну кількість правильних відповідей. Для зручності кнопка для перевірки відповідей зафіксована в межах нижньої частини екрану (див. Додаток 2). Вікно з результатами реалізовано не як діалогове вікно, що задумувалося спочатку, а як окреме вікно, щоб була можливість використовувати скроллбар основного вікна для співставлення питання та обраної відповіді.

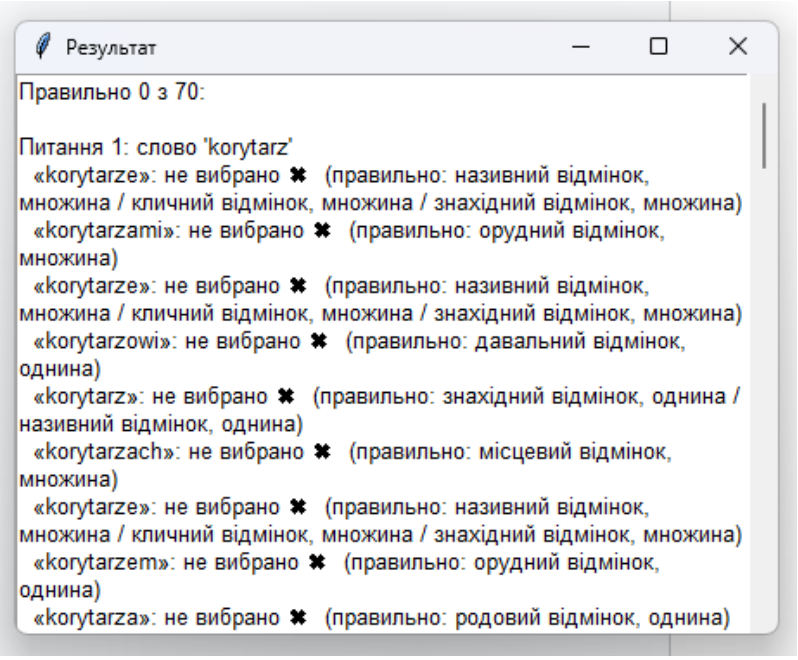
Для кожного завдання в тесті з множинним вибором користувач обирає один із варіантів через радіокнопки. Кожен варіант записується у змінну, і після завершення тесту усі відповіді порівнюються з еталонними:

```
user_choice = var_obj.get()
mark = "✔" if user_choice == correct_ans_val else "✖"
```

```
if user_choice == correct_ans_val:
    correct_count += 1
```

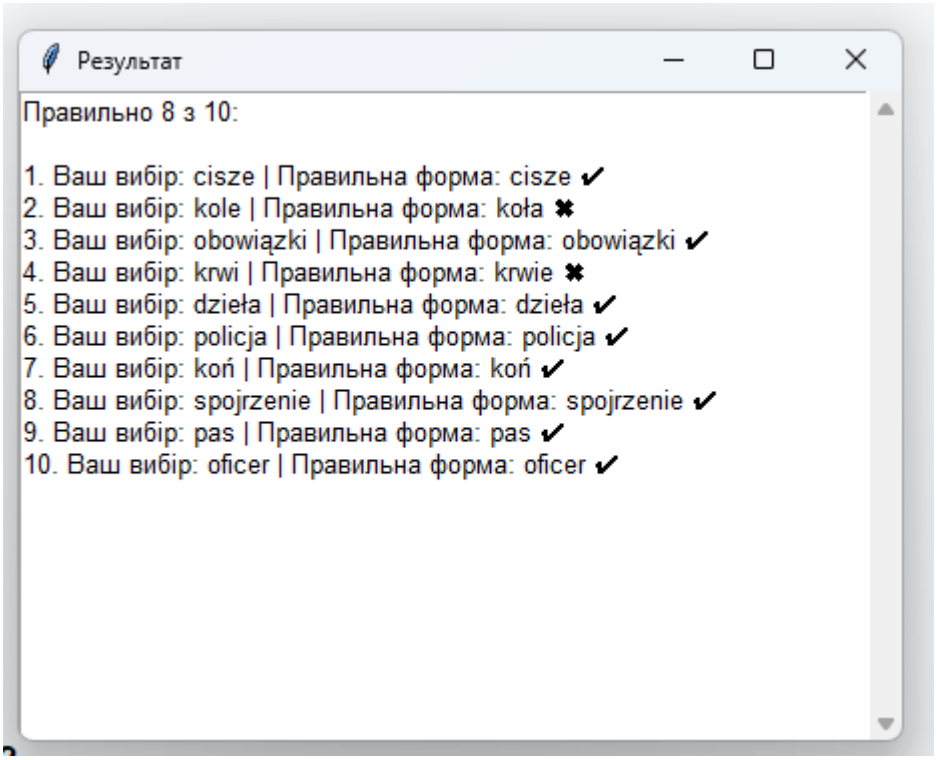
Для комплексного тесту програма дозволяє обрати правильний варіант лише один раз для кожної унікальної форми. Щоб уникнути неправильного зарахування однакових форм (якщо, наприклад, форма sgA і plN однакова), у коді реалізовано лічильники дозволених відповідей (Counter):

```
from collections import Counter
master_allowed_counters[form_text] = Counter(lbl for _, lbl in group_of_cbs_and_labels)
```



Малюнок 4.17. Вікно результатів комплексного тесту

За кожну правильну відповідь користувач отримує 1 бал. Після завершення перевірки формується докладний текстовий звіт із зазначенням правильних форм, обраних відповідей та позначками правильності (✔ або ✖).



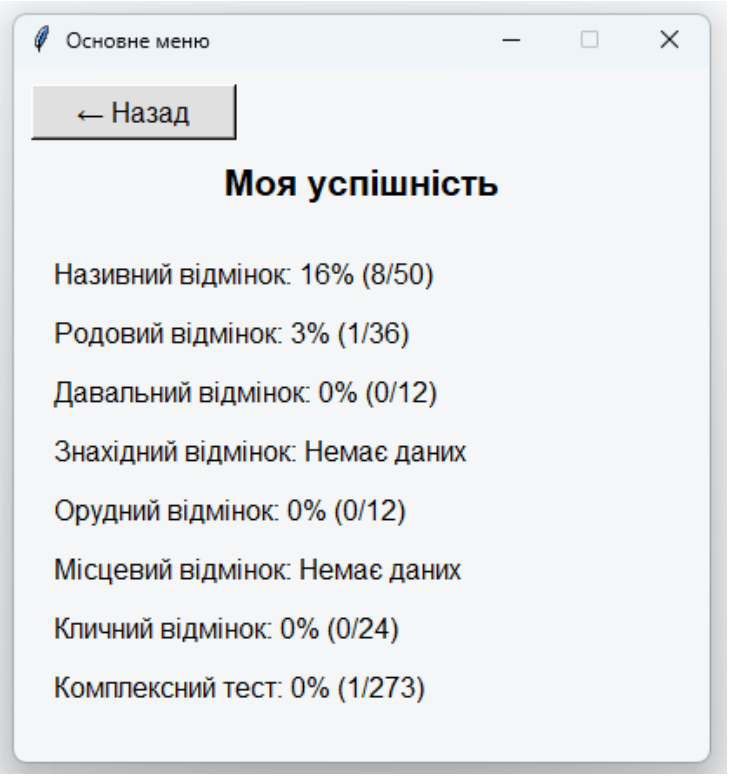
Малюнок 4.18. Вікно результатів тесту "Називний відмінок"

Результати кожного перевіреного тестування зберігаються у локальну базу даних **users.db**, у таблицю **stats**, де для кожного зареєстрованого користувача фіксується кількість правильних і загальних відповідей за кожним відмінком. Запис здійснюється через функцію **update_user_stats()**, яка спершу перевіряє, чи вже існує запис для поточного користувача і відповідного відмінка. Якщо такий запис є, система оновлює значення, додаючи нові результати до попередніх; якщо запису немає — створюється новий рядок. Таким чином, кожен користувач поступово накопичує власну статистику проходження тестів, яку може переглянути в розділі "Моя успішність". Це дозволяє відстежувати індивідуальний прогрес у розрізі кожного відмінка, що значно підвищує ефективність навчального процесу:

```
def update_user_stats(case_code, correct, total):
    user_id = get_user_id_by_email(current_user_email)
     users_cursor.execute("SELECT correct, total FROM stats WHERE user_id=? AND case_code=?", (user_id, case_code))
```

```
            row = users_cursor.fetchone()

            if row:
                new_correct = row[0] + correct
                new_total = row[1] + total
                  users_cursor.execute("UPDATE stats SET correct=?, total=? WHERE user_id=?
AND case_code=?", (new_correct, new_total, user_id, case_code))
            else:
                  users_cursor.execute("INSERT INTO stats (user_id, case_code, correct, total)
VALUES (?, ?, ?, ?)", (user_id, case_code, correct, total))
            users_conn.commit()
```



Малюнок 4.19. Вікно статистики успішності користувача

4.7. Оцінка ефективності розробленого додатка

У попередніх підрозділах було розглянуто процес реалізації основних функціональних можливостей програмного продукту. Значну увагу приділено технічним деталям, зокрема різноманітним типам тестування, системі контролю та збору статистики, а також впровадженню спеціальних механізмів і запобіжників для забезпечення коректності й об’єктивності перевірки знань.

Однак, щоб визначити реальну якість і корисність нашого електронного засобу навчання, необхідно провести комплексну оцінку його ефективності у практичному використанні. Такий аналіз дасть змогу оцінити, наскільки програмний продукт відповідає вимогам, що ставились на етапах планування, чи реалізовані всі необхідні функції, чи є інтерфейс зручним та зрозумілим для користувачів, чи дійсно програма допомагає досягати поставлених навчальних цілей. Для всебічної оцінки ефективності було враховано різні аспекти: продуктивність роботи, інтуїтивність і логіку інтерфейсу, повноту та якість реалізованих функцій, навчальну цінність, надійність та сучасність дизайну.

Оцінювання проводилося за кількома основними критеріями, кожен із яких аналізувався за п’ятибальною шкалою, де 1 бал означає недостатній рівень, а 5 балів — оптимальне втілення даного аспекту.

Таблиця 4.1. Оцінка ефективності "RzeczFlex"

Критерій оцінювання	**Оцінка (1–5)**	**Коментар**
Швидкість запуску та роботи додатка	5	Додаток запускається швидко, працює стабільно, без збоїв.
Швидкість обробки даних у базі	5	Запити до баз даних SQLite виконуються оперативно, забезпечуючи швидку авторизацію та генерацію тестів.
Інтуїтивність та простота навігації	4.5	Логічна структура та зрозуміла навігація.
Візуальний дизайн та відповідність сучасним стандартам UI	3.5	Функціональний, проте дещо простий дизайн; відповідає базовим вимогам, є простір для модернізації.
Повнота теоретичного матеріалу	5	Детально викладено теорію відмінювання іменників.
Якість тестових завдань	5	Реалізовано декілька типів тестів, що забезпечує всебічну перевірку знань.

Система реєстрації та статистики	5	Добре продуманий процес реєстрації; можна відстежувати індивідуальний прогрес за кожним відмінком.
Зворотний зв'язок після тестування	3.5	Відображаються правильні/неправильні відповіді та оцінка, але відсутні детальні пояснення помилок, що ускладнює самостійний аналіз.
Загальна навчальна цінність	4	Оцінка знижена через відсутність детальних пояснень помилок у тестах.

Середня оцінка: 4.5

Згідно з результатами оцінювання, розроблений додаток "RzeczFlex" продемонстрував високі показники за більшістю основних параметрів. Зокрема, швидкість запуску та стабільність роботи додатка отримали найвищу оцінку — додаток запускається швидко, працює стабільно, без збоїв, а всі запити до бази даних виконуються оперативно та коректно. Це суттєво підвищує комфорт користування, особливо в умовах повторних сесій.

Важливою перевагою є також логічна структура інтерфейсу та інтуїтивність навігації. Завдяки зрозумілому меню, продуманному переходу між модулями та простоті виконання основних дій, навіть користувач із базовим рівнем цифрової грамотності зможе швидко освоїти роботу з додатком. Високий рівень отримала і якість реалізації теоретичного блоку — вся необхідна інформація зібрана у зручному форматі, структурована і доступна для повторного перегляду в будь-який момент за потреби.

На наш погляд, особливої уваги заслуговує модуль тестування, який охоплює різноманітні типи завдань. Такий підхід сприяє залученню користувача до активного аналізу мовних явищ, а також дозволяє відпрацювати як прості навички впізнавання форм, так і складніші аспекти, наприклад,

ідентифікацію відмінка та числа для певної форми чи орфографію словоформ.

Крім того, модуль тестування тісно інтегрований зі статистичною системою додатка: усі результати автоматично фіксуються та зберігаються, що дозволяє користувачу оцінювати власний прогрес та акцентувати увагу на тих розділах, які потребують додаткового опрацювання.

Проте, аналізуючи результати тестування й зворотний зв’язок користувачів, варто також відзначити і певні недоліки розробленого додатка. Найсуттєвішим із них є відсутність детальних пояснень помилок у тестах. Після проходження тестування користувач отримує лише підсумкову оцінку, позначки правильності та правильну відповідь, але не бачить докладного пояснення помилки чи теоретичних коментарів до неправильної відповіді. Це дещо обмежує навчальну цінність програми.

Ще одним пунктом, який можна покращити, є візуальний дизайн інтерфейсу — хоча він і відповідає базовим сучасним стандартам, певна одноманітність оформлення, мінімалізм у використанні графічних елементів і невелика кількість візуальних акцентів можуть зробити взаємодію менш захопливою, особливо для тих користувачів, хто звик до сучасних освітніх платформ із багатшим візуальним оформленням. Загалом інтерфейс можна охарактеризувати як простий, лаконічний і доступний, але з точки зору естетики та інтерактивності має потенціал для подальшого удосконалення.

Висновки до розділу 4

Узагальнюючи підсумки проведеної роботи, можна стверджувати, що розроблений електронний засіб навчання "RzeczFlex" по більшості параметрів відповідає сучасним вимогам до освітніх програм і демонструє високий рівень реалізації основних функціональних компонентів.

Додаток успішно вирішує задачу комплексного засвоєння та перевірки знань з морфології польських іменників та забезпечує користувачам комфортну й зрозумілу взаємодію з програмою. Реалізовані рішення в архітектурі,

наповненні та логіці додатка свідчать про ретельне врахування специфіки морфології польської мови. Водночас, отримані результати аналізу підкреслюють наявність потенціалу для подальшого розвитку — як у напрямку розширення функціоналу, так і щодо покращення інтерфейсу та індивідуалізації навчання. Таким чином, додаток може стати цінним інструментом для тих, хто прагне ефективно опанувати польську граматику, і є гарною основою для майбутніх удосконалень.

ВИСНОВКИ

У представленій роботі було детально розглянуто процес створення електронного засобу навчання (ЕЗН) морфології іменників польської мови "RzeczFlex". Наше дослідження охоплює етапи від теоретичного обґрунтування й аналізу існуючих рішень до практичної реалізації та тестування власного програмного продукту. Створений додаток надає можливість користувачам ефективно засвоювати матеріал з відмінювання іменників польської мови через інтерактивні вправи.

У першому розділі роботи проведено огляд електронних засобів навчання, визначено їх класифікацію та вимоги, необхідні для створення якісного ЕЗН. Також здійснено аналіз найбільш популярних додатків для вивчення польської мови (Duolingo, Babbel, Mondly, Busuu), за результатами якого виявлено недостатнє висвітлення теми словозміни іменників. Це підтвердило актуальність створення спеціалізованого додатка, який би глибоко й системно розкривав особливості відмінювання іменників польської мови.

У другому розділі систематизовано та докладно проаналізовано граматичний матеріал за темою дослідження. Було визначено специфіку кожного з семи відмінків, основні правила та типові випадки чергування звуків, що лягли в основу теоретичного блоку ЕЗН. Особливу увагу приділено виняткам і складним випадкам словозміни.

У третьому розділі описано процес проєктування програмного продукту: визначено цільову аудиторію, яка включає українськомовних користувачів з різним рівнем мовної підготовки (від початкового до середнього рівня), розглянуто архітектуру додатка та визначено його функціональні можливості. До основних функціональних компонентів додатка увійшли модулі авторизації, теоретичної інформації, тестування й збереження статистики користувача.

Четвертий розділ присвячено практичній реалізації додатка "RzeczFlex" мовою програмування Python. Зокрема, детально описано процес створення та наповнення баз даних (**pol.sqlite3**, **users.db**), наведено їхню структуру. На прикладах фрагментів коду розглянуто процес реалізації модулів авторизації, реєстрації та збереження даних користувачів. Також докладно розглянуто реалізацію теоретичного блоку та різних типів тестування, включно з механізмами перевірки відповідей. За підсумками проведеної оцінки ефективності роботи додатка було визначено високу продуктивність і стабільність роботи, достатню повноту теоретичного матеріалу, інтуїтивну навігацію та продуману систему тестування й реєстрації користувачів. Водночас було виявлено певні недоліки у візуальному дизайні інтерфейсу та відсутність докладних пояснень помилок у тестах, що визначає напрями для майбутніх покращень.

Для подальшого вдосконалення додатка "RzeczFlex" доцільно розглянути можливість:

- розширення бази даних з новими іменниками;
- додавання більше видів тестувань;
- додавання детальних коментарів і пояснень до помилкових відповідей;
- оптимізації інтерфейсу з акцентом на візуальну привабливість і зручність;
- впровадження механізмів персоналізації процесу навчання залежно від успішності конкретного користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лобода Ю. Г. ЕЛЕКТРОННІ ЗАСОБИ НАВЧАННЯ: СТРУКТУРА, ЗМІСТ, КЛАСИФІКАЦІЯ. Information Technologies and Learning Tools. 2012. Т. 28, № 2. С. 3–9 URL: https://doi.org/10.33407/itlt.v28i2.649
2. Ilie V. Traditional Learning Versus E-Learning. EduWorld 2018 - 8th International Conference. 2019. P. 1193–1196. URL: https://doi.org/10.15405/epsbs.2019.08.03.146
3. Антонюк Д. С. Електронні засоби навчання: сутність поняття та їх класифікація. Фізико-математична освіта. 2019. Вип. 3 (21). С. 13–14. URL: http://nbuv.gov.ua/UJRN/fmo_2019_3_4
4. Mahanta D., Ahmed M. E-Learning Objectives, Methodologies, Tools and its Limitation. International Journal of Innovative Technology and Exploring Engineering. 2012. Vol. 2, no. 1. P. 46–50. URL: https://www.ijitee.org/wp-content/uploads/papers/v2i1/A0370112112.pdf.
5. Habbal F. Standards of Building e-Learning Systems in Higher Education. Emirati Journal of Business, Economics and Social Studies. 2021. Vol. 1, no. 1. P. 5–20. URL: https://doi.org/10.54878/ejbess.163
6. Duolingo. (n.d.). Language Learning Platform. URL: https://www.duolingo.com/.
7. Babbel. (n.d.). A Mobile Language Learning App. URL: https://ua.babbel.com/.
8. Mondly. (n.d.). Language Learning App. URL: https://www.mondly.com
9. Busuu. (n.d.). Learn Languages Online. URL: https://www.busuu.com
10. Varlamis I., Apostolakis I. The Present and Future of Standards for E-Learning Technologies. Interdisciplinary Journal of e-Skills and Lifelong Learning. 2006. Vol. 2. P. 059–076. URL: https://doi.org/10.28945/401
11. Буртовий С. Електронні засоби навчання – від теорії до практики : Метод. посіб. Кіровоград, 2014. 48 с.

12. Асоянц П. Г., Сердюков П. І., Чекаль Г. С. Основи методики створення і застосування комп'ютерних програм у навчанні іноземних мов : посібник. 1993. 106 с.
13. Офіційна документація Python. URL: https://docs.python.org/3.
14. Офіційна документація sqlite3. URL: https://docs.python.org/3/library/sqlite3.html.
15. Офіційна документація tkinter. URL: https://docs.python.org/3/library/tkinter.html.
16. Reddit. (n.d.). URL: https://www.reddit.com
17. Stelmach A., Stempek I. Polski krok po kroku Tablice gramatyczne. 2013. 34 p.
18. Левицька Ґ., Левицький Р. Самовчитель польської мови. Практичний курс. : Навч. посіб. Київ : Методика, 2011. 240 с.
19. Русіна О. В. 100 тем. Польська мова. АССА, 2018. 160 с.
20. Самовчитель польської мови для початківців онлайн. Вивчити польську з нуля. URL: https://popolsku.co.ua.
21. Levels of knowledge of the Polish language. MyPolish. URL: https://mypolish.link/rivni-znannya-polskoi-movy-vid-a1-do-c2/.
22. Kozachenko O. Advantages of teaching Ukrainians the grammar of the Polish language using a communicative approach. Scientific bulletin of South Ukrainian National Pedagogical University named after K. D. Ushynsky. 2024. Vol. 147, no. 2. P. 75–82. URL: https://doi.org/10.24195/2617-6688-2024-2-11
23. Король Л. РЕГІОНАЛЬНІ ОСОБЛИВОСТІ СЛОВОЗМІНИ ПОЛЬСЬКИХ ІМЕННИКІВ ТА ПРИКМЕТНИКІВ. ПРОБЛЕМИ СЛОВ'ЯНОЗНАВСТВА. 2009. № 58. С. 231–237.
24. A Review of Mobile Language Learning Applications: Trends, Challenges, and Opportunities / C. R. Heil et al. The EuroCALL Review. 2016. Vol. 24, no. 2. P. 32. URL: https://doi.org/10.4995/eurocall.2016.6402

25. Bączkowska A. An overview of popular website platforms and mobile apps for language learning. Forum Filologiczne Ateneum. 2021. No. 1(9)2021. P. 9–35. URL: https://doi.org/10.36575/2353-2912/1(9)2021.009
26. Raine P. Developing Web-based English Learning Applications: Principles and Practice. Computer-Assisted Language Learning Electronic Journal. 2024. Vol. 19, no. 2. P. 125–138. URL: https://callej.org/index.php/journal/article/view/252.
27. Levy M., and Steel C. (2015) Language learner perspectives on the functionality and use of electronic language dictionaries. ReCALL. 27(2). P. 177–196. doi:10.1017/S095834401400038X.
28. Developing Desktop-Based English for Accounting Teaching Materials to Support Computer-Assisted Language Learning (CALL) / P. D. Hudiananingsih et al. Advances in Economics, Business and Management Research. Dordrecht, 2024. P. 417–426. URL: https://doi.org/10.2991/978-94-6463-622-2_46
29. Warschauer M., Healey D. Computers and language learning: an overview. Language Teaching. 1998. 31(2), P. 57–71. URL: doi:10.1017/S0261444800012970.
30. Godwin-Jones R. Evolving technologies for language learning. Language Learning & Technology, 2021. 25(3). P 6–26. URL: http://hdl.handle.net/10125/73443.
31. Klimova B., Al-Obaydi L. H. Use of Mobile Applications for Foreign Language Learning Among University Students: Needs and Functions. International Journal of Interactive Mobile Technologies (iJIM). 2023. Vol. 17, no. 12. P. 28–39. URL: https://doi.org/10.3991/ijim.v17i12.38541
32. Nielson S. J., Monson C. K. Practical Cryptography in Python: Learning Correct Cryptography by Example. Apress, 2019. 388 p.
33. Byrne D. Full Stack Python Security: Cryptography, TLS, and Attack Resistance. Manning Publications Co. LLC, 2021. 304 p.

34. Tomlinson B. Testing to learn: a personal view of language testing. ELT Journal. 2005. Vol. 59, no. 1. P. 39–46. URL: https://doi.org/10.1093/elt/cci005
35. Jun Chen, Lei Wang. Computerized adaptive testing: A new trend in language testing. 2010 International Conference on Artificial Intelligence and Education (ICAIE), Hangzhou, China, 29–30 October 2010. 2010. URL: https://doi.org/10.1109/icaie.2010.5641509
36. Putting Education in "Educational" Apps / K. Hirsh-Pasek et al. Psychological Science in the Public Interest. 2015. Vol. 16, no. 1. P. 3–34. URL: https://doi.org/10.1177/1529100615569721
37. Staiano F. Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop. Packt Publishing, 2022. 382 p.
38. Костіков М. П. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ПРОЦЕСУ НАВЧАННЯ ГРАМАТИКИ ІНОЗЕМНОЇ МОВИ У ВНЗ : автореф. дис. канд. техн. наук : 05.13.06. Київ, 2016.
39. Najpopularniejsze słowa 1-10000 wersja Jerzego Kazojcia. Wikisłownik. URL: https://pl.wiktionary.org/wiki/Indeks:Polski_-_Najpopularniejsze_słowa_1-10000_wersja_Jerzego_Kazojcia
40. Odmiana.NET. URL: https://odmiana.net/

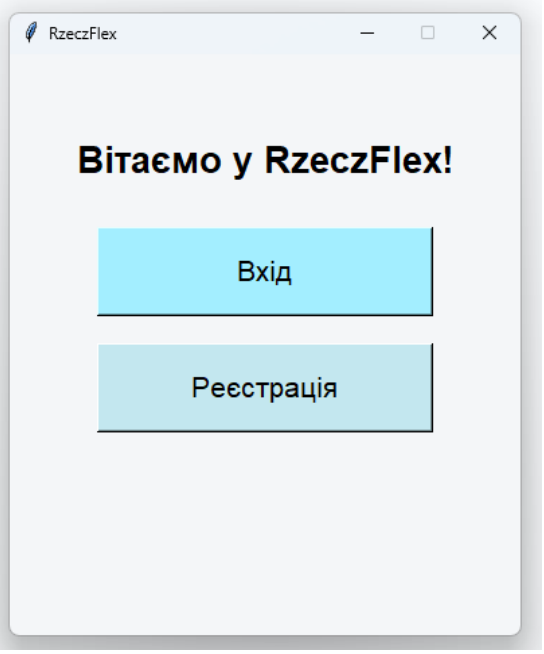
ДОДАТКИ

Додаток 1

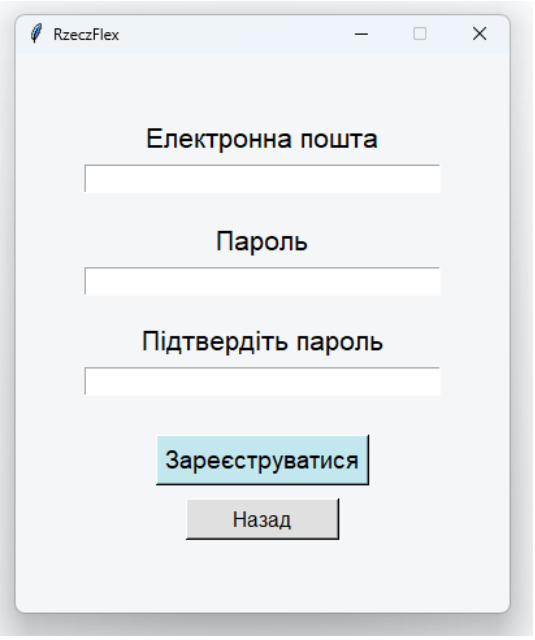
Посилання на GitHub проєкту: https://github.com/vikhhas/diploma

Додаток 2

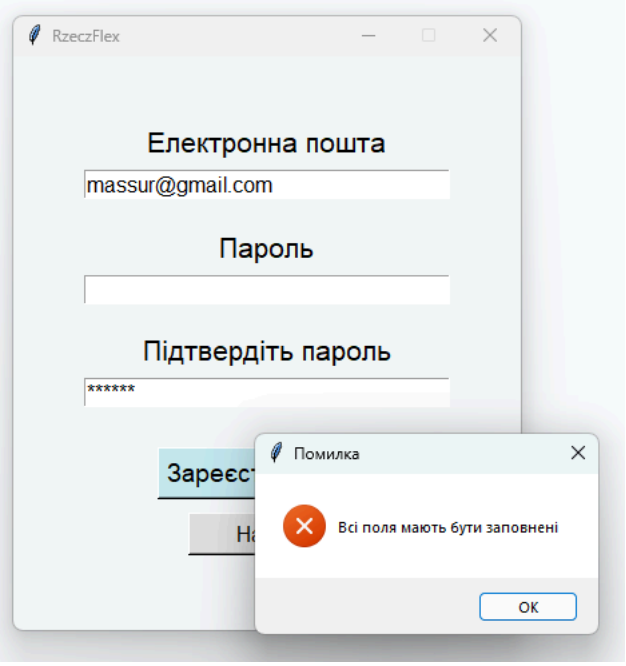
Скриншоти повного циклу роботи програми:



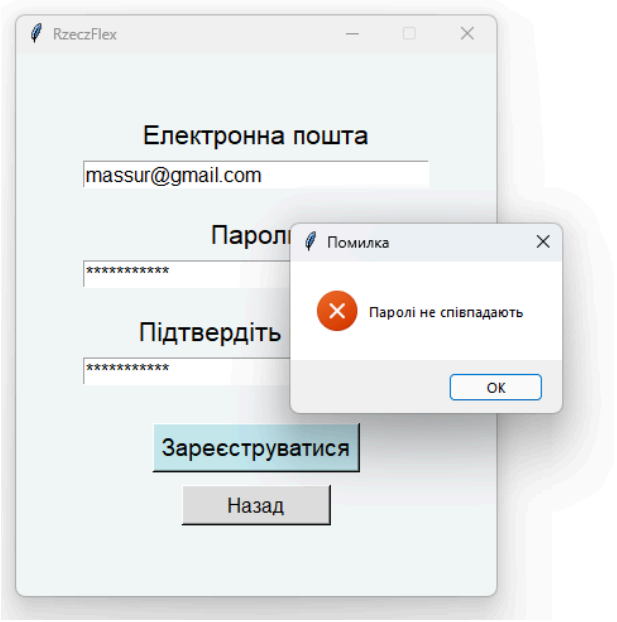
Мал. Д2.1. Початкове вікно входу



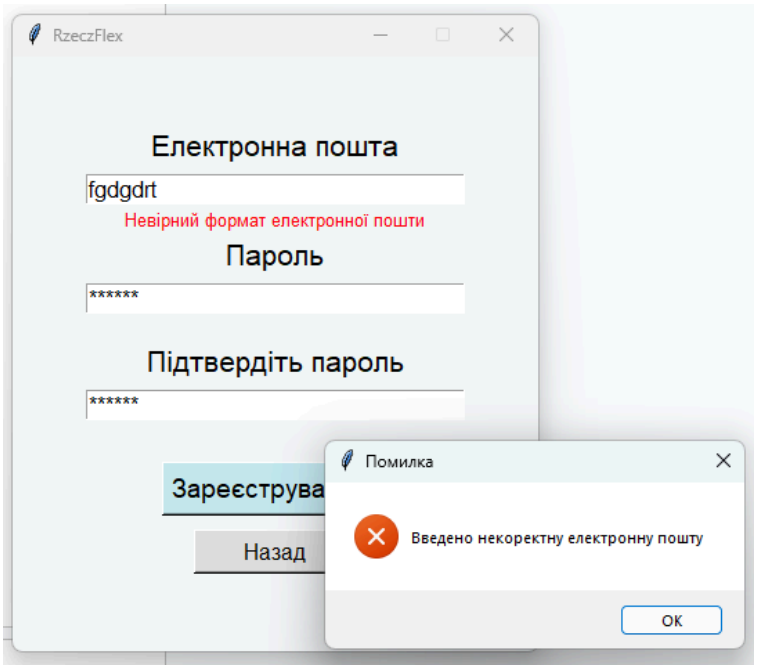
Мал. Д2.2. Вікно реєстарції



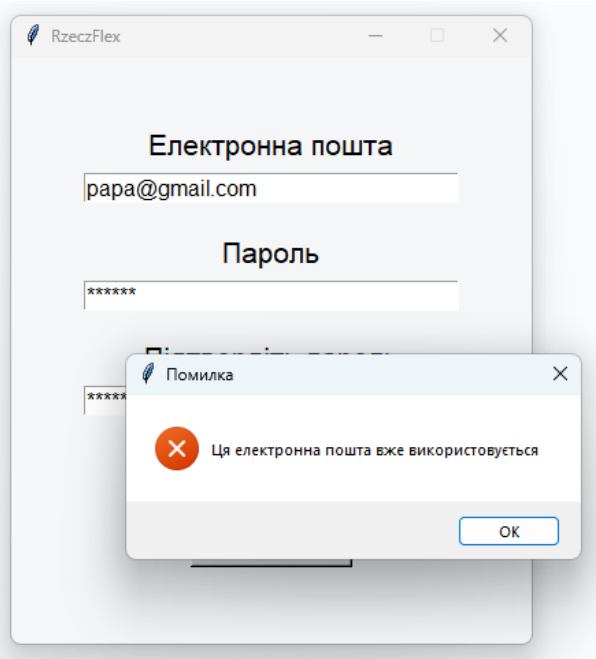
Мал. Д2.3. Повідомлення про помилку при незаповненні обов'язкових полів під час реєстрації



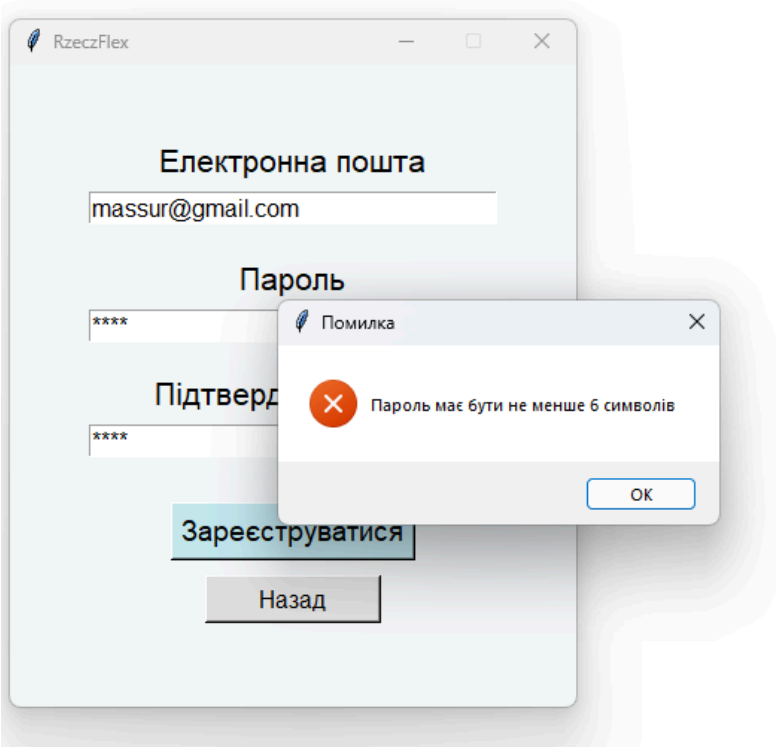
Мал. Д2.4. Повідомлення про помилку при неспівпадінні паролів під час реєстрації



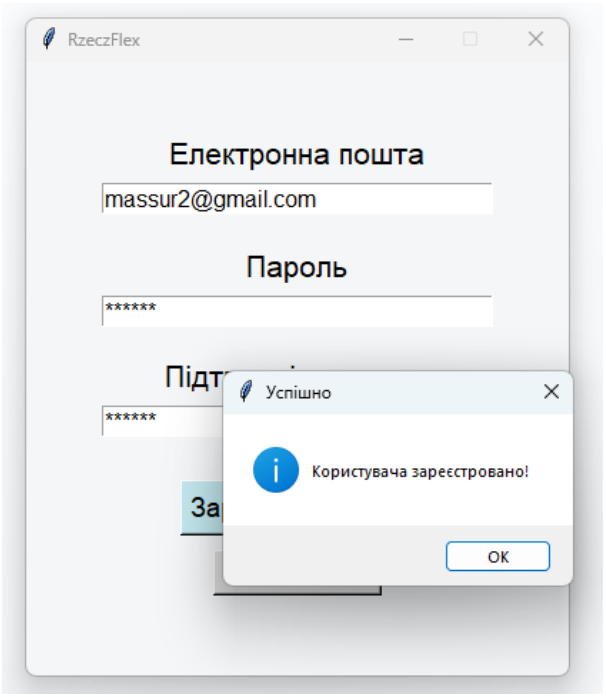
Мал. Д2.5. Повідомлення про некоректний формат ел. пошти під час реєстрації



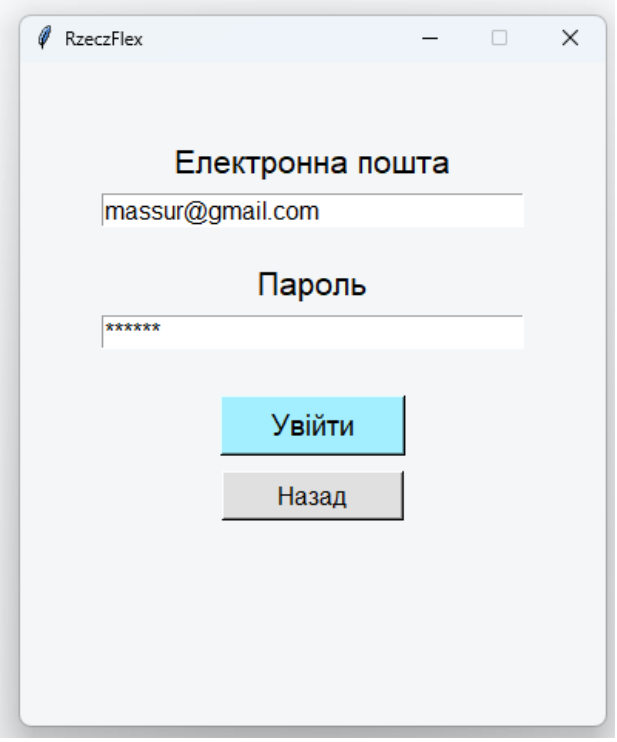
Мал. Д2.6. Помилка реєстрації: ця електронна пошта вже використовується



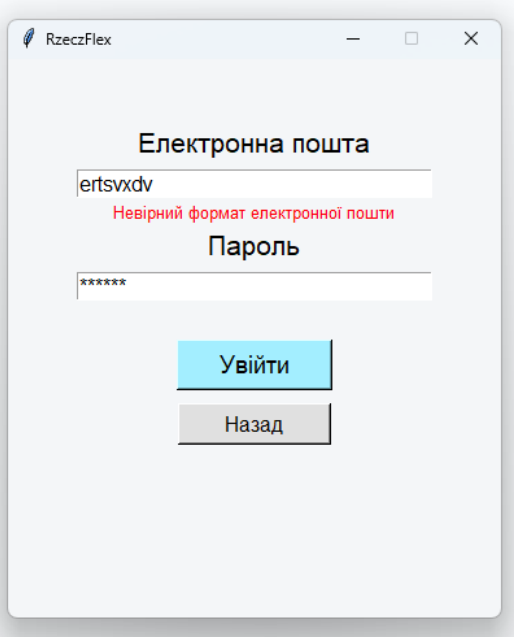
Мал. Д2.7. Помилка реєстрації: невідповідна довжина паролю



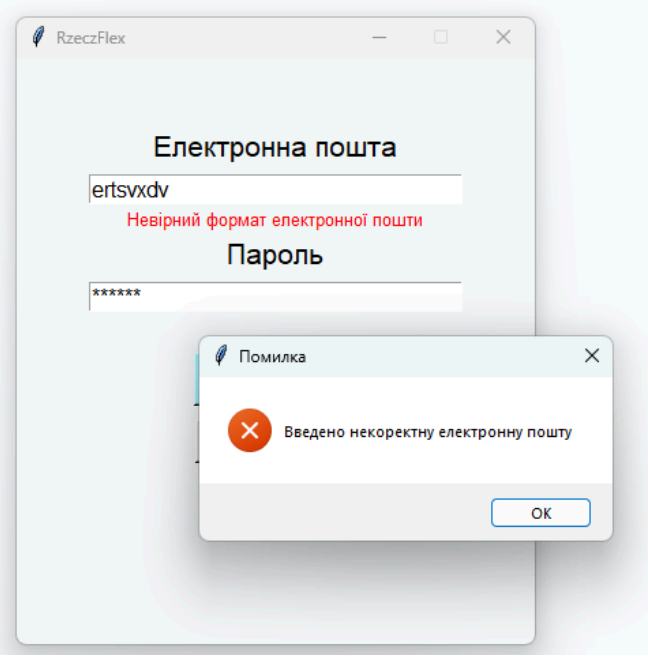
Мал. Д2.8. Успішна реєстрація



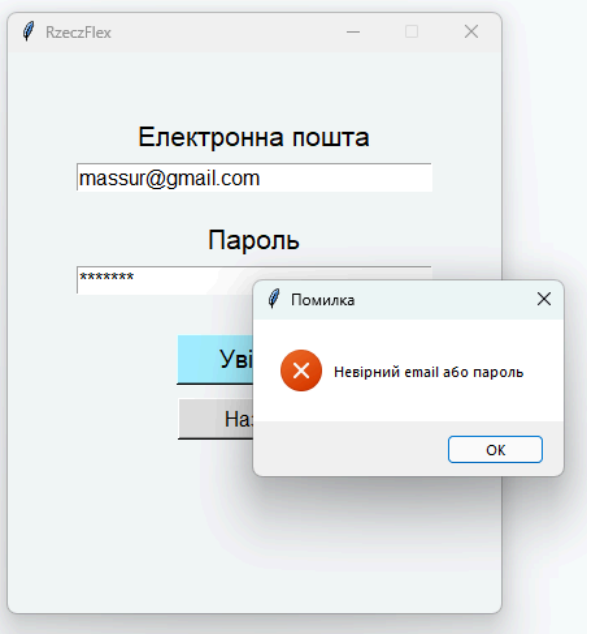
Мал. Д2.9. Вікно входу



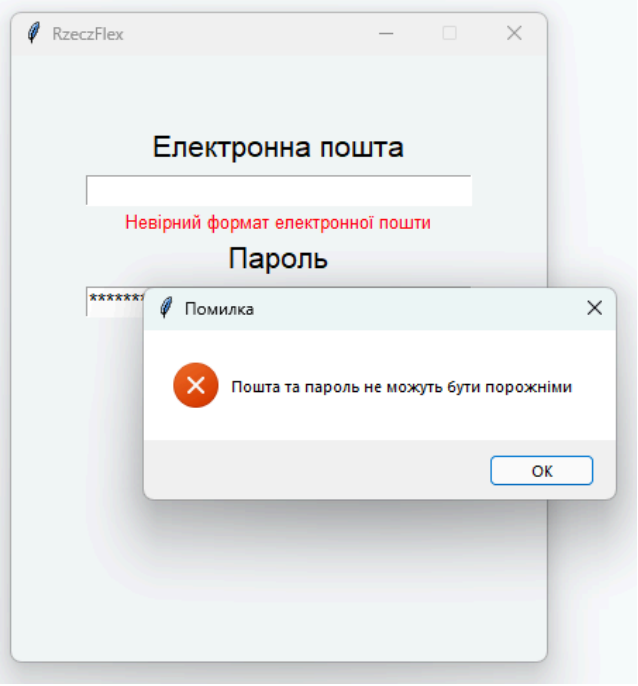
Мал. Д2.10. Некоректний формат електронної пошти під час авторизації



Мал. Д2.11. Повідомлення про некоректний формат електронної пошти під час авторизації

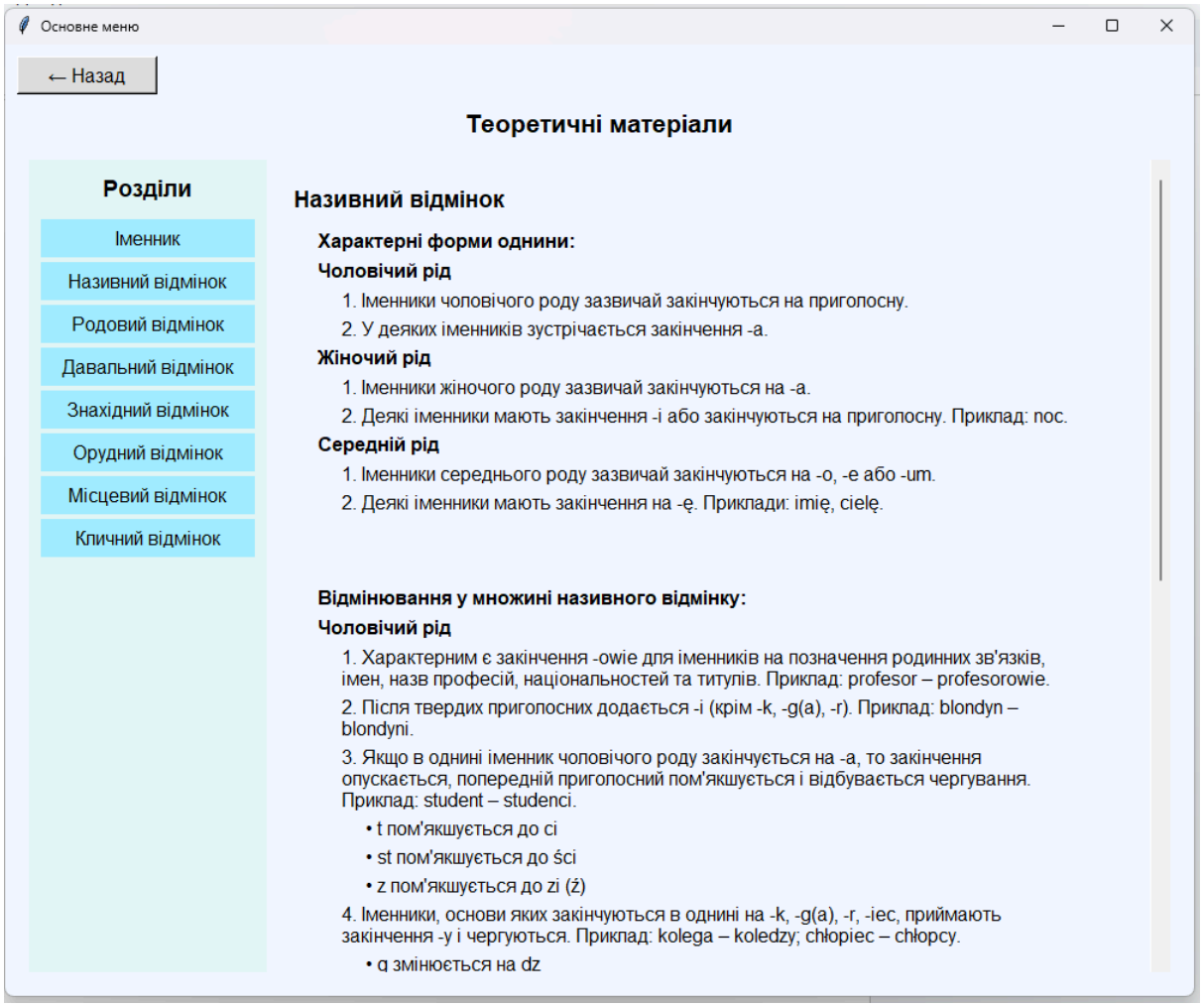


Мал. Д2.12. Повідомлення про помилку при введенні неправильних даних для входу

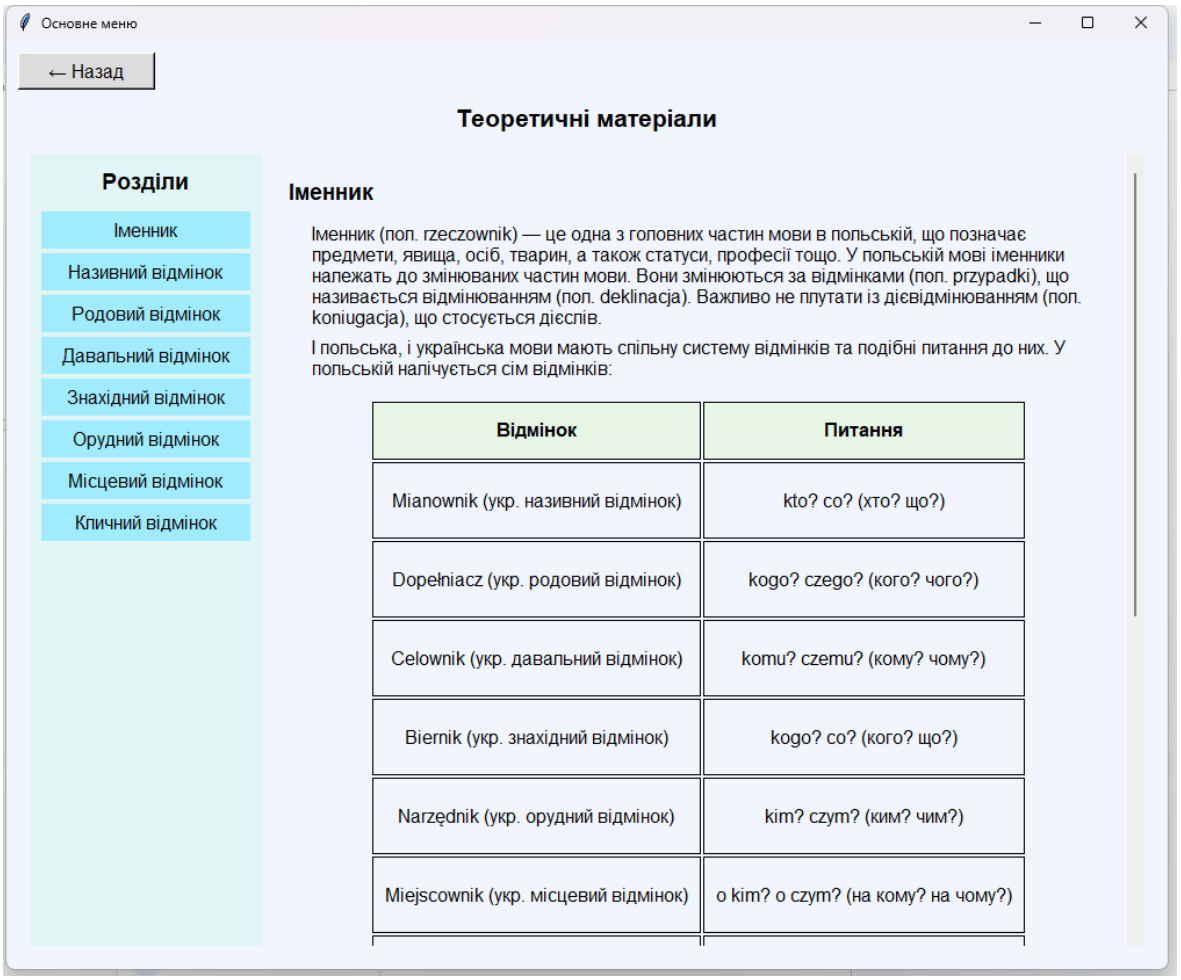


Мал. Д2.13. Повідомлення про помилку: електронна пошта або пароль не введені

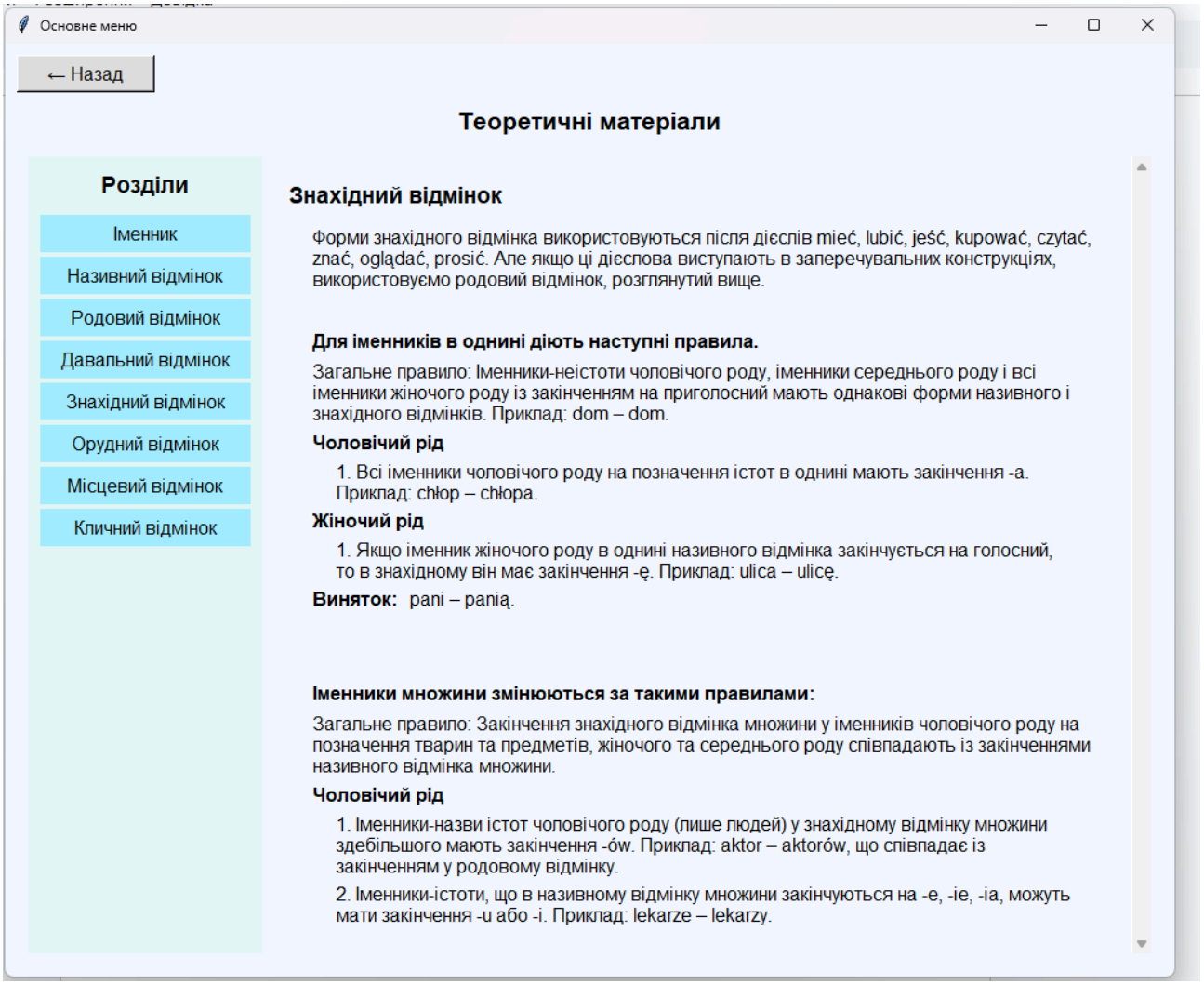
Мал. Д2.14. Основне меню після реєстрації



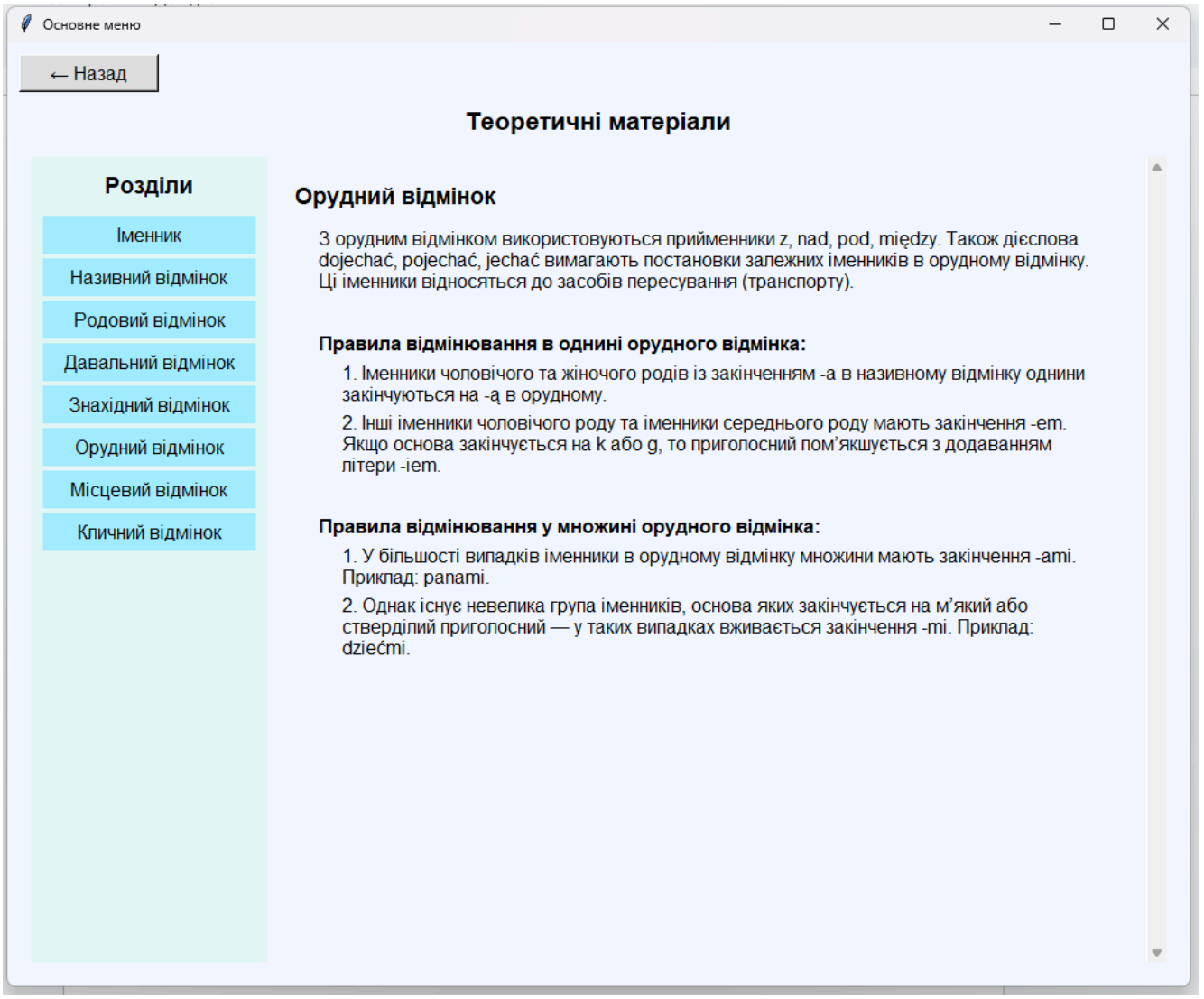
Мал. Д2.15. Приклад теоретичного розділу “Називний відмінок”



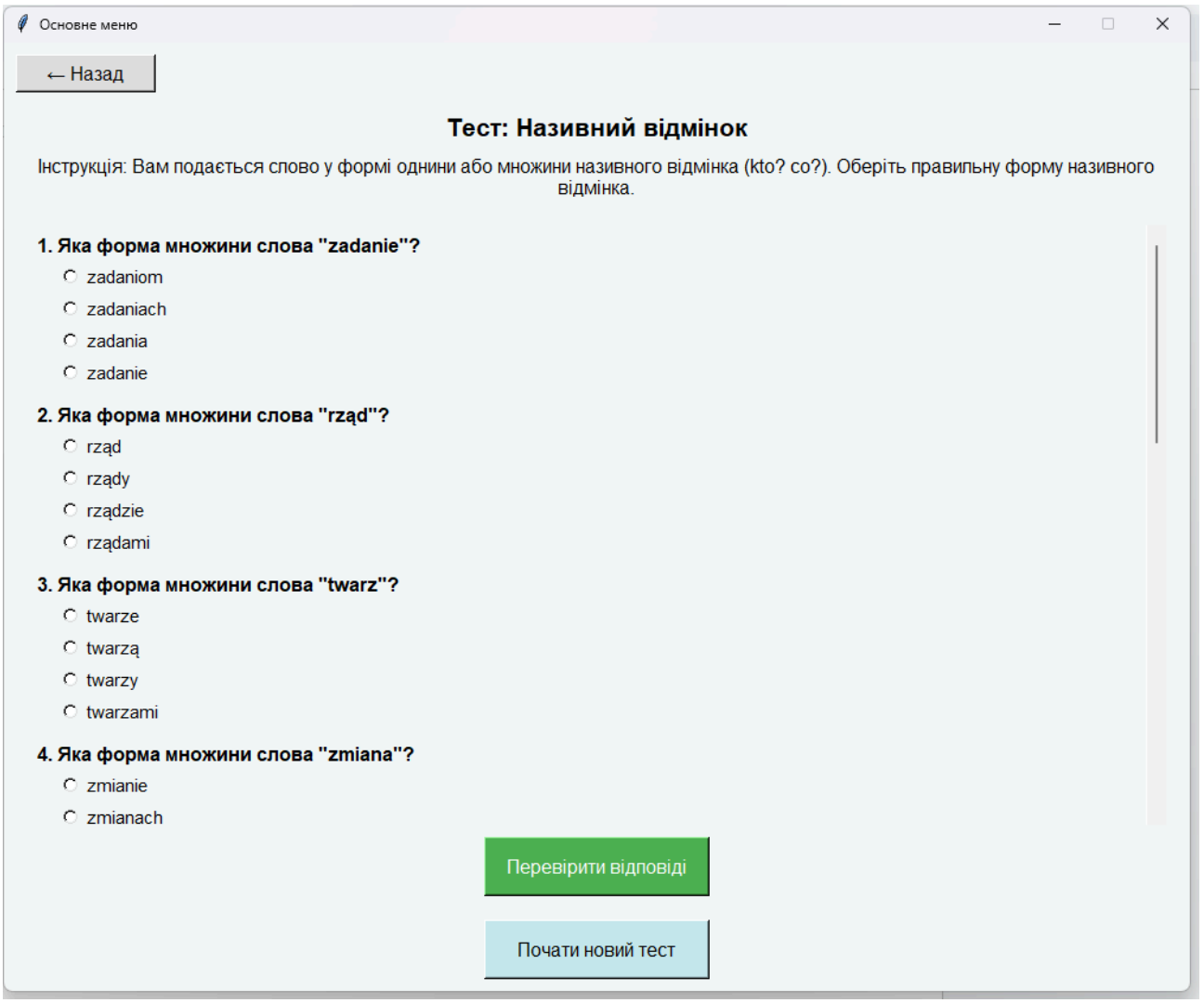
Мал. Д2.16. Приклад теоретичного розділу "Іменник" та реалізованої таблиці відмінків



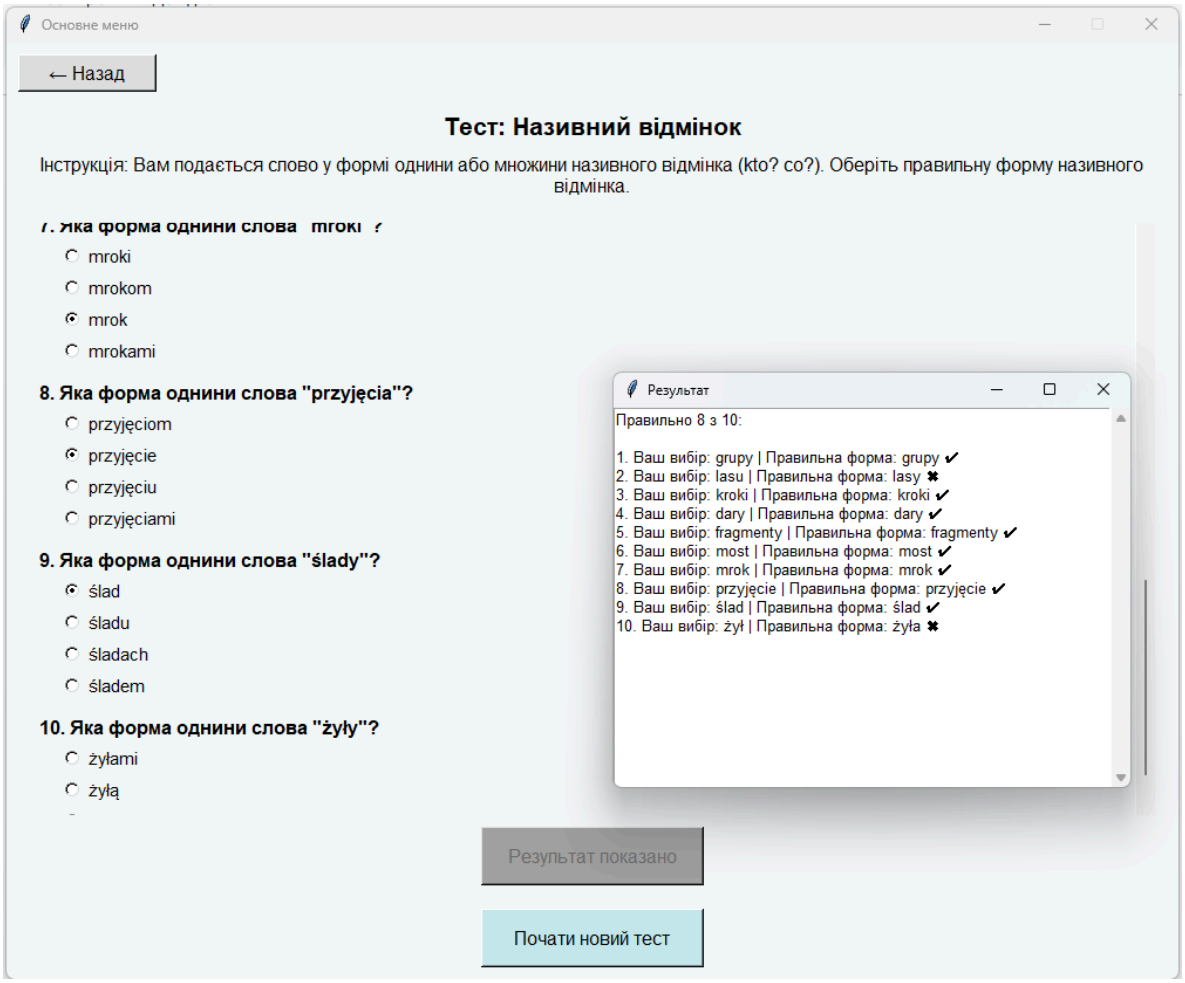
Мал. Д2.17. Приклад теоретичного розділу "Знахідний відмінок"



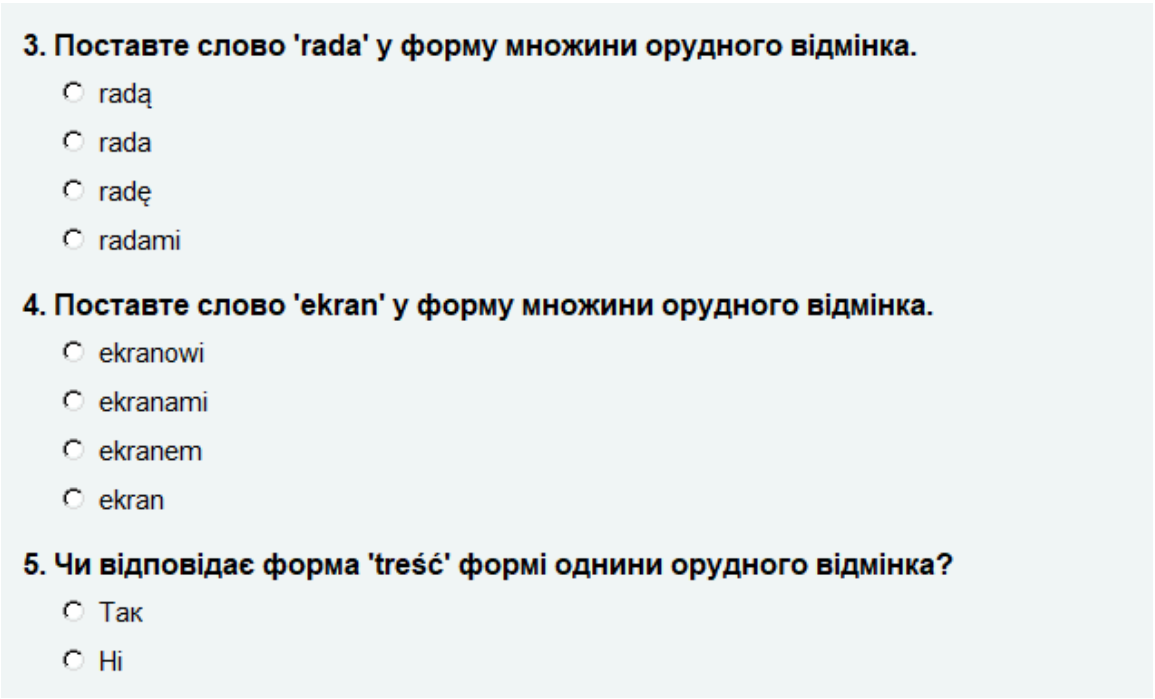
Мал. Д2.18. Приклад теоретичного розділу "Орудний відмінок"



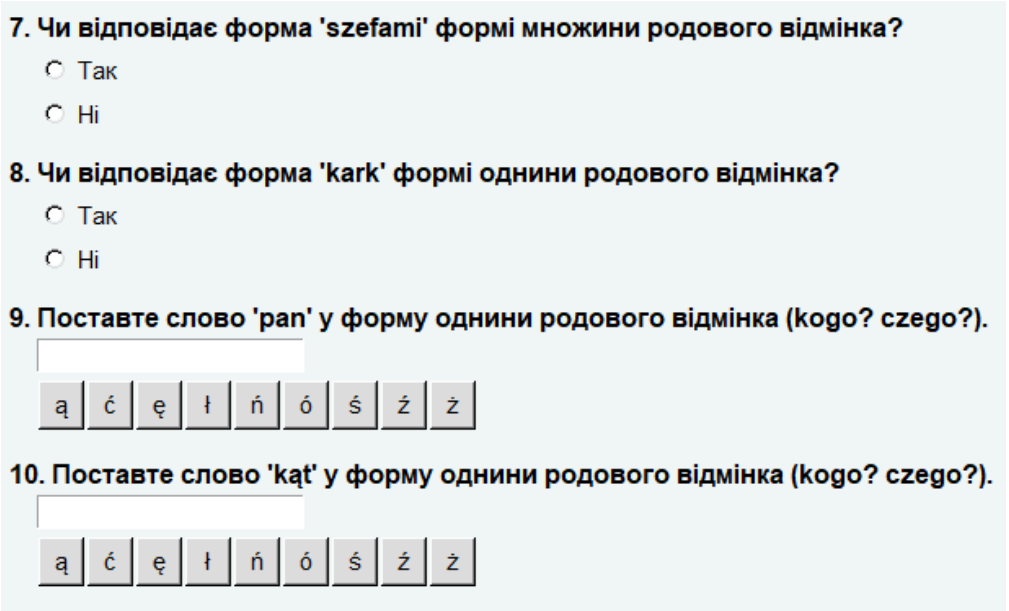
Мал. Д2.19. Приклад тесту на називний відмінок



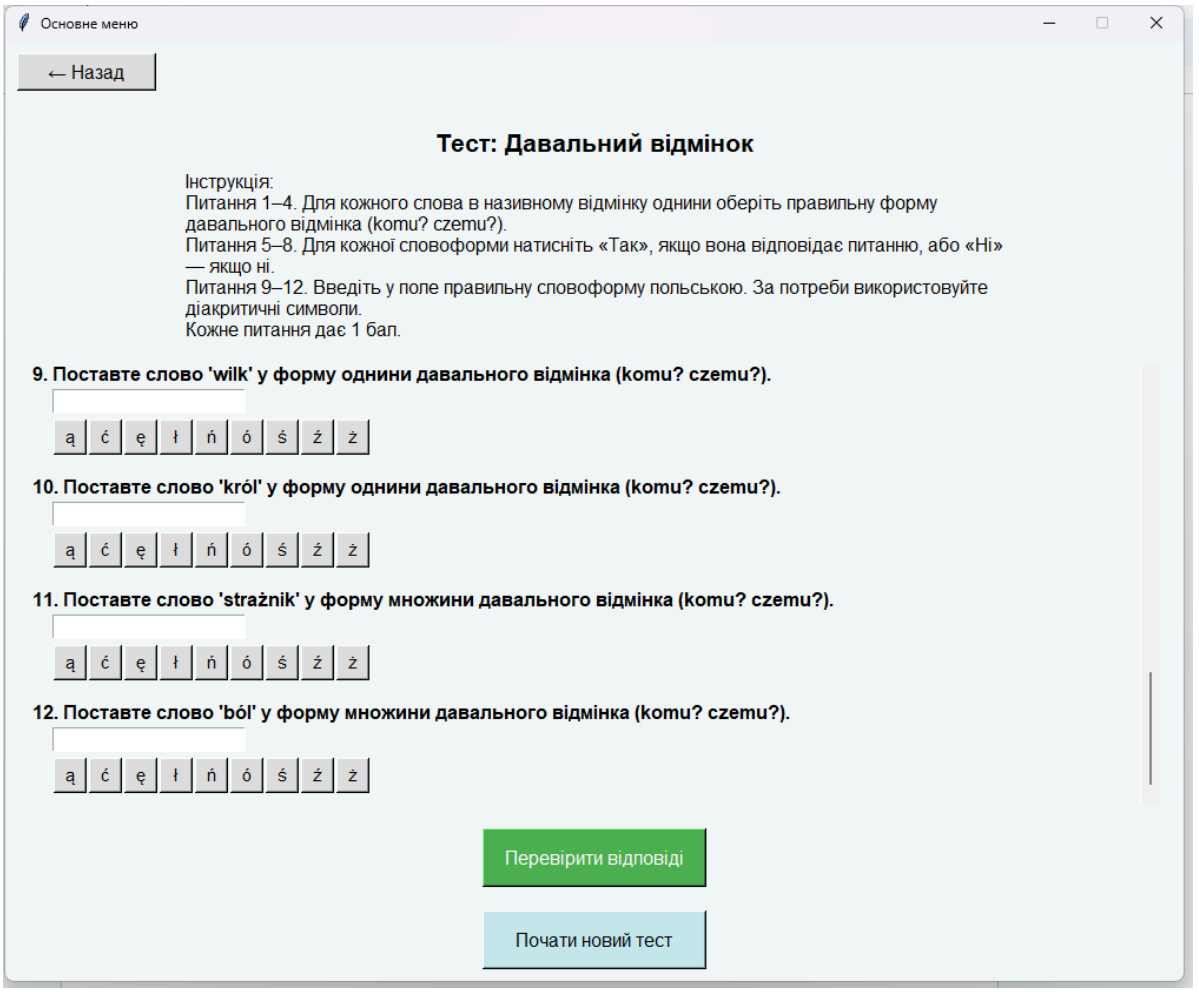
Мал. Д2.20. Приклад результатів тесту на називний відмінок



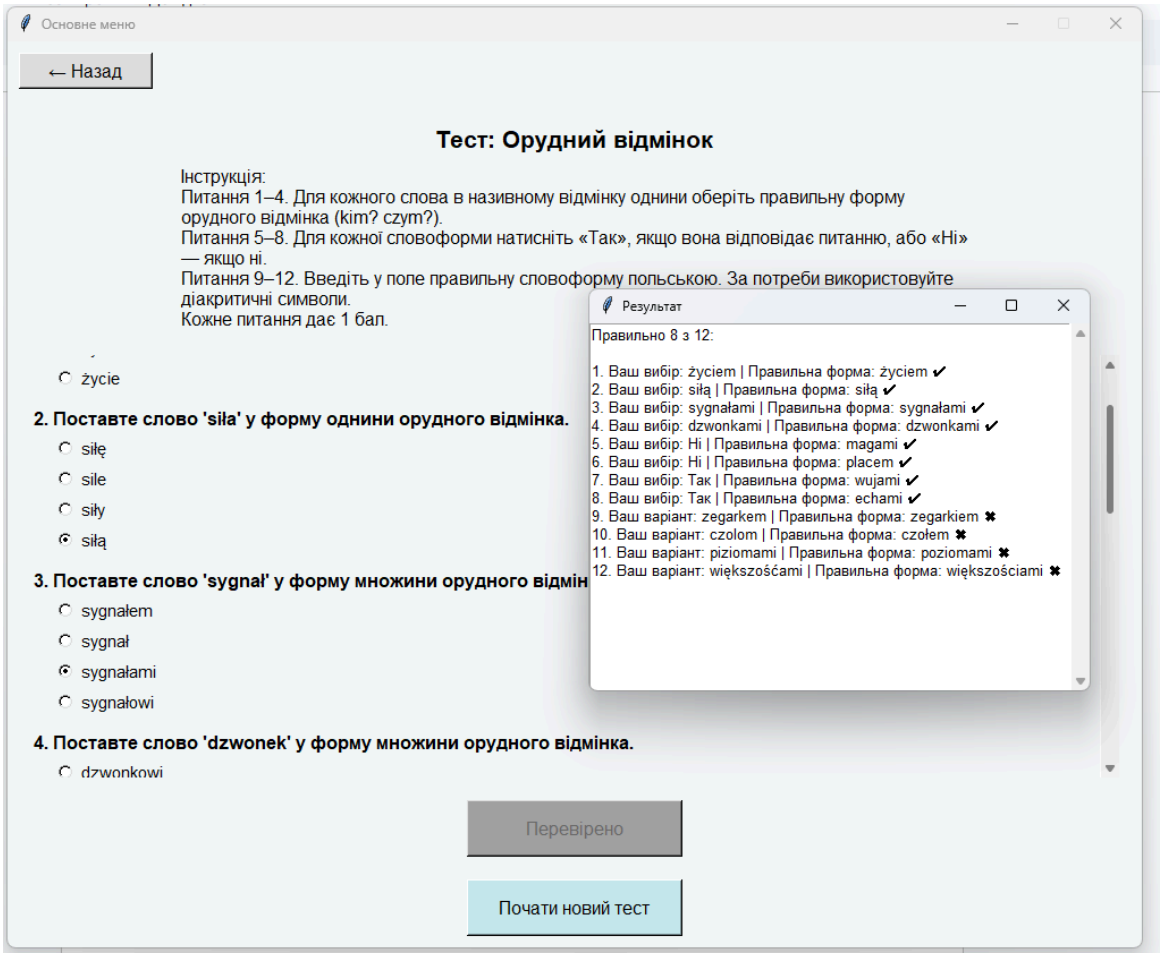
Мал. Д2.21. Приклад тесту на множинний вибір та формату "Так / Ні"



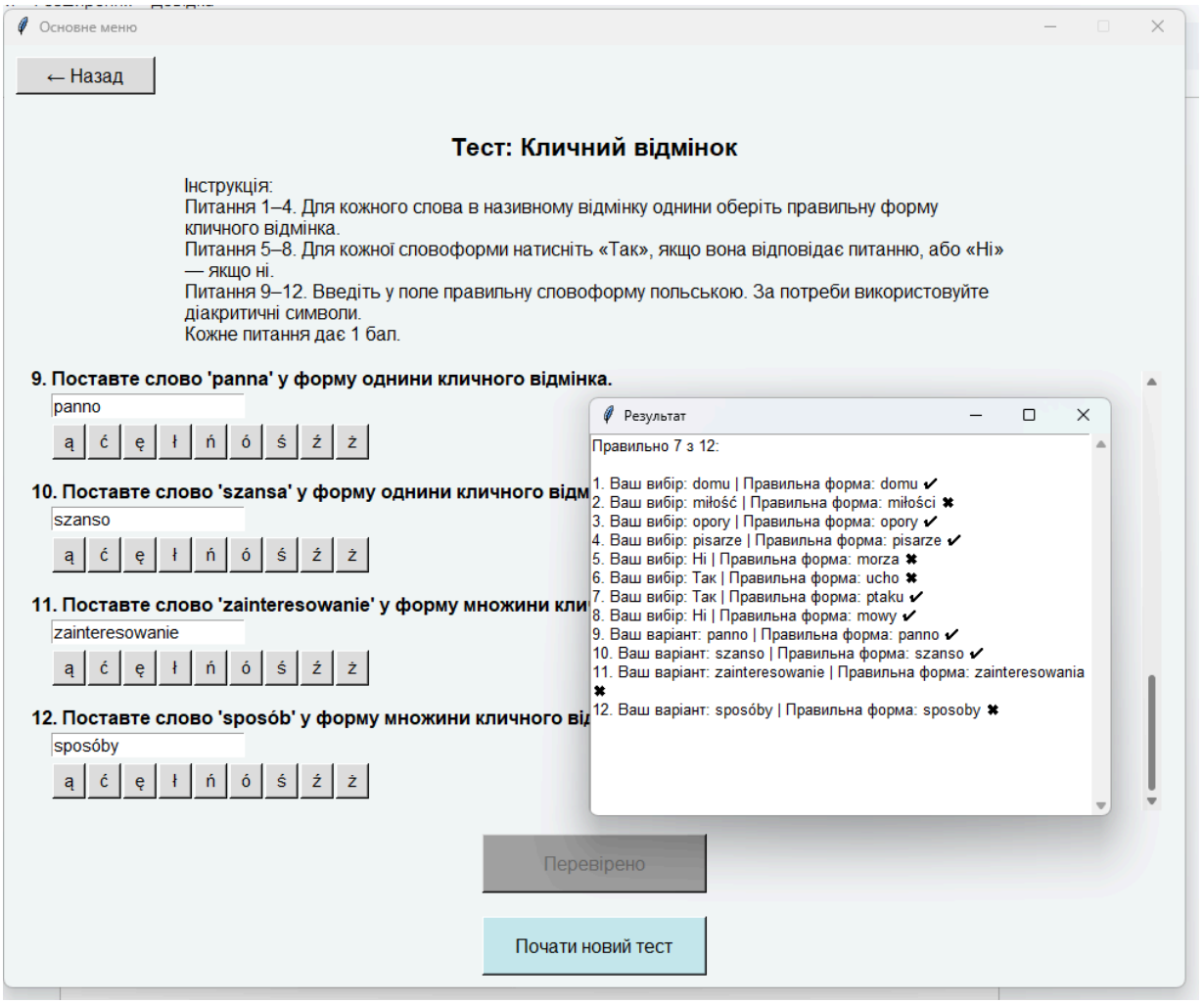
Мал. Д2.22. Приклад тесту формату "Так / Ні" та на відкриту відповідь



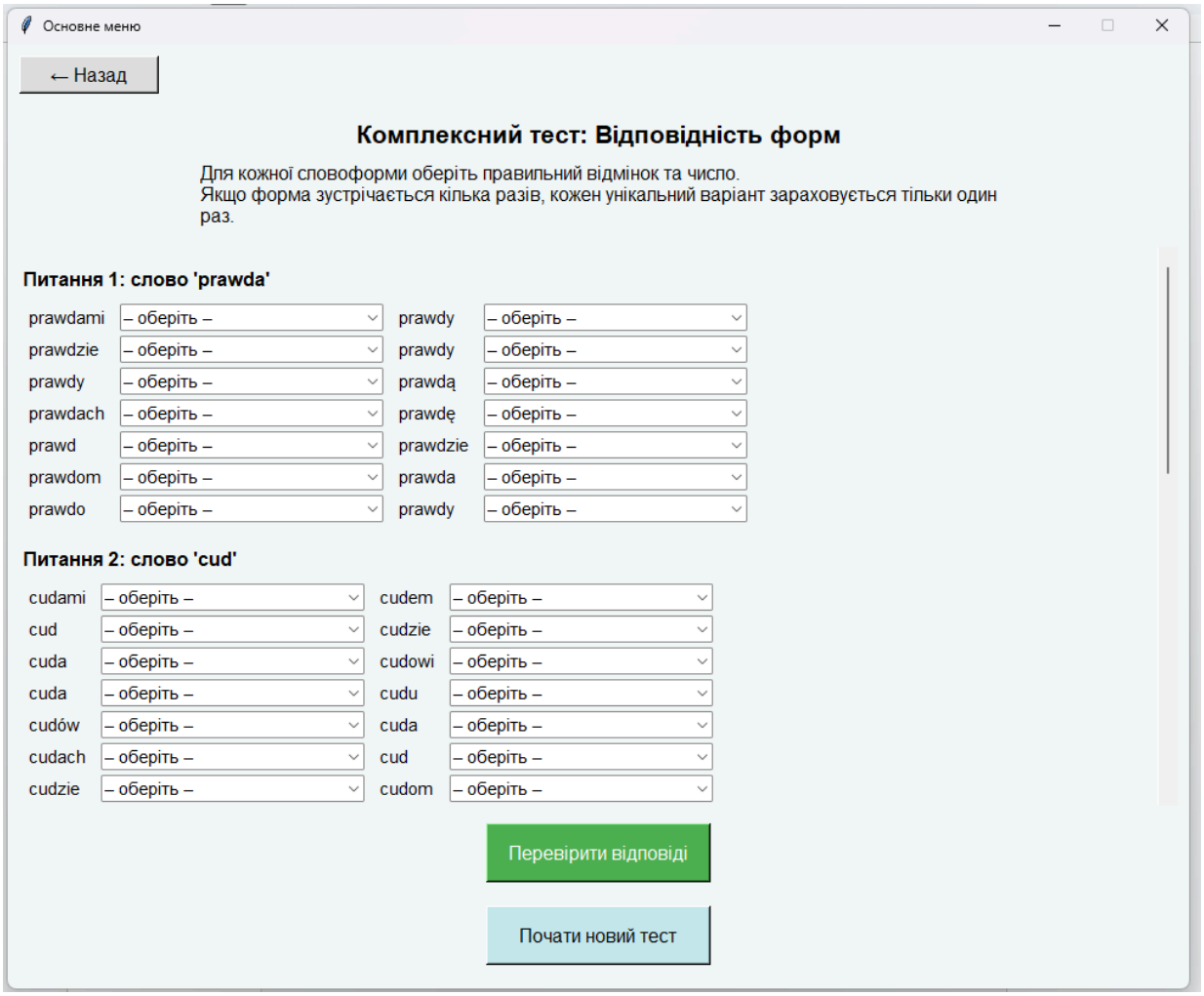
Мал. Д2.23. Приклад тесту на давальний відмінок



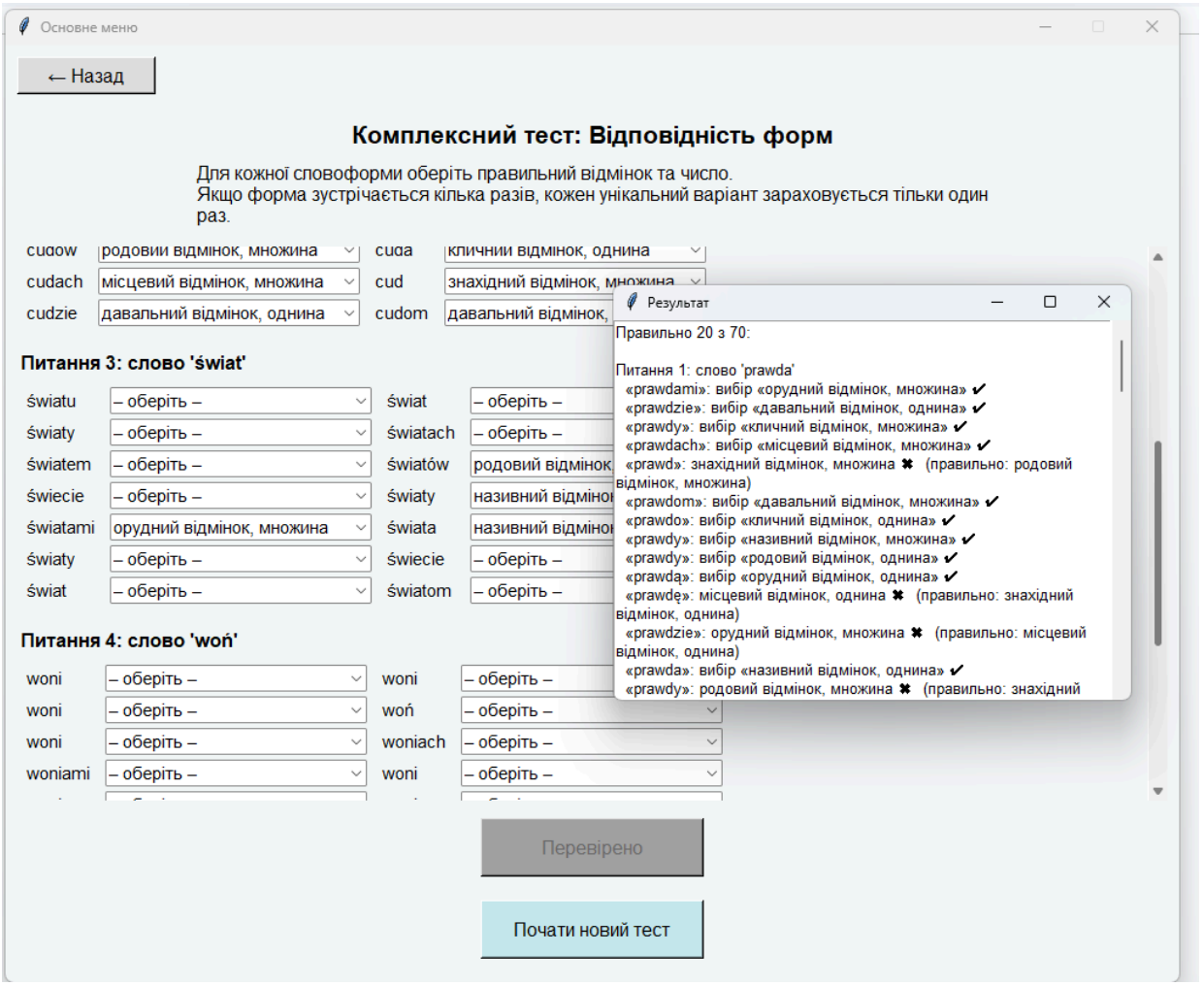
Мал. Д2.24. Приклад результатів тесту на орудний відмінок



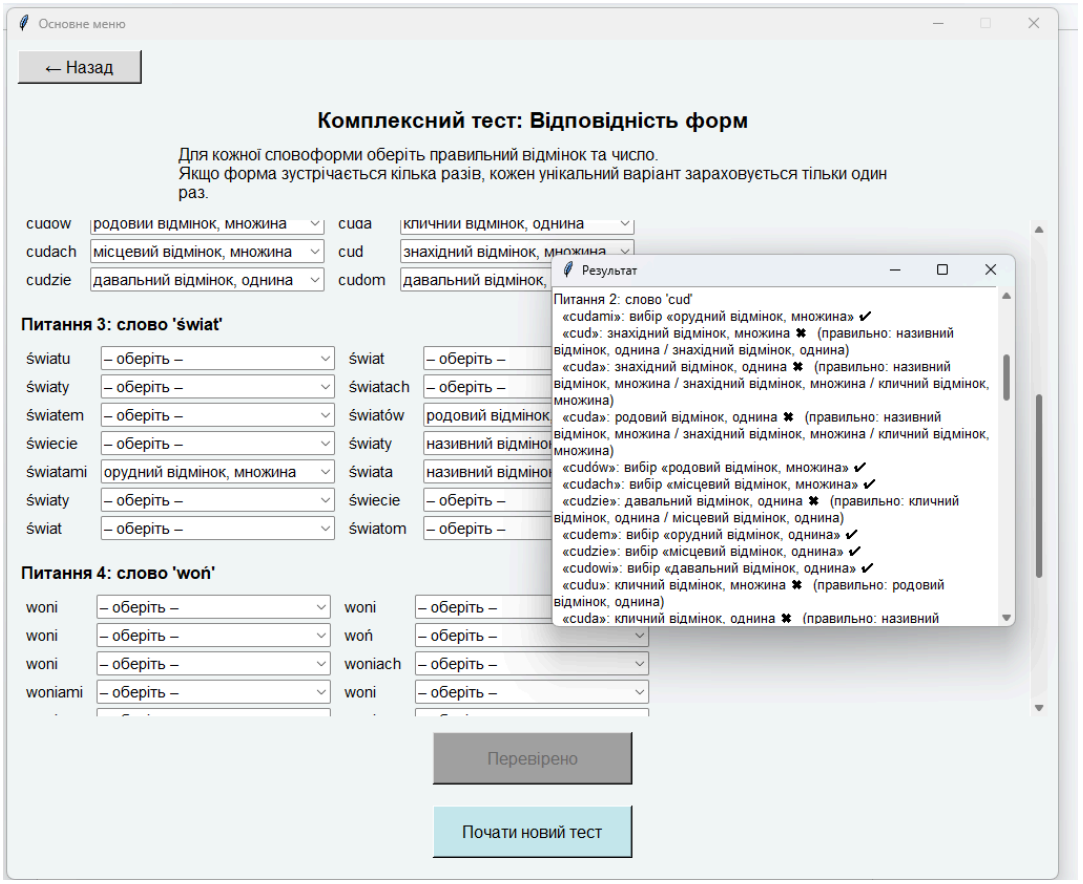
Мал. Д2.25. Приклад результатів тесту на кличний відмінок



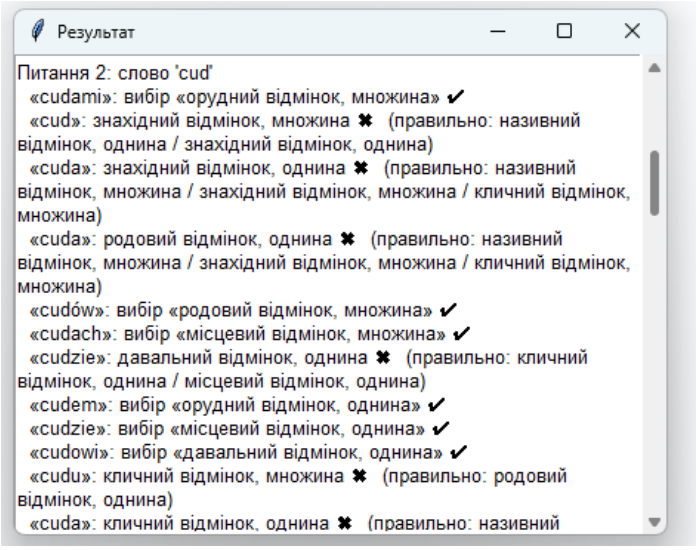
Мал. Д2.26. Приклад тесту на відповідність форм



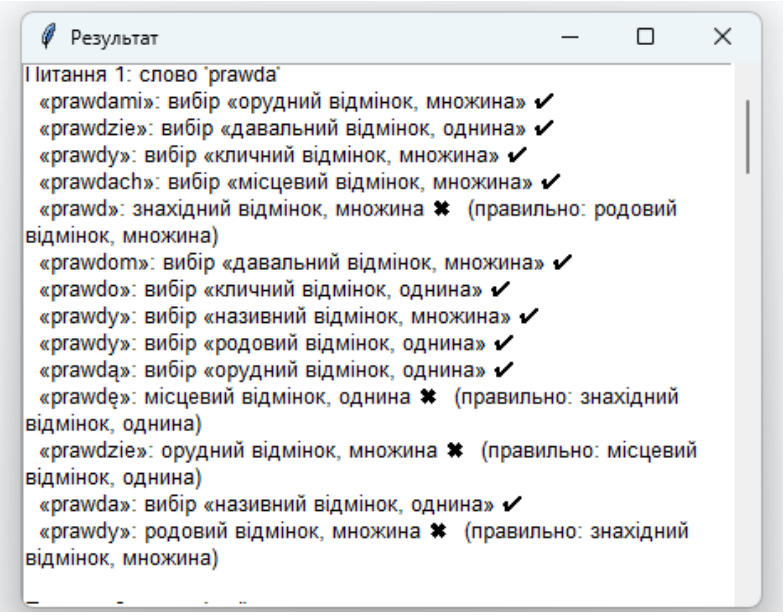
Мал. Д2.27. Приклад тесту на відповідність форм



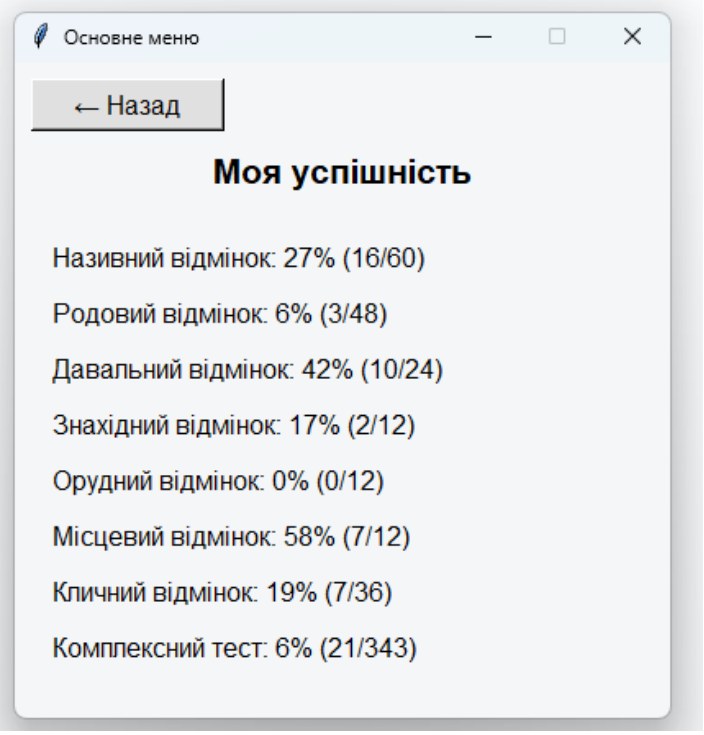
Мал. Д2.28. Приклад результату тесту на відповідність форм



Мал. Д2.29. Результат тесту на відповідність (словоформа “cud”)



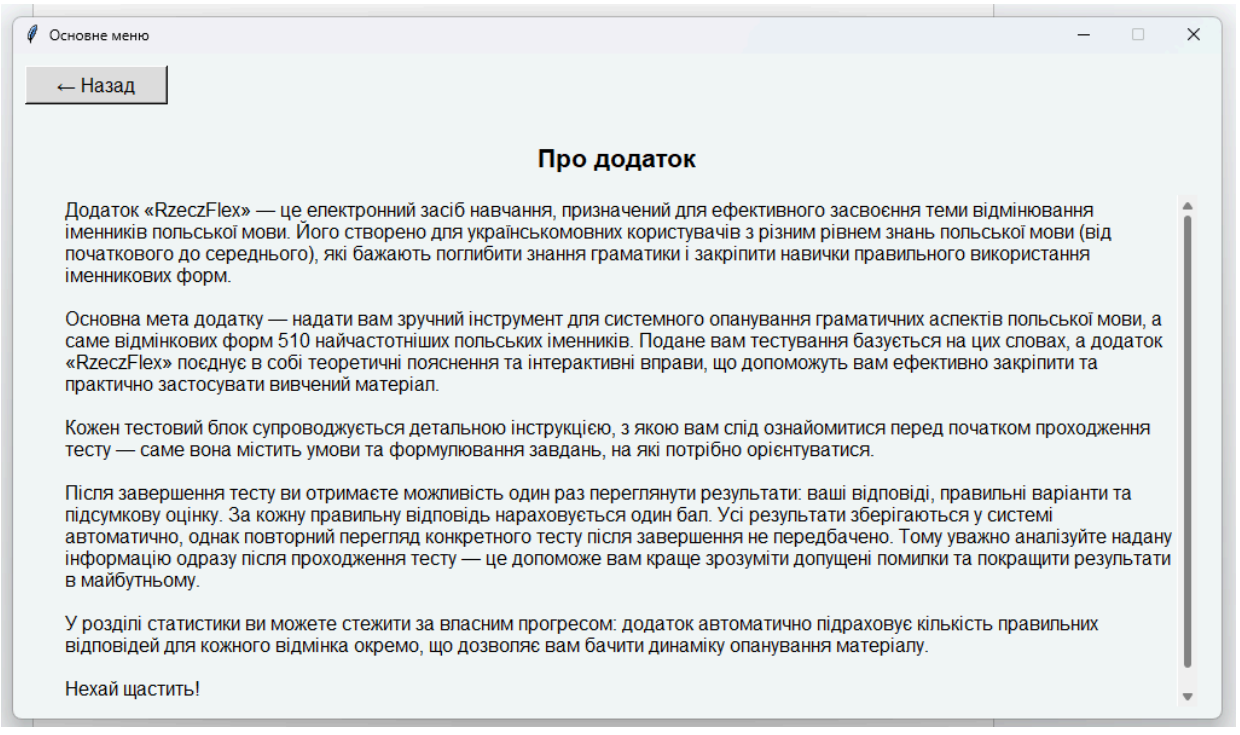
Мал. Д2.30. Результат тесту на відповідність (словоформа "prawda")



Мал. Д2.31. Приклад блоку "Моя успішність" в персональному акаунті користувача

	user_id	case_code	correct	total
	Filter	Filter	Filter	Filter
1	2	N	16	60
2	2	G	3	48
3	2	M	0	140
4	2	D	10	24
5	2	I	0	12
6	2	V	7	36
7	2	C	0	140
8	2	C	21	343
9	6	N	0	10
10	2	L	7	12
11	2	A	2	12

Мал. Д2.32. Фіксування результатів користувача у БД



Мал. Д2.32. Наповнення розділу “Про додаток”