

Міністерство освіти і науки України
Дрогобицький державний педагогічний університет імені Івана
Франка
кафедра фізики та інформаційних систем

«До захисту допускаю»

завідувач кафедри фізики

та інформаційних систем,

кандидат фіз.-мат. наук, доцент

_____ В. Б. Гольський

« ____ » _____ 2026 р.

Розроблення програми для вивчення іноземних слів

Спеціальність 122 Комп'ютерні науки

Випускова робота

на здобуття кваліфікації – бакалавр з комп'ютерних наук

Автор роботи Любас Віра Андріївна

_____ *підпис*

Науковий керівник: Доцент, кандидат фізико-математичних наук

Лешко Роман Ярославович

_____ *підпис*

Дрогобич, 2026

Анотація

Любас В. А. Розроблення програми для вивчення іноземних слів. Бакалаврська робота, Дрогобицький державний педагогічний університет імені Івана Франка, Дрогобич, 2026.

Робота присвячена розробці веборієнтованого програмного засобу для вивчення іноземних слів. Метою роботи є створення зручної та функціональної системи, яка дозволяє користувачам формувати власні словники, вивчати нову лексику та перевіряти рівень знань за допомогою інтерактивних вправ.

Для реалізації програмного засобу використані технології HTML, CSS та JavaScript. HTML застосовано для створення структури вебсторінок, CSS — для оформлення користувацького інтерфейсу та забезпечення адаптивного дизайну, JavaScript — для реалізації логіки роботи системи та взаємодії з користувачем. Для локального збереження даних використовується технологія LocalStorage.

У роботі реалізовано основний функціонал програмного засобу, зокрема: створення та керування мовними словниками, додавання слів із перекладами, поділ слів за рівнями складності A1–C2, імпорт словників із файлів формату JSON, CSV та TXT, автоматичне озвучування слів за допомогою Web Speech API, онлайн-переклад словників, а також навчальні модулі у вигляді флеш-карток, вікторини, анаграм та вправ на встановлення відповідностей між словами та перекладами.

Результатом виконання роботи є створений програмний засіб для вивчення іноземних слів, який забезпечує ефективне засвоєння нової лексики, контроль прогресу користувача та зручну організацію навчального процесу.

Загальний обсяг роботи включає аналіз предметної області, дослідження сучасних методів вивчення іноземних мов, проектування системи, реалізацію програмного забезпечення, тестування функціональних можливостей та оцінювання результатів роботи програмного засобу.

Annotations

Liubas V. Development of a Foreign Language Learning Application. Bachelor's Thesis, Drohobych Ivan Franko State Pedagogical University, Drohobych, 2026.

The thesis is devoted to the development of a web-based application for learning foreign words. The aim of the work is to create a convenient and functional system that allows users to create their own dictionaries, learn new vocabulary, and assess their knowledge through interactive exercises.

The application was developed using HTML, CSS, and JavaScript technologies. HTML was used to create the structure of web pages, CSS was used for interface design and responsive layout, and JavaScript was used to implement the application's functionality and user interaction. LocalStorage technology is used for local data storage.

The developed software includes the following functionality: creation and management of language dictionaries, adding words and translations, classification of vocabulary according to CEFR levels (A1–C2), importing dictionaries from JSON, CSV, and TXT files, automatic pronunciation of words using the Web Speech API, online translation of dictionaries, and learning modules such as flashcards, quizzes, anagrams, and matching exercises.

The result of this work is a software application for learning foreign words that provides effective vocabulary acquisition, user progress tracking, and convenient organization of the learning process.

The thesis includes an analysis of the subject area, research of modern foreign language learning methods, system design, software implementation, functional testing, and evaluation of the developed application

Зміст

Зміст

Вступ

1. Аналіз предметної області

1.1 Аналіз сучасних програм для вивчення іноземних мов

1.2 Методи та технології вивчення іноземних слів

1.3 Обґрунтування вибору технологій розробки програмного засобу

2.1 Постановка задачі та функціональні вимоги до системи

2.2 Архітектура та структура програмного засобу

2.3 Реалізація системи словників

2.4 Реалізація структури слова

2.5 Реалізація модуля вікторини

2.6 Реалізація модуля анаграм

2.7 Реалізація модуля встановлення відповідностей

2.8 Реалізація імпорту словників

2.9 Реалізація озвучування слів

2.10 Реалізація механізму LocalStorage

2.11 Опис інтерфейсу користувача

Головна сторінка програми

Навігаційне меню

Сторінка керування словниками

Сторінка словника

Інтерфейс флеш-карток

Інтерфейс вікторини

Інтерфейс анаграм

Інтерфейс встановлення відповідностей

ВИСНОВКИ

Використані джерела

Вступ

У сучасному світі знання іноземних мов є важливою складовою професійного та особистісного розвитку людини. Глобалізація, розвиток міжнародних відносин, поширення інформаційних технологій та активна взаємодія між країнами вимагають від фахівців володіння хоча б однією іноземною мовою. Особливо актуальним це є для студентів, програмістів, перекладачів, викладачів та інших спеціалістів, які працюють із міжнародними ресурсами, документацією або іноземними партнерами.

Одним із найскладніших етапів вивчення іноземної мови є запам'ятовування нових слів та їх подальше закріплення. Традиційні методи навчання, такі як використання паперових словників або записів у зошитах, поступово поступаються місцем сучасним цифровим технологіям. Використання комп'ютерних програм і веб-додатків дозволяє зробити процес навчання більш зручним, інтерактивним та ефективним.

Сьогодні існує значна кількість програм для вивчення мов, проте багато з них мають певні недоліки. Частина сервісів є платною або потребує оформлення підписки для отримання повного функціоналу. Інші програми не дозволяють користувачам створювати власні словники, налаштовувати навчальний матеріал відповідно до власних потреб або відстежувати особистий прогрес. Також багато існуючих рішень містять зайві функції, які ускладнюють використання програми та відволікають користувача від основної мети — ефективного вивчення нових слів.

У зв'язку з цим виникає необхідність створення програмного засобу, який забезпечить простий та зручний механізм додавання слів, формування власних словників, проходження різноманітних вправ та збереження результатів навчання. Така система повинна сприяти підвищенню ефективності засвоєння лексичного матеріалу та підтримувати інтерес користувача до процесу навчання.

Актуальність теми дипломної роботи обумовлена зростанням потреби у використанні сучасних інформаційних технологій для навчання іноземних мов. Розроблення програм для автоматизації процесу вивчення

слів дозволяє поєднати освітні методики з можливостями сучасних вебтехнологій, що забезпечує зручність використання та доступність програмного продукту для широкого кола користувачів.

Метою дипломної роботи є розроблення програмного засобу для вивчення іноземних слів, який забезпечує створення та редагування словників, проходження інтерактивних вправ різних типів, а також збереження результатів навчання користувача.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз предметної області та існуючих програм для вивчення іноземних мов;
- визначити функціональні та нефункціональні вимоги до програмного продукту;
- розробити структуру програмного забезпечення та інтерфейс користувача;
- реалізувати механізм створення та керування словниками;
- реалізувати систему збереження даних користувача;
- розробити модуль вправ для закріплення вивчених слів;
- реалізувати вправи у вигляді флеш-карток, тестування, анаграм та встановлення відповідностей;
- забезпечити можливість роботи з кількома мовними списками;
- провести тестування програмного продукту та оцінити коректність його роботи.

Об'єктом дослідження є процес вивчення та запам'ятовування іноземної лексики із застосуванням сучасних інформаційних технологій.

Предметом дослідження є методи та програмні засоби організації навчального процесу для вивчення іноземних слів за допомогою вебтехнологій.

Під час виконання дипломної роботи використовуються сучасні технології веброзробки, зокрема HTML для створення структури сторінок, CSS для оформлення інтерфейсу користувача та JavaScript для реалізації логіки роботи програми. Для збереження даних використовується механізм localStorage, що дозволяє зберігати інформацію без використання окремого сервера баз даних.

Практичне значення роботи полягає у створенні програмного продукту, який може використовуватися студентами, школярами та всіма користувачами, які вивчають іноземні мови. Програма дозволяє організувати навчальний процес, створювати власні набори слів, перевіряти рівень засвоєння матеріалу та відстежувати прогрес навчання. Результати роботи можуть бути використані як основа для подальшого розвитку системи та впровадження додаткових функцій, зокрема озвучування слів, використання технологій штучного інтелекту та синхронізації даних між різними пристроями.

Таким чином, розроблення програми для вивчення іноземних слів є актуальним та перспективним напрямом, що поєднує сучасні інформаційні технології та освітні методики, забезпечуючи ефективне засвоєння нового навчального матеріалу та підвищення мотивації користувачів до вивчення іноземних мов.

1. Аналіз предметної області

1.1 Аналіз сучасних програм для вивчення іноземних мов

У сучасному інформаційному суспільстві цифрові технології активно використовуються в освітній сфері. Особливого поширення набули програмні засоби для вивчення іноземних мов, які дозволяють організувати навчальний процес за допомогою комп'ютерів, смартфонів та планшетів. Завдяки таким програмам користувачі можуть отримувати доступ до навчальних матеріалів у будь-який час та з будь-якого місця, що значно підвищує ефективність самостійного навчання.

Основною метою програм для вивчення мов є допомога користувачу у засвоєнні нових слів, розвитку навичок читання, письма, аудіювання та мовлення. Більшість сучасних систем використовують інтерактивний підхід до навчання, що дозволяє підтримувати інтерес користувача та підвищувати його мотивацію.

Однією з найпопулярніших платформ для вивчення іноземних мов є Duolingo. Даний сервіс пропонує користувачам навчання у формі коротких інтерактивних вправ та тестів. Система використовує елементи гейміфікації, такі як накопичення балів, досягнення, щоденні завдання та рейтинги. Завдяки цьому процес навчання стає більш захоплюючим та мотивує користувача регулярно повертатися до занять.

Перевагами Duolingo є простий інтерфейс, підтримка великої кількості мов та можливість безкоштовного використання основного функціоналу. Разом із тим сервіс має певні недоліки. Зокрема, користувач не завжди може гнучко налаштувати власні словники та працювати виключно з власним набором слів. Частина функціональних можливостей доступна лише після оформлення платної підписки.

Іншим відомим сервісом є Memrise. Основною особливістю цієї платформи є використання методики повторення навчального матеріалу через певні проміжки часу. Такий підхід сприяє довготривалому запам'ятовуванню нових слів та фраз. Крім того, користувачі можуть працювати з готовими курсами або створювати власні набори лексики.

Перевагою Memrise є велика кількість навчальних матеріалів та можливість використання різноманітних способів повторення слів. Однак значна частина функцій також доступна лише у платній версії. Крім того, інтерфейс програми може бути складним для нових користувачів через велику кількість додаткових можливостей.

Широко використовується також сервіс Quizlet. Дана система дозволяє створювати власні картки для вивчення слів та використовувати різноманітні режими навчання. Користувач може самостійно формувати словники відповідно до своїх потреб, що є важливою перевагою для студентів та учнів.

Quizlet підтримує режим флеш-карток, тестування, письмові вправи та інші інструменти перевірки знань. Основною перевагою системи є її універсальність та можливість використання в освітньому процесі. Недоліком є обмеження безкоштовної версії та відсутність деяких функцій персоналізації навчання.

Окремої уваги заслуговує застосування мобільних додатків для вивчення мов. Сучасні смартфони дозволяють користувачам навчатися практично в будь-якому місці. Мобільні застосунки забезпечують швидкий доступ до навчальних матеріалів та дозволяють використовувати короткі проміжки часу для повторення слів.

Водночас багато існуючих програм орієнтовані на масового користувача та не враховують індивідуальні особливості навчання. Користувачі часто стикаються з обмеженнями щодо створення власних словників, налаштування вправ або відстеження особистого прогресу. Крім того, частина програм потребує постійного підключення до мережі Інтернет, що не завжди є зручним.

Аналіз існуючих програмних продуктів показав, що більшість із них використовують подібні методи навчання, серед яких флеш-картки, тести, вправи на встановлення відповідностей та повторення матеріалу. Саме ці інструменти демонструють високу ефективність при вивченні нових слів та активно використовуються в сучасних освітніх системах.

Проведене дослідження дозволило визначити основні функціональні можливості, які доцільно реалізувати у розроблюваній програмі. До них

належать створення кількох мовних словників, додавання та редагування слів, проходження інтерактивних вправ різних типів, збереження результатів навчання та підтримка декількох рівнів складності.

Таким чином, аналіз сучасних програм для вивчення іноземних мов підтвердив актуальність розробки власного програмного засобу. Майбутня система повинна поєднувати простоту використання, можливість персоналізації навчання та наявність інтерактивних механізмів перевірки знань, що сприятиме підвищенню ефективності процесу запам'ятовування іноземної лексики.

1.2 Методи та технології вивчення іноземних слів

Ефективне вивчення іноземної мови неможливе без постійного розширення словникового запасу. Саме знання лексики є основою для розвитку навичок читання, письма, аудіювання та усного мовлення. Незалежно від рівня володіння мовою користувач повинен регулярно засвоювати нові слова та повторювати вже вивчений матеріал. У зв'язку з цим особливого значення набувають методи та технології, які дозволяють зробити процес запам'ятовування більш ефективним.

Протягом багатьох років для вивчення нової лексики використовувалися традиційні методи навчання. До них належать робота зі словниками, записування нових слів у зошит, багаторазове переписування перекладів та механічне заучування. Незважаючи на свою простоту, такі підходи мають ряд недоліків. Вони часто потребують значних витрат часу та не забезпечують достатнього рівня зацікавленості користувача. Як наслідок, значна частина інформації швидко забувається.

З розвитком інформаційних технологій з'явилися нові підходи до організації навчального процесу. Комп'ютерні програми та вебдодатки дозволили автоматизувати значну частину навчання та зробити його більш інтерактивним. Сучасні системи використовують різноманітні вправи, елементи гейміфікації та механізми контролю знань, що позитивно впливає на якість засвоєння матеріалу.

Одним із найбільш поширених методів вивчення іноземних слів є використання флеш-карток. Даний підхід передбачає демонстрацію користувачеві слова іноземною мовою або його перекладу з подальшою перевіркою правильності відповіді. Картки можуть відображати слово, переклад, транскрипцію або приклади використання.

Перевагою флеш-карток є простота реалізації та висока ефективність. Користувач концентрує увагу на окремих словах, що сприяє кращому запам'ятовуванню. Крім того, флеш-картки можна використовувати як для первинного ознайомлення з новим матеріалом, так і для повторення вже вивченої лексики.

Ще одним ефективним методом є тестування. У тестових завданнях користувачеві пропонується декілька варіантів відповіді, серед яких необхідно обрати правильний переклад слова або фрази. Такий підхід дозволяє швидко перевірити рівень знань та виявити слова, які потребують додаткового повторення.

Тестові вправи мають ряд переваг. Вони забезпечують швидкий зворотний зв'язок, дозволяють автоматично оцінювати результати та створюють умови для самоконтролю користувача. Крім того, використання випадкового порядку запитань допомагає уникнути механічного запам'ятовування послідовності відповідей.

Важливу роль у процесі навчання відіграють вправи на встановлення відповідностей. У таких завданнях користувачеві необхідно знайти правильну пару між словом та його перекладом. Подібний метод сприяє розвитку асоціативного мислення та допомагає швидше формувати зв'язок між словами різними мовами.

Вправи на співставлення активно використовуються у сучасних освітніх системах завдяки своїй наочності та простоті. Вони дозволяють працювати з великою кількістю слів одночасно та забезпечують високу швидкість перевірки знань.

Одним із нестандартних способів вивчення лексики є використання анаграм. Анаграмою називається слово, літери якого перемішані у випадковому порядку. Завдання користувача полягає у відновленні правильного написання слова.

Подібний вид вправ сприяє розвитку орфографічних навичок та допомагає краще запам'ятовувати структуру слова. На відміну від звичайного читання або тестування, анаграми змушують користувача активно аналізувати послідовність літер, що позитивно впливає на якість запам'ятовування.

Окремої уваги заслуговує метод інтервального повторення. Його суть полягає в тому, що слова повторюються через певні проміжки часу. Якщо користувач правильно відповідає на запитання, інтервал до наступного повторення збільшується. Якщо відповідь неправильна, слово повторюється частіше.

Наукові дослідження показують, що людина поступово забуває нову інформацію після її вивчення. Регулярне повторення матеріалу дозволяє сповільнити цей процес та перевести знання до довготривалої пам'яті. Саме тому технологія інтервального повторення широко використовується у сучасних освітніх платформах.

Крім спеціалізованих методів навчання, важливим фактором є мотивація користувача. Для її підтримки багато програм використовують елементи гейміфікації. До таких елементів належать система балів, рівнів складності, досягнень, статистики та прогресу навчання.

Гейміфікація дозволяє зробити навчальний процес більш цікавим та захоплюючим. Користувач отримує додатковий стимул для регулярного використання програми та досягнення поставлених цілей. Особливо ефективним такий підхід є для студентів та молоді, які звикли до інтерактивних цифрових середовищ.

У сучасних системах також активно використовуються мультимедійні технології. Озвучування слів, використання зображень та інтерактивних елементів сприяє кращому сприйняттю інформації. Одночасне використання текстової, візуальної та звукової інформації дозволяє активізувати декілька каналів сприйняття та підвищити ефективність навчання.

Під час розробки програмного засобу для вивчення іноземних слів доцільно поєднати декілька методів навчання. Використання флеш-карток, тестування, вправ на співставлення та анаграм дозволить забезпечити

різноманітність навчального процесу та підтримувати інтерес користувача до роботи з програмою.

Таким чином, аналіз сучасних методів вивчення іноземних слів показав, що найбільш ефективними є інтерактивні підходи, які забезпечують активну участь користувача у навчальному процесі. Саме тому при розробці програмного продукту було прийнято рішення реалізувати декілька типів вправ та механізмів перевірки знань, що дозволить підвищити ефективність засвоєння нової лексики та зробити процес навчання більш зручним і результативним.

1.3 Обґрунтування вибору технологій розробки програмного засобу

Одним із важливих етапів створення будь-якого програмного продукту є вибір технологій розробки. Від правильності вибору програмних засобів залежить швидкість розробки, зручність супроводу, продуктивність роботи системи та можливість її подальшого вдосконалення. Для реалізації програмного засобу вивчення іноземних слів було обрано сучасні вебтехнології, які дозволяють створювати інтерактивні та зручні для користувача програмні продукти.

Основними технологіями, використаними під час розробки системи, є HTML, CSS та JavaScript. Даний набір технологій є стандартом для створення вебдодатків та забезпечує повний набір інструментів для реалізації інтерфейсу користувача, логіки роботи програми та взаємодії з локальними даними.

Використання HTML

HTML (HyperText Markup Language) є стандартною мовою розмітки, яка використовується для створення структури вебсторінок. Саме за допомогою HTML формуються всі елементи інтерфейсу користувача, включаючи заголовки, кнопки, текстові поля, таблиці, списки та інші компоненти.

У розроблюваній програмі HTML використовується для створення основних сторінок системи, таких як головна сторінка, сторінка словників,

флеш-картки, тестування, анаграми та вправи на встановлення відповідностей. Завдяки використанню HTML вдалося створити зрозумілу структуру програми та забезпечити зручну навігацію між окремими модулями.

Однією з переваг HTML є простота використання та універсальність. Створені сторінки можуть відкриватися практично у будь-якому сучасному браузері без необхідності встановлення додаткового програмного забезпечення. Це дозволяє зробити програму доступною для широкого кола користувачів.

Крім того, HTML підтримує семантичну розмітку, яка покращує читабельність коду та полегшує його подальше супроводження. Завдяки цьому розробник може швидше вносити зміни та розширювати функціональні можливості системи.

Використання CSS

Для оформлення зовнішнього вигляду програми використовується CSS (Cascading Style Sheets). Дана технологія дозволяє визначати кольори, шрифти, відступи, розміри елементів та інші параметри оформлення вебсторінок.

Використання CSS забезпечує відокремлення логіки роботи програми від її візуального представлення. Такий підхід спрощує модифікацію інтерфейсу та дозволяє швидко змінювати зовнішній вигляд системи без необхідності внесення змін до основного програмного коду.

У межах даного проєкту CSS використовується для створення сучасного та зрозумілого інтерфейсу користувача. Особлива увага приділяється зручності розташування елементів керування, читабельності тексту та адаптації сторінок до різних розмірів екранів.

Застосування каскадних таблиць стилів дозволило створити єдиний стиль оформлення для всіх сторінок програми. Це позитивно впливає на зручність використання системи та формує цілісне враження від програмного продукту.

Важливою перевагою CSS є підтримка адаптивного дизайну. Завдяки цьому інтерфейс програми може коректно відображатися як на персональних комп'ютерах, так і на мобільних пристроях, що значно розширює можливості використання розробленого програмного забезпечення.

Використання JavaScript

JavaScript є мовою програмування, яка забезпечує реалізацію логіки роботи вебдодатків. Саме за допомогою JavaScript здійснюється обробка дій користувача, створення інтерактивних елементів та управління даними.

У розроблюваній програмі JavaScript використовується для створення функціоналу словників, додавання та редагування слів, формування вправ, перевірки правильності відповідей та збереження результатів навчання.

Однією з головних переваг JavaScript є можливість виконання програмного коду безпосередньо у браузері користувача. Це дозволяє створювати швидкі та інтерактивні вебдодатки без необхідності використання серверного обладнання.

JavaScript також забезпечує роботу з елементами вебсторінки через об'єктну модель документа (DOM). Завдяки цьому програма може динамічно змінювати вміст сторінок без їх повторного завантаження. Такий підхід підвищує швидкодію системи та покращує взаємодію користувача з програмою.

Додатковою перевагою JavaScript є наявність великої кількості готових бібліотек та інструментів, які можуть використовуватися для подальшого розвитку системи. Це створює хороші можливості для масштабування програмного продукту у майбутньому.

Використання LocalStorage

Для збереження даних користувача у програмі використовується механізм LocalStorage. Дана технологія входить до складу сучасних веббраузерів та дозволяє зберігати інформацію без використання окремої бази даних.

LocalStorage забезпечує довготривале збереження даних навіть після закриття браузера або перезавантаження сторінки. Це дозволяє користувачу продовжувати навчання без втрати створених словників та результатів проходження вправ.

У межах програмного продукту LocalStorage використовується для зберігання мовних списків, словників, перекладів, рівнів складності та статистики навчання. Завдяки цьому користувач може працювати з програмою без підключення до мережі Інтернет.

Порівняно з використанням повноцінної бази даних LocalStorage має значно простішу реалізацію та не потребує встановлення серверної частини системи. Це особливо важливо для навчального проєкту, оскільки дозволяє зосередитися на реалізації основного функціоналу програми.

Водночас необхідно враховувати, що LocalStorage має певні обмеження щодо обсягу даних та рівня захисту інформації. Тому у майбутньому можлива модернізація системи шляхом підключення серверної бази даних, наприклад MySQL, PostgreSQL або MongoDB.

Переваги обраного технологічного стеку

Вибір HTML, CSS, JavaScript та LocalStorage обумовлений рядом переваг. По-перше, усі зазначені технології є безкоштовними та широко поширеними серед розробників. По-друге, вони дозволяють створити повноцінний програмний продукт без необхідності використання додаткового програмного забезпечення.

Обраний набір технологій забезпечує простоту розробки, швидкість виконання програми та зручність використання. Крім того, така архітектура дозволяє легко модернізувати систему в майбутньому шляхом підключення серверної частини, баз даних або технологій штучного інтелекту.

Таким чином, проведений аналіз показав, що використання HTML, CSS, JavaScript та LocalStorage є доцільним рішенням для реалізації програмного засобу вивчення іноземних слів. Дані технології забезпечують необхідний рівень функціональності, зручності використання та можливості подальшого розвитку програмного продукту.

Розділ 2. Проектування та розроблення програмного засобу для вивчення іноземних слів

2.1 Постановка задачі та функціональні вимоги до системи

У процесі виконання дипломної роботи поставлено завдання розробити веборієнтований програмний засіб для вивчення іноземних слів, який забезпечить користувачам можливість створення власних словників, збереження навчального матеріалу та проходження інтерактивних вправ для закріплення вивченої лексики.

Актуальність розробки обумовлена зростанням попиту на цифрові інструменти для самостійного вивчення іноземних мов. Сучасні користувачі потребують програмних засобів, які дозволяють не лише переглядати нові слова, але й організовувати власні словники, контролювати прогрес навчання та використовувати різноманітні методики запам'ятовування лексики.

Основною метою розробки є створення зручного та функціонального вебзастосунку, який надає можливість працювати з власними словниками без необхідності використання серверної бази даних або складної процедури реєстрації користувачів. Усі дані повинні зберігатися локально в браузері, що забезпечує автономність роботи системи та простоту використання.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- реалізувати систему створення та керування словниками;
- забезпечити можливість додавання, редагування та видалення слів;
- реалізувати підтримку декількох мовних списків;
- додати механізм імпорту словників із зовнішніх файлів;
- забезпечити можливість озвучування слів;
- реалізувати автоматичний переклад словників;
- створити навчальні модулі для закріплення лексики;
- реалізувати систему збереження даних користувача;

– забезпечити адаптивний та зручний інтерфейс користувача.

Функціональні можливості системи повинні забезпечувати повний цикл роботи зі словниковими даними. Користувач має можливість створювати власні словники, додавати нові слова та переклади, визначати рівень складності слів відповідно до шкали CEFR, імпортувати готові словникові бази та використовувати інтерактивні вправи для закріплення матеріалу.

До основних функціональних вимог системи належать:

- створення необмеженої кількості словників;
- підтримка рівнів складності A1, A2, B1, B2, C1 та C2;
- поділ слів на нові та вивчені;
- використання флеш-карток;
- проходження вікторин;
- виконання вправ типу анаграма;
- встановлення відповідностей між словами та перекладами;
- автоматичне озвучування слів;
- онлайн-переклад словників;
- імпорт словників із файлів формату JSON, CSV та TXT;
- збереження результатів роботи у LocalStorage.

Таким чином, поставлена задача полягає у створенні повноцінного програмного засобу для вивчення іноземних слів, який поєднує можливості керування словниками, інтерактивного навчання та локального збереження даних користувача.

2.2 Архітектура та структура програмного засобу

Розроблений програмний засіб для вивчення іноземних слів реалізований у вигляді вебзастосунку з використанням технологій HTML, CSS та JavaScript. Вибір даних технологій обумовлений їх доступністю,

простотою використання та можливістю створення інтерактивного користувацького інтерфейсу без використання серверної частини.

Архітектура програмного засобу побудована за клієнтським принципом. Усі дані користувача зберігаються локально в браузері за допомогою технології LocalStorage, що дозволяє використовувати програму без створення окремої бази даних або серверного програмного забезпечення.

Основними складовими системи є:

- інтерфейс користувача;
- модуль керування словниками;
- модуль збереження даних;
- модуль флеш-карток;
- модуль вікторини;
- модуль анаграм;
- модуль співвідношення;
- модуль імпорту словників;
- модуль озвучування слів;
- модуль автоматичного перекладу.

Загальна структура проєкту представлена набором HTML-сторінок, які взаємодіють із єдиним JavaScript-модулем та спільним файлом стилів.

Таблиця 2.1 – Структура проєкту

Файл	Призначення
index.html	Головна сторінка
words.html	Робота зі словниками
dictionary.html	Перегляд словника
flashcards.html	Флеш-картки

quiz.html	Вікторина
anagram.html	Анаграма
matching.html	Співвідношення
style.css	Оформлення інтерфейсу
script.js	Основна логіка програми

Для забезпечення зручності навігації всі сторінки використовують однакове меню переходів. Це дозволяє користувачеві швидко перемикатися між режимами навчання та словниками.

Інтерфейс програми реалізовано за допомогою HTML та CSS. Кожний модуль має окрему HTML-сторінку, що забезпечує зрозумілу структуру проєкту та спрощує подальшу підтримку програмного забезпечення.

Основна логіка системи реалізована у файлі script.js. Саме цей файл відповідає за створення словників, додавання нових слів, запуск навчальних вправ, імпорт словників, озвучування слів, автоматичний переклад та збереження інформації користувача.

Для локального збереження даних використовується LocalStorage. Такий підхід дозволяє відмовитися від використання серверної бази даних та забезпечує автономність роботи застосунку. Усі словники, результати вправ та налаштування користувача автоматично зберігаються у пам'яті браузера.

Особливістю архітектури розробленого програмного засобу є поєднання декількох навчальних модулів у межах єдиної системи. Користувач може працювати зі словниками, проходити флеш-картки, вікторини, анаграми та вправи на встановлення відповідностей без переходу до сторонніх сервісів.

Таким чином, обрана архітектура забезпечує простоту використання, зручність підтримки програмного коду, високу швидкість роботи та можливість подальшого розширення функціональних можливостей системи.

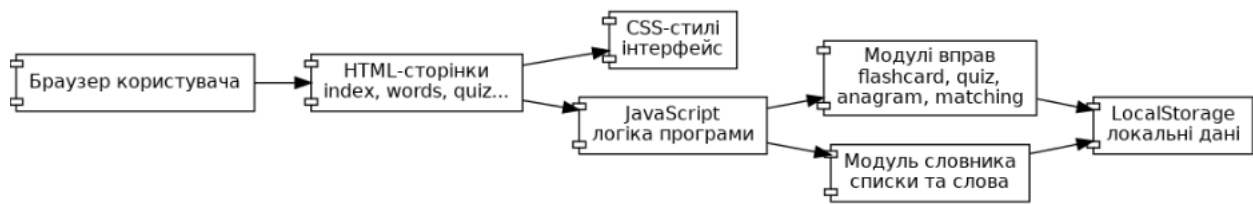


Рисунок 2.1 – Компонентна діаграма програмного засобу

Компонентна діаграма демонструє взаємодію між основними модулями системи. Центральним компонентом є файл `script.js`, який забезпечує обробку дій користувача та взаємодію між інтерфейсом і локальним сховищем даних. Усі HTML-сторінки використовують спільний файл стилів `style.css` та звертаються до єдиного JavaScript-модуля для виконання логіки роботи програми.

2.3 Реалізація системи словників

Одним із найважливіших компонентів розробленого програмного засобу є система керування словниками. Саме вона забезпечує можливість зберігання навчального матеріалу, організацію слів за мовами та подальше використання цих даних у навчальних вправах.

На відміну від багатьох аналогічних програм, створений застосунок підтримує роботу з декількома словниками одночасно. Користувач може створювати окремі списки для різних мов, наприклад англійської, польської, німецької або інших мов, які вивчаються.

Для зберігання словників використовується структура даних у вигляді об'єкта JavaScript, де кожний словник містить власний масив слів.

Під час запуску програми виконується перевірка наявності збережених словників у `LocalStorage`. Якщо інформація відсутня, автоматично створюється початковий словник `English`.

Приклад ініціалізації структури даних:

```

let allLists = JSON.parse(
  localStorage.getItem("allLists")
);
  
```

```
if (!allLists) {
  allLists = {
    "English": []
  };
}
```

Такий підхід забезпечує автоматичне створення початкового середовища для роботи користувача навіть під час першого запуску програми.

Створення нового словника здійснюється за допомогою функції `createList()`. Після введення назви нового списку система перевіряє коректність введених даних та наявність словника з аналогічною назвою.

У разі відсутності такого словника створюється новий порожній масив для зберігання слів.

Фрагмент програмного коду:

```
if (allLists[newListName]) {
  alert("Такий список уже існує!");
  return;
}

allLists[newListName] = [];
```

Після створення нового словника він автоматично додається до списку доступних словників та стає активним для подальшої роботи.

Для вибору активного словника використовується елемент `Select`, який автоматично оновлюється після створення нового списку. Перемикання між словниками реалізовано функцією `changeList()`.

Важливою перевагою даного рішення є можливість незалежного зберігання словникових баз для різних мов. Кожний словник містить власний набір слів та може використовуватися окремо від інших.

Після будь-якої зміни даних відбувається автоматичне збереження інформації до LocalStorage. Це дозволяє уникнути втрати даних у разі закриття браузера або перезавантаження сторінки.

Алгоритм створення нового словника можна представити таким чином:

1. Користувач вводить назву нового словника.
2. Система перевіряє правильність введених даних.
3. Виконується перевірка на існування словника.
4. Створюється новий масив для зберігання слів.
5. Дані зберігаються у LocalStorage.
6. Оновлюється список доступних словників.
7. Новий словник стає активним.

Використання такої системи керування словниками забезпечує гнучкість роботи користувача та дозволяє організувати навчальний матеріал відповідно до власних потреб.

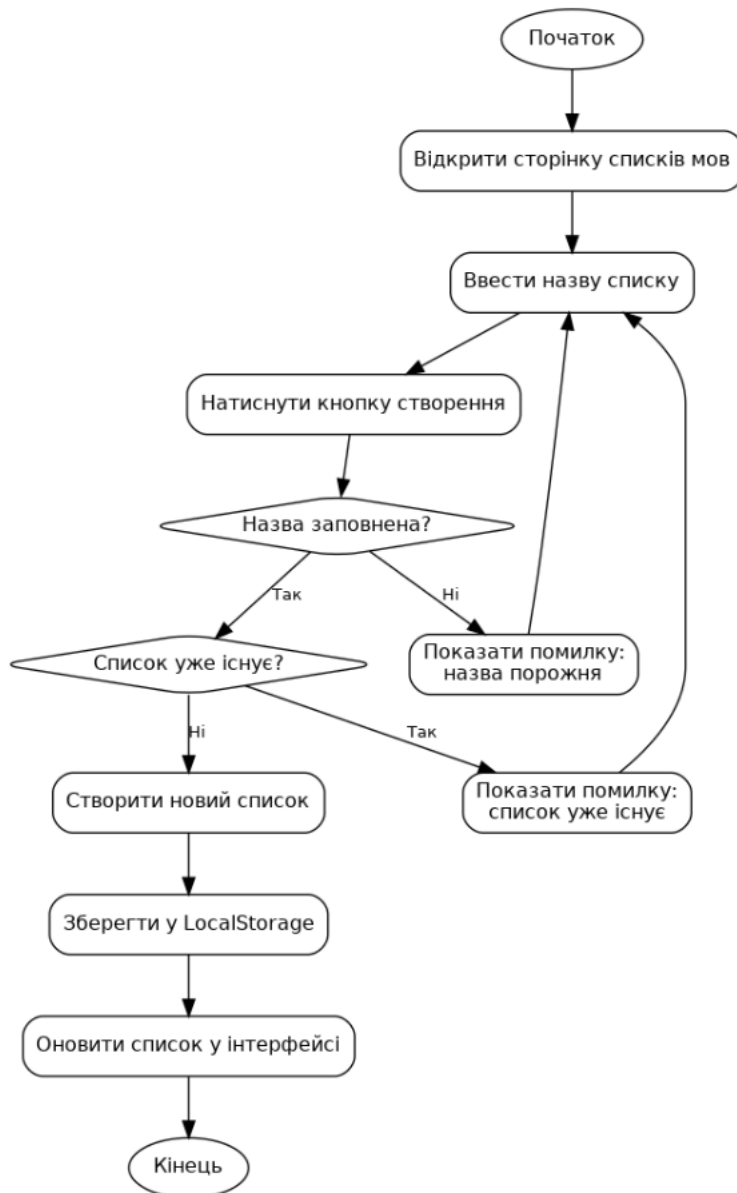


Рисунок 2.2 – UML-діаграма діяльності створення словника

На рисунку представлено послідовність дій під час створення нового словника. Діаграма демонструє процес введення назви списку, перевірки її коректності, створення нового словника та збереження результатів у локальному сховищі браузера.

Таким чином, реалізована система словників забезпечує зручне керування навчальними матеріалами та є основою функціонування всіх інших модулів програмного засобу.

2.4 Реалізація структури слова

Після створення системи словників необхідно було реалізувати механізм зберігання окремих слів та їх характеристик. Для цього в програмному засобі використовується об'єктна структура даних, яка дозволяє зберігати всю необхідну інформацію про кожне слово.

Кожне слово у системі представлено окремим об'єктом JavaScript. Такий підхід забезпечує зручність обробки даних та можливість подальшого розширення функціоналу без зміни загальної структури програми.

Під час додавання нового слова користувач вводить слово іноземною мовою, його переклад та вибирає рівень складності. Після цього система створює новий об'єкт та додає його до активного словника.

Структура слова має такий вигляд:

```
{
  word:                word,
  translation:         translation,
  level:                level,
  status: "new"
}
```

Дана структура містить чотири основні поля.

Поле **word** використовується для зберігання слова іноземною мовою. Саме це значення використовується під час роботи флеш-карток, вікторини, анаграм та інших навчальних вправ.

Поле **translation** містить переклад слова рідною мовою користувача. Переклад використовується для перевірки відповідей та формування навчальних завдань.

Поле **level** визначає рівень складності слова відповідно до міжнародної шкали CEFR. Використання рівнів дозволяє групувати лексику за складністю та організувати процес навчання більш ефективно.

У програмі підтримуються такі рівні:

- A1 — початковий рівень;

- A2 — базовий рівень;
- B1 — середній рівень;
- B2 — вище середнього;
- C1 — просунутий рівень;
- C2 — вільне володіння мовою.

Поле **status** використовується для відображення поточного стану слова у процесі навчання.

У системі реалізовано два основні статуси:

- new — нове слово;
- learned — вивчене слово.

Після додавання слова до словника йому автоматично присвоюється статус "new". У процесі проходження навчальних вправ статус може змінюватися залежно від результатів користувача.

Для додавання нового слова використовується функція `addWord()`. Перед створенням нового об'єкта система перевіряє коректність введених даних та наявність необхідних значень у полях введення.

Алгоритм додавання слова складається з таких етапів:

1. Отримання слова з форми введення.
2. Отримання перекладу.
3. Отримання рівня складності.
4. Перевірка правильності введених даних.
5. Створення нового об'єкта слова.
6. Додавання об'єкта до активного словника.
7. Збереження змін у `LocalStorage`.
8. Оновлення інтерфейсу користувача.

Особливістю реалізації є використання єдиної структури даних для всіх модулів програми. Незалежно від того, чи використовується слово у словнику, флеш-картках, вікторині або анаграмі, система працює з одним і тим самим об'єктом.

Такий підхід дозволяє уникнути дублювання даних та забезпечує узгодженість інформації між усіма частинами програмного засобу.

Крім того, структура слова легко масштабується. У майбутньому до неї можна додати нові характеристики, наприклад дату вивчення, кількість правильних відповідей, кількість повторень або індивідуальний рейтинг складності слова.

Таким чином, реалізована структура слова забезпечує зручне зберігання навчальної інформації, підтримує класифікацію лексики за рівнями складності та дозволяє відстежувати прогрес користувача під час навчання.

2.5 Реалізація модуля вікторини

Для перевірки рівня засвоєння нової лексики у програмному засобі реалізовано модуль вікторини. Даний режим навчання дозволяє користувачу перевіряти свої знання шляхом вибору правильної відповіді серед декількох запропонованих варіантів.

Модуль вікторини реалізовано на окремій вебсторінці `quiz.html`. Інтерфейс містить область для відображення запитання, блок із варіантами відповідей та поле для відображення результатів перевірки.

Основною функцією запуску вправи є `startQuiz()`.

Після запуску система автоматично отримує список слів із поточного словника та перевіряє їх кількість. Для коректної роботи вікторини необхідно, щоб словник містив щонайменше чотири слова.

Перевірка реалізується таким чином:

```
if (words.length < 4) {  
    alert("Для вікторини потрібно мінімум 4 слова!");  
    return;  
}
```

Дана умова дозволяє уникнути ситуацій, коли система не може сформулювати необхідну кількість варіантів відповіді.

Після успішної перевірки словника випадковим чином обирається слово, яке буде використане як запитання. Його переклад стає правильною відповіддю, а інші варіанти формуються з перекладів випадково вибраних слів.

Для створення випадкового порядку варіантів використовується алгоритм перемішування масиву:

```
options.sort(() => Math.random() - 0.5);
```

Такий підхід забезпечує випадкове розташування правильної відповіді та запобігає її запам'ятовуванню за позицією.

Після формування варіантів система динамічно створює кнопки відповідей та відображає їх користувачу.

Алгоритм роботи вікторини має такий вигляд:

1. Отримання слів із поточного словника.
2. Перевірка мінімальної кількості слів.
3. Випадковий вибір слова.
4. Формування правильної відповіді.
5. Генерація неправильних варіантів.
6. Перемішування відповідей.
7. Відображення варіантів користувачу.
8. Вибір відповіді.
9. Перевірка результату.
10. Збереження статистики.

Після натискання користувачем на один із варіантів відповіді виконується перевірка правильності вибору.

У випадку правильної відповіді система збільшує лічильник правильних відповідей та відображає повідомлення про успішне виконання завдання.

Якщо користувач обрав неправильний варіант, система збільшує лічильник помилок та повідомляє правильний переклад слова.

Однією з переваг реалізованого рішення є автоматичне накопичення статистики навчання. Користувач може поступово покращувати результати та відстежувати власний прогрес під час проходження вправ.

Використання тестових завдань дозволяє активізувати процес пригадування вивченого матеріалу та сприяє більш якісному закріпленню нової лексики.

Крім того, режим вікторини створює додаткову мотивацію до навчання, оскільки користувач отримує миттєвий зворотний зв'язок щодо правильності своїх відповідей.

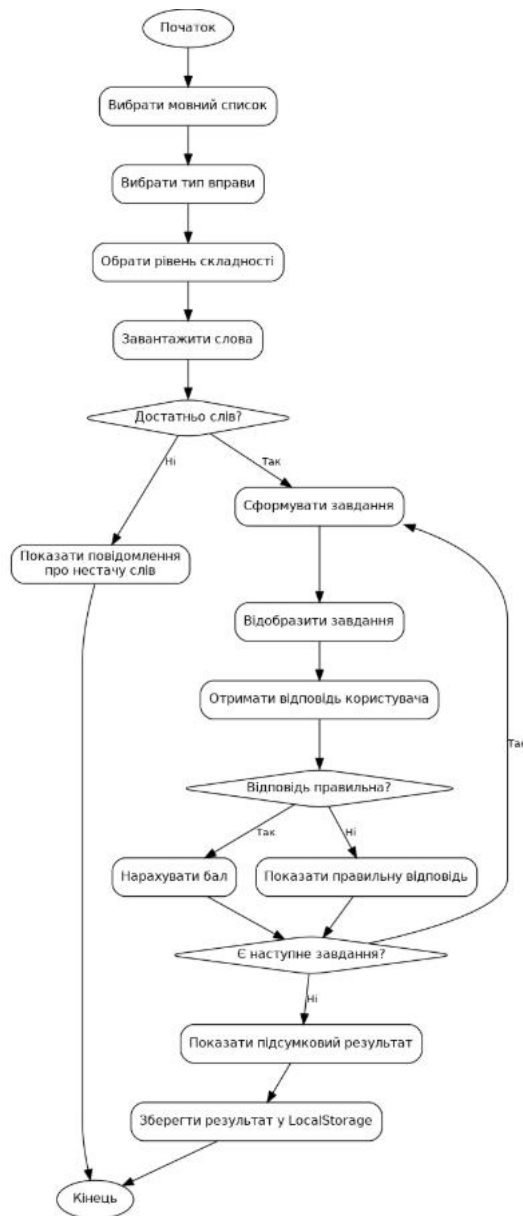


Рисунок 2.3 – UML-діаграма діяльності проходження вікторини

Діаграма діяльності демонструє процес створення тестового завдання, вибору відповіді користувачем та перевірки результату.

Таким чином, реалізований модуль вікторини забезпечує ефективний механізм контролю знань та дозволяє оцінити рівень засвоєння вивченого матеріалу.

2.6 Реалізація модуля анаграм

Для покращення навичок написання іноземних слів та розвитку асоціативного мислення у програмному засобі реалізовано модуль анаграм. Даний режим навчання базується на принципі перемішування літер слова, після чого користувач повинен відновити його правильне написання.

Модуль анаграм реалізовано на окремій вебсторінці `anagram.html`. Інтерфейс складається з області відображення анаграми, поля для введення відповіді та кнопок керування вправою.

Основною функцією запуску вправи є `startAnagram()`.

Після запуску система автоматично отримує список слів із поточного словника та випадковим чином вибирає одне з них для формування нового завдання.

Для створення анаграми використовується спеціальна функція перемішування символів слова.

```
function shuffleWord(word) {  
  return word  
    .split("")  
    .sort(() => Math.random() - 0.5)  
    .join("");  
}
```

Принцип роботи алгоритму полягає у розбитті слова на окремі символи, випадковому перемішуванні отриманого масиву та подальшому об'єднанні літер у новий рядок.

Наприклад, для слова:

`computer`

може бути створена анаграма:

otupmrec

або будь-який інший випадковий варіант розташування літер.

Після генерації анаграми користувач бачить на екрані набір перемішаних символів та повинен ввести правильний варіант слова у відповідне поле введення.

Перевірка відповіді здійснюється за допомогою функції `checkAnagram()`.

Після натискання кнопки «Перевірити» система порівнює введене користувачем слово з оригінальним значенням.

У випадку правильної відповіді користувач отримує повідомлення про успішне виконання завдання, а статистика правильних відповідей збільшується.

Якщо відповідь неправильна, система повідомляє про помилку та збільшує лічильник неправильних відповідей.

Алгоритм роботи модуля анаграм складається з таких етапів:

1. Завантаження слів із поточного словника.
2. Випадковий вибір слова.
3. Перемішування літер.
4. Відображення анаграми.
5. Введення відповіді користувачем.
6. Перевірка правильності відповіді.
7. Оновлення статистики.
8. Формування нового завдання.

Однією з переваг використання анаграм є розвиток навичок правильного написання слів. На відміну від флеш-карток або вікторини, де користувач переважно розпізнає правильну відповідь, у даній вправі необхідно самостійно відновити правильний порядок літер.

Крім того, регулярне використання анаграм сприяє покращенню зорової пам'яті та формуванню стійких асоціацій між написанням слова та його значенням.

Реалізований модуль може використовуватися для вивчення слів будь-якої мови, оскільки працює зі словниковими даними незалежно від обраного мовного списку.

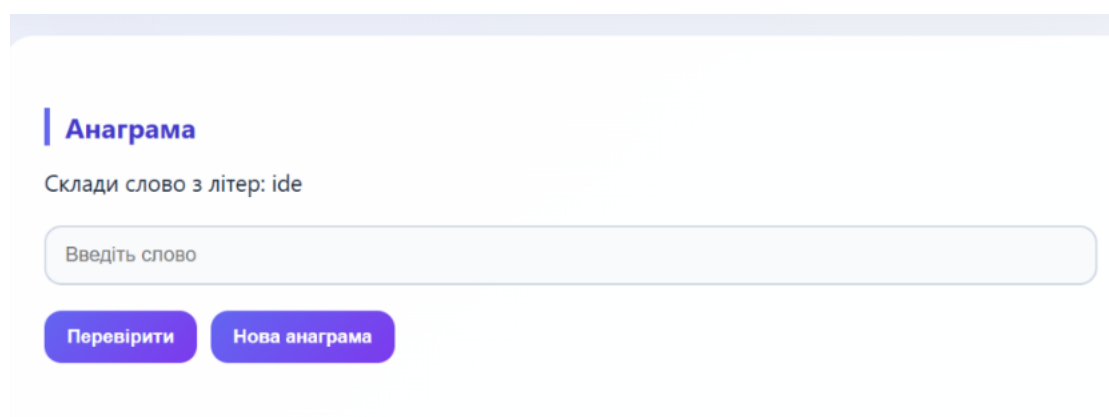


Рисунок 2.4 – Інтерфейс модуля анаграм

На рисунку представлено інтерфейс вправи з відображенням анаграми та полем введення відповіді.

Таким чином, модуль анаграм є важливим елементом навчальної системи та дозволяє ефективно закріплювати навички правильного написання іноземних слів.

2.7 Реалізація модуля встановлення відповідностей

Для закріплення знань користувача та розвитку асоціативного мислення у програмному засобі реалізовано модуль встановлення відповідностей між словами та їх перекладами. Даний тип вправ широко використовується у сучасних системах вивчення іноземних мов, оскільки дозволяє швидко формувати стійкі асоціативні зв'язки між новими словами та їх значеннями.

Модуль реалізовано на окремій вебсторінці `matching.html`. Інтерфейс вправи складається з двох незалежних блоків: списку слів

іноземною мовою та списку перекладів. Завдання користувача полягає у правильному співставленні відповідних елементів.

Основною функцією запуску вправи є `startMatching()`.

Після запуску система автоматично завантажує слова з активного словника та формує набір навчальних пар.

Для підвищення різноманітності завдань використовується випадковий вибір слів.

Фрагмент програмного коду:

```
matchingPairs = [...words]
.sort() => Math.random() - 0.5)
.slice(0, 4);
```

Даний алгоритм виконує перемішування словникового масиву та вибирає чотири випадкові слова для поточної вправи.

Після формування набору слів система створює дві незалежні групи кнопок:

- слова іноземною мовою;
- відповідні переклади.

Для уникнення запам'ятовування порядку відповідей переклади також додатково перемішуються випадковим чином.

У процесі виконання вправи користувач спочатку вибирає слово, після чого вибирає переклад, який, на його думку, відповідає даному слову.

Після вибору двох елементів система автоматично виконує перевірку правильності відповідності.

Алгоритм роботи вправи має такий вигляд:

1. Завантаження словника.
2. Випадковий вибір чотирьох слів.
3. Формування списку перекладів.
4. Перемішування перекладів.

5. Відображення елементів інтерфейсу.
6. Вибір слова користувачем.
7. Вибір перекладу.
8. Перевірка правильності відповідності.
9. Оновлення статистики.
10. Завершення вправи після знаходження всіх пар.

Для перевірки результату використовується функція `checkMatching()`.

Якщо користувач правильно визначив відповідність, система повідомляє про успішне виконання дії та збільшує кількість правильних відповідей.

У разі помилки користувач отримує повідомлення про неправильну відповідність, а статистика помилок оновлюється.

Особливістю реалізації є використання інтерактивного підходу до навчання. Користувач не просто вибирає правильний варіант серед готових відповідей, а самостійно знаходить відповідності між двома наборами даних.

Такий підхід сприяє кращому засвоєнню нових слів та активізує процес пригадування інформації.

Крім того, дана вправа дозволяє одночасно працювати з декількома словами, що позитивно впливає на швидкість формування словникового запасу.

Таким чином, модуль встановлення відповідностей забезпечує ефективний механізм закріплення нової лексики та сприяє розвитку асоціативного мислення під час вивчення іноземних мов.

2.8 Реалізація імпорту словників

Для підвищення зручності використання програмного засобу було реалізовано функцію імпорту словників із зовнішніх файлів. Дана можливість дозволяє користувачам швидко додавати великі обсяги навчального матеріалу без необхідності ручного введення кожного слова окремо.

Імпорт словників реалізовано на сторінці `words.html`. Користувач може вибрати файл зі свого комп'ютера та завантажити його до системи. Після обробки файлу всі слова автоматично додаються до активного словника.

Для забезпечення універсальності програмного засобу підтримуються такі формати файлів:

- JSON;
- CSV;
- TXT.

Під час завантаження словника користувач також має можливість вибрати рівень складності для всіх імпортованих слів. Це дозволяє швидко формувати навчальні словники відповідно до рівня підготовки користувача.

Для зчитування вмісту файлів використовується вбудований об'єкт JavaScript:

```
let reader = new FileReader();
```

Об'єкт `FileReader` дозволяє працювати з локальними файлами без необхідності використання серверної частини.

Після вибору файлу система автоматично визначає його розширення та обирає відповідний алгоритм обробки.

Алгоритм імпорту словника має такий вигляд:

1. Вибір файлу користувачем.

2. Визначення формату файлу.
3. Зчитування вмісту за допомогою FileReader.
4. Обробка даних.
5. Формування об'єктів слів.
6. Додавання слів до словника.
7. Збереження інформації у LocalStorage.
8. Оновлення інтерфейсу користувача.

Для роботи з файлами формату CSV використовується спеціальна функція `parseCSV(content)`.

Дана функція розділяє текст на окремі рядки, після чого кожний рядок обробляється як окремий запис словника.

Структура CSV-файлу може мати такий вигляд:

```
hello, привіт  
world, світ  
computer, комп'ютер
```

Перший елемент рядка розглядається як слово іноземною мовою, а другий — як його переклад.

Для роботи з текстовими файлами використовується функція:

```
parseTXT(content)
```

У текстовому форматі слова можуть зберігатися у вигляді окремих рядків.

Наприклад:

```
hello           -           привіт  
world           -           світ  
computer        -           комп'ютер
```

Після обробки інформації система автоматично створює відповідні об'єкти слів та додає їх до активного словника.

Окрему увагу приділено підтримці формату JSON. Даний формат дозволяє імпортувати повноцінні структури даних із додатковими параметрами, такими як рівень складності або статус слова.

Приклад JSON-запису:

```
{  
  "word": "hello",  
  "translation": "привіт",  
  "level": "A1",  
  "status": "new"  
}
```

Після завершення імпорту всі слова автоматично відображаються у словнику та стають доступними для використання в навчальних вправах.

Використання імпорту значно скорочує час наповнення словникових баз та дозволяє працювати з великими обсягами навчального матеріалу.

Особливо корисною дана функція є для користувачів, які вже мають готові словникові списки та бажають швидко інтегрувати їх у програму.

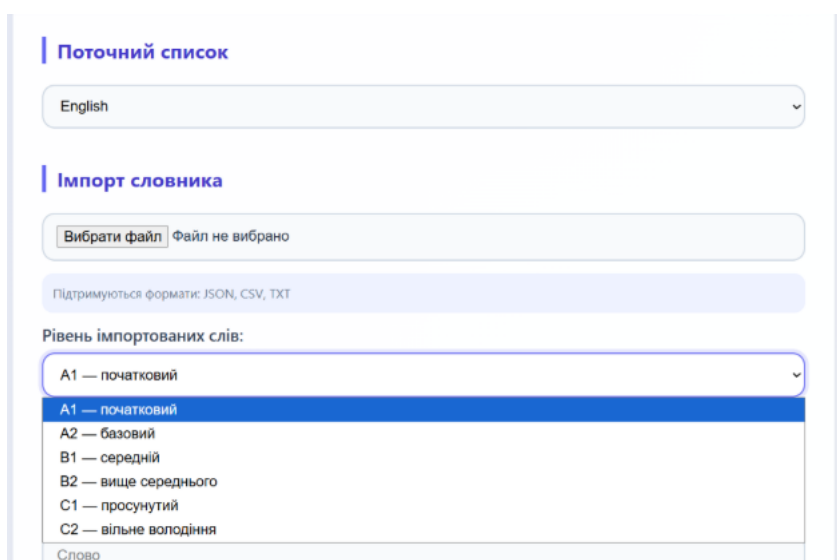


Рисунок 2.5 – Інтерфейс імпорту словників

На рисунку представлено інтерфейс завантаження словників із файлів різних форматів.

Таким чином, реалізований механізм імпорту словників значно розширює можливості програмного засобу та спрощує процес наповнення навчальних баз даних.

2.9 Реалізація озвучування слів

Однією з додаткових функціональних можливостей розробленого програмного засобу є озвучування слів. Дана функція дозволяє користувачеві не лише вивчати написання та переклад слів, але й знайомитися з їх правильною вимовою.

Вивчення правильної вимови є важливою складовою процесу опанування іноземної мови. Саме тому до складу програмного засобу було включено модуль синтезу мовлення, який забезпечує автоматичне відтворення слів за допомогою стандартних можливостей браузера.

Для реалізації озвучування використовується технологія Web Speech API. Даний інтерфейс програмування дозволяє створювати голосові повідомлення без необхідності встановлення додаткового програмного забезпечення або використання сторонніх бібліотек.

Перед початком озвучування система виконує перевірку підтримки Web Speech API браузером користувача.

Фрагмент програмного коду:

```
if (!("speechSynthesis" in window)) {  
    alert("Ваш браузер не підтримує озвучення");  
    return;  
}
```

У випадку відсутності підтримки технології користувач отримує відповідне повідомлення, а виконання функції припиняється.

Основною функцією озвучування є `speakWord(word)`.

Після отримання слова система створює спеціальний об'єкт мовлення:

```
let utterance =  
new SpeechSynthesisUtterance(word);
```

Об'єкт `SpeechSynthesisUtterance` використовується для формування голосового повідомлення та передачі його до системи синтезу мовлення браузера.

Однією з особливостей реалізації є автоматичне визначення мови озвучування залежно від активного словника.

Наприклад, для англійської мови використовується:

```
utterance.lang = "en-US";
```

Для польської:

```
utterance.lang = "pl-PL";
```

Для німецької:

```
utterance.lang = "de-DE";
```

Таким чином забезпечується більш природна та коректна вимова слів різними мовами.

Особливо активно функція озвучування використовується у модулі флеш-карток.

Для цього реалізовано окрему функцію `speakCurrentCard()`.

Після натискання кнопки «Озвучити» система автоматично визначає поточне слово та передає його до функції синтезу мовлення.

Алгоритм роботи озвучування має такий вигляд:

1. Отримання слова для озвучування.
2. Перевірка підтримки Web Speech API.
3. Створення голосового повідомлення.
4. Визначення мови вимови.
5. Передача повідомлення до браузера.
6. Відтворення звуку.

Перевагою використання Web Speech API є відсутність необхідності завантаження аудіофайлів. Уся генерація звуку виконується безпосередньо браузером користувача.

Крім того, даний підхід дозволяє значно зменшити обсяг програмного забезпечення та забезпечує підтримку великої кількості мов без додаткових ресурсів.

Використання озвучування позитивно впливає на ефективність навчання. Користувач не лише бачить написання слова, але й чує його правильну вимову, що сприяє розвитку навичок аудіювання та формуванню правильної фонетичної бази.

У сучасних системах вивчення мов функція озвучування є однією з найбільш затребуваних, тому її реалізація значно підвищує практичну цінність створеного програмного продукту.

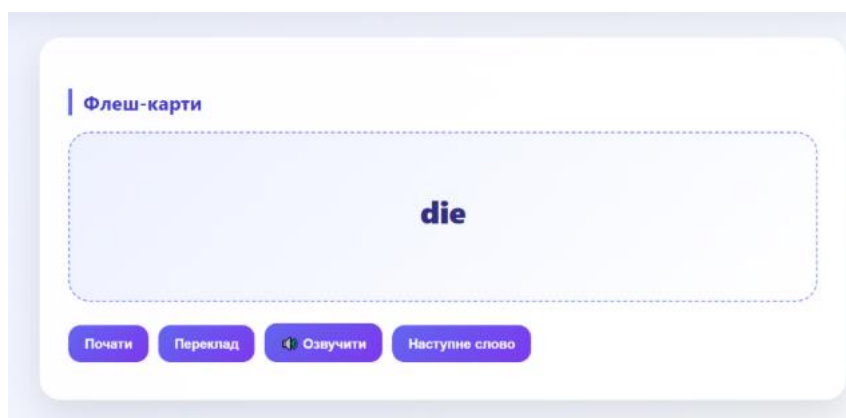


Рисунок 2.6 – Інтерфейс озвучування слів

На рисунку представлено інтерфейс модуля флеш-карток із можливістю озвучування поточного слова.

Таким чином, реалізований модуль озвучування забезпечує підтримку правильної вимови іноземних слів та робить процес навчання більш ефективним і наближеним до реального використання мови.

2.10 Реалізація механізму LocalStorage

Однією з ключових особливостей розробленого програмного засобу є можливість збереження даних без використання серверної бази даних. Для реалізації даного механізму використовується технологія LocalStorage, яка входить до складу Web Storage API та підтримується більшістю сучасних браузерів.

LocalStorage дозволяє зберігати інформацію локально на пристрої користувача у вигляді пар «ключ – значення». Дані не видаляються після закриття браузера та залишаються доступними під час наступного запуску програми.

Використання LocalStorage дозволило відмовитися від створення серверної частини та значно спростити архітектуру програмного засобу. Усі необхідні дані зберігаються безпосередньо у браузері користувача, що забезпечує автономність роботи системи.

У розробленому програмному засобі за допомогою LocalStorage зберігаються:

- словники користувача;
- активний словник;
- статистика правильних відповідей;
- статистика неправильних відповідей;
- рівні складності слів;
- статуси вивчення слів.

Під час запуску програми виконується зчитування даних із `LocalStorage`.

Для отримання інформації використовується наступний фрагмент програмного коду:

```
let allLists = JSON.parse(
  localStorage.getItem("allLists")
);
```

Функція `getItem()` дозволяє отримати дані за вказаним ключем, а метод `JSON.parse()` перетворює текстовий формат JSON у JavaScript-об'єкт.

Якщо словники відсутні, система автоматично створює початкову структуру даних.

```
if (!allLists) {
  allLists = {
    "English": []
  };
}
```

Такий підхід забезпечує коректну роботу програми навіть під час першого запуску.

Після будь-якої зміни словника виконується автоматичне збереження інформації.

Для цього використовується команда:

```
localStorage.setItem(
  "allLists",
  JSON.stringify(allLists)
);
```

Метод `JSON.stringify()` перетворює об'єкт JavaScript у текстовий формат JSON, який може бути збережений у `LocalStorage`.

Алгоритм збереження даних має такий вигляд:

1. Користувач виконує зміну даних.
2. Оновлюється структура словника.
3. Дані перетворюються у JSON.
4. Інформація записується у LocalStorage.
5. Збережені дані стають доступними після перезапуску програми.

Особливо важливим є збереження активного словника. Завдяки цьому після повторного відкриття програми користувач продовжує роботу саме з тим словником, який використовував останнім.

Крім словників, у LocalStorage зберігаються результати навчання. Це дозволяє накопичувати статистику правильних та неправильних відповідей і використовувати її для оцінки прогресу користувача.

Структура даних у LocalStorage може бути представлена таким чином:

```
{
  "English": [
    {
      "word": "hello",
      "translation": "привіт",
      "level": "A1",
      "status": "new"
    }
  ]
}
```

Перевагами використання LocalStorage є:

- відсутність необхідності встановлення сервера;
- висока швидкість доступу до даних;
- простота реалізації;
- автономність роботи застосунку;

– збереження інформації між сеансами роботи.

Разом із перевагами LocalStorage має певні обмеження. Зокрема, обсяг доступної пам'яті обмежений можливостями браузера, а дані зберігаються лише на поточному пристрої користувача.

Незважаючи на це, для навчального програмного засобу такого типу використання LocalStorage є цілком виправданим рішенням, оскільки забезпечує необхідний функціонал без ускладнення архітектури системи.

Рисунок 2.8 — Структура даних, що зберігаються у LocalStorage



Рисунок 2.7 – Структура даних у LocalStorage

На рисунку представлено структуру збереження словників та слів у локальному сховищі браузера.

Таким чином, використання LocalStorage забезпечує надійне локальне збереження інформації, спрощує архітектуру програмного засобу та дозволяє користувачу працювати із системою без необхідності підключення до серверної бази даних.

2.11 Опис інтерфейсу користувача

Одним із важливих етапів розробки програмного засобу є створення зручного та зрозумілого інтерфейсу користувача. Саме інтерфейс забезпечує взаємодію користувача із системою та впливає на комфортність використання програмного продукту.

Під час розробки було прийнято рішення створити єдиний стиль оформлення для всіх сторінок вебзастосунку. Для цього використано

каскадні таблиці стилів CSS, які забезпечують однаковий зовнішній вигляд усіх модулів програми.

Основною концепцією дизайну стало використання сучасного мінімалістичного інтерфейсу з акцентом на простоту використання та швидкий доступ до основних функцій системи.

Головна сторінка програми

Головна сторінка реалізована у файлі `index.html` та виконує роль стартового вікна програмного засобу.

На головній сторінці відображається назва програми та короткий опис її можливостей. Також користувач може ознайомитися з основними функціями системи.

До основних можливостей, представлених на головній сторінці, належать:

- створення словників;
- імпорт словників;
- використання флеш-карток;
- проходження вікторин;
- виконання анаграм;
- встановлення відповідностей між словами та перекладами.

Програма для вивчення іноземних слів

Навчальний застосунок для вивчення іноземних слів за допомогою флеш-карт, вікторин, анаграм та вправ на співвідношення.

Можливості програми

Створення словників різними мовами

Імпорт словників

Флеш-карти

Вікторини

Анаграми

Співвідношення слова і перекладу

Рисунок 2.8 – Головна сторінка програмного засобу

На рисунку представлено головне вікно програми після запуску.

Навігаційне меню

Для забезпечення швидкого переходу між модулями використовується навігаційне меню.

Меню присутнє на всіх сторінках програмного засобу та містить такі розділи:

- Головна;
- Списки мов;
- Словник;
- Флеш-карти;
- Вікторина;
- Анаграма;
- Співвідношення.

Завдяки використанню єдиного меню користувач може швидко переходити між різними режимами роботи програми без повернення на головну сторінку.

Сторінка керування словниками

Основним центром роботи із навчальними матеріалами є сторінка `words.html`.

На даній сторінці реалізовано:

- створення нових словників;
- вибір активного словника;
- імпорт словників;
- додавання нових слів;
- вибір рівня складності;
- онлайн-переклад словників.

Для створення словника використовується текстове поле введення назви та кнопка створення списку.

Користувач може створювати необмежену кількість словників та перемикається між ними за допомогою списку вибору.

Сторінка словника

Для перегляду навчального матеріалу використовується сторінка `dictionary.html`.

Інтерфейс сторінки складається з двох основних розділів:

- нові слова;
- вивчені слова.

Кожне слово відображається разом із перекладом та додатковими характеристиками.

Поділ слів за статусами дозволяє користувачу швидко оцінити власний прогрес та визначити слова, які потребують додаткового повторення.

Інтерфейс флеш-карток

Сторінка `flashcards.html` призначена для роботи з флеш-картками.

Основним елементом інтерфейсу є картка, у якій відображається слово або його переклад.

Користувач може:

- почати навчання;
- переглянути переклад;
- озвучити слово;
- перейти до наступної картки.

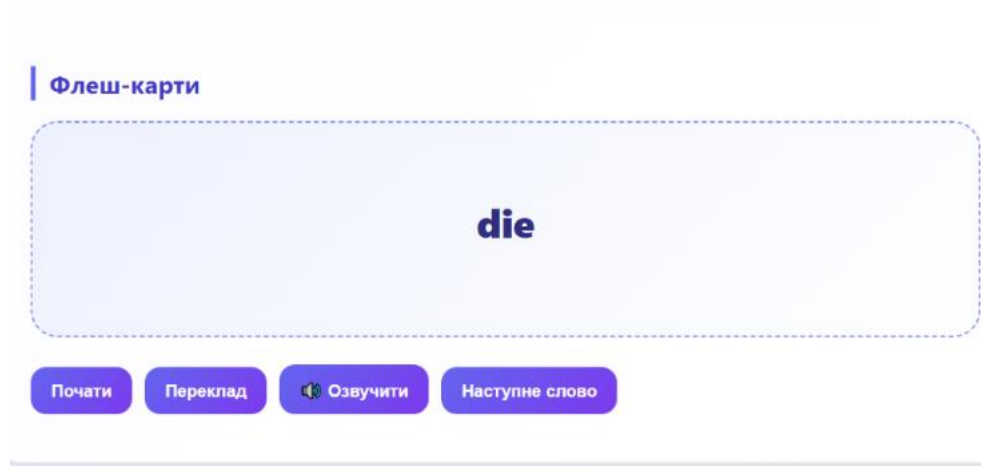


Рисунок 2.9 – Інтерфейс флеш-карток

На рисунку показано процес роботи з флеш-картками.

Інтерфейс вікторини

Сторінка `quiz.html` використовується для перевірки знань користувача.

Інтерфейс містить:

- запитання;
- варіанти відповідей;
- повідомлення про результат;
- кнопку запуску вправи.

Після вибору відповіді система автоматично перевіряє правильність результату та повідомляє користувача про успішність виконання завдання.

Інтерфейс анаграм

Сторінка `anagram.html` реалізує вправу на відновлення правильного написання слова.

Інтерфейс складається з:

- області відображення анаграми;
- поля введення відповіді;
- кнопки перевірки;
- кнопки генерації нової анаграми.

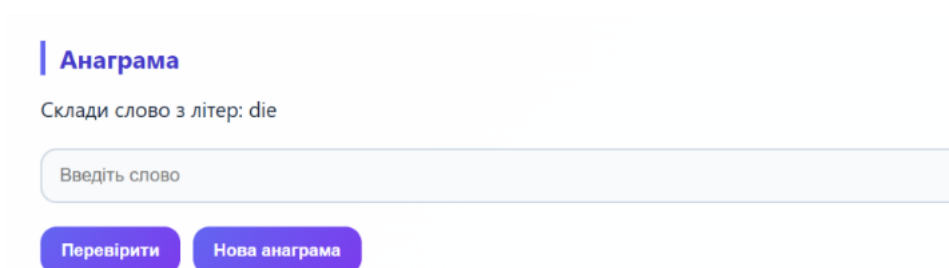


Рисунок 2.10 – Інтерфейс анаграми

На рисунку представлено приклад виконання вправи з анаграмами.

Інтерфейс встановлення відповідностей

Для виконання вправи на встановлення відповідностей використовується сторінка `matching.html`.

Інтерфейс складається з двох окремих блоків:

- слова;
- переклади.

Користувач послідовно вибирає відповідні пари, після чого система виконує перевірку правильності відповіді.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено веборієнтований програмний засіб для вивчення іноземних слів, який забезпечує зручну організацію навчального процесу та сприяє ефективному засвоєнню нової лексики. Під час виконання роботи було проведено аналіз предметної області, досліджено сучасні методи вивчення іноземних мов та виконано огляд існуючих програмних рішень.

У першому розділі роботи було розглянуто сучасні програмні засоби для вивчення іноземних мов, проаналізовано їхні переваги та недоліки. Проведене дослідження показало, що більшість існуючих систем мають значний функціонал, однак часто обмежують можливості користувача щодо створення власних словників та адаптації навчального процесу під індивідуальні потреби. Також було досліджено сучасні методики вивчення іноземної лексики, зокрема використання флеш-карток, тестових завдань, анаграм, вправ на встановлення відповідностей та технологій синтезу мовлення. Крім того, виконано аналіз сучасних вебтехнологій та обґрунтовано вибір HTML, CSS і JavaScript як основних засобів реалізації програмного продукту.

У другому розділі було виконано проектування та розробку програмного засобу. Створено архітектуру системи, реалізовано механізм керування словниками та структуру зберігання слів. Для забезпечення зручності роботи користувача було розроблено інтерфейс із навігаційним меню та окремими сторінками для різних режимів навчання.

У процесі реалізації програмного засобу було створено систему роботи зі словниками, яка дозволяє користувачеві створювати необмежену кількість мовних списків та додавати власні слова й переклади. Для кожного слова передбачено збереження рівня складності відповідно до міжнародної шкали CEFR та статусу вивчення.

У межах роботи реалізовано декілька навчальних модулів. Модуль флеш-карток забезпечує послідовне вивчення нових слів та перевірку знань шляхом самостійного пригадування перекладу. Модуль вікторини дозволяє оцінювати рівень засвоєння матеріалу за допомогою тестових завдань із декількома варіантами відповідей. Модуль анаграм сприяє закріпленню

навичок правильного написання слів, а модуль встановлення відповідностей дозволяє формувати асоціативні зв'язки між словами та їх перекладами.

Для підвищення функціональності програмного засобу реалізовано можливість імпорту словників із файлів форматів JSON, CSV та TXT. Це значно спрощує наповнення словникових баз та дозволяє використовувати вже підготовлені набори слів.

Однією з особливостей створеного програмного засобу є підтримка автоматичного озвучування слів за допомогою Web Speech API. Реалізація даної функції дозволяє користувачам ознайомлюватися з правильною вимовою слів та покращувати навички аудіювання. Також у системі реалізовано автоматичний переклад словників із використанням зовнішнього сервісу перекладу, що значно спрощує створення багатомовних словникових баз.

Для збереження інформації було використано технологію LocalStorage, яка дозволяє працювати із системою без використання серверної бази даних. Усі словники, слова та результати роботи користувача автоматично зберігаються у браузері та залишаються доступними після повторного запуску програми.

Під час виконання роботи було створено UML-діаграми, які дозволили формалізувати структуру системи, описати взаємодію користувача із програмним засобом та візуалізувати логіку роботи окремих функціональних модулів.

Проведене тестування підтвердило коректність роботи всіх основних компонентів програмного засобу. Було перевірено функції створення словників, додавання слів, імпорту даних, роботи флеш-карток, вікторини, анаграм, вправ на встановлення відповідностей, автоматичного перекладу, озвучування слів та механізму локального збереження даних. За результатами тестування критичних помилок виявлено не було, а всі функціональні можливості працювали відповідно до поставлених вимог.

Таким чином, поставлена мета дипломної роботи була повністю досягнута. У результаті виконання роботи створено повноцінний вебзастосунок для вивчення іноземних слів, який поєднує сучасні методики навчання, зручний інтерфейс користувача та сучасні вебтехнології. Розроблений програмний засіб може використовуватися для самостійного вивчення іноземних мов, розширення словникового запасу та закріплення набутих знань. Подальший розвиток проєкту може бути пов'язаний із впровадженням авторизації користувачів, використанням баз даних, реалізацією системи статистики навчання, синхронізацією даних між пристроями та створенням мобільної версії застосунку.

Використані джерела

1. JavaScript Tutorial [Електронний ресурс]. Режим доступу до ресурсу: <https://www.w3schools.com/js/>
2. JavaScript Tutorial [Електронний ресурс]. Режим доступу до ресурсу: <https://www.youtube.com/watch?v=RS4LF577fj4&list=PL7vq4D0vOpQbGerLS1pEgvPXSLUXgvFEo>
3. Ефективні методики запам'ятовування іноземних слів [Електронний ресурс]. Режим доступу до ресурсу: <https://greencountry.com.ua/journal/read/sekreti-pamyati-abo-yak-efektivno-vivchati-i-zapamyatovuvati-inozemni-slova>
4. Документація для Visual Studio Code [Електронне джерело]. Режим доступу: <https://code.visualstudio.com/docs>
5. Побудова UML-діаграм [Електронне джерело]. Режим доступу: <https://dou.ua/forums/topic/40575/>
6. Visual Studio Code (VS Code) [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/uk-ua/power-apps/maker/portals/vs-code-extension>