

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»


Факультет електроніки

Кафедра акустичних та мультимедійних електронних систем

УДК 621.3

«До захисту допущено»

Завідувач кафедри


(підпис) Найда С.А.
(ініціали, прізвище)

“ 16 ” грудня 2021 р.

Магістерська дисертація

зі спеціальності 171 Електроніка

на тему: Система озвучення контенту з використанням семантичної розмітки сайтів на базі CMS WordPress з підтримкою користувачів голосовим чатом

Виконав: студент 2 курсу, групи ДГ-01мп

Осадчук Олександр Русланович

Керівник доц. к.т.н. доц. Богданов О. В.

Рецензент доцент каф. ЕІ к.т.н. доц. Катерина Олегівна Іванько



Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2021 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Факультет електроніки

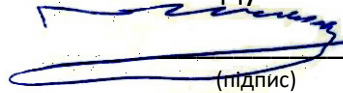
Кафедра акустичних та мультимедійних електронних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
«Електроніка»

Спеціальність 171 «Електроніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 С. А. Найда
(підпис) (ініціали, прізвище)

«_1_» вересня 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Осадчуку Олександрю Руслановичу

1. Тема дисертації: "Система озвучення контенту з використанням семантичної розмітки сайтів на базі CMS WordPress з підтримкою користувачів голосовим чатом"

науковий керівник дисертації Богданов Олексій Вікторович, к.т.н. доц.
затвержені наказом по університету від «03» листопада 2021 р. № 3666-с

2. Строк подання студентом дисертації 06 грудня 2021 р.

3. Об'єкт дослідження: Наукові джерела з питань моделювання, програмного забезпечення, наявні програмні рішення аналізу обвідної звуку, що генерується голосовим трактом

4. Предмет дослідження: Виконати аналіз літературних джерел та визначити сучасні методи діагностики стану голосового тракту, проаналізувати можливості сучасного програмного забезпечення для діагностики патологій голосу, розробити програмно-апаратний макет системи аналізу часових параметрів мовного сигналу, порівняти результати роботи створеної системи із такими для систем-прототипів.

5. Перелік завдань, які потрібно розробити: 1) аналітичний огляд літературних джерел до розділу 1 роботи; 2) опис інструментарію та його можливостей для створення програмного забезпечення

6. Перелік графічного (ілюстративного) матеріалу: Узагальнені блок-схеми існуючих діагностичних систем. Узагальнена блок-схема діагностичної системи, що розробляється.

7. Орієнтовний перелік публікацій: Алгоритм розпізнавання природного мовного сигналу, Електроніка 2021, готується до публікації.

8. Консультанти розділів дисертації^{1*}

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 03.09.2021

^{1*} Консультантом не може бути зазначено наукового керівника

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Огляд літературних джерел	30.09.2021	виконано
2	Опис інструментарію та його можливостей для створення програмного забезпечення	26.10.2021	виконано
3	Опанування технічного інструментарію та програмного забезпечення	02.11.2021	виконано
4	Розробка програмних методів для використання Google [®] Cloud Platform на веб сайті	08.11.2021	виконано
5	Розробка практичної частини	22.11.2021	виконано
6	Оформлення результатів робіт	21.11.2021	виконано
7	Оформлення магістерської дисертації	03.12.2021	виконано

Студент

(підпис)

О. Р. Осадчук

Науковий керівник дисертації



О. В.

Богданов

(підпис)

АНОТАЦІЯ

Осадчук О. Р. Система озвучення контенту з використанням семантичної розмітки сайтів на базі CMS WordPress з підтримкою користувачів голосовим чатом. : магістерська дис. : 171 Електроніка. / Осадчук Олександр Русланович. — Київ, КПІ ім. Ігоря Сікорського», 2021. — 60 с.

Ключові слова: семантика, веб-сервіси, мовні кодеки, автоматичне розпізнавання мовлення, аналіз мовлення, мовлення.

Сучасні інформаційні технології дозволяють людині з вадами зору отримувати інформацію нарівні зі здоровими завдяки ряду технічних рішень, однак вибір методів відтворення такої інформації повинен повністю забезпечуватись самими людьми з обмеженими можливостями, це являє собою значну проблему через значні витрати часу на споживання інформації.

Для спрощення сприйняття інформації слабозорими при користуванні веб сайтами розроблено міжнародний стандарт для вебмайстрів – Web Content Accessibility Guidelines. Стандарт детально описує вимоги людей з вадами зору які рекомендується задовольняти. Для реалізації таких рекомендацій вебмайстрам необхідно вивчати нові принципи та алгоритми програмування. Часто потребує додаткового підвищення кваліфікації, що несе за собою недотримання вебмайстрами таких вимог.

Метою магістерської дисертації є розробка простої, для вебмайстрів, в інсталяції та використанні системи споживання контенту на веб сторінках для слабозорих.

Система була розроблена на базі глибинних нейронних мереж та має можливість інтегруватися в найпопулярнішу в світі систему управління контентом веб сайтів WordPress© і інтеграція голосового чату на сайт.

ABSTRACT

Osadchuk O. R. Sound system of content using semantic markup of sites based on CMS WordPress with support for users of voice chat. : Master's thesis : 171 Electronics. / Osadchuk Oleksandr Ruslanovych. — Kyiv, Igor Sikorsky KPI, 2021. — 60 p.

Keywords: semantic web, web services, speech codecs, automatic speech recognition, speech analysis, speech.

Modern information technologies allow visually impaired people to receive information along with healthy ones due to a number of technical solutions, but the choice of methods of reproducing such information should be fully provided by people with disabilities, this is a significant problem due to significant time consumption.

To simplify the perception of information by the visually impaired when using websites, an international standard for webmasters - Web Content Accessibility Guidelines has been developed. The standard describes in detail the requirements of visually impaired people that are recommended to be met. To implement such recommendations, webmasters need to learn new principles and programming algorithms. It often requires additional training, which entails non-compliance by webmasters with such requirements.

The aim of the master's dissertation is to develop a simple, for webmasters, to install and use the system of content consumption on web pages for the visually impaired.

The system was developed on the basis of deep neural networks and has the ability to integrate into the world's most popular content management system for WordPress[®] websites and website voice chat integration.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
1. ВСТУП	10
2. МЕТОДИ ТА МАТЕРІАЛИ	12
2.1. Вимоги до сайтів для людей з вадами зору.	12
2.2. Огляд допоміжних засобів для перегляду сайтів людям з вадами зору.	18
2.3. Огляд CMS WordPress®.	20
2.4. Обробка природної мови	22
2.5. Машинний переклад	23
2.6. Консоль Dialogflow	24
2.7. Глибоке навчання в NLP	25
2.8. Нейромережевий машинний переклад	26
2.9. Розробка голосового чат боту та тренування нейронної мережі.	30
3. РЕЗУЛЬТАТИ	33
3.1. Розробка плагіну для CMS WordPress®.	33
3.2. Конфігурація передачі та обробки мовного сигналу	34
3.3. Конфігурація декодування.	36
3.4. Запит на обробку.	37
3.5. Взаємодія користувачів з інтеграціями	37
3.6. Взаємодія користувачів з API	38
3.7. Обробка запитів	39
4. ОБГОВОРЕННЯ І ВИСНОВКИ	43
4.1. Маршрутизація запитів	45
4.2. Обробка запитів	49
4.3. Навчальні фрази та контексти	50
3.4. Налаштування WordPress® REST API	53
4.4. Налаштування безпеки	53
5. РОЗРОБКА СТАРТАП-ПРОЕКТУ	54
5.1. Опис ідеї проекту.	54
5.2. Визначення характеристик проекту.	54
5.3. Технологічний аудит ідеї проекту.	55

	8
5.4. Аналіз ринкових можливостей запуску стартап-проекту.	55
5.5. Розроблення маркетингової програми стартап-проекту.	56
Висновки	57
ПЕРЕЛІК ПОСИЛАНЬ	58
Додаток А. Код файлу voicer.php	A1
Додаток Б. Код файлу afminscrips.php	B1
Додаток В. Код файлу afminscrips.php	B1
Додаток Г. Код файлу XMLHelper.php	G1
Додаток Д. Код файлу admin.saass	D1

УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВООЗ — Всесвітня організація охорони здоров'я.

NVDA — Non Visual Desktop Access, англ. аббревіатура.

ОС — Операційна система.

NLP — Natural language processing (Обробка природної мови).

RNN — Рекурентна нейронна мережа (Recurrent Neural Network).

DNM — Мережа Динамічної Пам'яті (Dynamic Memory Network).

LSTM — Короткострокова / довгострокова пам'ять (Long / short term memory).

1. ВСТУП

На даний момент люди з обмеженими можливостями зору значно обмежені в користуванні проєктами мережі Інтернет тому що більшість веб-ресурсів не пристосовані під їхні потреби. Сучасні інформаційні технології дозволяють таким людям отримувати інформацію нарівні зі здоровими завдяки рішенням розробленим вченими і інженерами різних країн [1].

Текстову інформацію можна доставити людям зі слабким зором за допомогою екранної лупи або за рахунок збільшення шрифту програмними засобами, сліпим – шляхом озвучування тексту за допомогою комп'ютерних програм або виведення текстів на брайлівський екран монітора. Це ефективне рішення однак вибір методів відтворення такої інформації повинні повністю забезпечуватись самими людьми з обмеженими можливостями, це являє собою значну проблему через значні витрати часу на споживання інформації.

Для спрощення сприйняття інформації слабозорими при користуванні веб сайтами розроблено міжнародний стандарт для вебмайстрів – керівництво по доступності веб-контенту яке називається консорціум W3 Web Content Accessibility Guidelines (WCAG) 2.0 [2]. Стандарт детально описує вимоги людей з вадами зору які рекомендується задовольняти для безпроблемного перегляду ними веб сайту. Основні положення WCAG 2.0 описуть параметри і алгоритми масштабування, кластерезування та відділення інформації програмами для слабозорих і надають рекомендації по написанню коду веб сайту [3].

Однак для реалізації таких рекомендацій вебмайстрам необхідно вивчати нові принципи та алгоритми програмування і використовувати додаткові інструменти розробки. Часто це складно, потребує додаткового підвищення кваліфікації, що несе за собою недотримання вебмастерами таких вимог.

Метою магістерської дисертації є розробка простої, для вебмайстрів, в інсталяції, налаштуванні та використанні системи споживання контенту на веб сторінках для слабозорих, на момент написання дисертації не знайдено аналогів такої системи.

Система була розроблена на базі глибинних нейронних мереж та має можливість інтегруватися в найпопулярнішу в світі систему управління контентом веб сайтів WordPress[®] і однієї системи Google[®] Cloud Platform[®].

2. МЕТОДИ ТА МАТЕРІАЛИ

2.1. Вимоги до сайтів для людей з вадами зору.

Міжнародним стандартом є документ консорціуму W3 Web Content Accessibility Guidelines (WCAG) 2.0 (Керівництво по доступності веб-контенту) [4].

Основні положення документа полягають в тому, що користувач повинен мати можливість:

- Сприймати компоненти користувальницького інтерфейсу і інформацію таким чином, щоб він був здатний сприймати їх;
- Оперувати всіма компонентами користувальницького інтерфейсу і навігації;
- Ясно розуміти структуру користувальницького інтерфейсу і представлену інформацію;
- Використовувати різні веб-браузери, включаючи спеціальні браузери для людей з обмеженими можливостями.

Тепер докладно про кожен з цих вимог до форми представлення тексту:

Сприйняття

- Контраст - не менше 4.5: 1;
- Розмір шрифту може бути збільшений в 2 і більше разів, при збереженні функціональності і без появи горизонтальної прокрутки;
- Можливість вибору кольору тексту і фону;
- Ширина текстового блоку не повинна перевищувати 80 знаків;
- Вирівнювання тексту по ширині блоку або вікна не допускається;
- Міжрядковий інтервал полуторний або вище, відстань між параграфами в півтора рази більше, ніж інтервал між рядками.

Операбельність.

- Всі функції повинні бути доступні за допомогою клавіатури;
- Навігація, пошук і визначення поточного положення користувача на сторінках сайту повинні бути простими і зручними;

- Можливість пропустити повторюється блок з переходом до головного контенту сторінки;
- При управлінні за допомогою клавіатури, поточний фокус повинен бути видимим.

Ясність.

- Вміст сайту має бути зрозумілим і читабельним;
- Визначення рідкісних слів, скорочень, незвичайних словосполучень, аббревіатур і т.п. повинні бути приведені в тексті або на полях;
- Веб-сторінки повинні «вести» себе передбачувано;
- Допомога користувачеві при введенні даних: перевірка на помилки, підказки, контекстна допомога, інструкції щодо заповнення полів.

Надійність.

- Сайт повинен бути сумісним з різними платформами і браузерами, в тому числі з перспективними, а також підтримувати спеціальні технології для людей з обмеженими можливостями;
- Як користувачі з ослабленим зором користуються інтернетом.

Повністю сліпі користувачі використовують спеціальні програми (скрін-рідери), які синтезують мова, ці програми аналізують html-розмітку і відсівають то, що потрібно прочитати, а що ні. Інший спосіб читання пов'язаний з використанням дисплея Брайля, який складається з програми-рідера і пристрої, на корпусі якого розташовані рельєфно-точкові осередки [5].

У користувачів з ослабленим зором є кілька варіантів:

- веб-браузер + спеціальна програма, яка збільшує частину екрану;
- програмні засоби масштабування вікна браузера;
- додаткові можливості самого веб-сайту, які передбачені розробниками при створенні або модернізації сайту;
- текстовий браузер - це браузер, який відображає тільки текстовий зміст веб-сайту.

Дальтоніки можуть використовувати звичайні браузері з колірними налаштуваннями, які дозволяють змінити стандартні кольори на видимі конкретним користувачем. Існує багато додаткових способів для спрощення сприйняття інформації користувачами з обмеженими можливостями по зору за рахунок відділення важливих частин від другорядних [5].

Кольори

Кольорове кодування дозволяє поліпшити сприйняття інформації будь-якими користувачами. Для прикладу: кнопка «ОК» може бути зеленою, а кнопка «СКАСУВАТИ» - червоною. Користувачеві можна запропонувати вибирати кольору для різних завдань: дуже зручно, коли на екрані з'являються кольорові квадратики з текстом і зображенням, які можна натиснути. Це набагато краще, ніж меню, що випадає.

Однак колір - це не єдиний спосіб поліпшити сприйняття інформації. На червоній кнопці (квадратику) повинен бути відповідний текст - «Видалити». Ще один варіант: виділення поля червоним бордюром, якщо воно неправильно заповнене. Колірного кодування в цьому випадку недостатньо: необхідно вказати неправильні поля і повідомити користувачеві, в чому конкретно помилка (не вистачає цифр в телефонному номері, адреса електронної пошти не відповідає формату) [6].

Аудіо-контент.

Необхідно виключити появу нав'язливих банерів та інших елементів, які не тільки показують, але і відтворюють звуки. Якщо ж без них обійтися ніяк не можна, то необхідно поряд розмістити кнопку відключення звуку.

Від фонового звуку бажано зовсім відмовитися або зробити його тихим.

Текстовий контент

- Основний текст повинен бути контрастним;
- Повинна бути можливість збільшити текст на 200%, при цьому сторінка повинна залишитися читабельною;

- Без необхідності не варто виводити текст у вигляді картинки;
- При верстці необхідно дотримуватися правила типографіки для веб-сторінок: ширина рядка повинна бути не більше 80 символів, текст не вирівнюється по краях, міжрядковий інтервал не повинен бути більшим і істотно менше інтервалу між абзацами.

У користувача повинно бути достатньо часу для ознайомлення з контентом.

Йдеться про змінюваних картинках зі спец-пропозиціями, онлайн-тестами, новинками, онлайн-іграми, які обмежують користувача в часі, додатки для перегляду презентацій і та інших мультимедійних матеріалів. Ці обмеження можуть ускладнити роботу на комп'ютері не тільки зі слабким зором, але дітям і літнім людям, для яких веб-контент не є природним по формі. Тимчасових обмежень по можливості краще уникати, а якщо це не виходить, то повинна бути можливість вручну продовжити термін появи контенту або поставити паузу [6].

Існують онлайн-сервіси, на яких повністю ці вимоги врахувати неможливо: аукціони, бронювання та інше. Однак в цих випадках можна збільшити час на прийняття користувачем рішення, блокувати ресурси на час для підтвердження користувачем свого вибору.

Якщо обмеження часу — це вимога безпеки (онлайн-банк, інтернет-магазин), то необхідно попередити користувача про закінчення сесії, і після повторної авторизації важливо повернути користувача на те саме місце, де він був до припинення сесії.

Не слід використовувати спалахи і миготіння сторінок і блоків, а також різкі звуки. Навіть здорові люди реагують на такі речі негативно і непередбачувано, не кажучи вже про хворих.

Сторінки сайту повинні бути прості в навігації і користувач повинен мати можливість зорієнтуватися — де він знаходиться.

Ці вимоги природні і для звичайних сайтів:

- Всі сторінки сайту повинні мати заголовок, відповідний змісту;
- Текст посилання повинен відповідати сторінці, на яку посилається;
- У користувача повинно бути кілька варіантів переходу на інші сторінки (звичайна навігація, рядок пошуку, карта сайту);
- Активне поле форми повинно бути виділено.

Користувачі вже звикли, що на сучасних сайтах навіть солідних електронних видань основного контенту на сторінці передує другорядна інформація (елементи навігації, реклама, шапка і т.п.). Однак вимогливий користувач точно скаже, що ховати контент під спливаючими вікнами на третю прокрутку екрану — це поганий тон. Якщо без нав'язливої реклами ніяк не обійтись, то на сторінці на видному місці повинен бути елемент, який дозволяє користувачеві перейти до вивчення основного вмісту, замість того щоб спішно закривати спливаючі банери і відключати звук [7].

Текст повинен бути читабельним і зрозумілим [8].

- Мову сторінки необхідно визначити в HTML-кодi. Якщо на сторінці є текст на іншій мові, то необхідно використовувати атрибут `xml: lang`, який визначає мову.
- Якщо в тексті є рідкісні слова, специфічні значення слів, аббревіатури, рекомендується там же пояснити їх зміст.
- Спеціалізований контент (з формулами, науковими і галузевими термінами), який орієнтований не тільки на професіоналів, а й для ширшої аудиторії, рекомендується доповнити альтернативним спрощеним за змістом варіантом.

Веб-сторінки повинні бути передбачуваними

- При заповненні форми фокус не повинен самостійно переміщатися на неправильно заповнений поле
- Поле, яке потребує пояснення, повинно мати підказку при фокусуванні на поле або спеціальному елементі поруч з полем
- Пояснення до поля введення повинно з'являтися всередині нього, зникати при фокусуванні
- Навігація на різних сторінках повинна бути стандартною на всьому сайті та вести себе однаково і передбачувано
- Перехід по одній з посилань списку повинен проводитися після вибору посилання і натискання стандартних для це випадку клавіш: пробілу або Enter.

Зміна зовнішнього вигляду сторінки (перехід на іншу сторінку, поява нового вікна, динамічна зміна контенту за кількістю) має бути передбачуваним для читача; дію користувача, яке викликало це зміна (переклад фокуса, відправка форми, прокрутка, наведення мишки на елемент), має бути зрозуміло користувачеві і асоціюватися з наслідками.

Необхідно стежити за сумісністю контенту з сучасними і перспективними для користувача додатками, в тому числі з додатками використовують допоміжні технології.

Слід уникати не стандартні засоби для розробки призначених для користувача інтерфейсів: невидимий шар, який з'являється при активації елемента; піктограми із зображенням включених / виключених гуртків замість радібаттона, зображення з onclick-му замість кнопки сабмита. Можливості HTML / CSS достатні для створення найрізноманітніших візуальних ефектів.

2.2 Огляд допоміжних засобів для перегляду сайтів людям з вадами зору.

Програми екранного доступу) — це категорія програм або програмних комплексів, що забезпечують доступ (як правило, тактильний або мовної) незрячих і слабозорих користувачів до інформації, яка відображається на екрані комп'ютера, а також доступ до спеціалізованої допоміжної інформації, необхідної таким користувачам для ефективної роботи [9].

Програма екранного доступу NVDA[®]

Програма NVDA[®] є безкоштовною, створена для забезпечення візуального доступу до ПК. Працює софт завдяки мовному оповіщення. Це, по суті, потужний інструмент для користувачів, які мають проблеми зору. Людина може отримати будь-яку інформацію, замовити квитки, різні речі з інтернет-магазину [9].

В кінцевому підсумку користувач ніколи не самотній, так як може поговорити з іншими людьми. Програма переводить текст в шрифт Брайля. NVDA[®] функціонує на Windows[®]. Якщо є дисплей Брайля, тоді можна перекладений текст завантажити на флешку [10].

В даний час NVDA[®] працює з Miranda Instant Messenger і забезпечує підтримку Mozilla Firefox (з яким працює краще, ніж з Internet Explorer). NVDA[®] також непогано взаємодіє з останньою версією Skype (необхідно в головному вікні Skype активувати пункт меню accessible, натиснувши alt, потім v, потім a, потім s) [11].

NVDA[®] побудований за модульним принципом, що дозволяє легко модифікувати існуючий вихідний код і додавати новий, щоб забезпечити доступ до нових програм і елементів управління Windows. Модулі додатки (App Modules) можуть бути додані, щоб забезпечити повну підтримку конкретного додатка, віртуальні буфери (virtual buffers) можуть бути додані, щоб дозволити NVDA[®] відображати складні документи та інші дані, і об'єкти NVDA[®] можуть бути додані, щоб забезпечити підтримку певних елементів управління або вікон [12].

NVDA[®] завжди намагається зробити елементи програми або операційної системи доступними, так щоб користувачі могли відшукати будь-яку потрібну інформацію.

NVDA[®] вільний від ринкової кон'юнктури і маркетингових трюків. Це означає, що розробники NVDA[®], додаючи нові можливості, виходять з практичних потреб незрячих користувачів, а не на основі ринкової кон'юнктури. NVDA[®] не може бути настільки стійкою, як інші програми екранного доступу, однак вона є не поганим засобом для дослідження і тестування різних нових можливостей, які відсутні в інших програмах екранного доступу для операційної системи Windows або які користувач не може самостійно додати. Наприклад, NVDA[®] використовує звуковий сигнал, щоб повідомити користувачеві про рух смуги індикатора виконання (progress bar). Чим вище звук, тим ближче підійшла смужка до 100 відсотків [13].

В даний момент інтерфейс NVDA[®], крім англійської, перекладений на бразильський діалект португальської та фінський мови. Оскільки не існує будь-яких специфічних проблем з перекладом NVDA[®] на інші мови, то передбачається, що список підтримуваних мов буде рости.

Программа екранного доступа Jaws

Являє собою платне додаток для ОС Windows для сліпих, що працює за допомогою перекладу на шрифт Брайля. До появи NVDA[®], Jaws була найбільш затребуваною програмою. Сьогодні немає сенсу переплачувати, коли у безкоштовного конкурента є все необхідне для забезпечення комфортного проведення часу за комп'ютером [14].

У JAWS непоганий функціонал, але як відзначають багато користувачів занадто багато «наворотів», які безглузді.

Брайлівський дисплей

Брайлівський дисплей або дисплей Брайля — спеціальний електронний пристрій, яке переводить текстову інформацію, представлену на екрані ПК, планшета або смартфона, в рельєфно-крапковий шрифт за системою Брайля.

Такий дисплей — це своєрідна клавіатура, кнопки якої відзначені звичайними літерами, а шеститочкові символами абетки Брайля [15].

Принтери Брайля

Брайлівські принтери дозволяють конвертувати звичайний текст в рельєфно-крапковий шрифт по системі Брайля, а потім роздруковувати його на спеціальному папері.

2.3. Огляд CMS WordPress®.

WordPress® — це система керування вмістом сайту (CMS) з відкритим вихідним кодом.

Найпопулярнішим на даний момент способом створення сайту є CMS. CMS (Content Management System) — це система створення та управління сайтом. Простими словами, це програма із зручним інтерфейсом для створення веб-ресурсу. У ній можна використовувати код, але це необов'язково. Завдяки такій технології усі бажаючі можуть створити собі сторінку [16].

На ринку програм зі створення інтернет-ресурсів за першість борються Joomla, Drupal, 1С-Бітрікс, проте найпопулярніша CMS WordPress®. За даними W3techs [4], WP використовують 64,7% всіх веб-сайтів, зроблених на CMS. Це 41,1% усіх існуючих у світі сайтів. Цю платформу для своїх блогів використовують The New York Times та Forbes. Таку популярність Вордпрес отримав за зручність інтерфейсу та великі можливості [17].

Переваги:

Безкоштовна платформа. CMS WordPress® повністю безкоштовна. Оплачується лише за домен та хостинг (або VPS), де зберігатимуться файли сайту.

Відкритий вихідний код. Вам не потрібно платити за програмне забезпечення WordPress[®].

Просте встановлення. Налаштування відбувається в пару кліків. Якщо вийдуть нові оновлення, програма сама повідомить про них та запропонує оновитись.

Легкість в управлінні. Вам не потрібні знання програмування для таких щоденних завдань, як написання та редагування публікацій, завантаження та редагування зображень, встановлення плагінів [18].

Зрозумілий інтерфейс. Достатньо подивитися на назву кнопки і відразу стане зрозумілим, що можна зробити з її допомогою.

Готові теми для різних веб-сайтів. У середині CMS є багато безкоштовних шаблонів для різних типів ресурсів (корпоративних, блогових, новиних). Не має значення, чи робите ви портфоліо для фотографа або сайт медичної компанії. Шаблони знайдуться для будь-якого бізнесу. Також сторонні розробники продають самостійно розроблені теми. Якщо ви купуєте послугу REG.Site, ви отримуєте доступ до десятків безкоштовних ексклюзивних тем, база яких постійно поповнюється [11].

Підтримка багатьох хостинг-провайдерів. Хостинг — важлива частина сайту. На ньому зберігаються всі файли ресурсу. WP «дружить» з усіма великими хостинг-провайдерами, тому ви можете вибрати будь-яку компанію, що сподобалася.

Велика спільнота. Користувачів WordPress[®] дуже багато, і вони постійно обмінюються знахідками на офіційному форумі. Якщо ви чогось не знаєте, або у вас є якісь проблеми, ви швидко зможете знайти рішення [19].

Недоліки:

Багато сторонніх програм. Безліч плагінів і тим самим WordPress[®] створені сторонніми розробниками, і в них можуть бути помилки. Перед тим як

встановлювати нове доповнення, почитайте відгуки, запитайте у спільноти думку щодо плагіну або теми [18].

Завантаження сторінки може тривати багато часу. Багато плагінів впливає на завантаження сторінки. Проте встановлення плагіна кешування зазвичай вирішує цю проблему [20].

Несумісність оновлень та плагінів. Великий вибір плагінів — це не лише плюс WordPress[®], а й справжній біль. Деякі плагіни та їх оновлення можуть викликати різні баги (помилки). Помилки можуть виникнути і через оновлення самого WordPress[®]. Може загубитися сумісність із встановленими доповненнями та темами. В цьому випадку доведеться відкочуватися на попередню версію сайту.

Бувають проблеми із безпекою. Оскільки програма розповсюджується з відкритим вихідним кодом, зловмисникам простіше знайти вразливість. Також підозрілі плагіни можуть спричиняти віруси. Розробники WordPress[®] борються із цією проблемою. Вони регулярно випускають патчі, щоб посилити захист системи. Також є багато ефективних плагінів, які захищають сайт.

Відсутня техпідтримка. Відсутність технічної підтримки може спричинити деякі труднощі. Вивчати інтерфейс та розбиратися із проблемами потрібно буде самостійно. Однак WordPress[®] має дуже активну спільноту, тому знайти вирішення проблеми можна там.

Не синхронізується з 1С[®]. У середині програми немає інструменту для роботи з 1С[®]. Якщо все-таки це потрібно зробити, доведеться запрошувати програмістів для роботи з кодом. Однак це складне завдання навіть для спеціалістів [21].

2.4. Обробка природної мови

Обробка природної мови (далі NLP - Natural language processing) — область, що знаходиться на перетині computer science, штучного інтелекту та лінгвістики. Мета

полягає в обробці і "розумінні" природної мови для перекладу тексту і відповіді на питання.

З розвитком голосових інтерфейсів і чат-ботів, NLP стала однією з найважливіших технологій штучного інтелекту. Але повне розуміння і відтворення сенсу мови — надзвичайно складне завдання, так як людську мову має особливості:

- Людська мова — спеціально сконструйована система передачі сенсу сказаного або написаного. Це не просто екзогенний сигнал, а усвідомлена передача інформації. Крім того, мова кодується так, що навіть маленькі діти можуть швидко вивчити його.
- Людська мова — дискретна, символна або категоріальна сигнальна система, що володіє надійністю.
- Категоріальні символи мови кодуються як сигнали для спілкування по декількох каналах: звук, жести, лист, зображення і так далі. При цьому мова здатна виражатися будь-яким способом.

2.5. Машинний переклад

Машинний переклад (Machine translation) — перетворення тексту на одному природною мовою в еквівалентний за змістом текст на іншій мові. Робить це програма або машина без участі людини. У машинному перекладі використовуються статистика використання слів по сусідству. Системи машинного перекладу знаходять широке комерційне застосування, так як переклади з мов світу — індустрія з об'ємом \$40 мільярдів на рік . Деякі відомі приклади:

- Google[©] Translate перекладає 100 мільярдів слів в день.
- Facebook[©] використовує машинний переклад для автоматичного перекладу текстів в постах і коментарях, щоб зруйнувати мовні бар'єри і дозволити людям з різних частин світу спілкуватися один з одним.

- eBay[©] використовує технології машинного перекладу, щоб зробити можливим транскордонну торгівлю і з'єднати покупців і продавців з різних країн.
- Microsoft[©] застосовують переклад на основі штучного інтелекту до кінцевих користувачів і розробникам на Android, iOS і Amazon Fire незалежно від доступу в Інтернет.
- Systran[©] став першим постачальником софта для запуску механізму нейронного машинного перекладу на 30 мов в 2016 році.

У традиційних системах машинного перекладу доводиться використовувати паралельний корпус — набір текстів, кожен з яких перекладено на один або кілька інших мов. Наприклад, маючи вихідних мову f (Французький) і цільової e (Англійська), потрібно побудувати статистичну модель, що включає вірогідну формулювання для правила Байеса, модель перекладу $p(f | e)$, навчену на паралельному корпусі, і модель мови $p(e)$, навчену тільки на корпусі з англійською мовою. Зайве говорити, що цей підхід пропускає сотні важливих деталей, вимагає великої кількості спроектованих вручну ознак, складається з різних і незалежних завдань машинного навчання [16].

2.6. Консоль Dialogflow

Dialogflow надає веб-інтерфейс користувача, який називається Dialogflow Console. Можна використати цю консоль для створення, запуску та тестування агентів.

Консоль Dialogflow відрізняється від консолі Google[©] Cloud Platform. Консоль Dialogflow використовується для управління агентами Dialogflow, тоді як консоль GCP використовується для керування налаштуваннями діалогового потоку, характерними для GCP та іншими ресурсами GCP [22].

У більшості випадків слід використовувати консоль Dialogflow для створення агентів, але ви також можете використовувати API Dialogflow для створення агентів для розширених сценаріїв.

2.7. Глибоке навчання в NLP

Істотна частина технологій NLP працює завдяки глибокому навчанню (deep learning) — області машинного навчання, яка почала набирати обертів тільки на початку цього десятиліття з наступних причин [23]:

- Накопичені великі обсяги тренувальних даних;
- Розроблено обчислювальні потужності: багатоядерні CPU і GPU;
- Створені нові моделі і алгоритми з розширеними можливостями і поліпшеною продуктивністю, с гнучким навчанням на проміжних уявленнях;
- З'явилися навчальні методи с використанням контексту, нові методи регуляризації та оптимізації [23].

Більшість методів машинного навчання добре працюють через розроблених людиною уявлень (representations) даних і вхідних ознак, а також оптимізації ваг, щоб зробити фінальне пророцтво краще [23].

У глибокому навчанні алгоритм намагається автоматично витягти кращі ознаки або подання з сирих вхідних даних. Створені вручну ознаки часто занадто спеціалізовані, неповні і вимагають час на створення та затвердження. На противагу цьому, виявлені глибоким навчанням ознаки легко пристосовуються [23].

Глибоке навчання пропонує гнучкий, універсальний і той, якого навчають фреймворк для подання світу як у вигляді візуальної, так і лінгвістичної інформації. Спочатку це призвело до проривів в областях розпізнавання мови і комп'ютерному зорі. Ці моделі часто навчаються за допомогою одного поширеного алгоритму і не вимагають традиційного побудови ознак під конкретну задачу [24].

2.8. Нейромережевий машинний переклад

Підхід до моделювання перекладу за допомогою рекурентної нейронної мережі (Recurrent Neural Network, RNN). RNN — нейронна мережа з залежністю від попередніх станів, в якій має зв'язки між проходами. Нейрони отримують інформацію з попередніх шарів, а також з самих себе на попередньому етапі. Це означає, що порядок, в якому подається на вхід дані і тренується мережа, важливий: результат подачі "Дональд" - "Трамп" не збігається з результатом подачі "Трамп" - "Дональд".

Стандартна модель нейро-машинного перекладу є наскрізною нейромережею, де вихідне пропозицію кодується RNN, званої кодировщик (encoder), а цільове слово передбачається за допомогою іншої RNN, званої декодер (decoder). Кодувальник «читає» вихідне пропозицію зі швидкістю один символ в одиницю часу, далі об'єднує вихідне пропозицію в останньому прихованому шарі. Декодер використовує зворотне поширення помилки для вивчення цього об'єднання і повертає перекладену варіант. Дивно, що знаходився на периферії дослідницької активності в 2014 році нейро-машинний переклад став стандартом машинного перекладу в 2016 році [25].

Нижче представлені досягнення перекладу на основі нейронної мережі:

- Наскрізне навчання: параметри в NMT (Neural Machine Translation) одночасно оптимізуються для мінімізації функції втрат на виході нейромережі.
- Розподілені уявлення: NMT краще використовує схожості в словах та фразах.
- Найкраще дослідження контексту: NMT працює більше контексту — вихідний і частково цільової текст, щоб переводити точніше.
- Більш швидке генерування тексту: переклад тексту на основі глибокого навчання набагато перевершує за якістю метод паралельного корпусу.

Головна проблема RNN — проблема зникнення градієнта, коли інформація втрачається з часом. Інтуїтивно здається, що це не є серйозною проблемою, так як це тільки ваги, а не стану нейронів. Але з плином часу ваги стають місцями, де зберігається інформація з минулого. Якщо вага прийме значення 0 або 100000, попереднє стан не буде занадто інформативно. Як наслідок, RNN будуть зазнавати труднощів у запам'ятовуванні слів, що стоять далі в послідовності, а передбачення будуть робитися на основі крайніх слів, що створює проблеми [21].

Мережі короткострокової-довгострокової пам'яті (Long / short term memory, далі LSTM) намагаються боротися з проблемою градієнта зникнення вводючи гейти (gates) і вводючи осередок пам'яті. Кожен нейрон представляє з себе осередок пам'яті з трьома Гейт: на вхід, на вихід і забування (forget). Ці затвори виконують функцію охоронців для інформації, дозволяючи або забороняючи її потік.

- Вхідний гейт визначає, яку кількість інформації з попереднього шару буде зберігатися в цьому осередку;
- Вихідний гейт виконує роботу на іншому кінці і визначає, яка частина наступного шару дізнається про стан поточної комірки.
- Гейт забування контролює міру збереження значення в пам'яті: якщо при вивченні книги починається нова глава, іноді для нейромережі стає необхідним забути деякі слова з попередньої глави.

Було показано, що LSTM здатні навчатися на складних послідовності і, наприклад, писати в стилі Шекспіра чи складати примітивну музику. Зауважимо, що кожен з гейтов з'єднаний з осередком на попередньому нейроні з певною вагою, що вимагають більше ресурсів для роботи. LSTM поширені і використовуються в машинному перекладі. Крім цього, це стандартна модель для

більшості завдань маркування (labeling) послідовності, які складаються з великої кількості даних [20].

Закриті рекурентні блоки (Gated recurrent units, далі GRU) відрізняються від LSTM, хоча теж є розширенням для нейросетевого машинного навчання. У GRU на один гейт менше, і робота будується по-іншому: замість вхідного, вихідного і забування, є гейт поновлення (update gate). Він визначає, скільки інформації необхідно зберегти з останнього стану і скільки інформації пропускати з попередніх шарів [22].

Функції скидання гейта (reset gate) схожі на затвор забування у LSTM, але розташування відрізняється. GRU завжди передають своє повне стан, не мають вихідний затвор. Часто ці затвор функціонує як і LSTM, однак, відмінністю полягає в наступному: в GRU затвор працюють швидше і легше в управлінні (але також менш інтерпретовані). На практиці вони прагнуть нейтралізувати один одного, так як потрібна велика нейросеть для відновлення виразності (expressiveness), яка зводить нанівець природи в результаті. Але у випадках, де не потрібно екстра виразності, GRU показують кращий результат, ніж LSTM [25].

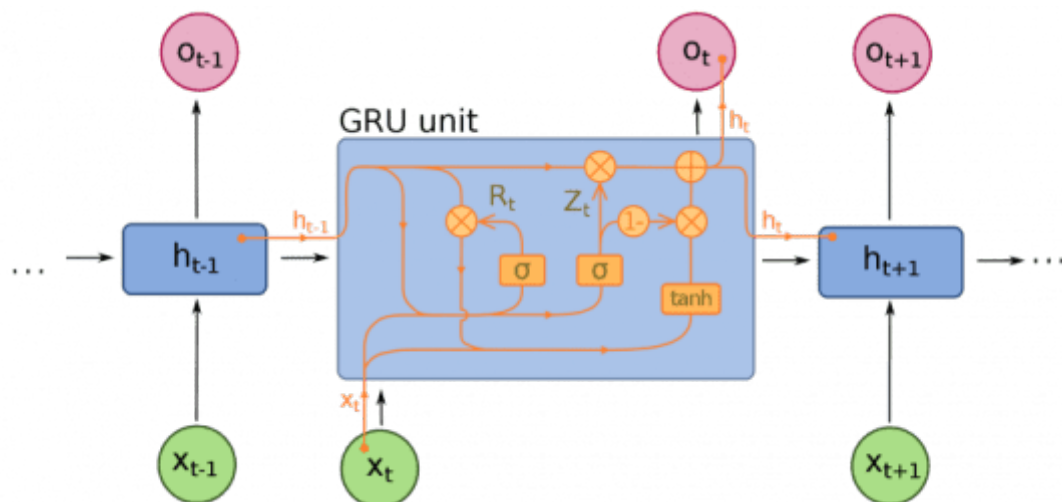


Рисунок 2.1 – Функція скидання гейта.

Крім цих трьох головних архітектур, за останні кілька років з'явилося багато поліпшень в нейромережевому машинному перекладі.

Нижче представлені деякі примітні розробки:

- Sequence-to-Sequence Learning with Neural Networks довели ефективність LSTM для нейронного машинного перекладу. Цей метод використовує багат шарову LSTM, щоб відобразити вхідні послідовності у вигляді вектора з фіксованою розмірністю, далі йде застосування іншої LSTM для декодування цільової послідовності з вектора.
- Neural Machine Translation by Jointly Learning to Align and Translate представили механізм уваги (attention mechanism) в NLP (який буде розглянуто в наступній частині). Визнаючи факт, що використання вектора фіксованої довжини є вузьким місцем в поліпшенні результативності NMT, автори пропонують вирішувати моделі автоматично шукати частини вихідного пропозицію, що релевантні до передбачення цільового слова, без необхідності явного формування цих частин [26].
- Convolutional over Recurrent Encoder for Neural Machine Translation підсилюють стандартний RNN кодировщик в NMT за допомогою додаткового сверточного шару, щоб захоплювати більш широкий контекст на виході кодувальника.
- Google[©] створила власну NMT систему, звану Google's Neural Machine Translation, яка вирішує завдання точності і простоти застосування. Модель складається з глибокої LSTM мережі з 8 кодують і 8 декодуючими шарами і використовує як залишкові зв'язку, так і attention-зв'язку від декодер- до кодер-мережі.
- Замість використання рекурентних нейромереж, Facebook AI Researchers використовують нейронну мережу для задач sequence-to-sequence навчання в NMT.

2.9. Розробка голосового чат боту та тренування нейронної мережі.

Для розробки використовується Dialogflow. Це хмарний сервіс розпізнавання природної мови від Google[©], який дозволяє навчати власті нейронні мережі. Він має безкоштовні ліміти використання, а для роботи з API можна скористатися бібліотеками для різних мов, тому його досить легко інтегрувати у свої проекти.

Також Dialogflow можна інтегрувати з месенджерами, так що для простих кожен проект Cloud Platform може містити один додаток App Engine [6].

Веб сайт повинен увійти в папку DiplomaBot, а його файл конфігурації називається agent.json, але крім того, він може приймати будь-яку форму. Файл app.yaml — це файл конфігурації, який повідомляє App Engine, як зіставляти URL-адреси статичних файлів. Його не потрібно редагувати [9].

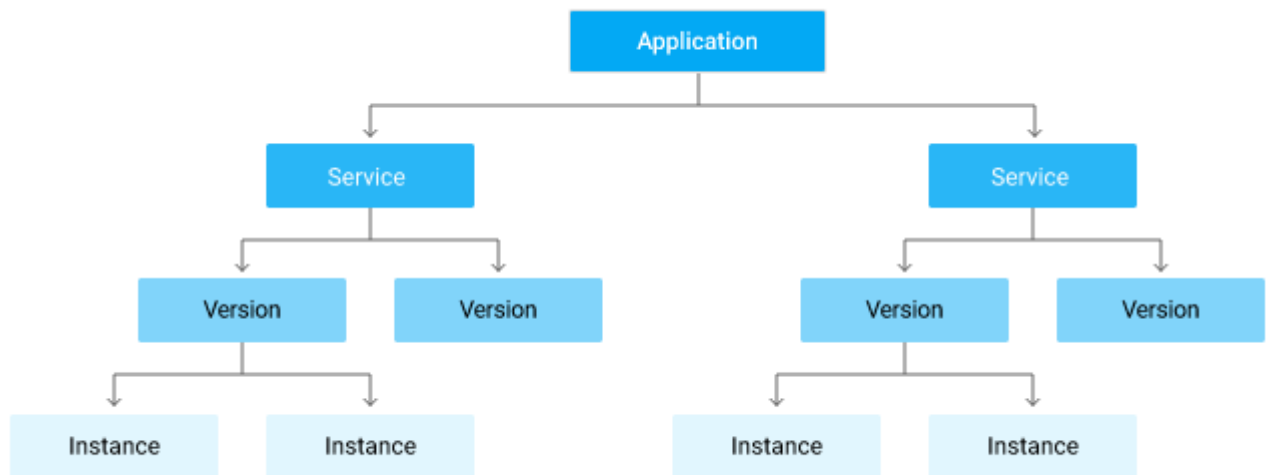
Додаток App Engine складається з одного ресурсу додатка, який має частини з однієї або декількох служб. Кожна служба може бути налаштована на використання різних режимів виконання та для роботи з різними налаштуваннями продуктивності. У межах кожної служби ви розгортаєте її версії. Потім кожна версія працює в одному або декількох екземплярах, залежно від того, скільки трафіку ви налаштували на обробку.

Компоненти програми

Додаток App Engine створено в рамках проекту Google[©] Cloud під час створення ресурсу програми. Додаток App Engine - це контейнер верхнього рівня, що включає сервіси, версію та ресурси екземпляра, які складають програму. Коли створюється додаток App Engine, усі ресурси користувача створюються у вибраному регіоні, включаючи код додатка разом із колекцією налаштувань, облікових даних та метаданих додатка [3].

Кожна програма App Engine включає щонайменше одну послугу — default сервіс, який може містити багато версій. Наступна схема ілюструє ієрархію програми App Engine, що працює з декількома службами. На цій діаграмі у

додатку є два сервіси, що містять кілька версій, і дві з цих версій активно



працюють у кількох екземплярах (рис 2.2):

Рисунок 2.2 – Схема версій

Використовуються сервіси в App Engine, щоб розподілити великі програми на логічні компоненти, які можуть надійно ділитися функціями App Engine і спілкуватися між собою. Як правило, служби App Engine поведуться як мікросервіси. Таким чином, можна запустити весь додаток в одній службі, або можна спроектувати та розгорнути кілька сервісів для запуску як набір мікросервісів [12].

Наприклад, додаток, який обробляє запити клієнтів, може включати окремі сервіси, кожен з яких вирішує різні завдання, наприклад:

- Запити API від мобільних пристроїв
- Внутрішні, адміністративні запити
- Обробка бекенду, така як оплата трубопроводів та аналіз даних

Кожна служба в App Engine складається з вихідного коду додатка та відповідних файлів конфігурації App Engine. Набір файлів, які розгортаються в

службі, являють собою єдину версію цієї служби, коли додаток розгортається у цій службі, створюються додаткові версії в межах тієї самої служби [24].

Версії

Наявність декількох версій програми в межах кожної послуги дозволяє швидко перемикається між різними версіями цього додатка для відкатів, тестування чи інших тимчасових подій. Можна переміщувати трафік до однієї або декількох конкретних версій програми, переміщуючи або розділяючи трафік.

Екземпляри

Версії служб запускаються в одному або кількох екземплярах. За замовчуванням App Engine масштабує програму щоб відповідати навантаженню. Додатки дозволять збільшити кількість примірників, які працюють, щоб забезпечити стабільну продуктивність, або зменшити їх, щоб мінімізувати непрацюючі випадки та зменшити витрати [14].

Запити

Кожна з послуг додатка та кожна з версій цих служб повинні мати унікальні назви.

Вхідні запити користувачів направляються до служб або версій, налаштованих для обробки трафіку. Також можна орієнтувати та направляти запити на конкретні служби та версії.

Потім можна використовувати ці унікальні імена для націлювання та спрямування трафіку на конкретні ресурси за допомогою URL-адреси [23].

Наприклад:

https://VERSION_ID-dot-SERVICE_ID-dot-PROJECT_ID.REGION_ID.r.appspot.com

3. РЕЗУЛЬТАТИ

За допомогою використання та навчання глибоких нейронних мереж було адаптовано процедуру синтезу мовлення під українську мову, а також розроблено пакетне рішення (архів плагіну додається до дипломної роботи) для інтеграції такої системи синтезу мовлення в сайти на базі системи управління контентом WordPress[®], практична частина роботи реалізована на веб сайті www.voice.uttermouse.com. Код програми надано в додатках 1-5.

Для подальшої зручності користування слабозорим на сайт був інтегрований голосовий чат, який розроблено за допомоги сервісів Google[®] Cloud Platform та Google[®] Dialogflow та Google[®] Speech to Text API і адаптований для роботи з веб сайтом за допомогою мови PHP. Алгоритм роботи чатботу:

3.1. Розробка плагіну для CMS WordPress[®].

Щоб створити плагін WordPress[®]:

- Створено папку плагіна. У ній лежать нашого плагіна. Створюється вона у папці всіх плагінів WordPress[®]. /wp-content/plugins/my-plugin-name.
- Основний файл плагіна на мові PHP.
- Створити опис плагіна — заголовки плагіна. Вони потрібні щоб wordpress розпізнав плагін як плагін, інакше він просто не працюватиме.

Використано два типи хуків у WordPress[®]:

- Події (actions) — дозволяють додавати або змінювати функціонал WordPress[®].
- Фільтри (filters) — дозволяють змінювати дані.

Хуки потрібні не тільки для розробників плагінів, але і тим, хто буде використовувати плагін. Хуки використовуються скрізь: у самому ядрі WordPress[®], у плагінах та темах. Саме хуки роблять WordPress[®] таким гнучким. Безпосередньо до плагіну належать три функції:

- register_activation_hook() — реєструє функцію, яка спрацює під час активації плагіна. Використовується для додавання налаштувань плагіна тощо.

- `register_deactivation_hook()` — реєструє функцію, яка має запускатися після деактивації плагіна. Використовується для видалення тимчасових даних плагіна.
- `register_uninstall_hook()` — реєструє функцію, що викликається при видаленні плагіна.

3.2. Конфігурація передачі та обробки мовного сигналу

Для того, щоб нейронна мережа могла обробити аудіосигнал та мати можливість його кодувати/декодувати було розроблено алгоритм наведений нижче:

```

/**
<code>.Google.cloud.texttospeech.v1.AudioEncoding audio_encoding =
1 [(.Google.api.field_behavior) = REQUIRED];</code>
    * @return int*/
    public function getAudioEncoding()
    { return $this->audio_encoding;}
/**
<code>.Google.cloud.texttospeech.v1.AudioEncoding
audio_encoding = 1 [(.Google.api.field_behavior) = REQUIRED];</code>
    * @param int $var
    * @return $this
    */
    public function setAudioEncoding($var)
    {
GPBUtil::checkEnum($var,
\Google\Cloud\TextToSpeech\V1\AudioEncoding::class);
        $this->audio_encoding = $var;
        return $this;
    }
<code>double speaking_rate = 2 [(.Google.api.field_behavior) =
INPUT_ONLY, (.Google.api.field_behavior) = OPTIONAL];</code>
    * @return float
    */
    public function getSpeakingRate()
    {

```

```

        return $this->speaking_rate;
    }
    /**
    <code>double speaking_rate = 2
[ (.Google.api.field_behavior) = INPUT_ONLY,
(.Google.api.field_behavior) = OPTIONAL];</code>
    * @param float $var
    * @return $this
    */
    public function setSpeakingRate($var)
    {
        GPBUtil::checkDouble($var);
        $this->speaking_rate = $var;
        return $this;
    }

```

Данні які отримуються:

- `@type int $audio_encoding` — Обов'язково. Формат потоку аудіо байтів.
- `@type float $speaking_rate` — Швидкість/швидкість мовлення в діапазоні [0,25, 4,0]. 1 – нормальна швидкість, яку підтримує певний голос, 2.0 – вдвічі більше, а 0,5 вдвічі менше. Якщо значення `unset(0.0)`, за замовчуванням встановлюється вихідний 1.0. Будь-які інші значення `< 0,25` або `> 4,0` повернуть помилку.
- `@type float $pitch` — Висота мовлення, в діапазоні [-20,0, 20,0]. 20 означає збільшення на 20 півтонів від початкової висоти. -20 означає зменшення на 20 півтонів вихідної висоти.
- `@type float $volume_gain_db` — Підсилення гучності (в дБ) від нормальної власної гучності, підтримується конкретним голосом, у діапазоні [-96,0, 16,0]. Якщо не встановлено, або встановлене на значення 0,0 (дБ), відтворюватиметься зі звичайною амплітудою

рідного сигналу. А значення -6,0 (дБ) буде відтворене приблизно на половині амплітуди від рідного сигналу. Відтворюватиметься значення +6,0 (дБ).

- `@type int $sample_rate_hertz` — Частота дискретизації синтезу (у герцах) для цього аудіо. Коли це
- `@type string[]\Google\Protobuf\Internal\RepeatedField $effects_profile_id` — ідентифікатор, який вибирає профілі "аудіоефектів" які застосовуються до (після синтезованого) тексту до мовлення. Застосовуються ефекти в тому порядку, в якому вони подані на <https://cloud.google.com/text-to-speech/docs/audio-profiles>.

3.3. Конфігурація декодування.

```
<?php
<code>Google.cloud.texttospeech.v1.SynthesizeSpeechRequest</code>{
  /**
Синтезатор вимагає або звичайний текст, або SSML як вхідні дані.
  <code>.Google.cloud.texttospeech.v1.SynthesisInput input = 1
  [(.Google.api.field_behavior) = ОБОВ'ЯЗКОВО];</code>
  */
  приватний $input = нуль;
  /**
   * Голос синтезованого аудіо.
   * Згенеровано з поля protobuf
<code>.Google.cloud.texttospeech.v1.VoiceSelectionParams voice = 2
  [(.Google.api.field_behavior) = ОБОВ'ЯЗКОВО];</code>
  */
  приватний $voice = нуль;
  /**
   * Конфігурація синтезованого аудіо.
   *
   * Згенеровано з поля protobuf
<code>.Google.cloud.texttospeech.v1.AudioConfig audio_config = 3
  [(.Google.api.field_behavior) = ОБОВ'ЯЗКОВО];</code>
  */
  приватний $audio_config = null;

  /**
   * @param масив $data {
   * Необов'язково. Дані для заповнення об'єкта Message.
  @type \Google\Cloud\TextToSpeech\V1\SynthesisInput $input
```

Синтезатор вимагає або звичайний текст, або SSML як вхідні дані.

```

* @type \Google\Cloud\TextToSpeech\V1\VoiceSelectionParams
$voice
Потрібний голос синтезованого аудіо.
* @type \Google\Cloud\TextToSpeech\V1\AudioConfig $audio_config
* }

/**
* Вимагається. Синтезатор вимагає або звичайний текст, або SSML
як вхідні дані.
*
* Згенеровано з поля protobuf
<code>.Google.cloud.texttospeech.v1.SynthesisInput input = 1
[ (.Google.api.field_behavior) = ОБОВ'ЯЗКОВО];</code>
* @return \Google\Cloud\TextToSpeech\V1\SynthesisInput
*/

```

3.4. Запит на обробку.

```

/**
* Повідомлення повернуто клієнту методом `SynthesizeSpeech`.
<code>Google.cloud.texttospeech.v1.SynthesizeSpeechResponse</code>
*/ клас SynthesizeSpeechResponse розширює
\Google\Protobuf\Internal\Message

```

Байти аудіоданих, закодовані, як зазначено в запиті, включаючи заголовок для кодувань, які загорнуті в контейнери (наприклад, MP3, OGG_OPUS).

Для аудіо LINEAR16 включаємо заголовок WAV, як із усіма полями байтів протобуфери використовують чисте двійкове представлення, тоді як у поданнях JSON використовується base64.

3.5. Взаємодія користувачів з інтеграціями

Виконання інтеграцій

За замовчуванням агент відповідає на відповідні розпізнані звернення статичною відповіддю. Якщо можна використати один із варіантів інтеграції, можна надати більш динамічну відповідь. Коли вмикається виконання наміру, Dialogflow відповідає на цей намір, викликаючи службу, яка визначається. Наприклад, якщо кінцевий користувач хоче запланувати стрижку в п'ятницю,

служба може перевірити базу даних і відповісти кінцевому користувачеві з інформацією про доступність на п'ятницю [16].

Кожен намір має налаштування для включення виконання. Якщо намір вимагає певних дій з боку системи або динамічної відповіді, потрібно включити виконання наміру. Якщо інтенція без включеного виконання збігається, Dialogflow використовує статичну відповідь, яка визначена для наміру (рис 3.1).

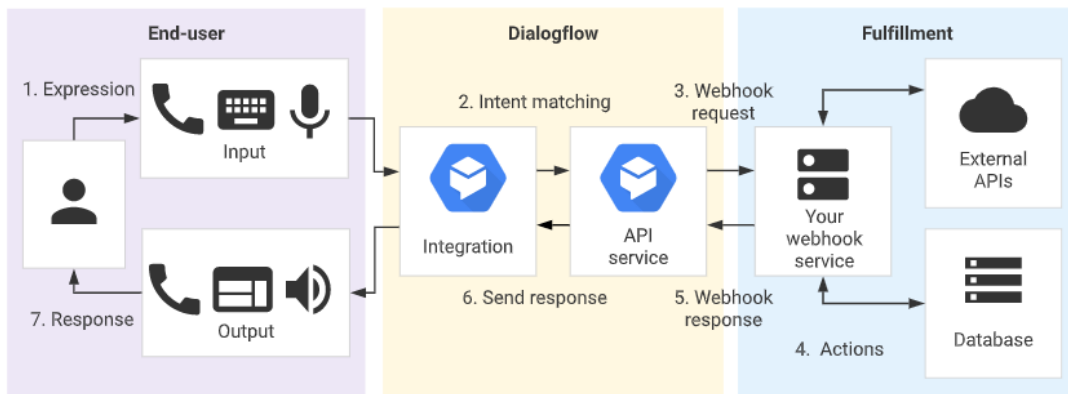


Рисунок 3.1 – Кінцевий користувач набирає або говорить вираз.

DialogFlow відповідає виразу кінцевого користувача за наміром і витягує параметри, далі Dialogflow API надсилає повідомлення із запитом webhook на службу webhook (описану у розділі 3.4). Це повідомлення містить інформацію про відповідні наміри, дії, параметри та відповідь, визначені для наміру.

Служба виконує необхідні дії, наприклад запити до бази даних або зовнішні дзвінки API.

Служба надсилає відповідь на відповідь веб-камери в Dialogflow. Це повідомлення містить відповідь, яку слід надіслати кінцевому користувачеві.

- Dialogflow надсилає відповідь кінцевому користувачеві.
- Кінцевий користувач бачить, або чує відповідь [27].

3.6. Взаємодія користувачів з API

Якщо не використовується один із варіантів інтеграції, потрібно написати код, який безпосередньо взаємодіє з кінцевим користувачем. Також треба

безпосередньо взаємодіяти з API Dialogflow для кожного ходу, щоб надсилати вирази кінцевого користувача та отримувати відповідні наміри (рис 3.2). На наступній схемі показаний потік обробки під час взаємодії з API.

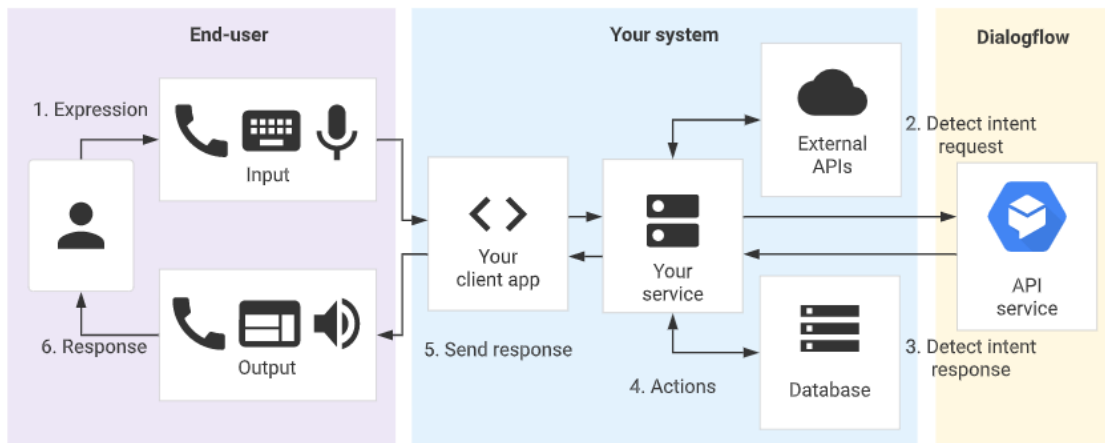


Рисунок 3.2 – Схема взаємодії. Кінцевий користувач набирає або говорить вираз.

Служба надсилає це вираження кінцевого користувача Dialogflow у повідомленні про запит щодо намірів.

Dialogflow надсилає службі повідомлення щодо відповіді про виявлення. Це повідомлення містить інформацію про відповідні наміри, дії, параметри та відповідь, визначені для наміру.

Служба виконує необхідні дії, наприклад запити до бази даних або зовнішні дзвінки API.

- Служба надсилає відповідь кінцевому користувачеві.
- Кінцевий користувач або бачить, або чує відповідь [27]

3.7. Обробка запитів

Програма несе відповідальність за запуск веб-сервера та обробку запитів. можна використовувати будь-яку веб-структуру, доступну для мови розробки.

App Engine запускає кілька примірників програми, і кожен примірник має власний веб-сервер для обробки запитів. Будь-який запит може бути перенаправлений до будь-якого примірника, тому послідовні запити від одного і

того ж користувача не обов'язково надсилаються до одного і того ж екземпляра. Екземпляр може одночасно обробляти кілька запитів.

Кількість екземплярів може автоматично регулюватися у міру зміни трафіку. Наступний зразок містить код JavaScript для запуску сервера та відповіді на всі GET запити від веб-клієнтів до кореневого шляху ('/'), відображаючи "Hello, world!" повідомлення через сервер, який працює на порт 8080:

```
const express = require('express');
const app = express();
app.get('/', (req, res) => {
  res.status(200).send('Hello, world!').end();
});
// Start the server
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
  console.log(`App listening on port ${PORT}`);
  console.log('Press Ctrl+C to quit.');
```

Важливо, що в останніх рядках код надає серверу прослуховування порту, визначеного `process.env.PORT` змінною. Це змінна середовища, встановлена програмою App Engine — якщо сервер не слухає цей порт, він не зможе отримувати запити [24].

Вхідний запит HTTP включає заголовки HTTP, надіслані клієнтом. З метою безпеки деякі заголовки дезінфікуються або доповнюються проміжними проксі-серверами, перш ніж вони дісталися до програми.

Існують обмеження, які застосовуються до генерованих відповідей, і відповідь може бути змінена до повернення клієнту.

Вимкнення буферизації

За замовчуванням всі відповіді від App Engine буферизовані у 64-х блоках контейнерів. У деяких випадках може бути доцільним відключити буферизацію та безпосередньо передавати байти клієнту. Це, як правило, краще використовувати

підвісні GET або серверні події (SSE). Щоб відключити буферизацію, можна встановити X-Accel-Buffering заголовок відповіді на no.

HTTPS-з'єднання

З міркувань безпеки всі програми повинні заохочувати клієнтів до з'єднання https. Можна використовувати заголовок Strict-Transport-Security, щоб доручити браузеру надавати перевагу https над http певною сторінкою або всім доменом [28].

3.8. Використання семантичної верстки та семантичної розмітки при розробці веб сайтів для людей з вадами зору.

За допомогою правильного коду можна приховати неточності семантики від людини, але ось від пошукових систем не можна. Тому, при розробці сайтів з можливістю адаптації для людей з вадами зору, варто уважно прописувати ключові пункти:

- Теги структури документа: <html>, <head>, <body>.
- Текстові теги: <p>, <h1>, <blockquote>.
- Медіа теги: , <video>.
- Кореляційні теги: , .

Якісна семантична верстка допоможе спокійно розбиратися в інформації на сайті, дозволяє змінювати версії на варіант для людей з вадами зору та назад без збоїв. У тому числі і сам адаптований режим сайту буде робочим і читаним.

Медіа теги з описом alt

Не завжди на сайтах необхідно видаляти зображення. Багато незрячі або слабозорі люди використовують програми екранного доступу. Наприклад, Jaws, NVDA[®]. Саме тому до кожної картинки, файлу, полю форм, запланування або будь-якого іншого додатку потрібно текстовий супровід. Тобто, суть будь-якого медіа-файлу повинна бути передана в текстовому форматі.

ТСП для серфінгу в інтернеті.

Крім програм екранного доступу, слабозорі можуть використовувати брайлівський дисплей, програми збільшення екрану, принтер для друку рельєфно-крапковим шрифтом.

4. ОБГОВОРЕННЯ І ВИСНОВКИ

Були досліджені проблеми споживання контенту в мережі Інтернет у слабозорих та людей з вадами зору.

За допомогою використання та навчання глибинних нейронних мереж було адаптовано процедуру синтезу мовлення під українську мову, а також розроблено пакетне рішення (архів плагіну додається до дипломної роботи) для інтеграції такої системи синтезу мовлення в сайти на базі системи управління контентом WordPress[®], практична частина роботи реалізована на веб сайті www.voice.uttermouse.com.

Синтез мовлення з тексту у вигляді пакетного рішення дозволить розробникам сайтів створювати синтетичну людську мову, яка звучить природно і гарно. Ця система буде інтегрується як плагін до CMS. Розробка надає можливість використовувати файли аудіоданих, які будуть створюватися за допомогою синтезу мовлення, для забезпечення роботи програм або доповнення мультимедіа, як аудіозаписи що вирішить проблему споживання контенту слабозорими та людьми з вадами зору.

Перетворення тексту в мовлення перетворює текст або введену мову, інтегрує в нього семантичну розмітку і кодує в аудіодані, такі як MP3 або LINEAR16 (кодування, що використовується у файлах WAV). Синтез мовлення з тексту створює вихідні аудіодані природної людської мови. Тобто він створює звук, схожий на розмову людини.

У системі використовується модель WaveNet [30], яка використовується для створення голосу. Моделі WaveNet навчаються з використанням необроблених звукових зразків реальних людей, які надаються ліцензією використання сервісу хмарних обчислень Google[®] Cloud Platform. В результаті ці моделі генерують синтетичне мовлення з більш схожим на людину наголосом і флексією на складах, фонемах і словах.

За допомогою використання Google[©] Text-to-Speech API реалізована можливість налаштування інших аспектів виведення аудіоданих, створених шляхом синтезу мовлення. Функція синтезу мовлення підтримує налаштування частоти мовлення, висоти, гучності та частоти дискретизації в герцах. Необхідність реалізації налаштувань виникла через інтеграцію семантичної розмітки для покращення вимови. Програма може прочитати двійкові дані безпосередньо з відповіді у форматі gRPC, однак формат JSON використовується під час відповіді на запит REST.

Оскільки JSON є текстовим форматом, який безпосередньо не підтримує двійкові дані, функція перетворення тексту в мовлення повертає відповіді, закодовані у Base64. На стороні плагіну перетворюється текстові дані, закодовані як base64, з відповіді в двійкові, перш ніж зможуть відтворитись на пристрої.

Для подальшої допомоги слабозорим на сайт був інтегрований голосовий чат, який розроблено за допомоги сервісів Google[©] Cloud Platform та Google[©] Dialogflow та Google[©] Speech-to-Text API і адаптований для роботи з веб сайтом за допомогою мови PHP. Алгоритм роботи чатботу:

1. Користувач надсилає текстове/голосове повідомлення на пристрій або програму.
2. Програма / Пристрій передає повідомлення Dialogflow.
3. Повідомлення класифікується та зіставляється з відповідним наміром (наміри визначаються вручну розробником у Dialogflow).
4. Визначаються наступні дії для кожного наміру (Webhook).
5. Коли Dialogflow виявляє певний намір, веб-перехоплювач використовуватиме зовнішні API для пошуку відповіді у зовнішніх базах даних.
6. Зовнішні бази даних надсилають на веб-перехоплювач необхідну інформацію.
7. Webhook відправляє форматовану відповідь на намір.

8. Intent генерує дієві дані з різних каналів.
9. Активні дані надсилаються у вихідні програми/пристрої.
10. Користувач отримує текст/зображення/голосову відповідь.

4.1. Маршрутизація запитів

Запити можна направити двома способами:

- Правила маршрутизації за замовчуванням App Engine застосовуються до запитів із URL-адресою, яка закінчується на рівні домену.
- Крім того, можна використовувати диспетчерський файл, який спрямовує конкретні шаблони URL відповідно до встановлених правил.

Ці параметри стосуються лише розгорнутих додатків. Під час тестування локальна поведінка маршрутизації залежить від конкретного середовища виконання та розробки, яке використовується.

Запити та домени

Після запуску програми в App Engine, можна використовувати наступну URL-адресу для надсилання HTTP-запитів у програму:

`https://PROJECT_ID.REGION_ID.r.appspot.com`

де PROJECT_ID ідентифікатор проекту Google[®] Cloud, який містить додаток.

Ця URL-адреса надсилає запити до версії додатка, налаштованої на отримання трафіку.

У кожній версії додатка також є своя URL-адреса, тому можна розгорнути та протестувати нову версію, перш ніж налаштувати її на отримання трафіку.

URL-адреса, що стосується версії, використовує ідентифікатор конкретної версії на додаток до ідентифікатора проекту, ідентифікатора регіону та appspot.com доменного імені [26].

Наприклад: `https://VERSION_ID-dot-default-dot-PROJECT_ID.REGION_ID.r.appspot.com`

Субдомени

В appspot.com області також підтримують субдомени форми, де може бути будь-який рядок дозволений в одній частині доменного імені, за винятком символу.

Запити, які таким чином надсилаються до будь-якого субдомену, перенаправляються до програми.

SUBDOMAIN-dot-PROJECT_ID.REGION_ID.r.appspot.comSUBDOMAIN
можна також використовувати піддомени з версіями конкретного URL:

https://SUBDOMAIN-dot-VERSION_ID-dot-
PROJECT_ID.REGION_ID.r.appspot.com

Спеціальні домени

Можна налаштувати спеціальний домен верхнього рівня за допомогою програми Google[®] Workspace, а потім призначити субдомени різним додаткам, таким як Google[®] Mail або сайти. Користувач також може прив'язати додаток App Engine з піддоменом.

Доменне ім'я включене в дані запиту

Ім'я домену, яке використовується для запиту, включається в дані запиту, які передаються вашій програмі.

Таким чином, можна використовувати дані запиту, щоб контролювати, як програма реагує на основі доменного імені у запиті. Наприклад, якщо користувач хоче перенаправити на офіційний домен, можна кодувати свою програму, щоб перевірити Host заголовок запиту, а потім відповісти відповідно до доменного імені.

Ідентифікатор регіону

Це REGION_ID код, який Google[®] присвоює на основі регіону, що був обраний під час створення програми. Включення REGION_ID.r в URL-адреси App Engine не обов'язково для існуючих додатків і буде потрібно для всіх нових додатків.

Щоб забезпечити плавний перехід App Engine повільно оновлюється для використання ідентифікаторів регіону.

Оскільки ідентифікатор необов'язковий для існуючих додатків, не потрібно оновлювати URL-адреси чи вносити інші зміни, коли ідентифікатор регіону буде доступний для існуючих додатків [29].

Можна націлити HTTP-запит з різним ступенем конкретності. У наведених нижче прикладах REGION_ID.r.appspot.com можна замінити спеціальним доменом програми.

В URL підрядка VERSION_ID, SERVICE_ID і PROJECT_ID кожен представляє ідентифікатори ресурсів застосування.

Маршрутизація за замовчуванням

Наведені нижче шаблони URL мають поведінку маршрутизації за замовчуванням. Маршрутизація за замовчуванням перевизначається, якщо у диспетчерському файлі є відповідна схема :

- Надсилає запит на доступний примірник default служби:
https://PROJECT_ID.REGION_ID.r.appspot.com
- https: // CUSTOM_DOMAIN
- Запити отримує будь-яка версія, налаштована для трафіку в default сервісі.
- Надсилає запит на доступний примірник певної послуги:
https://SERVICE_ID-dot-PROJECT_ID.REGION_ID.r.appspot.com
- https: // SERVICE_ID . CUSTOM_DOMAIN
- Запити отримуються будь-якою версією, налаштованою для трафіку в цільовій службі. Якщо послуги, на яку ви орієнтуєтесь, не існує, запит буде переадресовано.
- Надсилає запит на доступний екземпляр певної версії в default послуга:
https://VERSION_ID-dot-default-dot-PROJECT_ID.REGION_ID.r.appspot.com
- https: // VERSION_ID . CUSTOM_DOMAIN
- Якщо послуга не націлена, запити надсилаються до неї default [30].

М'яка маршрутизація

Якщо запит відповідає частині імені хоста, але включає в себе службу, версію чи ім'я екземпляра, яка не існує, запит буде перенаправлений до служби. М'яка маршрутизація не застосовується до користувацьких доменів, запити до них повернуть код статусу HTTP, якщо ім'я хоста недійсне.
PROJECT_ID.REGION_ID.r.appspot.comdefault404

Цільова маршрутизація

Наступні шаблони URL-адрес гарантовано досягають своєї мети, якщо вони існують. Ці запити ніколи не перехоплюються та не переспрямовуються за шаблонами, визначеними у диспетчерському файлі:

- Надсилає запит на доступний примірник певної послуги та версії:
`https://VERSION_ID-dot-SERVICE_ID-dot-PROJECT_ID.REGION_ID.r.appspot.com`
- `https://VERSION_ID.SERVICE_ID.PROJECT_ID.CUSTOM_DOMAIN`

Маршрутизація з відправним файлом

Для URL-адрес, які використовують шаблони, описані раніше, можна створити файл відправки, щоб замінити правила маршрутизації App Engine і визначити власні правила маршрутизації. За допомогою диспетчерського файлу можна надсилати вхідні запити до певної служби на основі шляху або імені хоста в URL-адресі запиту [9].

Файл відправки повинен бути розміщений або в корені каталогу проекту, або в кореневому каталозі default служби. можна визначити до 20 правил маршрутизації у диспетчерському файлі, і кожне правило складається з елементів service та url елементів [24].

Наприклад, можна створити диспетчерський файл для маршрутизації мобільних запитів, таких `https://simple-sample.uc.r.appspot.com/mobile/` як мобільний інтерфейс, до статичного бекенда:

```
dispatch:
  # Send all mobile traffic to the mobile frontend.
  - url: "*/mobile/*"
```



```

    service: mobile-frontend
# Send all work to the one static backend.
- url: "*/work/*"
    service: static-backend

```

4.2. Обробка запитів

Програма несе відповідальність за запуск веб-сервера та обробку запитів. можна використовувати будь-яку веб-структуру, доступну для мови розробки.

App Engine запускає кілька примірників програми, і кожен примірник має власний веб-сервер для обробки запитів. Будь-який запит може бути перенаправлений до будь-якого примірника, тому послідовні запити від одного і того ж користувача не обов'язково надсилаються до одного і того ж екземпляра. Екземпляр може одночасно обробляти кілька запитів.

Кількість екземплярів може автоматично регулюватися у міру зміни трафіку. Наступний зразок містить код JavaScript для запуску сервера та відповіді на всі GET запити від веб-клієнтів до кореневого шляху ('/'), відображаючи "Hello, world!" повідомлення через сервер, який працює на порт 8080:

```

const express = require('express');
const app = express();
app.get('/', (req, res) => {
  res.status(200).send('Hello, world!').end();
});
// Start the server
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
  console.log(`App listening on port ${PORT}`);
  console.log('Press Ctrl+C to quit.');
```

Важливо, що в останніх рядках код надає серверу прослуховування порту, визначеного `process.env.PORT` змінною. Це змінна середовища, встановлена програмою App Engine — якщо сервер не слухає цей порт, він не зможе отримувати запити [24].

Вхідний запит HTTP включає заголовки HTTP, надіслані клієнтом. З метою безпеки деякі заголовки дезінфікуються або доповнюються проміжними проксі-серверами, перш ніж вони дісталися до програми.

Існують обмеження, які застосовуються до генерованих відповідей, і відповідь може бути змінена до повернення клієнту.

Вимкнення буферизації

За замовчуванням всі відповіді від App Engine буферизовані у 64-х блоках контейнерів. У деяких випадках може бути доцільним відключити буферизацію та безпосередньо передавати байти клієнту. Це, як правило, краще використовувати підвісні GET або серверні події (SSE). Щоб відключити буферизацію, можна встановити X-Accel-Buffering заголовок відповіді на no.

HTTPS-з'єднання

З міркувань безпеки всі програми повинні заохочувати клієнтів до з'єднання https. Можна використовувати заголовок Strict-Transport-Security, щоб доручити браузеру надавати перевагу https над http певною сторінкою або всім доменом [23].

4.3. Навчальні фрази та контексти

Навчальні фрази: це приклади фраз для того, що можуть сказати кінцеві користувачі. Коли вираз кінцевого користувача нагадує одну з цих фраз, DialogFlow відповідає наміру. Не потрібно визначати кожен можливий приклад, оскільки вбудоване машинне навчання Dialogflow розширюється у списку іншими, подібними фразами [13].

Дія: можна визначити дію для кожного наміру. Коли інтент збігається, Dialogflow надає дію вашій системі, і можна використовувати дію для запуску певних дій, визначених у вашій системі.

Параметри: коли намір підходить під час виконання, Dialogflow надає вилучені значення з виразу кінцевого користувача як параметри. Кожен параметр має тип, який називається типом сутності, який точно диктує спосіб отримання

даних. На відміну від необмеженого введення кінцевого споживача, параметри - це структуровані дані, які можна легко використовувати для виконання певної логіки або генерування відповідей [25].

Відповіді: визначаються текстові, мовленнєві чи візуальні відповіді, щоб повернутися до кінцевого користувача. Вони можуть надати кінцевому користувачеві відповіді, попросити кінцевого користувача отримати додаткову інформацію або припинити розмову [25]. На наступній схемі показаний базовий потік для відповідності намірів та відповіді для кінцевого користувача (рис. 4.1):

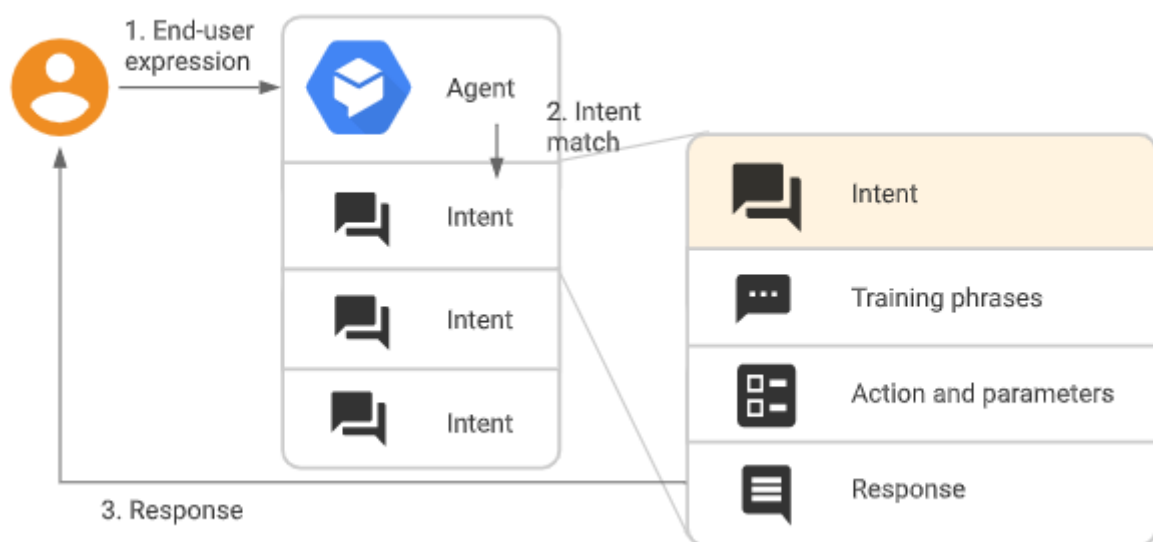


Рисунок 4.1 – Базовий потік для відповідності намірів та відповіді для кінцевого користувача

Суб'єкти

Кожен параметр наміру має тип, який називається типом сутності, який точно диктує, як витягуються дані з виразу кінцевого користувача.

DialogFlow забезпечує попередньо визначені системні об'єкти, які можуть відповідати багатьом загальним типам даних. Наприклад, є системні об'єкти для відповідності дат, часу, кольорів, адрес електронної пошти тощо. Ви також можете створити власні спеціалізовані об'єкти для відповідності користувацьким даним. Наприклад, можна визначити овочеву організацію, яка може відповідати видам овочів, доступних для придбання, з агентом продуктового магазину [10].

Контексти

Dialogflow контексти схожі на контекст природної мови. Якщо людина каже "вони оранжеві", вам потрібен контекст, щоб зрозуміти, на що вони мають на увазі. Аналогічно, щоб Dialogflow обробляв вираження кінцевого користувача таким чином, йому потрібно забезпечити контекст, щоб правильно відповідати наміру.

Використовуючи контексти, можна контролювати потік розмови. Можна налаштувати контексти для намірів, встановивши вхідні та вихідні контексти, які ідентифікуються іменами рядків. Коли інтент збігається, будь-який налаштований вихідний контекст для цього наміру стає активним. Хоча будь-які контексти активні, DialogFlow швидше співпадає з намірами, налаштованими на вхідні контексти. На наступній схемі показаний приклад, який використовує контекст для агента (рис. 4.2)

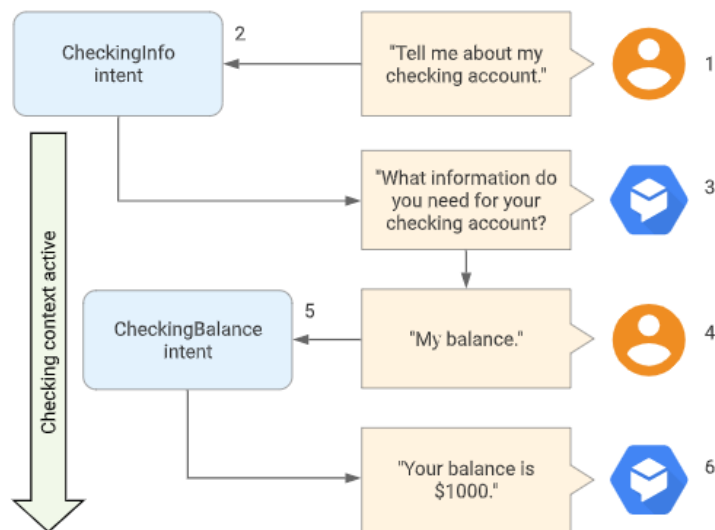


Рисунок 4.2 – Схема використання контексту для агента

Можна створити багато заздалегідь заданих наступних намірів для загальних відповідей кінцевого користувача, таких як "так", "ні" або "скасувати". Також можна створити власні подальші наміри для обробки користувацьких відповідей [24].

3.4. Налаштування WordPress® REST API

REST API або WP API розроблений на базі технології REST, щоб мати зрозумілі URL-адреси та використовувати коди HTTP відповідей для вказівки успіхів чи помилок. Використовує вбудовані функції HTTP, такі як http-автентифікація та http-команди (GET, POST), які можуть бути зрозумілі багатьма клієнтами HTTP. WP API дозволяє зручно та безпечно працювати з WordPress® із клієнтської програми. REST API може замінити спосіб обробки ajax запитів [27].

4.4. Налаштування безпеки

Безпечне введення

Щоразу, коли PHP код отримує якісь дані:

- користувач надсилає дані в WordPress® через форму.
- дані імпортуються із внутрішнього або зовнішнього джерела.
- обробляється AJAX запит.
- дані надходять у WordPress® через API.

Дуже важливо переконатись, що це безпечно. Зробити це можна шляхом перевірки та очищення даних.

Щоразу, коли заголовок запису, мета дані, чи будь-які інші дані надаються користувачеві — виводяться на екран, їх потрібно очищати. Очищення потрібне для запобігання атакам типу: Cross-Site Scripting (XSS - міжсайтовий скриптинг).

У WordPress® для такого очищення передбачено ряд функцій, всі вони починаються з префікса `esc_`, наприклад `esc_html()`.

5. РОЗРОБКА СТАРТАП-ПРОЕКТУ

5.1. Опис ідеї проекту.

Ідея проекту полягає в розробці простого та зручного для розробників пакетного рішення для забезпечення можливості озвучення тексту на веб сайті для людей з вадами зору. Рішення повинно бути простим для використання розробником та мати можливість інтеграції в CMS WordPress[®]. Особлива увага приділяється виразливості згенерованого мовлення.

5.2. Визначення характеристик проекту.

Таблиця 5.1 – Визначення характеристик ідеї проекту

№ п/п	Технікоєкономічні характеристик и ідеї	Потенційні конкуренти	Слабка сторона	Нейтральна сторона	Сильна сторона
1	Інтеграція з WordPress [®] і зручна та швидка інсталяція	Ні	Необхідність в постійному оновленні	Потрібні навички веб-програмування	Конкурента перевага та можливість гарного старту продажів
2	Підтримка української	Ні	Значна собівартість при високому трафіку	Лише жіночі голоси	Не має аналогів
3	Інтеграція с редактором WordPress [®]	Ні	Неможливість інтеграціх з іншими CMS	Потрібні навички веб-програмування	Підтримка всіх наявних редакторів
4	Виразливе озвучення	Ні	Потребує ручного наповнення сайтів	Лише жіночі голоси	Не має аналогів

5.3. Технологічний аудит ідеї проекту.

Таблиця 5.2 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Інтеграція в CMS WordPress [®]	Підтримка баз даних	Наявна	Доступна
2	Підтримка внутрішніх API WordPress [®]	Підтримка REST API	Наявна	Доступна
3	Підтримка зовнішніх API	Підтримка Google [®] Cloud API	Наявна	Доступна
4	Зручна настройка	Адмін-панель	Наявна	Доступна

5.4. Аналіз ринкових можливостей запуску стартап-проекту.

Таблиця 5.3 – Попередня характеристика потенційного ринку стартап проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1
2	Динаміка ринку (якісна оцінка)	Зростає
3	Наявність обмежень для входу (вказати характер обмежень)	Немає
4	Специфічні вимоги до стандартизації та сертифікації	Підтримка REST API

Таблиця 5.4 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)
1	Потреби демонстрації контенту для людей з вадами зору	Топ менеджери та менеджери компаній, програмісти та програмісти аутсорсери
2	Необхідність відповідати регламенту GDPR	

5.5. Розроблення маркетингової програми стартап-проекту.

Таблиця 5.5 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, сервісність	Якість
2	Дешевизна	Раціональне використання коштів	Дешевизна

Таблиця 5.6 – Визначення меж встановлення ціни

№	Рівень цін на товари замітники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	20 у.о.	25 у.о.	Високий	Нижня - 20 у.о. Верхня - 30 у.о.

Висновки

У процесі роботи було здійснено опис ідеї проекту, що полягає розробці пакетного рішення для CMS WordPress[®] для озвучування контенту на веб сайтах. Розробка зручного інтерфейсу та системи інсталяції допоможе надавати можливість слабозором споживати контент на сайтах .

ПЕРЕЛІК ПОСИЛАНЬ

1. Rabiner, L. Fundamentals of Speech Recognition / L. Rabiner, B.H. Juang. – Prentice-Hall International, Inc. - 1993.
2. Hinton, G., Deep Neural Networks for Acoustic Modeling in Speech Recognition / G. Hinton, L. Deng, D. Yu, G.E. Dahl // IEEE Signal Processing Magazine. - November 2012. - p. 82-97.
3. Вычислительный центр им. А. А. Дороницына Российской академии наук: Модели, методы, алгоритмы и архитектуры систем распознавания речи / [отв. ред. Рязанов В.В.]. - Вычислительный центр им. А. А. Дороницына. - Москва, 2006.
4. Furtuna, T. F. Dynamic Programming Algorithms in Speech Recognition [Электронне джерело] / T.F. Furtuna, // Revista Informatica Economica. - Nr. 2(46). - 2008. - p. 94-99. – Режим доступу до ресурсу: <http://revistaie.ase.ro/index.html>
5. Listen, Attend and Spell [Электронне джерело] // W. Chan, N. Jaitly, Q. V. Le, O. Vinyals – 2015 – p.16 - Режим доступу до ресурсу: <https://arxiv.org/pdf/1508.01211.pdf>
6. A Comparison of Sequence-to-Sequence Models for Speech Recognition [Электронне джерело] // R. Prabhavalkar, K. Rao, T. N. Sainath, Bo Li, L. Johnson, N. Jaitly – 2017 – p. 5 - Режим доступу до ресурсу: https://www.isca-speech.org/archive/Interspeech_2017/pdfs/0233.PDF
7. MINIMUM WORD ERROR RATE TRAINING FOR ATTENTION-BASED SEQUENCE-TO-SEQUENCE MODELS [Электронне джерело] // R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. Chiu, A. Kannan – 2017 – p. 5 Режим доступу до ресурсу: <https://arxiv.org/pdf/1712.01818.pdf>
8. Large Scale Language Modeling in Automatic Speech Recognition [Электронне джерело] // С. Chelba – 2012 - Режим доступу до ресурсу: <https://ai.Googleblog.com/2012/10/large-scale-language-modeling-in.html>

9. Мазуренко И.Л. Компьютерные системы распознавания речи / И.Л. Мазуренко // Интеллектуальные системы. – Москва, 1998. – № 1-2. - с.117-134.
10. Федосин С.А. Классификация систем распознавания речи [Электронное джерело]: / С.А. Федосин, А.Ю. Еремин. // Электронное научное издание «Электроника и информационные технологии». - 2010. – Режим доступа до ресурсу: <http://fetmag.mrsu.ru/2010-2/pdf/SpeechRecognition.pdf>
11. Large Scale Language Modeling in Automatic Speech Recognition [Электронное джерело] // С. Chelba, D. Bikel, M. Shugrina, P. Nguyen, S. Kumar - Режим доступа до ресурсу: <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/40491.pdf>
12. Quickstart: Setup [Электронное джерело]// cloud.Google.com - Режим доступа до ресурсу: <https://cloud.google.com/dialogflow/docs/quick/setup>
13. Create a Project and Dialogflow Agent [Электронное джерело] // developers.Google.com - Режим доступа до ресурсу: <https://developers.google.com/assistant/actions/dialogflow/project-agent>.
14. Dialogflow Documentation [Электронное джерело] // cloud.Google.com - Режим доступа до ресурсу: <https://cloud.google.com/dialogflow/docs>.
15. Quickstart: Build an agent [Электронное джерело] // cloud.Google.com - Режим доступа до ресурсу: <https://cloud.google.com/dialogflow/docs/quick/build-agent>.
16. Overview, Conversational Actions [Электронное джерело] // developers.Google.com - Режим доступа до ресурсу: <https://developers.google.com/assistant/discovery>.
17. Web Content Accessibility Guidelines (WCAG) Overview [Электронное джерело] // w3.org - Режим доступа до ресурсу: <https://www.w3.org/WAI/standards-guidelines/wcag/>
18. How visually impaired people navigate the web by Busayomi Oyewole [Электронное джерело] //UXCollective.com - Режим доступа до ресурсу: <https://uxdesign.cc/how-visually-impaired-people-navigate-the-web-7f9eab9d9c37>

19. WAVENET: A GENERATIVE MODEL FOR RAW AUDIO [Електронне джерело] //arxiv.org- Режим доступу до ресурсу: <https://arxiv.org/pdf/1609.03499.pdf>
20. Yoshioka, T. Making Machines Understand Us in Reverberant Rooms // T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, W. Kellerman / IEEE Signal Processing Magazine. - November 2012. - p. 114-126.
21. Lau, P. The Lombard Effect as a Communicative Phenomenon / P. Lau // UC Berkeley Phonology Lab Annual Report. – UC Berkeley. - 2008.
22. Juang, B. H. Speech recognition in adverse environments [Електронне джерело] / Juang B. H. // Computer Speech & Language. - July 1991. - No 5(3). - p. 275-294. - Режим доступу до ресурсу: https://www.researchgate.net/publication/222192250_Speech_recognition_in_adverse_environments
23. Kukharicheva, K. Quality Assessment of Automatic Speech recognition Systems / K. Kukharicheva // Innovations in Science and Technology: the XV All-Ukrainian R&D Students Conference Proceeding, (Kyiv, November 24, 2015) / National Technical University of Ukraine 'Kyiv Polytechnic Institute'. – Kyiv, 2015. – Part I. – 124 p., p. 109-111.
24. Дідковський В.С. Комп'ютерна обробка акустичних сигналів: Навчальний посібник / В.С. Дідковський, М.В. Дідковська, А.М. Продеус. – Київ, 2010. – 430 с.
25. Хайкин С. Нейронные сети: полный курс : пер. с англ. / С. Хайкин. — [2-е изд.]. — М. : Издательский дом «Вильямс». — 2010. — 1104 с.
26. The Birth Of The Vocoder And Its Use In Modern Music (With Audio Examples) [Електронне джерело] – 2020 - Режим доступу до ресурсу: <https://reaktorplayer.wordpress.com/tag/homer-dudley/>
27. Скрытая марковская модель — Википедия. [Електронне джерело] // wikipedia.org – Режим доступу до ресурсу: http://ru.wikipedia.org/wiki/Скрытая_марковская_модель

28. Attention-Based Models for Speech Recognition [Электронный ресурс] // J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio – 2015 – р.19 - Режим доступа до ресурсу: <https://arxiv.org/pdf/1506.07503.pdf>
29. Speech to text basis [Электронне джерело] // cloud.Google.com - Режим доступа до ресурсу: <https://cloud.Google.com/speech-to-text/docs/basics>
30. An All-Neural On-Device Speech Recognizer [Электронне джерело]// ai.Googleblog.com - Режим доступа до ресурсу: <https://ai.Googleblog.com/2019/03/an-all-neural-on-device-speech.html>

ДОДАТКИ

Додаток А. Код файлу voicer.php

```
?php
/**
 * Voicer
 *
 * @encoding UTF-8
 * @version 1.1
 * @copyright (C).
 * @license None
 * @contributors Alex
 * @support No
 * @license None
 *
 * @wordpress-plugin
 * Plugin Name: Voicer
 * Plugin URI: https://uttermouse.com.ua
 * Description: Чудова дипломна робота. ВПЕРШЕ Є УКРАЇНСЬКА!
 * Version: 1.1
 * Requires at least: 3.0
 * Requires PHP: 7.1
 * Alex Uttermouse
 * Author URI: uttermouse.com.ua
 * License: None
 * License URI: None
 * Text Domain: voicer
 * Domain Path: /languages
 * Tested up to: 5.8
 * Elementor tested up to: 3.9
 * Elementor Pro tested up to: 3.9
 **/

namespace Merkulove;

/** Exit if accessed directly. */
if ( ! defined( 'ABSPATH' ) ) {
header( 'Status: 403 Forbidden' );
header( 'HTTP/1.1 403 Forbidden' );
exit;}
/** Include plugin autoloader for additional classes. */
require __DIR__ . '/src/autoload.php';

use Merkulove\Voicer\Caster;
use Merkulove\Voicer\Config;
use Merkulove\Voicer\Unity\Unity;

/**
 * SINGLETON: Core class used to implement a plugin.
 *
 * This is used to define internationalization, admin-specific
hooks, and public-facing site hooks.
 *

```

```

    * @since 1.0.0
    *
    **/
final class Voicer {

/**
 * The one true Voicer.
 *
 * @var Voicer
 * @since 1.0.0
 * @access private
 **/
private static $instance;

/**
 * Sets up a new plugin instance.
 *
 * @since 1.0.0
 * @access private
 *
 * @return void
 **/
private function __construct() {

/** Initialize Unity and Main variables. */
Unity::get_instance();}
/**
 * Setup the plugin.
 *
 * @since 1.0.0
 * @access public
 *
 * @return void
 **/
public function setup() {

/** Do critical compatibility checks and stop work if fails. */
if ( ! Unity::get_instance()->initial_checks( ['php56', 'curl'] ) )
{ return; }
/** Prepare custom plugin settings. */
Config::get_instance()->prepare_settings();

/** Setup the Unity. */
Unity::get_instance()->setup();

/** Custom setups for plugin. */
Caster::get_instance()->setup();}
/**
 * Called when a plugin is activated.
 *
 * @static
 * @since 1.0.0
 * @access public

```



```

*
* @return void
**/
public static function on_activation() {

/** Call Unity on plugin activation. */
Unity::on_activation();

/** Call Voicer on plugin activation */
Caster::get_instance()->activation_hook();}
/**
* Called when a plugin is deactivated.
*
* @static
* @since 1.0.0
* @access public
*
* @return void
**/
public static function on_deactivation() {

/** MP on plugin deactivation. */
Unity::on_deactivation();}
/**
* Main Instance.
*
* Insures that only one instance of plugin exists in memory at any
one time.
*
* @static
* @since 1.0.0
*
* @return Voicer
**/
public static function get_instance() {

if ( ! isset( self::$instance ) && ! ( self::$instance instanceof
self ) ) {

self::$instance = new self;}
return self::$instance;}}
/** Run 'on_activation' when the plugin is activated. */
register_activation_hook( __FILE__, [ Voicer::class, 'on_activation'
] );

/** Run 'on_deactivation' when the plugin is deactivated. */
register_deactivation_hook( __FILE__, [ Voicer::class,
'on_deactivation' ] );

/** Run Plugin class once after activated plugins have loaded. */
add_action( 'plugins_loaded', [ Voicer::get_instance(), 'setup' ] );

```

```
<?php
/**
 * Voicer
 * Чудова дипломна робота. ВПЕРШЕ Є УКРАЇНСЬКА!
 * Exclusively on https://uttermouse.com.ua
 *
 * @encoding      UTF-8
 * @version       1.1
 * @copyright     (C).
 * @license       None
 * @contributors  Alex
 * @support       No
 **/

namespace Merkulove\Voicer;

/** Exit if accessed directly. */
if ( ! defined( 'ABSPATH' ) ) {
    header( 'Status: 403 Forbidden' );
    header( 'HTTP/1.1 403 Forbidden' );
    exit;
}
use Merkulove\Voicer\Unity\Plugin;

final class AdminScripts {

    /**
     * The one true AdminScripts.
     *
     * @var AdminScripts
     * @since 1.0.0
     */
    private static $instance;

    /**
     * Sets up a new AdminScripts instance.
     *
     * @since 1.0.0
     * @access public
     */
    private function __construct() {

        /** Add plugin scripts. */
        add_action( 'admin_enqueue_scripts', [ $this,
'admin_scripts' ] );}
    /**
     * Add plugin admin scripts.
     *
     * @since 1.0.0
     *
     * @return void
     */
}
```

```

public function admin_scripts() {

    /** Add styles only on setting page */
    $screen = get_current_screen();
    if ( null === $screen ) { return; }
    /** Add styles only on plugin settings page */
    if ( ! in_array( $screen->base, Plugin::get_menu_bases(),
true ) ) { return; }
        wp_enqueue_script( 'mdp-jquery-data-tables',
Plugin::get_url() . 'js/jquery.dataTables' . Plugin::get_suffix() .
'.js', [], Plugin::get_version(), true );}
    /**
     * Main AdminScripts Instance.
     *
     * Insures that only one instance of AdminScripts exists in
memory at any one time.
     *
     * @static
     * @return AdminScripts
     * @since 1.0.0
     */
    public static function get_instance() {

        if ( ! isset( self::$instance ) && ! ( self::$instance
instanceof self ) ) {

            self::$instance = new self;

        }
        return self::$instance;}
} // End Class AdminScripts.

```

```

<?php
/**
 * Voicer
 * Чудова дипломна робота. ВПЕРШЕ Є УКРАЇНСЬКА!
 * Exclusively on https://uttermouse.com.ua
 *
 * @encoding      UTF-8
 * @version       1.1
 * @copyright     (C).
 * @license       None
 * @contributors  Alex
 * @support       No
 **/
namespace MVoicer;
use Voicer\Unity\Tab;use Merkulove\Voicer\Unity\Plugin;use
Merkulove\Voicer\Unity\Settings;use
Merkulove\Voicer\Unity\TabGeneral;use Merkulove\Voicer\Unity\UI;
use Google@\Cloud\TextToSpeech\V1\AudioConfig;use
Google@\Cloud\TextToSpeech\V1\AudioEncoding;use
Google@\Cloud\TextToSpeech\V1\SsmlVoiceGender;use
Google@\Cloud\TextToSpeech\V1\SynthesisInput;use
Google@\Cloud\TextToSpeech\V1\TextToSpeechClient;use
Google@\Cloud\TextToSpeech\V1\VoiceSelectionParams;
/** Exit if accessed directly. */
if ( ! defined( 'ABSPATH' ) ) {    header( 'Status: 403 Forbidden'
);    header( 'HTTP/1.1 403 Forbidden' );    exit;}/**
 * SINGLETON: Settings class used to modify default plugin settings.
 *
 * @since 1.0.0
 *
 **/
final class Config extends Tab {
    /**
     * The one true Settings.
     *
     * @since 1.0.0
     * @access private
     * @var Config
     **/
    private static $instance; /**
     * Google@ languages
     * @var \string[][]
     */
    public static $languages = [
        'da-DK' => 'Danish (Dansk)',
        'nl-NL' => 'Dutch (Nederlands)',
        'en-AU' => 'English (Australian)',
        'en-GB' => 'English (UK)',
        'en-US' => 'English (US)',
        'fr-CA' => 'French Canada (Français)',
        'fr-FR' => 'French France (Français)',
        'de-DE' => 'German (Deutsch)',
    ]
}

```

```

        'it-IT' => 'Italian (Italiano)',
'ja-JP' => 'Japanese (日本語)',
'ko-KR' => 'Korean (한국어)',
        'nb-NO' => 'Norwegian (Norsk)',
        'pl-PL' => 'Polish (Polski)',
        'pt-BR' => 'Portuguese Brazil (Português)',
        'pt-PT' => 'Portuguese Portugal (Portugal)',
        'ru-RU' => 'Russian (Русский)',
        'sk-SK' => 'Slovak (Slovenčina)',
        'es-ES' => 'Spanish (Español)',
        'sv-SE' => 'Swedish (Svenska)',
        'tr-TR' => 'Turkish (Türkçe)',
        'uk-UA' => 'Ukrainian (Українська)',
        'ar-XA' => 'Arabic (العربية)',
        'cs-CZ' => 'Czech (Čeština)',
        'el-GR' => 'Greek (Ελληνικά)',
        'en-IN' => 'Indian English',
        'fi-FI' => 'Finnish (Suomi)',
        'vi-VN' => 'Vietnamese (Tiếng Việt)',
        'id-ID' => 'Indonesian (Bahasa Indonesia)',
        'fil-PH' => 'Philippines (Filipino)',
'hi-IN' => 'Hindi (हिन्दी)',
        'hu-HU' => 'Hungarian (Magyar)',
'cmn-CN' => 'Chinese (官话)',
'cmn-TW' => 'Taiwanese Mandarin (中文(台灣))',
'bn-IN' => 'Bengali (বাংলা)',
'gu-IN' => 'Gujarati (ગુજરાતી)',
'kn-IN' => 'Kannada (ಕನ್ನಡ)',
'ml-IN' => 'Malayalam (മലയാളം)',
'ta-IN' => 'Tamil (தமிழ்)',
'tel-IN' => 'Telugu (తెలుగు)',
        'th-TH' => 'Thai (ภาษาไทย)',
        'yue-HK' => 'Yue Chinese',
        'ro-RO' => 'Romanian (Română)',
        'ca-ES' => 'Catalan (Català)',
        'af-ZA' => 'Afrikaans (South Africa)',
        'bg-BG' => 'Bulgarian (Български)',
        'lv-LV' => 'Latvian (Latvietis)',
        'sr-RS' => 'Serbian (Српски)',
        'is-IS' => 'Icelandic (Íslensk)',
        'es-US' => 'US Spanish (Hispanoamericano)',
        'ms-MY' => 'Malay (Malaysia)',
        'nl-BE' => 'Dutch (Belgium)',
        'pa-IN' => 'Punjabi (India)'
]; /**
 * Prepare plugin settings by modifying the default one.
 *
 * @since 1.0.0
 * @access public
 *
 * @return void

```

```

    **/
    public function prepare_settings() {
        /** Reset API Key on fatal error. */
        if ( isset( $_GET['reset-api-key'] ) && '1' ===
$_GET['reset-api-key'] ) {
            $this->reset_api_key();
        }
        /** Get default plugin settings. */
        $tabs = Plugin::get_tabs();
        /** Remove 'Delete plugin, settings and data' option from
Uninstall tab. */
        unset(
        $tabs['uninstall']['fields']['delete_plugin']['options']['plugin+set
tings+data'] );
        /** Set System Requirements. */
        $tabs['status']['reports']['server']['allow_url_fopen'] =
false;
        $tabs['status']['reports']['server']['dom_installed']
= true;
        $tabs['status']['reports']['server']['xml_installed']
= true;
        $tabs['status']['reports']['server']['bcmath_installed'] = true;
        $tabs['status']['reports']['server']['mbstring_installed'] = true;
        $tabs['status']['reports']['server']['server_time'] = true;
        /** Short hand access to plugin settings. */
        $options = Settings::get_instance()->options;
        if ( isset( $options['dnd-api-key'] ) && $options['dnd-api-
key'] ) {
            $tabs[ 'general' ][ 'fields' ][ 'multi' ] = [
                'type' => 'switcher',
                'label' => esc_html__( 'Multilingual
support', 'voicer' ),
                'placeholder' => esc_html__( 'Multilingual
website', 'voicer' ),
                'description' => esc_html__( 'The Site locale
will be used to generate speech. Standard-A voice will be used.',
'voicer' ),
                'default' => 'off',
            ];
            $tabs[ 'general' ][ 'fields' ][ 'current_language' ] = [
                'type' => 'custom',
                'render' => [ self::get_instance(),
'render_current_language' ],
                'label' => esc_html__( 'Now used:', 'voicer' ),
                'default' => '',
            ];
            $tabs[ 'general' ][ 'fields' ][ 'language' ] = [
                'type' => 'custom',
                'render' => [ self::get_instance(),
'render_language' ],
                'label' => esc_html__( 'Select Language:', 'voicer'
),
                'default' => 'en-US-Standard-B',
            ];
            $tabs[ 'general' ][ 'fields' ][ 'language-code' ] = [ //
Hidden language code input
                'type' => 'text',

```

```

        'placeholder'      => esc_html__( 'Language Code',
'voicer' ),
        'default'          => 'en-US',
        'attr'             => [
            'id'           => 'mdp-voicer-general-settings-
language-code',
        ]
    ];
    $tabs[ 'general' ][ 'fields' ][ 'audio-profile' ] = [
        'type' => 'select',
        'label' => esc_html__( 'Audio Profile', 'voicer' ),
        'placeholder' => esc_html__( 'Audio Profile',
'voicer' ),
        'description' => esc_html__( 'Optimize the synthetic
speech for playback on different types of hardware.', 'voicer' ),
        'default' => 'handset-class-device',
        'options' => [
            'wearable-class-device' => esc_html__( 'Smart
watches and other wearables', 'voicer' ),
            'handset-class-device' => esc_html__(
'Smartphones', 'voicer' ),
            'headphone-class-device' => esc_html__( 'Earbuds
or headphones', 'voicer' ),
            'small-bluetooth-speaker-class-device' =>
esc_html__( 'Small home speakers', 'voicer' ),
            'medium-bluetooth-speaker-class-device' =>
esc_html__( 'Smart home speakers', 'voicer' ),
            'large-home-entertainment-class-device' =>
esc_html__( 'Home entertainment systems', 'voicer' ),
            'large-automotive-class-device' => esc_html__(
'Car speakers', 'voicer' ),
            'telephony-class-application' => esc_html__(
'Interactive Voice Response', 'voicer' ),
        ]
    ];
    $key = 'speaking-rate';          $default = 1;
    $tabs[ 'general' ][ 'fields' ][ $key ] = [
        'type' => 'slider',
        'label' => esc_html__( 'Speaking Rate/Speed:',
'voicer' ),
        'description' => esc_html__( 'Speaking rate:',
'voicer' ) .
            ' <strong>' .
            esc_html( ( isset( $options[ $key ] ) ) ) ?
    $options[ $key ] : $default ) .
            '</strong>',
        'min' => 0.1,
        'max' => 4,
        'step' => 0.1,
        'default' => $default,
        'discrete' => false,
    ];

```

```

        $key = 'pitch';                $default = 0;
$tabs[ 'general' ][ 'fields' ][ $key ] = [
    'type' => 'slider',
    'label' => esc_html__( 'Pitch:', 'voicer' ),
    'description' => esc_html__( 'Current pitch:',
'voicer' ) .
        ' <strong>' .
        esc_html( ( isset( $options[ $key ] ) ) ?
$options[ $key ] : $default ) .
        '</strong>',
    'min' => -20,
    'max' => 20,
    'step' => 1,
    'default' => $default,
    'discrete' => true,
];
$tabs[ 'general' ][ 'fields' ][ 'use' ] = [
    'type'                => 'switcher',
    'label'                => esc_html__( 'Enable for
site', 'voicer' ),
    'placeholder'        => esc_html__( 'Use
everywhere', 'voicer' ),
    'description'        => esc_html__( 'In the off
mode, the Voicer will only work for blocks with CSS classes',
'voicer' ) . ' <br/><code>voicer-voice-voicename</code> ' .
        esc_html__( 'Learn more
about this from the' ) . ' <a href="https://uttermouse.com/"
target="_blank">' . esc_html__( 'Online Documentation' ) . '</a>',
    'default'            => 'on',
];
}
$tabs['general']['fields']['dnd-api-key'] = [
    'type'                => 'custom',
    'render'              => [ self::get_instance(),
'render_dnd_api_key' ],
    'label'                => esc_html__( 'API Key File:',
'voicer' ),
    'default'              => '',
];
/** Design Tab */

$offset = 1;                $tabs = array_slice( $tabs, 0, $offset,
true ) +
    ['design' => [
        'enabled'        => true,
        'class'          => TabGeneral::class, // Handler
        'label'          => esc_html__( 'Design', 'voicer' ),
        'title'          => esc_html__( 'Design Settings',
'voicer' ),
        'show_title'    => true,
        'icon'           => 'palette', // Icon for tab
        'fields'        => []
    ] ] +

```



```

        array_slice( $tabs, $offset, NULL, true );
$backend_options = is_array( get_option(
'mdp_stellar_design_settings' ) );
        $key = 'radius';           $default = 100;           $value =
$this->get_slider_value( 'design', $key );
$tabs['design']['fields'][$key] = [
        'type'                       => 'slider',
        'label'                       => esc_html__( 'Border radius:',
'voicer' ),
        'description'                 => esc_html__( 'Border radius:',
'voicer' ) .
        ' <strong>' .
        esc_html( is_string( $value ) ? $value : $default )
        .
        '</strong>' .
        esc_html__( ' px', 'voicer' ),
        'min'                         => 1,
        'max'                         => 100,
        'step'                        => 1,
        'default'                     => $default,
        'discrete'                   => true,
];
$tabs['design']['fields']['bgcolor'] = [
        'type'                       => 'colorpicker',
        'label'                       => esc_html__( 'Background Color',
'voicer' ),
        'placeholder'                => esc_html__( 'Background Color',
'voicer' ),
        'description'                 => esc_html__( 'Select tooltip
background-color', 'voicer' ),
        'default'                     => '#0253ee',
        'attr'                        => [
                'readonly'           => 'readonly',
        ]
];
$tabs['design']['fields']['iconcolor'] = [
        'type'                       => 'colorpicker',
        'label'                       => esc_html__( 'Icon Color',
'voicer' ),
        'placeholder'                => esc_html__( 'Icon Color',
'voicer' ),
        'description'                 => esc_html__( 'Select the Icon
color', 'voicer' ),
        'default'                     => '#ffffff',
        'attr'                        => [
                'readonly'           => 'readonly',
        ]
];
$tabs['design']['fields']['play_icon'] = [
        'type'                       => 'icon',
        'label'                       => esc_html__( 'Play icon', 'voicer'
),
        'placeholder'                => '',

```

```

        'description'      => esc_html__( 'Select Play Icon.',
'voicer' ),
        'default'         => 'font-awesome/play.svg',
        'meta'            => [
            'font-awesome.json',
            'material.json'
        ]
    ];
    $tabs['design']['fields']['stop_icon'] = [
        'type'             => 'icon',
        'label'            => esc_html__( 'Stop icon', 'voicer'
),
        'placeholder'     => '',
        'description'     => esc_html__( 'Select Stop Icon.',
'voicer' ),
        'default'         => 'font-awesome/stop.svg',
        'meta'            => [
            'font-awesome.json',
            'material.json'
        ]
    ];
    $key = 'icon_size';      $default = 16;      $value =
$this->get_slider_value( 'design', $key );
    $tabs['design']['fields'][$key] = [
        'type'             => 'slider',
        'label'            => esc_html__( 'Icon size:',
'voicer' ),
        'description'     => esc_html__( 'Icons size:',
'voicer' ) .
            ' <strong>' .
            esc_html( is_string( $value ) ? $value : $default )
            .
            '</strong>' .
            esc_html__( ' px', 'voicer' ),
        'min'             => 1,
        'max'              => 50,
        'step'             => 1,
        'default'         => $default,
        'discrete'        => true,
    ];
    $key = 'icon_margin';   $default = 16;      $value =
$this->get_slider_value( 'design', $key );
    $tabs['design']['fields'][$key] = [
        'type'             => 'slider',
        'label'            => esc_html__( 'Icon margin:',
'voicer' ),
        'description'     => esc_html__( 'Icons margin:',
'voicer' ) .
            ' <strong>' .
            esc_html( is_string( $value ) ? $value : $default )
            .
            '</strong>' .
            esc_html__( ' px', 'voicer' ),

```

```

        'min'                => 1,
        'max'                => 50,
        'step'               => 1,
        'default'            => $default,
        'discrete'           => true,
    ];
    /** Set updated tabs. */
    Plugin::set_tabs( $tabs );
    /** Refresh settings. */
    Settings::get_instance()->get_options();{    /**
 * Get values for slider
 * @param $stab
 * @param $key
 *
 * @return false|mixed
 */
private function get_slider_value( $stab, $key )
{
    $stab_settings = get_option( 'mdp_voicer_' . $stab .
'_settings', false );
    return ( is_array( $stab_settings ) && array_key_exists(
$key, $stab_settings ) ) ?
        $stab_settings[ $key ] :
        false;{    /**
 * Render Now used field.
 *
 * @param string $key - Field key.
 * @param string $stab_slug - Tab slug to which the field
belongs.
 *
 * @since 1.0.0
 * @access public
 *
 * @return void
 *
 * @noinspection PhpUnused
 */
public function render_current_language( $key, $stab_slug ) {
    /** Shorthand for options. */
    $options = Settings::get_instance()->options;
    ?>
    <div class="mdp-now-used">
        <div>
            <strong><?php echo esc_attr( $options['language'] );
?></strong>
        </div>
        <div>
            <audio controls="">
                <source src="https://cloud.Google.com/text-to-
speech/docs/audio/<?php esc_attr_e( $options['language'] ); ?>.mp3"
type="audio/mp3">
                <source src="https://cloud.Google.com/text-to-
speech/docs/audio/<?php esc_attr_e( $options['language'] ); ?>.wav"
type="audio/wav">

```

```

        <?php esc_html_e( 'Your browser does not support
the audio element.', 'voicer' ); ?>
    </audio>
    </div>
</div>
<?php
{ /** @noinspection PhpUnused */
/**
 * Render Drag & Drop API Key field.
 *
 * @param string $key - Field key.
 * @param string $tab_slug - Tab slug to which the field
belongs.
 *
 * @since 3.0.0
 * @access public
 */
public function render_dnd_api_key( $key, $tab_slug ) {
    $key_exist = false; if ( array_key_exists( 'dnd-api-
key', Settings::get_instance()->options ) && strlen(
Settings::get_instance()->options['dnd-api-key'] ) > 0 ) {
$key_exist = true; } ?>
    <div class="mdp-dnd">
        <!--suppress HtmlFormInputWithoutLabel -->
        <div class="mdc-text-field mdc-input-width mdc-text-
field--outlined mdc-hidden">
            <!--suppress HtmlFormInputWithoutLabel -->
            <input type="text"
                class="mdc-text-field__input"
                name="mdp_voicer_general_settings[dnd-api-
key]"
                id="mdp-voicer-general-settings-dnd-api-key"
                value="<?php esc_attr_e(
Settings::get_instance()->options['dnd-api-key'] ); ?>"
            >
            <div class="mdc-notched-outline mdc-notched-outline-
-upgraded mdc-notched-outline--notched">
                <div class="mdc-notched-outline__leading"></div>
                <div class="mdc-notched-outline__notch">
                    <label for="mdp-voicer-general-settings-dnd-
api-key" class="mdc-floating-label mdc-floating-label--float-
above"><?php esc_html_e( 'API Key', 'voicer' ); ?></label>
                </div>
                <div class="mdc-notched-
outline__trailing"></div>
            </div>
        </div>
        <div id="mdp-api-key-drop-zone" class="<?php if (
$key_exist ) : ?>mdp-key-uploaded<?php endif; ?>">
            <?php if ( $key_exist ) : ?>
                <span class="material-
icons">check_circle_outline</span><?php esc_html_e( 'API Key file
exist', 'voicer' ); ?>

```

```

        <span class="mdp-drop-zone-hover"><?php
esc_html_e( 'Drop Key file here or click to upload', 'voicer' );
?></span>
        <?php else : ?>
            <span class="material-icons">cloud</span><?php
esc_html_e( 'Drop Key file here or click to upload.', 'voicer' ); ?>
            <?php endif; ?>
        </div>
        <?php if ( $key_exist ) : ?>
            <div class="mdp-messages mdc-text-field-helper-line
mdc-text-field-helper-text mdc-text-field-helper-text--persistent">
                <?php esc_html_e( 'Drag and drop or click on the
form to replace API key. |', 'voicer' ); ?>
                <a href="#" class="mdp-reset-key-btn"><?php
esc_html_e( 'Reset API Key', 'voicer' ); ?></a>
            </div>
        <?php else : ?>
            <div class="mdp-messages mdc-text-field-helper-line
mdc-text-field-helper-text mdc-text-field-helper-text--persistent">
                <?php esc_html_e( 'Drag and drop or click on the
form to add API key.', 'voicer' ); ?>
            </div>
        <?php endif; ?>
        <input id="mdp-dnd-file-input" type="file" name="name"
class="mdc-hidden" />
    </div>
    <?php
{ /**
 * Render settings current_language field.
 *
 * @since 2.0.0
 * @access public
 *
 * @return void
 */
public function render_language() {
    /** Setting custom exception handler. */
    set_exception_handler( [ ErrorHandler::class,
'exception_handler' ] );
    /** Includes the autoloader for libraries installed with
Composer. */
    require Plugin::get_path() . 'vendor/autoload.php';
    /** Create client object. */
    $client = new TextToSpeechClient();
    /** Perform list voices request. */
    $response = $client->listVoices();           $voices =
$response->getVoices();
    if ( count( $voices ) === 0 ) {                ?>
        <div class="mdp-key-error"><?php esc_html_e( 'Failed to
get the list of languages. The request failed. It looks like a
problem with your API Key File. Make sure that you are using the
correct key file, and that the quotas have not been exceeded. If you

```

```

set security restrictions on a key, make sure that the current
domain is added to the exceptions.', 'voicer' ); ?></div><?php
        return;          }          /** Prepare Languages Options.
*/
        $options = [];          $options[] = esc_html__( 'Select
Language', 'voicer' );          foreach ( $voices as $voice ) {
$lang = $this->get_lang_by_code( $voice->getLanguageCodes() );
$options[$lang] = $lang;          }          /** Render Language select.
*/

        UI::get_instance()->render_select(
            $options,
            '',
            esc_html__( 'Language', 'voicer' ),
            '',
            [
                'name' => 'mdp_voicer_language_filter',
                'id' => 'mdp-voicer-language-filter'
            ]
        );
        /** Shorthand for plugin settings. */
        $options = Settings::get_instance()->options;
        ?>

        <div class="mdc-text-field-helper-line mdp-voicer-helper-
padding">
            <div class="mdc-text-field-helper-text mdc-text-field-
helper-text--persistent"><?php esc_html_e( 'The list includes both
standard and', 'voicer' ); ?>
                <a href="https://cloud.Google.com/text-to-
speech/docs/wavenet"
                    target="_blank"><?php esc_html_e( 'WaveNet
voices', 'voicer' ); ?></a>.
                <?php esc_html_e( 'WaveNet voices are higher quality
voices with different', 'voicer' ); ?>
                <a href="https://cloud.Google.com/text-to-
speech/pricing"
                    target="_blank"><?php esc_html_e( 'pricing',
'voicer' ); ?></a>;          <?php esc_html_e( 'in the list,
they have the voice type "WaveNet".', 'voicer' ); ?>
            </div>
        </div>

        <table id="mdp-voicer-settings-language-tbl" class="display
stripe hidden">
            <thead>
                <tr>
                    <th><?php esc_html_e( 'Language', 'voicer' );
?></th>
                    <th><?php esc_html_e( 'Voice', 'voicer' ); ?></th>
                    <th><?php esc_html_e( 'Gender', 'voicer' ); ?></th>
                </tr>
            </thead>
            <tbody>

```

```

        <?php foreach ( $voices as $voice ) : ?>
            <tr <?php if ( $voice->getName() ===
$options['language'] ) { echo 'class="selected"'; } ?>>
                <td class="mdp-lang-name">
                    <?php echo esc_html( $this-
>get_lang_by_code( $voice->getLanguageCodes() ) ); // Language. ?>
                    </td>
                <td>
                    <span class="mdp-lang-code" title="<?php
echo esc_html( $voice->getLanguageCodes()[0] ); // Language Code
?>"><?php echo esc_html( $voice->getLanguageCodes()[0] ); //
Language Code ?></span>
                    -
                    <span><?php echo $this->get_voice_type(
$voice->getName() ); //Voice Type ?></span>
                    -
                    <span class="mdp-voice-name" title="<?php
echo esc_html( $voice->getName() ); // Voice name ?>"><?php echo
esc_html( substr( $voice->getName(), -1 ) ); // Voice Variant
?></span>
                </td>
                <td>
                    <?php
                    /** SSML voice gender values from
TextToSpeech\V1\SsmlVoiceGender. */
                    $ssmlVoiceGender = [
'SSML_VOICE_GENDER_UNSPECIFIED', 'Male', 'Female', 'Neutral' ];
                    echo '<span title="" . esc_attr(
$ssmlVoiceGender[ $voice->getSsmlGender() ], 'voicer' ) . "'><img
src="" . Plugin::get_url() . 'images/' . strtolower(
$ssmlVoiceGender[ $voice->getSsmlGender() ] ) . '.svg" alt="" .
esc_attr( $ssmlVoiceGender[ $voice->getSsmlGender() ], 'voicer' ) .
"'>' . esc_html( $ssmlVoiceGender[ $voice->getSsmlGender() ] ) .
'</span>'; ?>
                </td>
            </tr>
        <?php endforeach;
        $client->close();
        ?>
    </tbody>
</table>

    <input id="mdp-voicer-general-settings-language"
type='hidden' name='mdp_voicer_general_settings[language]'
value='<?php echo esc_attr( $options['language'] );
?>'>

    <?php

    /** Restore previous exception handler. */
    restore_exception_handler();{ /**
* Return Voice Type.
*

```

```

* @param $lang_name - Google© voice name.
*
* @return string
* @since 2.0.0
* @access public
**/
public function get_voice_type( $lang_name ) {
    $pos = strpos( $lang_name, 'Wavenet' );
    if ( false === $pos ) {
        return '' .
esc_html( 'Standard' );
    }
    return '' . esc_html(
'WaveNet' );{ /**
* Return language name by code.
*
* @param $lang_code - Google© language code.
*
* @return mixed
* @since 2.0.0
* @access public
*/
public function get_lang_by_code( $lang_code ) {
    if ( is_object( $lang_code ) ) {
        $lang_code =
$lang_code[0];
    }
    $languages = self::$languages;
    /** If we inside, this means that a new language has added.
*/
    if ( ! array_key_exists( $lang_code, $languages ) ) {
        /** Render "New Languages" message. */
        self::show_new_languages_snackbar();
        return false;
    }
    return $languages[ $lang_code ];{ /**
* Render "New Languages" message.
*
* @return string
* @access public
*
* @return void|null
**/
private static function show_new_languages_snackbar()
{
    /** Run only Once. */
    static $already_run = false;
    if ( $already_run ===
true ) { return; }
    $already_run = true;
    UI::get_instance()->render_snackbar( //Render "New
Languages" message
        esc_html__( 'There are new languages available. Update
your copy of the Voicer plugin to ensure the best compatibility.',
'voicer' ),
        'info', // Type
        -1, // Timeout
        true, // Is Closable
        [ [ 'caption' => 'Check Updates', 'link' =>
'https://uttermouse.com.ua' ] ] // Buttons
    );{ /**

```



```

* Reset API Key on fatal error.
*
* @since 2.1.0
* @access public
* @return void
**/
private function reset_api_key() {
    /** Remove API Key. */
    $options = get_option( 'mdp_voicer_general_settings' );
$options['dnd-api-key'] = '';
    /** Save new value. */
    update_option( 'mdp_voicer_general_settings', $options );
    /** Go to first tab. */
    wp_redirect( admin_url(
'/admin.php?page=mdp_voicer_settings&tab=general' ) );      exit;
}    /**
    * Main Settings Instance.
    * Insures that only one instance of Settings exists in memory
at any one time.
    *
    * @static
    * @since 1.0.0
    * @access public
    *
    * @return Config
    **/
    public static function get_instance() {
        if ( ! isset( self::$instance ) && ! ( self::$instance
instanceof self ) ) {
            self::$instance = new self;
        }
        return self::$instance;
    }
}

```

Додаток Г. Код файлу XMLHelper.php

```

<?php
/**
 * Voicer
 * Чудова дипломна робота. ВПЕРШЕ Є УКРАЇНСЬКА!
 * Exclusively on https://uttermouse.com.ua
 *
 * @encoding      UTF-8
 * @version       1.1
 * @copyright     (C).
 * @license       None
 * @contributors  Alex
 * @support       No
 **/

namespace Merkulove\Voicer;
/** Exit if accessed directly. */
if ( ! defined( 'ABSPATH' ) ) {
    header( 'Status: 403 Forbidden' ); header( 'HTTP/1.1 403
Forbidden' ); exit;}
use DOMElement;
/**
 * SINGLETON: Class used to implement work with HTML and XML files.
 *
 * @since 2.0.0
 **/
final class XMLHelper {

    /**
     * The one true XMLHelper.
     *
     * @var XMLHelper
     * @since 2.0.0
     **/
    private static $instance;
    /**
     * Sets up a new XMLHelper instance.
     *
     * @since 2.0.0
     * @access public
     **/
    private function __construct() {

    }
    /**
     * The easiest way to get inner HTML of the node.
     *
     * @param DOMElement $node - Return inner html.
     *
     * @return string
     * @since 2.0.0
     * @access public
     */
}

```

```

public function get_inner_html( $node ) {
    $innerHTML= '';          $children = $node->childNodes;
    foreach ( $children as $child ) {
        $innerHTML .= $child->ownerDocument->saveXML( $child
);
    }
    return $innerHTML; }
/**
 * Prepare HTML for Google© TTS. Remove unnecessary html tags.
 *
 * @param $post_content - Post/Page content.
 *
 * @return string|string[]|null
 * @since 2.0.0
 * @access public
 **/
public function clean_html( $post_content ) {

    /** Strip Tags except contents tags and SSML. */
    $post_content = strip_tags( $post_content,
'<div><p><pre><ul><ol><li><table><span><i><b><strong><em><code><brea
k><say-as><sub><emphasis><voice>' );
    /** Remove inline styles. */
    $post_content = preg_replace( '/(<[^>]+) style=".*?"/i',
'$1', $post_content );
    /** Decoding HTML entities. */
    $post_content = html_entity_decode( $post_content );
    /** Remove empty tags. */
    $pattern      = "/<[^\\/>]*>([\s?]*)<\/[^>]*>/"; // Pattern
to remove any empty tag.
    $post_content = preg_replace( $pattern, '', $post_content
);

    /** Remove spaces, tabs, newlines. */
    $post_content = preg_replace( '~>\s+<~', '> <',
$post_content );
    return $post_content;
}
/**
 * Main XMLHelper Instance.
 *
 * Insures that only one instance of XMLHelper exists in memory
at any one time.
 *
 * @static
 * @return XMLHelper
 * @since 2.0.0
 **/
public static function get_instance() {
    if ( ! isset( self::$instance ) && ! ( self::$instance
instanceof XMLHelper ) ) {
        self::$instance = new XMLHelper;
    }
    return self::$instance; }
} // End Class XMLHelper.

```

Додаток Д. Код файлу admin.saass

```

/** * Voicer * Чудова дипломна робота. ВПЕРШЕ Є УКРАЇНСЬКА! *
Exclusively on https://uttermouse.com.ua * * @encoding UTF-8 *
@version 1.1 * @copyright (C). * @license None * @contributors Alex
* @support No */ $table-border: 1px solid rgba(2,116,230,.25) //
Voice choice table #mdp-voicer-settings-language-tbl border-
collapse: separate border-spacing: 0 12px width: 100% !important
clear: both max-width: none !important font-family: "Roboto", sans-
serif tbody border: none tr border-radius: 8px background: white td
border-top: $table-border border-bottom: $table-border &:first-of-
type border-radius: 8px 0 0 8px border-left: $table-border &:last-
of-type border-radius: 0 8px 8px 0 border-right: $table-border span
background: white border-radius: 4px padding: 4px 6px font-size:
11px border: 1px solid #ccc white-space: nowrap font-weight: 500
letter-spacing: .01rem display: inline-block margin: 2px 0 img
margin-right: 4px width: 10px height: 10px vertical-align: middle
margin-top: -2px &.mdp-voice-name white-space: nowrap &:hover box-
shadow: 0 0 8px 0 rgba(2,116,230,.08) transition: .4s background:
rgba(2,116,230,.08) td border-color: rgba(2,116,230,.12) tr.selected
background-color: #0253ee color: white span background:
rgba(10,33,56,.6) border-color: rgba(10,33,56,.6) thead background-
color: rgba(0, 0, 0, .04) border-radius: 8px th padding: 10px 14px
width: auto !important &:first-of-type border-radius: 8px 0 0 8px
&:last-of-type border-radius: 0 8px 8px 0 &:focus outline: none
table.display thead > tr > th.sorting_asc, table.display thead > tr
> th.sorting_desc, table.display thead > tr > th.sorting,
table.display thead > tr > td.sorting_asc, table.display thead > tr
> td.sorting_desc, table.display thead > tr > td.sorting padding-
right: 30px .dataTables_wrapper .dataTables_length margin-bottom:
10px table.dataTable tbody td:first-of-type, th:first-of-type font-
weight: 500 padding-left: 1.5rem table.dataTable tfoot td,
table.dataTable tfoot th padding: 10px 8px .mdp-key-file-fld
display: block overflow: hidden height: 0 transition: all 0.5s ease-
in-out .mdp-show-btn position: absolute text-decoration: none font-
size: 12px text-transform: uppercase border-bottom: 1px dashed
&.active height: 60px .mdp-show-btn display: none #mdp-voicer-
settings-language-tbl_wrapper position: relative #mdp-voicer-
settings-language-tbl_filter display: none thead width: 100% tr
width: 100% th font-size: 12px font-weight: 500 tbody width: 100% *
cursor: pointer .dataTables_length position: absolute bottom: 0 //
Below are styles by MD .mdp-voicer-helper-padding padding-bottom:
14px .mdp-upload div display: flex align-items: center .mdp-key-
file-name margin-left: 16px font-weight: bold button height: 54px
padding: 0 16px font-size: .9rem border-radius: 8px .mdp-now-used
display: flex align-items: stretch background-color: #f1f3f4 border-
radius: 8px width: fit-content div display: flex align-items: center
&:first-of-type background: #23282d border-radius: 8px 0 0 8px
padding: 0 1.25rem audio:focus outline: none strong color: white
font-family: "Roboto", sans-serif font-weight: 500 .mdp-key-error
background: #fff1f0 color: #d85030 border: 1px solid rgba(216, 80,
48, .3) padding: 10px border-radius: 4px text-shadow: 0 1px 0 #fff
.mdp-lang-name padding: 1.25rem 1rem !important .mdp-plugin-version
font-size: 0.8rem padding-left: 20px /* Responsiveness */ @media

```

```

screen and (max-width: 782px) #mdp-voicer-settings-language-tbl
tbody tr td border-top: none border-bottom: none border-left:
$stable-border border-right: $stable-border padding: 0 10px &:first-
of-type border-top: $stable-border border-radius: 8px 8px 0 0
padding: 10px 10px 4px 10px !important &:last-of-type border-bottom:
$stable-border border-radius: 0 0 8px 8px padding: 0 10px 10px 10px
// Drag & Drop JSON KEY #mdp-api-key-drop-zone border: 2px dashed
#d3d7db border-radius: 8px text-align: center color: #a5a5a5 cursor:
pointer position: relative display: flex flex-wrap: wrap align-
items: center align-content: center justify-content: center height:
80px transition: 1s &:hover height: 160px transition: 1s border: 2px
dashed lighten( #0274e6, 20% ) &.mdp-key-uploaded width: 100%
height: 80px padding: 0 border: 1px solid #d3d7db transition: 1s
color: #0274e6 -webkit-font-smoothing: antialiased .mdp-drop-zone-
hover display: none opacity: 0 color: rgb(170, 170, 170) width: 100%
font-size: 13px margin-top: 12px transition: 1s &:hover, &:focus
width: 100% height: 160px border: 2px dashed lighten( #0274e6, 20% )
justify-content: center justify-items: center color: rgb(170, 170,
170) .mdp-drop-zone-hover display: block opacity: 1 &.mdp-hover
width: 100% height: 200px border: 2px dashed #d3d7db transition: 1s
color: rgb(170, 170, 170) align-items: center justify-content:
center .mdp-drop-zone-hover display: block opacity: 1 &.mdp-busy
pointer-events: none &:before content: ' ' display: block position:
absolute left: -2px top: -2px bottom: -2px right: -2px background:
#fff opacity: 0.7 z-index: 1 &:after content: 'refresh' font-size:
22px color: rgba(10,33,56,.6) font-family: 'Material Icons' font-
weight: normal font-style: normal line-height: 1 letter-spacing:
normal text-transform: none display: block position: absolute top:
50% left: 50% width: 22px height: 22px margin-left: -11px margin-
top: -11px z-index: 1 animation: mdp-rotate 2s infinite linear .mdp-
dnd .material-icons margin: 0 8px 0 16px font-size: 18px .mdp-
messages font-size: 12px &.mdp-error color: #ff495c &.mdc-text-
field-helper-line display: block // Reset button spin animation.
.mdp-spin .material-icons animation: mdp-rotate 2s infinite linear
@keyframes mdp-rotate 0% transform: rotate(0) 100% transform:
rotate(359deg) // Hide language code .mdp-tab-name-general table
tr:nth-last-child(6) display: none // Code .mdc-switcher-helper code
font-size: 11px color: black border-radius: 8px line-height: 2

```