

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
«___» _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

**Тема: «Розширення для автоматичного перекладу та озвучення
субтитрів на YouTube»**

Виконавець:

Владислав ЗЕЛІНСЬКИЙ

Керівник:

к.т.н., доцент Олег ЗУДОВ

Нормоконтролер:

к.т.н., доцент Олена ТОЛСТІКОВА

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних наук та технологій*

Кафедра *комп'ютерних інформаційних технологій*

Спеціальність *122 «Комп'ютерні науки»*

Освітньо-професійна програма *«Інформаційні технології проектування»*

ЗАТВЕРДЖУЮ:
завідувач кафедри КІТ

Аліна САВЧЕНКО

(підпис)

«__» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Зелінського Владислава Олександровича

(ПІБ випускника)

1. Тема роботи: «Розширення для автоматичного перекладу та озвучення субтитрів на YouTube» затверджена наказом ректора № 623/ст від 01.05.2023р.

2. Термін виконання роботи: з 15 травня 2023 року по 25 червня 2023 року.

3. Вихідні дані до роботи: розширення для перекладу та озвучення субтитрів на платформі YouTube.

4. Зміст пояснювальної записки: 1. Огляд предметної області. 2. Програмна реалізація розширення для браузера. 3. Тестування розширення.

5. Перелік обов'язкового ілюстративного матеріалу: 1. Історія відео-хостингу YouTube. 2. Проблема доступності іноземних відео. 3. Методи вирішення проблеми. 4. Вибір та обґрунтування інструментів проектування. 5. Методи створення розширення. 6. Розробка серверної частини додатку. 7. Розробка клієнтської частини. 8. Тестування та перевірка.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Огляд та аналіз предметної області. Написання 1 розділу, представлення керівнику.	15.05.2023- 20.05.2023	
2	Вибір та опис використаних технологій. Розробка розширення. Написання 2 розділу, представлення керівнику.	21.05.2023- 27.05.2023	
3	Написання 3 розділу, представлення керівнику.	28.05.2023- 04.06.2023	
4	Загальне редагування та друк пояснювальної записки.	05.06.2023- 12.06.2023	
5	Проходження нормоконтролю, перепліт пояснювальної записки.	12.06.2023- 15.06.2023	
6	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	16.06.2023- 18.06.2023	

7. Дата видачі завдання 15.05.2023р.

Керівник кваліфікаційної роботи _____ Олег ЗУДОВ
(підпис керівника)

Завдання прийняв до виконання _____ Владислав ЗЕЛІНСЬКИЙ
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Розширення для автоматичного перекладу та озвучення субтитрів на YouTube» містить: 63 сторінки, 32 рисунки, 1 таблицю, 12 інформаційних джерел.

Об'єкт дослідження – розширення для браузера.

Предмет дослідження – переклад та озвучення субтитрів за допомогою розширення для браузера.

Мета кваліфікаційної роботи – розробити розширення для браузера, що перекладає та озвучує субтитри з використанням сучасних інструментів.

Методи дослідження – порівняльний, обробка літературних джерел, аналітичний.

Результати кваліфікаційної роботи можуть бути використані для ознайомлення з процесом та методами створення розширення а також для перекладу субтитрів на платформі YouTube.

РОЗШИРЕННЯ ДЛЯ БРАУЗЕРА, JAVASCRIPT, PYTHON, ТЕСТУВАННЯ, YOUTUBE, WEB.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП	7
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Історія відео-хостингу YouTube	9
1.2. Огляд відео-хостингу YouTube	11
1.3. Проблема доступності іноземних відео	14
1.4. Методи вирішення проблеми.....	15
1.5. Постановка завдання.....	17
ВИСНОВКИ ДО РОЗДІЛУ 1	18
РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗШИРЕННЯ ДЛЯ БРАУЗЕРА ..	19
2.1. Вибір та обґрунтування інструментів проектування	19
2.2. Методи створення розширення	28
2.3. Опис файлової структури та принципу роботи розширення	31
2.4. Розробка серверної частини додатку	33
2.5. Розробка клієнтської частини	38
ВИСНОВОК ДО РОЗДІЛУ 2	53
РОЗДІЛ 3. ТЕСТУВАННЯ РОЗШИРЕННЯ.....	54
3.1. Встановлення та запуск розширення	54
3.2. Тестування та перевірка	55
3.3. Можливі покращення	58
ВИСНОВОК ДО РОЗДІЛУ 3	60
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

HTML	– Hyper Text Markup Language – мова гіпертекстової розмітки
CSS	– Cascade Style Sheet – каскадні таблиці стилів
DOM	– Document Object Model – модель об'єктів документу
API	– Application Programming Interface - інтерфейс прикладного програмування
JSON	– JavaScript Object Notation – формат запису об'єктів JavaScript
HTTP	– Hyper Text Transfer Protocol – протокол передачі гіпертексту
URL	– Uniform Resource Locator - Уніфікований покажчик інформаційного ресурсу
SVG	– Scalable Vector Graphics - Масштабована векторна графіка
BEM	– Block, Element, Modifier – методологія найменування класів

ВСТУП

У сучасному світі важко уявити життя без інтернету. Завдяки ньому є можливість здійснювати покупки не виходячи з дому, спілкуватися з людьми, що знаходяться на іншому кінці світу, знаходити необхідну інформацію на будь-які теми, співпрацювати з колегами та партнерами, дивитися фільми та телебачення, навчатися онлайн.

Інтернет створив умови для глобалізації та культурного обміну. Проте з новими можливостями виникають і нові проблеми. Справа в тому, що в різних країнах світу розмовляють на різних мовах. Англійська мова є однією із найпоширеніших на планеті. Відповідно, велика кількість книг, навчального матеріалу, відео, та ін. представлена в інтернеті саме цією мовою, що може стати перешкодою для людей, котрі не володіють іноземними мовами.

Актуальність теми кваліфікаційної роботи «Розширення для автоматичного перекладу та озвучення субтитрів на YouTube» ґрунтується на тому, що YouTube є одним з найпопулярніших відеосервісів у світі, який має мільйони користувачів. Однак не всі відео мають субтитри або озвучення на різних мовах, що обмежує доступність та розуміння інформації для багатьох людей. Розробка розширення для автоматичного перекладу та озвучення субтитрів на YouTube дозволить покращити якість та доступність відео для широкої аудиторії і сприятиме культурному та мовному обміну.

Об'єктом дослідження кваліфікаційної роботи є розширення для браузера.

Предмет – переклад та озвучення субтитрів за допомогою розширення для браузера.

Мета кваліфікаційної роботи – розробити розширення для браузера Google Chrome, що перекладає та озвучує субтитри, з використанням сучасних інструментів розробки.

У відповідності з метою роботи визначено основні **завдання дослідження**:

- проаналізувати існуючі рішення для перекладу та озвучення тексту та визначити їх переваги та недоліки;
- обрати оптимальні технології та алгоритми для реалізації розширення, такі як машинний переклад, синтез мови тощо;
- розробити архітектуру та дизайн розширення, враховуючи вимоги до функціональності, зручності, безпеки та продуктивності;
- реалізувати розширення за допомогою мов програмування JavaScript та Python;
- протестувати розширення на різних веб-браузерах та відео на YouTube з різними мовами субтитрів.

Для досягнення мети та виконання поставлених завдань використано наступні методи: порівняльний, обробка літературних джерел, аналітичний.

Наукова новизна отриманих результатів полягає в тому, що запропоновано рішення проблеми локалізації відео на платформі YouTube.

Практичне значення отриманих результатів. Результати кваліфікаційної роботи можуть бути використані для ознайомлення із прикладом створення розширення для браузера Google Chrome, а також для перекладу та озвучення субтитрів на платформі для перегляду відео YouTube. Проєкт сприяє покращенню доступності та якості інформації для користувачів, які не володіють мовою оригінального відео.

Розширення дозволить переглядати відео авторів зі всього світу, що значно розширює кількість інформації на довільну тему. Воно може бути корисним для навчання іноземних мов, популяризації культурного та наукового контенту, а також для інклюзивної освіти та розваг людей з особливими потребами.

РОЗДІЛ 1

ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Історія відео-хостингу YouTube

YouTube - це відеохостингова платформа, яка дозволяє користувачам з довільного місця на планеті переглядати, завантажувати та ділитися відео з іншими користувачами. YouTube був заснований 14 лютого 2005 року Чардом Герлі, Стівеном Чені, Джаведом Карімом – колишніми працівниками PayPal. Існують різні версії щодо виникнення ідеї створення даного сервісу. Згідно однієї з них, засновники хотіли обмінятися відео, записаними на вечірці, однак не змогли знайти зручний сервіс для цього. 23 квітня 2005 року Джавед Карім виклав на сайт перший відеоролик, що тривав 19 секунд [6].

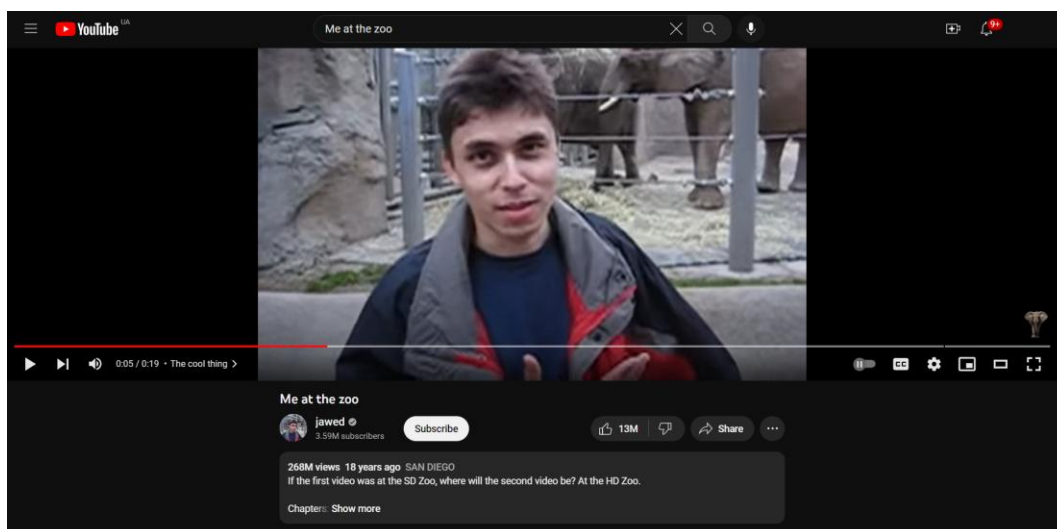


Рис. 1.1. Перше відео на YouTube

Кафедра КІТ				НАУ 23 10 14 000 ПЗ			
	<i>ПІБ</i>			РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Розроб.</i>	Зелінський В.О.					9	10
<i>Керівник</i>	Зудов О. М.				ТП-415Б – 122		
<i>Н. Контр.</i>	Толстікова О.В.						

Перший час сервіс не був популярним, проте згодом, після нововведень та покращення функціоналу на сайт почало заходити все більше користувачів. В травні 2005 року кількість відвідувачів за день сягала близько 30000. У грудні місяці 2005 року кількість переглядів відео на день перевищувала 2 млн, а ще через місяць це число перевищило 25 млн переглядів на день [7].

Незабаром у проєкту почали з'являтися інвестори. Першим стала компанія Sequoia Capital. За пів року величина вкладень сягнула 11,5 млн доларів. У 2006 році на YouTube з'явилася перша контекстна реклама. Таким чином засновники вирішили монетизувати сервіс, оскільки для підтримки сервісу були необхідні кошти. Попри це, в перші роки існування сайт майже не приносив дохід своїм власникам.

В листопаді 2006 року компанія Google викупила YouTube за 1,65 млрд доларів. З того часу компанія доклала чимало зусиль для розвитку сервісу. Відбувалось впровадження нових функцій, користувацький інтерфейс сайту та довідкова інформація перекладалися різними мовами світу. Велика Британія, Франція, Італія, Іспанія, Японія, Нідерланди стали першими країнами, в яких з'явилися локальні версії відеохостингу. В кінці вересня – на початку жовтня 2010 року користувацький інтерфейс та довідка з використання сервісу були перекладені українською мовою.

Нині YouTube є одним із найпопулярніших сервісів у всьому світі. За даними веб ресурсу www.semrush.com [8] відеохостинг посів друге місце у рейтингу найбільш відвідуваних сайтів.

Таблиця 1.1.

Рейтинг сайтів за відвідуваністю

№	Назва сайту	Кількість відвідувань
1	Google.com	99 323 026 768
2	Youtube.com	87 199 825 620
3	Facebook.com	12 516 682 581

4	Twitter.com	8 404 126 302
5	Wikipedia.org	6 887 562 598

1.2. Огляд відео-хостингу YouTube

Назва YouTube з англійської мови перекладається як you – «ти», tube – «камера, телевізор». А гаслом є «Broadcast Yourself», що означає «транслуй себе». Тобто сама назва дає зрозуміти, що сайт створює можливість будь-кому ділитись відео, проводити прямі трансляції та дивитися відео інших авторів.

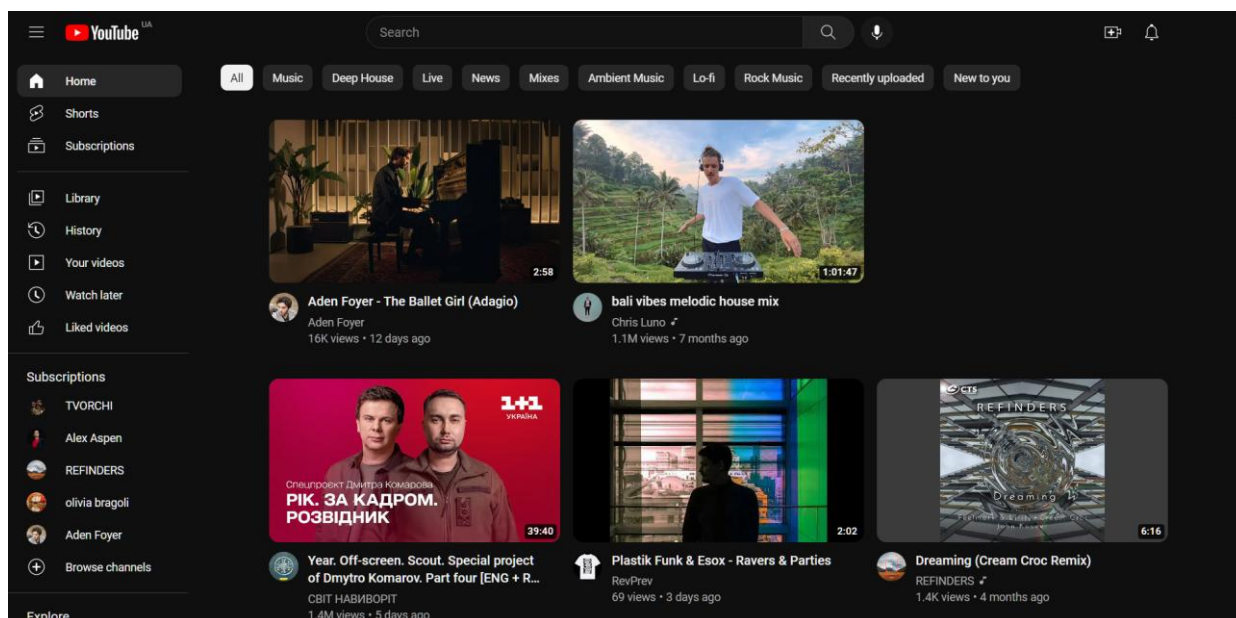


Рис. 1.2. Головна сторінка YouTube

На головній сторінці (рис. 1.2) розміщені відео із списку «Рекомендовані». Це ті відео, які YouTube рекомендує на основі історії переглядів. Також доступні вкладки «Підписки», «Бібліотека», «Історія», «Ваші відео» та «Переглянути пізніше». На вкладці «Підписки» відображаються останні відео авторів, за якими користувач слідкує. Відео відсортовані за датою публікації починаючи з новіших.

Через вкладку «Бібліотека» можна отримати доступ до історії переглядів, списку «Переглянути пізніше», а також до списків відтворення,

які користувач може створювати сам або зберігати публічні, створені іншими користувачами.

Окрім рекомендованих та збережених відео, YouTube надає можливість шукати відео на основі текстових запитів. На верхній панелі сайту знаходиться поле для введення тексту. Ввівши тему, що цікавить, та натиснувши клавішу «Enter» YouTube перейде на сторінку з результатами пошуку (рис. 1.3). Як результат будуть відображені відео, що відповідають заданій пошуковій фразі.

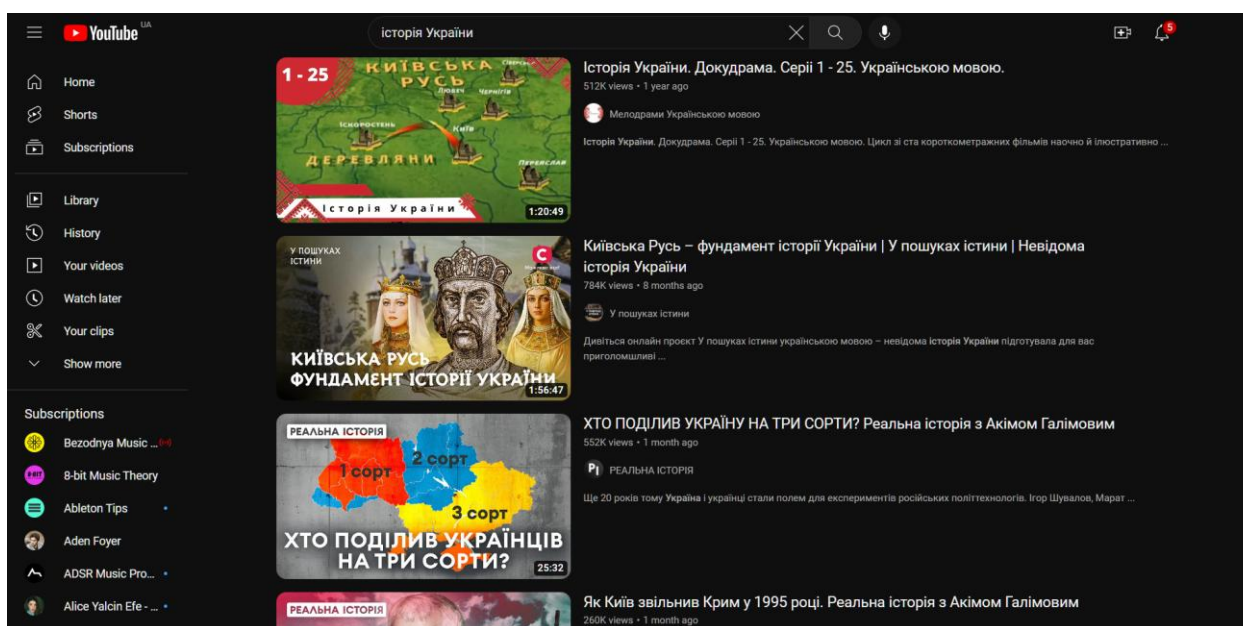


Рис. 1.3. Результати пошуку на YouTube

Також є можливість фільтрувати результати пошуку за допомогою налаштування фільтра. Таким чином є можливість сконцентруватися на конкретних відео, що відповідають конкретним запитам. Результати пошуку можна відфільтрувати за датою публікації, типом (відео, канал, список відтворення, фільм), тривалістю (менше 4 хв., 4 – 20 хв., більше 20 хв.), а також є можливість сортувати за різними параметрами.

При натисканні на назву відео відбувається перехід на сторінку з програвачем відео. Сторінка умовно поділена на декілька структурних елементів. Головним елементом є власне програвач відео. В ньому

відтворюється відео, а також представлені органи керування та налаштування.

На рис 1.4 зображено загальний вигляд елемента, в якому відтворюється відео. Присутні кнопки для паузи та продовження відтворення. Також є можливість перейти до наступного або попереднього відео. При наведенні на значок «динамік» з'являється слайдер для зміни гучності. Вище знаходиться індикатор відтворення – лінія що вказує, яка частина відео вже була відтворена, та скільки ще залишилося. За допомогою даного елемента є можливість переходити до окремої частини відео.

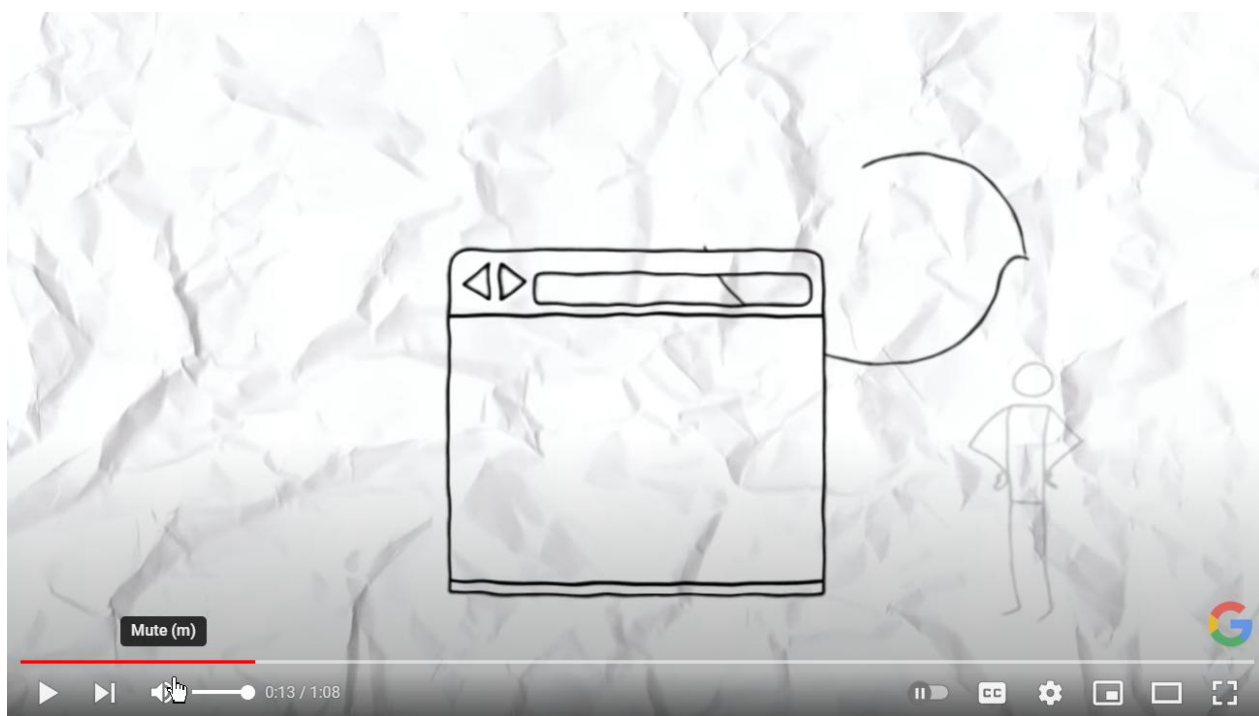


Рис. 1.4. Відео програвач YouTube

В нижній правій частині знаходяться органи для налаштування відео та зміни відображення плеєра. Є можливість переглядати відео в повноекранному режимі, розтягнути плеєр по ширині сторінки, або відкрити міні програвач. Серед опцій налаштування можна виділити наступні пункти:

- кінематографічне освітлення: опція, пов'язана зі зміною дизайну, при ввімкненні навколо програвача відображається тінь, колір якої залежить від кольорової гами у відео;

- швидкість відтворення: призначена для зміни швидкості програвання відео, можна вибрати зі списку, або вказати власноруч;

- субтитри: у даному пункті можна обрати мову субтитрів, що будуть відображатися поверх відео;

- якість: дане налаштування призначене для вибору якості відео. YouTube при завантаженні відео автоматично створює копії у гіршій якості. Опція є корисною при низькій швидкості інтернет підключення.

Нижче програвача розміщений інформаційний блок. В ньому відображається назва відео, коротка інформація про автора (назва каналу, кількість підписників), кількість вподобань, а також кнопки для поширення відео в інших соціальних мережах та збереження у списках відтворення. Також представлений блок з описом відео. В даному блоці автори зазвичай коротко описують відео, вставляють корисні посилання та ін.

Однією з важливих функцій є коментування відео. Під кожним відео представлено поле, в яке можна вписати коментар (якщо автор не вимкнув таку можливість в налаштуваннях). Доступний перегляд коментарів інших глядачів, а також написання відповіді. Таким чином під відеороликом на YouTube в коментарях можуть розпочатися дискусії та обговорення, в яких автор може взяти участь.

Праворуч знаходиться секція з відео, що пов'язані з темою, а також інші відео автора. Також в даній секції можуть знаходитися відео, що входять до поточного списку відтворення.

1.3. Проблема доступності іноземних відео

Зі зростанням популярності YouTube та поширенням сайту у різних країнах світу, постала проблема доступності відео. Користувачі ділилися своїми враженнями, думками, робили навчальні та пізнавальні відео. На

платформі почали з'являтися відео не тільки англійською мовою, а й іспанською, французькою, японською та ін. Це призвело до умовного поділу аудиторії (користувачів) YouTube. Часто трапляється так, що потрібна інформація або якісний навчальний матеріал представлені іноземною мовою. Це стає перешкодою для тих користувачів, що не розуміють мову оригіналу.

Однією із найпоширеніших мов світу є англійська мова. Велика кількість відео на платформі представлена саме цією мовою. Тому знання цієї мови створює можливість переглядати безліч якісних відео, розважальних шоу, пізнавальних передач. Згідно дослідженню порталу ef.com Україна у 2022 році набрала 539 балів та посіла 35 місце із 111 у рейтингу країн за рівнем знання англійської мови. Близько 25% українців розуміють англійську мову. Дещо більша кількість розуміє англійську на достатньому рівні, щоб приблизно зрозуміти загальні поширені теми і лише приблизно 10% знають мову на високому рівні. Тому для українського регіону дана проблема є актуальною [9].

1.4. Методи вирішення проблеми

Одним із можливих методів рішення проблеми є субтитри. Дану функцію на YouTube було добавлено ще у 2006 році. Користувачі можуть добавити власноруч створені субтитри у форматі «.srt» або створити субтитри автоматично. У другому випадку YouTube за допомогою технології розпізнавання голосу перетворює аудіо у текст та синхронізує його із відео. Точність тексту в даному випадку не є 100% і залежить від багатьох факторів, серед яких: наскільки чітко та виразно говорить оповідач, як швидко він вимовляє текст, та терміни, які він вживає, якість аудіо та присутність сторонніх звуків (музика, шум та ін.).

При перегляді відео користувач має можливість обрати субтитри із списку доступних. На рис. 1.5 представлено меню для вибору субтитрів. Проте виникає інша проблема – в більшості випадків субтитри представлені лише мовою оригіналу.

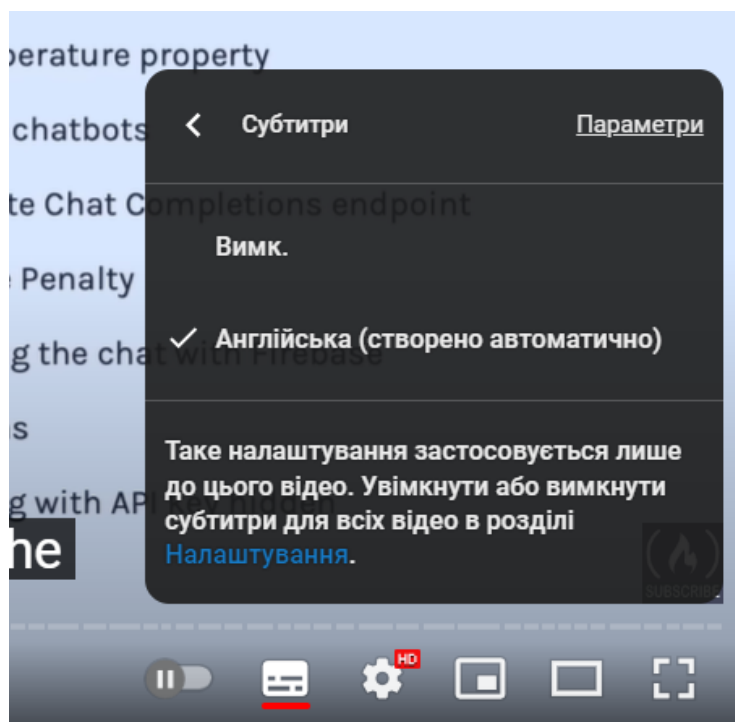


Рис. 1.5. Меню вибору субтитрів

Існують сайти та додатки, що дозволяють завантажити субтитри. Наприклад downsub.com. У поле пошуку потрібно вставити посилання на відео YouTube, натиснути «download» і вже за кілька секунд на сторінці буде відображено доступні субтитри із посиланням для їх завантаження. Додатково сайт пропонує автоматично перекласти субтитри на іншу мову. Завантажити субтитри можна у форматі «.txt» або «.srt». Незручність даного методу полягає в тому, що користувачу потрібно відволікатись для завантаження та відкриття субтитрів.

Загалом субтитри сприяють кращому розумінню та можуть бути корисними як для тих, що не розуміють або погано розуміють мову оригіналу, так і для людей із вадами слуху.

Інший метод полягає у створенні та популяризації українських каналів, які перекладають та озвучують відео іноземною мовою на YouTube. Такі канали допомагають розширити знання та горизонти україномовних глядачів, а також підтримують розвиток української мови та культури. Прикладом є «Цікава наука» - YouTube канал, автор якого перекладає та озвучує науково-

популярні та освітні відео з фізики, астрономії, біології, географії та математики.

Якість перекладу в такому випадку буде вище, ніж автоматичного. Проте недоліком є низька кількість каналів, на яких публікуються переклади іноземних відео. Відповідно і кількість перекладених відео значно нижча від кількості іноземних. Тому ймовірність знайти переклад матеріалу, що цікавить користувача дуже низька.

1.5. Постановка завдання

Під час роботи над дипломним проєктом було запропоновано рішення проблеми доступності іноземних відео – розширення для браузерів, що перекладає та озвучує субтитри поверх відео. Додаток значно полегшить розуміння іноземних відео. Основні завдання, що необхідно виконати:

- обрати оптимальні технології та алгоритми для реалізації розширення, такі як машинний переклад, синтез мови тощо;
- розробити архітектуру та дизайн розширення, враховуючи вимоги до функціональності, зручності, безпеки та продуктивності;
- розробити програмну реалізацію розширення за допомогою обраних технологій та мов програмування;
- протестувати розширення на різних веб-браузерах та відео на YouTube з різними мовами субтитрів;
- провести аналіз отриманих результатів та порівняти їх з існуючими рішеннями.

ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі розглянуто предметну область. Зокрема, описано історію створення відеохостингової платформи YouTube, основний функціонал, а також проблему локалізації та доступності відео.

YouTube був заснований у 2006 році Чардом Герлі, Стівеном Чені, Джаведом Карімом. Сайт був створений з метою поширення відео. З того часу мережа значно розрослась, та стала однією із найпопулярніших у світі. Користувачі з легкістю можуть завантажувати відео, а також переглядати відео інших авторів, коментувати їх та зберігати у списках відтворення. Дана ідея сподобалась людям, що стало причиною росту популярності додатку не тільки на території США, а й по всьому світу.

Проте виникла проблема доступності. Так люди, що не знають іноземних мов, можуть зіткнутися з перешкодами при пошуку відео на YouTube, оскільки велика кількість контенту представлена англійською та іншими мовами. Частковим рішенням даної проблеми є субтитри – текст, який промовляє автор. Проте в більшості випадків субтитри представлені лише мовою оригіналу.

Також в інтернеті є сайти, що дають можливість перекласти та завантажити субтитри. Все що потрібно зробити – це вказати посилання на відео та обрати бажану мову перекладу. Але такий варіант виявляється не надто зручним, оскільки потрібно відволікатись на пошук та завантаження перекладу.

Ще одним варіантом є YouTube канали, автори яких перекладають та переозвучують іноземні відео. Якість перекладу краща від машинного, проте шанси знайти потрібне відео в перекладі доволі низькі.

РОЗДІЛ 2

ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗШИРЕННЯ ДЛЯ БРАУЗЕРА

2.1. Вибір та обґрунтування інструментів проектування

2.1.1. Середовище розробки

Для розробки розширення для браузера було обрано редактор коду Visual Studio Code (рис. 2.1). Даний редактор створений на мові програмування JavaScript з використанням фреймворку Electron.

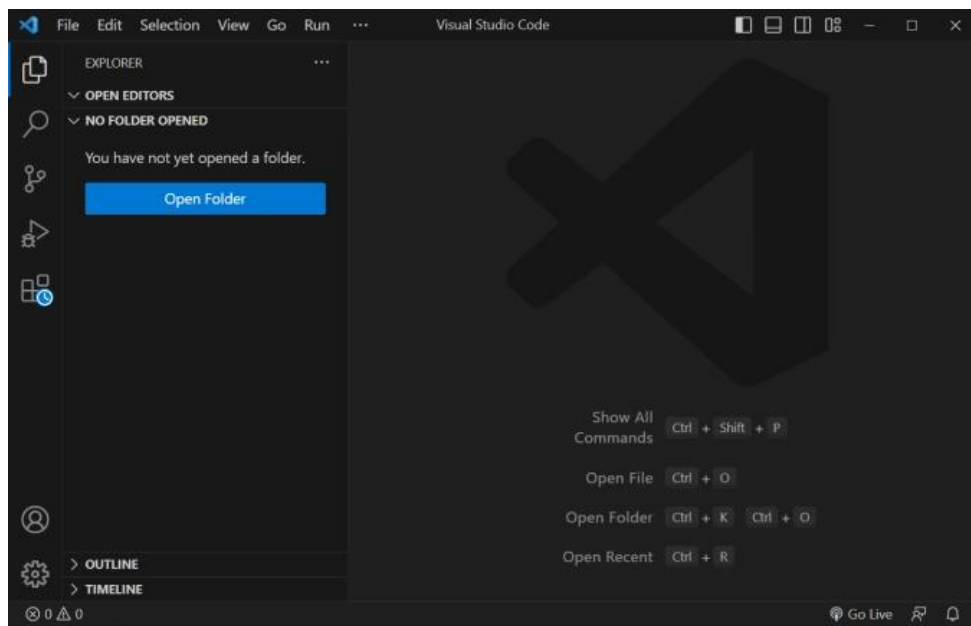


Рис. 2.1. Середовище розробки VS Code

VS Code - це безкоштовний відкритий редактор коду, розроблений компанією Microsoft. VS Code підтримує багато мов, включаючи JavaScript, Python, HTML та CSS, а також має можливості для відлагодження коду, автодоповнення, збереження інтерактивних сеансів, керування версіями за допомогою Git та інші корисні функції.

Кафедра КІТ				НАУ 23 10 14 000 ПЗ			
	ПІБ			РОЗДІЛ 2. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗШИРЕННЯ ДЛЯ БРАУЗЕРА	Літ.	Аркуш	Аркушів
Розроб.	Зелінський В.О.					19	34
Керівник	Зудов О. М.				ТП-415Б – 122		
Н. Контр.	Толстікова О.В.						

Одна з головних переваг VS Code полягає в тому, що він є дуже легким та інтуїтивним, що дозволяє швидко та легко редагувати код. Крім того, VS Code має велику кількість опцій для налаштування, тобто користувач може додавати розширення та налаштовувати редактор згідно своїх потреб.

Також редактор має різноманітну кількість функцій для взаємодії з кодом. Наприклад, користувач може використовувати зручну систему пошуку та заміни тексту, налаштовувати гарячі клавіші, використовувати функцію "розгортання" коду, щоб згорнути довгі блоки коду, виконувати швидкий пошук і заміну фрагментів коду та багато інших корисних функцій.

Вбудований термінал у VS Code став у нагоді при розробці серверної частини застосунку. У ньому виконувався запуск скриптів, написаних на мові програмування Python.

2.1.2 Мова гіпертекстової розмітки HTML

HTML (англ. HyperText Markup Language — мова розмітки гіпертексту) — це мова, що дозволяє створювати веб-сторінки для Інтернету. HTML використовує теги, що описують структуру та зміст документа, а також можуть включати посилання, зображення, медіафайли, форми та інші елементи. HTML є стандартом, що розробляється та підтримується організацією W3C (World Wide Web Consortium).

HTML складається з декількох частин: заголовка (head), тіла (body) та опціональної частини DOCTYPE, що визначає версію HTML. Заголовок містить метадані про документ, такі як його назву, кодування символів, стилі та скрипти. Тіло містить власне вміст документа, який поділяється на розділи (sections), абзаци (paragraphs), заголовки (headings), списки (lists) та інші елементи. DOCTYPE вказує браузеру, як інтерпретувати HTML-код [1].

На рис. 2.2 представлено приклад HTML розмітки. В першому рядку декларується тип документу. Таким чином для браузера вказується інформація про тип документу. Всі теги html документа повинні бути знаходитися поміж тегів <html></html>. Документ розділяється на дві

частини: заголовок та тіло. У заголовку представлено дані про назву, опис та автора документу, а також за допомогою елементу `<link>` підключено зовнішній файл стилів.

```
<!DOCTYPE html>
<html>
<head>
  <title>Головна сторінка</title>
  <meta name="description" content="Приклад HTML документу">
  <meta name="keywords" content="HTML, гіпертекст, розмітка">
  <meta name="author" content="Владислав Зелінський">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Вітаю на моєму веб-сайті!</h1>

  <p>Це перший абзац мого тексту.</p>

  <h2>Список моїх улюблених кольорів:</h2>
  <ul>
    <li>Червоний</li>
    <li>Зелений</li>
    <li>Синій</li>
  </ul>

  <h3>Зображення автомобіля:</h3>
  
</body>
</html>
```

Рис. 2.2. Приклад HTML розмітки

Тіло документу – це частина документу, що відображається в браузері. У прикладі наведено приклади заголовків, абзаців, списків.

HTML застосовується в різних сферах веб-розробки і дизайну, таких як:

- статичні веб-сайти: HTML є основою будь-якого веб-сайту, який не змінюється динамічно. Статичні веб-сайти показують однаковий зміст усім користувачам і не потребують серверної обробки;

- динамічні веб-сайти: HTML є складовою частиною динамічних веб-сайтів, які змінюються в залежності від запиту користувача, часу, місця або інших факторів;

- веб-додатки: HTML є необхідним елементом сучасних веб-додатків, які надають користувачам інтерактивний і функціональний інтерфейс. Веб-додатки використовують клієнтські технології, такі як JavaScript, CSS, AJAX тощо, для покращення користувацького досвіду і забезпечення зв'язку з сервером.

2.1.3 Мова каскадних таблиць стилів CSS

CSS - каскадні таблиці стилів (Cascading Style Sheets) - це мова, що використовується для опису вигляду та форматування веб-сторінок, які написані з використанням мови розмітки гіпертексту (HTML) або розширеної мови розмітки (XML). За допомогою CSS можна задавати різні стилі для елементів HTML-коду, такі як кольори, розміри, шрифти, відступи, фонові зображення, анімації та інші ефекти.

```
h1 {
  color: blue;
  font-size: 24px;
}

p {
  color: green;
}

ul {
  color: red;
}

img {
  width: 200px;
  height: 200px;
}

.main-container {
  background-image: url(../media/img/background.jpg);
}
```

Рис. 2.3. Приклад CSS

На рис. 2.3 представлено приклад CSS коду для html документу. Для заголовку першого рівня задано синій колір тексту та розмір шрифту 24 пікселі. Для абзацу (елемент <p></p>) задано зелений колір тексту. Колір тексту елементів маркованого (ненумерованого) списку вказано червоний. Також задано ширину та висоту для зображення.

Синтаксис мови CSS складається із селектора та блоку декларації. Селектор вказує на HTML елемент, до якого повинні застосуватися правила css. Блок декларації містить в собі css правила. Кожне правило складається із назви властивості та значення, що розділені двокрапкою. Самі правила розділені між собою крапкою з комою [2]. Нижче наведено загальна структура синтаксису:

```
селектор {  
    властивість: значення;  
    властивість: значення;  
    ...  
}
```

CSS можна прописувати безпосередньо для елемента HTML, поміж тегів <style>...</style> або в окремому файлі з розширенням “.css”. Останній спосіб дозволяє відокремити опис зовнішнього вигляду сторінки від її структури та контенту, що дозволяє розробникам бути більш гнучкими та зберігати час при внесенні змін до веб-сайту.

CSS працює за принципом каскаду. Це означає, що якщо кілька правил задають однаковий стиль для одного й того ж елемента, то буде використовуватися правило з найвищим пріоритетом.

2.1.4 Мова програмування JavaScript

JavaScript (JS) - це високорівнева мова програмування, яка використовується для створення динамічних веб-сторінок та веб-додатків. Вона є однією з найпопулярніших мов програмування в світі і широко використовується для розробки як клієнтських, так і серверних додатків. Код

виконується на стороні клієнта (браузер), але також може використовуватися для програмування на стороні сервера (за допомогою платформи Node.js). JavaScript є інтерпретованою мовою. Це означає, що код виконується відразу, без необхідності компіляції (Just In Time) [3].

JavaScript може взаємодіяти з HTML-кодом та CSS-стилями, що дозволяє змінювати зовнішній вигляд сторінки та динамічно змінювати її вміст без необхідності перезавантаження сторінки. JavaScript також може взаємодіяти з сервером, що дозволяє створювати динамічні веб-додатки, що змінюються залежно від даних, які надходять від користувача або сервера.

Синтаксис JS подібний до синтаксису мови програмування C/C++. До основних елементів належать:

- змінні. Використовуються для збереження значень. В JavaScript змінні можна створювати за допомогою ключового слова `var`, `let` або `const`. Мова має динамічну типізацію, тобто тип змінної змінюється в залежності від типу даних. Всього у мові програмування представлено 8 типів даних: рядок (`string`), число (`number`), велике число (`bigint`), булевий (`boolean`), невизначений (`undefined`), нуль (`null`), символ (`symbol`), об'єкт (`object`);

- оператори. Використовуються для виконання різних операцій над змінними або значеннями. У JavaScript оператори поділяються на декілька типів: арифметичні (`+`, `-`, `*`, `**`, `/`, `%`, `++`, `--`), оператори присвоєння (`=`, `+=`, `-=`, `*=`, `/=`, `%=`, `**=`), порівняння (`==`, `===`, `!=`, `>`, `<`, `>=`, `<=`, `?`), логічні (`&&`, `||`, `!`);

- функції – це блоки коду, які виконують певні дії. Функції можуть приймати параметри та повертати значення. Декларація функції відбувається за допомогою ключового слова `function`, після якого вказується назва функції, а в дужках можна вказати значення, які функція приймає. Тіло функції знаходиться поміж фігурних дужок `{ ... }`. За замовчуванням функція повертає значення `undefined`. Проте є можливість повернути результат обчислень за допомогою ключового слова `return`;

- умовні конструкції використовуються для виконання коду в залежності від заданої умови. Синтаксис умовної конструкції виглядає наступним чином:

```
if (умова) {  
    // якщо умова задовільняється, то виконати наступний код  
    ...  
} else {  
    // інакше виконати альтернативний код  
    ...  
}
```

- об'єкти використовуються для збереження зв'язаних даних та функцій у вигляді ключ – значення. Нижче наведено приклад синтаксису об'єкту:

```
let назваОб'єкту = {  
    ключ1: "значення1",  
    ключ2: "значення2",  
    ...  
}
```

Дана мова програмування має велику кількість бібліотек та фреймворків що допомагають розробникам створювати більш складні веб-додатки з високою ефективністю та масштабованістю.

2.1.5 Мова програмування Python

Python – це високорівнева, інтерпретована мова програмування загального призначення. Python використовується для різних завдань, таких як розробка веб-додатків, наукових досліджень, аналізу даних, машинного навчання та багатьох інших. Він має простий синтаксис, що дозволяє швидко та легко розуміти код. Python підтримує об'єктно-орієнтований та

процедурний підходи до програмування, що дозволяє розробникам використовувати різні підходи для розв'язання завдань.

Python має багатий стандартний набір бібліотек, що дозволяють розробникам використовувати готові рішення для різних задач. Також, він має велику кількість сторонніх бібліотек, що дозволяють розробникам швидко та легко розширювати можливості мови.

Архітектура мови Python характеризується такими особливостями:

- це інтерпретована мова програмування: вихідний код виконується без попереднього компілювання. Це сприяє швидкій розробці та налагодженню програм, а також переносності коду між різними платформами;

- це об'єктно-орієнтована мова програмування: Python підтримує створення та використання користувацьких типів даних, які називаються класами. Класи можуть мати атрибути (дані) та методи (функції), які визначають поведінку об'єктів цього класу. Python також дозволяє наслідування, поліморфізм та інкапсуляцію, які є основними принципами об'єктно-орієнтованого програмування [4];

- це мова з динамічною типізацією: типи даних змінних визначаються під час виконання програми, а не попередньо. Це дозволяє більшою гнучкістю та експресивністю коду, але також потребує більшої уваги до можливих помилок типізації;

- це мова з багатим набором структур даних високого рівня, таких як списки, словники, множини та кортежі. Це дозволяє зберігати та обробляти різноманітну інформацію ефективно та зручно;

- це мова з поглибленою підтримкою функціонального програмування, що означає, що вона дозволяє використовувати функції як об'єкти першого класу.

Згідно з опитуванням, проведеним редакцією DOU [5], мова програмування Python знаходиться на 3 місці за популярністю, а у сфері

Back-End розробки посіла 4 місце. Це говорить про поширеність мови та зручність застосування для розробки різноманітних проектів.

2.1.6 Веб фреймворк Flask

Flask - це легкий фреймворк для веб-розробки на мові програмування Python. Даний фреймворк надає інструменти для швидкої розробки веб-додатків та API, дозволяючи розробникам створювати високоякісні веб-додатки з невеликими зусиллями. Він призначений для швидкого та простого створення веб-додатків з можливістю масштабування до складних проектів.

Flask дозволяє розробникам зосередитися на функціональності додатку, не турбуючись про більш складні аспекти веб-розробки, такі як обробка запитів, маршрутизація та керування станом сесії. Розробник сам обирає інструменти та бібліотеки, які він хоче використовувати. Існує багато розширень, наданих спільнотою, які полегшують додавання нової функціональності.

Flask також має вбудовану підтримку тестування, що дозволяє розробникам легко тестувати свій код та перевіряти правильність його роботи.

До переваг фреймворку можна віднести:

- мінімалізм і гнучкість. Flask не накладає жодних обмежень на організацію коду або вибір компонентів. Є можливість використовувати будь-яку базу даних, шаблонізатор, систему аутентифікації тощо;

- швидке та просте розгортання. Flask має вбудований сервер розробки, який дозволяє запускати додаток локально без потреби в налаштуванні Apache або Nginx. Також існують зручні способи розгортання додатку на хмарних платформах, таких як Heroku або Google App Engine;

- багата документація та підтримка спільноти. Flask має чудову офіційну документацію, яка покриває всі аспекти роботи з фреймворком. Крім того, існує багато практичних посібників, курсів та прикладних проектів, доступних на ресурсах, таких як PyPI, Real Python або Auth0.

2.2. Методи створення розширення

Розширення – це програмний застосунок, призначений для додавання нового функціоналу у браузер. Такі додатки роблять користування браузером зручнішим, безпечнішим та ефективнішим. Розширення доступні для різних веб-браузерів, таких як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari і Opera. Вони можуть бути завантажені та встановлені з веб-магазинів або офіційних веб-сайтів браузера.

В основному, такі додатки створюють на мові програмування JavaScript з використанням різноманітних API (JavaScript API, Chrome API та ін.). Також можливе використання мови HTML та CSS для зміни зовнішнього вигляду окремих елементів веб сторінки.

2.2.1. Chrome API

Chrome APIs дають змогу взаємодіяти з браузером. Наприклад, відтворення звуку, доступ до історії, взаємодія з відкритими вкладками та багато іншого функціоналу. Нижче наведено перелік деяких API з їх коротким описом [10]:

- tabs – це API для взаємодії із системою вкладок браузера. Надає змогу маніпулювати вкладками (створення нових, перезавантаження, визначення мови, вимкнення звуку та ін.);
- tts – це API для озвучення тексту. Підтримується тикими операційними системами, як Windows, Mac OS X, та ChromeOS;
- storage – надає зручний спосіб зберігати та отримувати дані в локальному сховищі на комп'ютері користувача, такі як історія відвідувань, обрані закладки та інші;
- messaging – дозволяє спілкуватися між різними частинами розширення, які можуть працювати в різних процесах. Зокрема, це API дозволяє обмінюватися повідомленнями між фоновими та відображуваними сторінками;

- cookies – дозволяє розробникам отримувати доступ до cookies (файли з інформацією про користувача, які зберігаються на його комп'ютері) та працювати з ними.

2.2.2. Контентний скрипт (contentScript.js)

Зазвичай розширення складаються з трьох основних компонентів: фонового скрипту (background.js), контентного скрипту (contentScript.js), файлу конфігурації (manifest.json) та інтерфейсу користувача. Контентний скрипт виконується в контексті веб-сторінки і може змінювати її зміст та поведінку. Їх використовують для зміни зовнішнього вигляду веб-сторінки, взаємодії з користувачем та інших дій на сторінці.

У contentScript.js можуть бути визначені функції та змінні, які потім використовуються в розширенні. Також цей файл може взаємодіяти з іншими скриптами та HTML-елементами на веб-сторінці, щоб додати або змінити функціональність сторінки.

На рис. 2.4 представлено приклад коду у файлі contentScript.js. За допомогою методу getElementById() отримано елемент з ідентифікатором “myButton” та додано йому прослуховувач подій. При натисканні на кнопку фоновий колір сторінки зміниться на рожевий.

```
// встановлення обробника події для кнопки на сторінці
document.getElementById('myButton').addEventListener('click', function() {
  // зміна колір фону сторінки
  document.body.style.backgroundColor = 'pink';
});
```

Рис. 2.4. contentScript.js – приклад коду

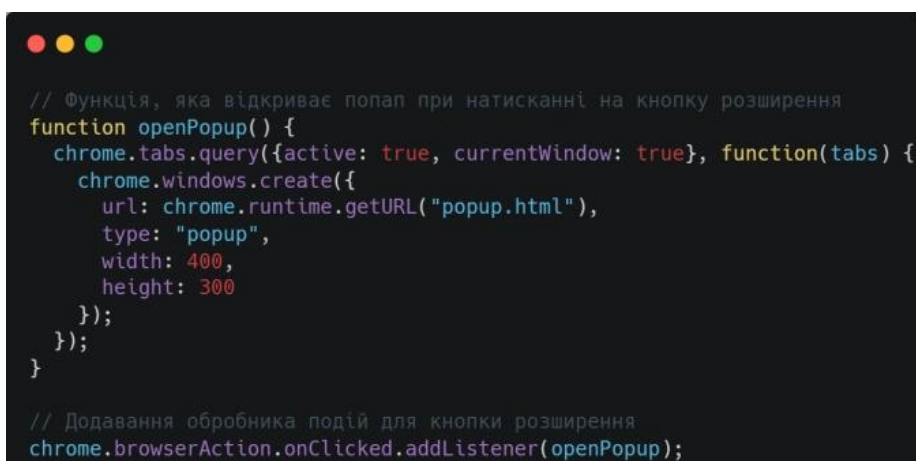
Даний приклад демонструє роботу скрипту, який виконується у контексті веб сторінки, тобто має доступ до всіх елементів сторінки.

2.2.3. Фоновий скрипт (background.js)

Фоновий скрипт виконується в окремому процесі і може взаємодіяти з браузером та іншими розширеннями.

Цей файл містить скрипти, які запускаються під час завантаження розширення і залишаються активними протягом всього часу роботи розширення. Їх функції можуть бути викликані з інших скриптів у розширенні. Наприклад, у background.js можна створити обробник подій для отримання повідомлень від розширення або веб-сторінки, який містить інформацію про стан розширення, або зберігати дані в локальному сховищі, яке доступне для всіх скриптів у розширенні.

На рис. 2.5 представлено приклад фонового скрипта. Даний код додає обробник подій до кнопки розширення, який викликає функцію openPopup(), яка відкриває вікно з файлу popup.html.



```
// Функція, яка відкриває попап при натисканні на кнопку розширення
function openPopup() {
  chrome.tabs.query({active: true, currentWindow: true}, function(tabs) {
    chrome.windows.create({
      url: chrome.runtime.getURL("popup.html"),
      type: "popup",
      width: 400,
      height: 300
    });
  });
};

// Додавання обробника подій для кнопки розширення
chrome.browserAction.onClicked.addListener(openPopup);
```

Рис. 2.5. Background.js – фоновий скрипт

2.2.4. Manifest.json – файл конфігурації

Також обов'язковим для розширення є файл manifest.json. Це файл конфігурації, що містить основну інформацію про розширення, таку як назва, версія, автор, опис, а також список файлів і функцій, які розширення використовує. Цей файл є обов'язковим для всіх розширень Chrome і

дозволяє браузеру зрозуміти, які ресурси потрібні для завантаження розширення та його функціоналу.

У `manifest.json` визначається, які сторінки та сайти повинні бути доступні для взаємодії з розширенням, які повідомлення мають відображатися в користувальницькому інтерфейсі розширення та інші параметри. Деякі API, які надає Chrome, також можуть вимагати відповідних дозволів у `manifest.json` [11].

На рис. 2.6 зображено загальний вигляд файлу конфігурації, в якому вказано версію розширення, назву, опис, дозволи та контекст виконання скриптів.

```
{
  "manifest_version": 3,
  "name": "YouTube Polyglot",
  "description": "Translate youtube close captions and voice them",
  "version": "1.0",
  "action": {
    "default_popup": "popup.html"
  },
  "permissions": ["storage", "tabs", "tts"],
  "host_permissions": ["https://*.youtube.com/*", "http://localhost/*"],
  "content_scripts": [
    {
      "matches": ["https://*.youtube.com/*"],
      "js": ["library.js", "contentScript.js"]
    }
  ],
  "background": {
    "service_worker": "background.js"
  }
}
```

Рис. 2.6. Manifest.json – приклад конфігурації розширення

Файл `manifest.json` зазвичай знаходиться в кореневій папці розширення і може бути відредагований в будь-який час для зміни параметрів розширення.

2.3. Опис файлової структури та принципу роботи розширення

Основна функція розширення полягає в автоматичному перекладі та озвученні субтитрів. Користувач може обрати субтитри зі списку доступних і

вибрати мову, на яку їх потрібно перекласти. Після цього, натиснувши кнопку увімкнення, розширення починає озвучувати перекладені субтитри.

З програмної точки зору розширення має дві складові частини: клієнтську та серверну. Клієнтська частина виконується у веб-браузері користувача і реалізована з використанням JavaScript. Ця частина відповідає за взаємодію користувача з розширенням, включаючи вибір субтитрів та мови перекладу.

Серверна частина відповідає за пошук та переклад субтитрів. Після того як користувач обрав субтитри та мову, клієнтська частина надсилає запит до сервера, який виконує пошук відповідних субтитрів та їх переклад. Перекладені субтитри потім надсилаються назад до клієнтської частини, де вони озвучуються.

Цей поділ на клієнтську та серверну частини дозволяє розширенню працювати ефективно і забезпечує зручний інтерфейс для користувача. Клієнтська частина взаємодіє з браузером користувача і керує процесом відтворення субтитрів, тоді як серверна частина виконує завдання пошуку та перекладу субтитрів на віддаленому сервері.

Розширення – це папка, в якій містяться скрипти, написані мовою програмування JavaScript, а також інші необхідні файли (стили css, зображення, html документи і т.д.). Обов'язковим файлом для кожного розширення є «manifest.json». Це файл типу JSON, в якому вказано інформація про розширення, а також файли, що необхідно підключити.

У папці містяться файли «contentScript.js» та «background.js». У першому файлі прописані скрипти, що виконуються у контексті веб сторінки. За допомогою даних скриптів можна змінювати структуру сторінки, додавати та видаляти елементи, а також модифікувати їх.

Фонові скрипти виконуються глобально у контексті усього браузера. Вони виконуються відразу після запуску розширення та продовжують працювати після закриття веб сторінки. За допомогою них можна отримувати

доступ до системи вкладок, керування закладками та історією переходів та ін.

У папці «css» знаходиться файл «style.css» – каскадні таблиці стилів. У даному файлі описані стилі для відображення інтерфейсу користувача. Дизайн меню повторює загальний дизайн меню налаштувань YouTube.

У папці «js» містяться скрипти «library.js» «svg.js» «menu.js». Це додаткові скрипти, що виконуються у контексті веб сторінки. «library.js» – це невеличка бібліотека, що полегшує створення елементів DOM. У файлі «svg.js» з використанням бібліотеки створено функції для швидкого створення svg елементів.

Файлова структура представлена на рис 2.7.



Рис. 2.7. Файлова структура клієнтської частини

2.4. Розробка серверної частини додатку

Серверна частина складається із єдиного файлу «server.py». В даному скрипті на мові програмування Python з використанням фреймворку Flask реалізовано пошук та переклад субтитрів. Сервер отримує, обробляє запит та відправляє відповідь клієнту. Відповідь – це список доступних субтитрів для відео, або субтитри, що перекладені на вказану мову.

Для розробки серверної частини використано підхід REST (з англ. «REpresentational State Transfer») – це стиль архітектури програмного забезпечення, який використовується для побудови веб-сервісів. Він визначає набір принципів та обмежень, що регулюють взаємодію між клієнтами та серверами. На основі даного підходу побудовано REST API додаток [12]. REST API використовує протокол HTTP для передачі даних між клієнтом і сервером. Він базується на понятті ресурсів, які є ідентифікаторами або адресами, і можуть бути отримані, створені, оновлені або видалені за допомогою стандартних HTTP методів, таких як GET, POST, PUT, DELETE, та ін.

REST API передає дані у форматі, як правило, JSON (з англ. «JavaScript Object Notation») або XML (з англ. «eXtensible Markup Language»), що дозволяє легко обмінюватися даними між різними системами.

З метою полегшення розробки додатку використано методи фреймворку Flask, а саме «Flask» «request» «jsonify».

На рис. 2.8 представлено приклад створення об'єкту додатку Flask. «*Flask(__name__)*» створює інстанцію класу Flask з бібліотеки Flask. Аргумент «*__name__*» вказує на поточний модуль або пакет, де цей рядок коду виконується. Це потрібно для правильної ідентифікації шляху до статичних і шаблонних файлів, які можуть бути використані в додатку.

```
7 app = Flask(__name__)
8 CORS(app)
```

Рис. 2.8. Створення об'єкту додатку Flask

Створений об'єкт «*app*» є екземпляром класу Flask і представляє базовий фреймворк додатку Flask. Він використовується для конфігурування та управління додатком, встановлення маршрутів (URL-шляхів), обробки запитів і відповідей, обробки шаблонів, налаштування підключень до бази даних та багато іншого.

«*CORS(app)*» застосовує налаштування CORS до додатку Flask. CORS (з англ. «Cross-Origin Resource Sharing») - це механізм, який дозволяє браузерам контролювати доступ до ресурсів на веб-сторінках, розташованих на інших доменах або портах. Пакет CORS надає можливість налаштовувати додаток Flask для дозволу або відмови в доступі до його ресурсів з різних джерел.

Після цього постає потреба в обробці запитів від клієнта. Для запитів до сервера використовується метод GET. Метод GET є одним із стандартних методів HTTP-протоколу. Основна характеристика методу полягає в тому, що він передає дані через параметри URL-адреси. Метод GET не передає тіло запиту, тому всі дані, включаючи параметри запиту, додаватимуться безпосередньо до URL-адреси.

Клієнт звертається до сервера у двох випадках: коли необхідно отримати список субтитрів для відео та коли необхідно перекласти субтитри на вказану мову. Для обробки цих двох запитів створено дві функції: «*get_caption_list()*» та «*return_caption()*».

Для того, щоб зв'язати ці функції із запитами використовується команда «*@app.get()*». Для першої функції вказано шлях «*/captionlist*». При зверненні за цим шляхом буде виконано пошук субтитрів. Для другої функції вказано «*/caption*». Дана функція перекладає субтитри вказаною мовою.

При здійсненні запитів обов'язково передаються додаткові параметри. При запиті за посиланням «*/captionlist*» додатково вказується ідентифікатор (унікальний код) відео. Загальне посилання виглядає наступним чином: «*http://127.0.0.1:5000/captionlist?id='ідентифікатор'*». Для отримання перекладу субтитрів у запиті необхідно вказати ідентифікатор відео, обрані субтитри та мову перекладу. Загальне посилання виглядає наступним чином: «*http://127.0.0.1:5000/caption?id='ідентифікатор'&lang='субтитри'&translateto='мова перекладу'*»

Для отримання параметрів запиту використовується «*Flask.request*» – об'єктом, який надає доступ до інформації про HTTP-запит, що надійшов до

сервера у контексті виконання додатку Flask. Він є частиною Flask і дозволяє отримувати доступ до різних атрибутів та даних, пов'язаних з запитом.

До основних цілей використання `Flask.request` включають:

Отримання методу запиту – за допомогою `Flask.request.method` можна отримати метод, який був використаний у відправленому запиті, такий як GET, POST, PUT або DELETE. Це дозволяє визначити, яку дію потрібно виконати відповідно до типу запиту.

Отримання параметрів запиту – `Flask.request.args` дозволяє отримати параметри запиту, передані через URL-рядок запиту або через запит POST. Наприклад, можна отримати значення параметра `id` з URL-строки запиту за допомогою `Flask.request.args['id']`.

Отримання даних форм: якщо запит був виконаний з методом POST і містить дані форми, то `Flask.request.form` надає доступ до цих даних. Можна отримати значення конкретного поля форми, використовуючи `Flask.request.form['field_name']`.

Робота з файлами: якщо ваш запит містить відправку файлів, то `Flask.request.files` дозволяє отримати доступ до цих файлів. Наприклад, можна отримати файл за його ім'ям, використовуючи `Flask.request.files['file_name']`.

Інші атрибути та методи: `Flask.request` також надає доступ до додаткових атрибутів та методів, таких як `headers` (заголовки запиту), `cookies`, `url` (URL-адреса запиту) та багато інших, які дозволяють отримувати детальну інформацію про запит.

Загалом, `request` є потужним інструментом для отримання доступу до даних, пов'язаних з HTTP-запитом, і використовується для обробки запитів і взаємодії з даними, які передаються до вашого сервера Flask.

Для пошуку та перекладу субтитрів використовується бібліотека `YouTubeTranscriptApi`. Дана бібліотека є «обгорткою» навколо офіційного YouTube API, що дає змогу отримати доступ не тільки до субтитрів, а й до інших функцій YouTube.

При отриманні запиту «/captionlist» створюється список *list*, а також виконується запит «*YouTubeTranscriptApi.list_transcripts(request.args["id"])*». «*request.args["id"]*» повертає ідентифікатор відео, що був переданий через запит, а сам метод *list_transcripts* повертає список субтитрів, що зберігаються у змінну *transcript_list*.

Список доступних субтитрів за допомогою циклу переписується у список *list*, після чого конвертується у формат JSON для повернення даних на клієнтську частину. На рис. 2.9 представлено загальну структуру списку *list*, що повертається після виконання функції.

```
[
  {
    "language" : "Англійська",
    "language_code" : "en",
    "is_generated" : "false",
    "translation_languages" : []
  },
  {
    "language" : "Іспанська",
    "language_code" : "es",
    "is_generated" : "false",
    "translation_languages" : []
  }
]
```

Рис. 2.9. Список *list*

Код отримання та повернення списку субтитрів розміщений в конструкції `try ... except ...`. Дана конструкція дає змогу обробляти помилки, а також додаток не припинить свою роботу після невдалого запиту та продовжить прослуховувати порт.

При запиті «/caption» виконується функція «*return_caption()*». Дана функція також виконується у конструкції `try...except...`. Спершу виконується запит на отримання списку субтитрів для відео за ідентифікатором. Далі

виконується наступний запит для отримання субтитрів вказаною мовою. За допомогою методу `translate` виконується переклад субтитрів. Далі результат повертається на клієнтську частину. При виникненні помилки, на випадок, якщо дані в запиті вказані невірно (невірний ідентифікатор), або субтитри для вказаного відео недоступні повернеться повідомлення "Transcript is not found!" (з англ. «Субтитри не знайдено!»).

2.5. Розробка клієнтської частини

У ролі клієнтської частини виступає розширення для браузера. Основною мовою програмування для розробки такого додатку є JavaScript. З технічної точки зору розширення вбудовує у плеєр перегляду відео додаткову кнопку для відкриття меню налаштувань плагіна. Після налаштування та натискання на перемикач «on/off» до сервера надсилається запит на отримання перекладених субтитрів. В разі успіху субтитри починають озвучуватись.

2.5.1. Дизайн та шаблон інтерфейсу користувача

Для взаємодії із розширенням необхідно розробити інтерфейс користувача. Існує два основні способи для представлення інтерфейсу в розширенні: впливаюче вікно або вбудовування меню безпосередньо в сторінку.

Перший варіант полягає у створенні HTML документа, що буде з'являтися у впливаючому вікні при натисканні на іконку розширення. На рис. 2.10 наведено приклад такого меню.

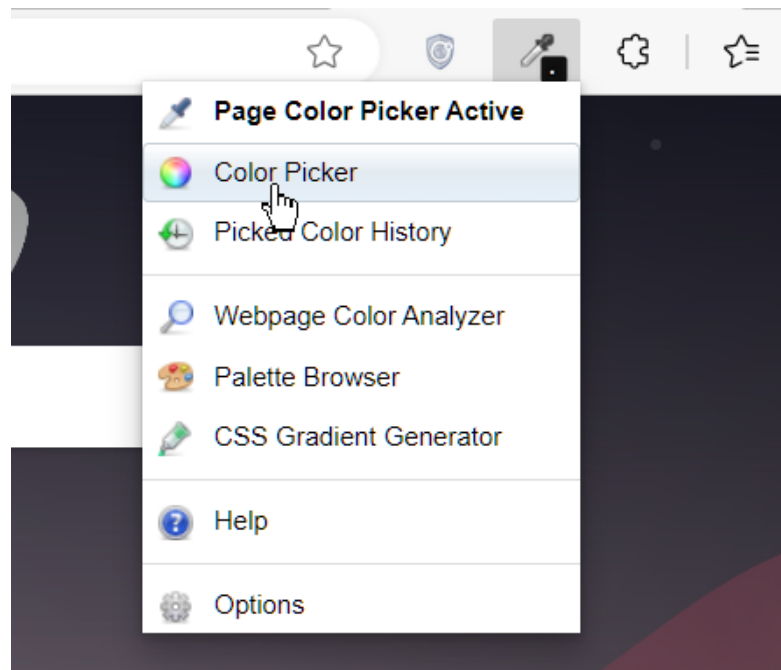


Рис. 2.10. Меню налаштувань розширення

При цьому в файлі `manifest.json` потрібно вказати, що даний документ буде відкриватись при натисканні на іконку.

Для вбудовування меню у сторінку потрібно скористатися методами мови JavaScript. Дані скрипти повинні виконуватись у контексті веб сторінки для того, щоб мати доступ до елементів сторінки і відповідно змінювати вміст сторінки. Проте спершу необхідно розробити дизайн.

При розробці дизайну як зразок було взято дизайн меню налаштувань відео програвача YouTube. На рис. 2.11 представлено дане меню.

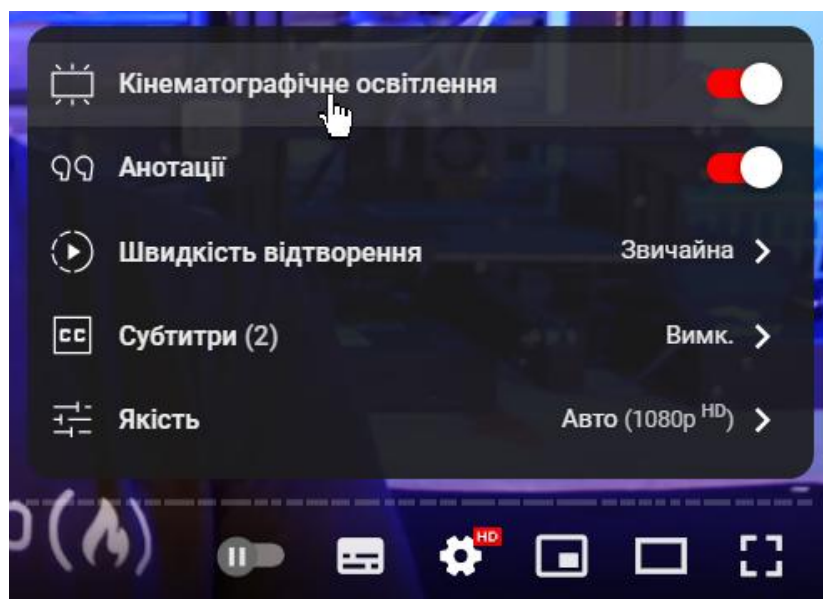


Рис. 2.11. Меню налаштувань YouTube

За допомогою програми Figma було відтворено дане меню. Така необхідність полягає тому, що за допомогою даного редактора можна швидко зробити графічний дизайн, який потім зручно реалізовувати за допомогою HTML та CSS коду. В редакторі в боковій панелі відображаються основні параметри обраного елемента, CSS код, а також є можливість експортувати SVG файли.

Після створення дизайну у редакторі Figma було створено окремі файли menu.html, style.css та script.js. У файлі menu.html описано структуру меню. Структурно меню поділене на три частини: головне меню, та два додаткові меню. Головне меню містить наступні опції: «Voice Over Subtitle» (Перекладати субтитри), «Volume» (гучність), «Subtitle» (субтитри) та «Translate into» (перекласти на). Перемикач «Voice Over» вмикає та вимикає роботу плагіна. За допомогою повзунка можна вказати гучність закадрового голосу. Опції «Subtitle» та «Translate into» відповідають за вибір субтитрів і мову їх перекладу.

Додаткові меню розміщені в окремих структурних блоках та за замовчуванням невидимі. При натисканні на опцію «Subtitle» відкривається додаткове меню, в якому перелічено список доступних субтитрів. Якщо

натиснути на «Translate into», то відкриється список мов, якими можна перекласти субтитри.

В окремому файлі «style.css» описано стилі для візуального оформлення шаблону. Позиціонування відбувається відносно головного блоку, в якому знаходяться всі інші елементи. Таким чином налаштування відступів, зміщення та інших параметрів завжди будуть залежати від стилів головного блоку. При стилізації в якості селекторів використовувались класи. Такий спосіб дає змогу призначати однакові стилі для кількох елементів, наприклад, для списку опцій, в якому кожен елемент виглядає однаково.

Назви класів створені за методологією BEM (з англ. «Block, Element, Modifier» - блок, елемент, модифікатор) – це методологія іменування класів в CSS, яка допомагає створювати структурований і легко розширюваний код. BEM надає чіткі правила для іменування класів, що забезпечують зрозумілість в розробці веб-інтерфейсів. Основна ідея BEM полягає в тому, щоб іменувати класи відповідно до їх функціональності і місця в розмітці сторінки. Кожен елемент сторінки розділяється на блоки, елементи та модифікатори.

Найменування відбувається за наступними правилами:

- назва блоку повинна характеризувати, для чого призначений даний блок. Якщо назва складається з кількох слів, то їх можна розділити символом «-». Наприклад «yt-polyglot-menu»;

- назва елемента повинна містити в собі назву блоку, в якому знаходиться елемент та назву елемента. Для розділення використовується подвійний символ нижнього підкреслення. Наприклад, «yt-polyglot-menu__item»;

- модифікатор повинен містити в собі назву блоку чи елемента та назву модифікатора. Розділення відбувається за допомогою подвійного дефісу «--». Наприклад, «.yt-polyglot-menu__item--active».

Анімація переходів між головним та додатковим меню повторює анімацію меню налаштувань YouTube. Для її реалізації було використано такі властивості CSS, як *transition*, *transform*, а також зміну властивостей при зміні модифікатора. Також для розширення можливостей анімації використано методи мови JavaScript. За допомогою методу *addEventListener* прослуховується подія «click». При спрацьовуванні методу виконується функція, що змінює CSS властивості окремих елементів, як наприклад *height* та *transform*. Це необхідно для динамічної зміни висоти відповідно до вмісту контейнера. Таким чином, якщо вміст додаткового меню складатиметься лише з 2 пунктів, то висота меню автоматично зменшиться з плавною анімацією.

На рис. 2.12 зображено результат розробленого HTML шаблону.

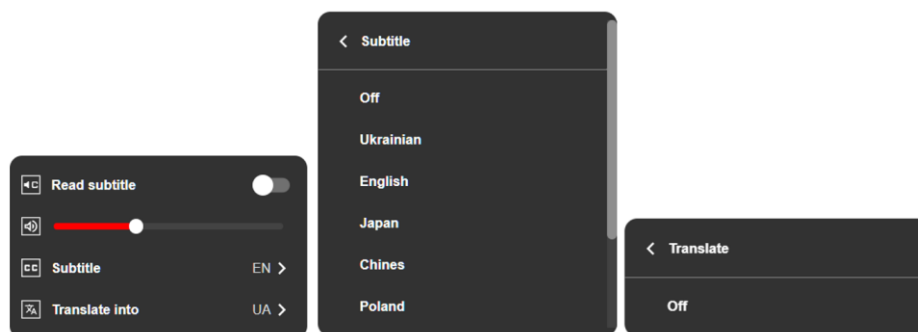
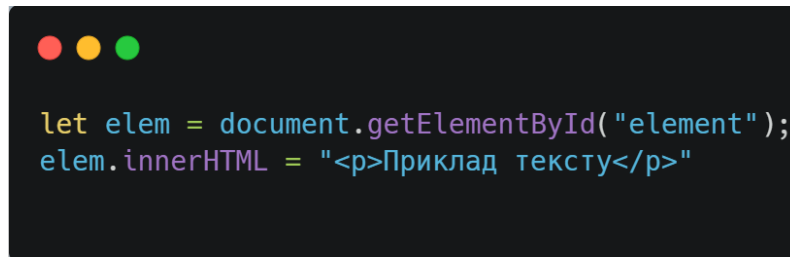


Рис. 2.12. HTML шаблон меню плагіна

2.5.2. Імплементация шаблону на сторінку

У мові програмування JavaScript існує декілька способів вставки HTML елементів та шаблонів на сторінку. Всі вони зводяться до двох типів: безпосереднього вставлення HTML коду на сторінку та створення DOM елементів за допомогою методів JavaScript.

Перший варіант передбачає використання властивості *innerHTML*. За допомогою даної властивості можна отримати доступ до HTML коду в середині елемента та перезаписати його. Нижче наведено приклад використання даного методу.

A screenshot of a code editor with a dark background. At the top left, there are three colored circles (red, yellow, green) representing window control buttons. The code is written in a light-colored font and consists of two lines:

```
let elem = document.getElementById("element");  
elem.innerHTML = "<p>Приклад тексту</p>"
```

Рис. 2.13. Приклад Використання `innerHTML`

На рис. 2.13 у першому рядку коду за допомогою методу `getElementById` отримано доступ до елемент із ідентифікатором «`element`». Далі за допомогою `innerHTML` в даний елемент вбудовано HTML код «`<p>Приклад тексту</p>`». Даний приклад наочно демонструє простоту та зручність даного методу. Попри це метод має і свої недоліки:

- при заміні вмісту елемента за допомогою `innerHTML`, всі обробники подій, прикріплені до дочірніх елементів, будуть втрачені. Це означає, що якщо є обробники подій, пов'язані з елементами, які будуть замінені через `innerHTML`, їх доведеться повторно прикріпити після заміни;
- деякі старі версії браузерів можуть не повністю підтримувати `innerHTML` або мають різні відмінності у виконанні. Це може призвести до непередбачуваної поведінки або помилок у деяких випадках;
- використання `innerHTML` може створювати потенційну вразливість до атак типу "Cross-Site Scripting" (XSS). Якщо вміст, що вставляється через `innerHTML`, містить шкідливий код JavaScript, це може призвести до виконання небажаного скрипту на сторінці.

Інший метод полягає у створенні елементів DOM за допомогою методів JavaScript. Для цього використовуються такі методи об'єкту `document`, як: `createElement`, `createTextNode`, `createElementES` та ін. За допомогою методу `setAttribute` елементу можна вказати додаткові атрибути.

При розробці було обрано другий метод з використанням вбудованих у JavaScript методів створення елементів DOM. Для того, щоб спростити створення елементів було розроблено невеличку бібліотеку, що полегшує

взаємодію із методами. В даній бібліотеці представлено два класи «DOMElement» та «SVGElement». У конструкторі класу «DOMElement» створюється новий елемент за допомогою `document.createElement(config.tag)`, де `config.tag` є значенням тегу HTML елемента, який потрібно створити. Далі, за допомогою функції `addAttributes`, додаються атрибути до створеного елемента згідно значень, переданих в `config.attributes`. Функція `addAttributes` може встановлювати атрибути елемента, такі як `class`, `id`, `style` і т.д.

Клас «SVGElement» реалізований аналогічно, різниця лише у використанні метода `createElementNS`, який призначений для створення SVG елементів.

Функція «`addAttributes`» отримує два параметри: `element`, `attributes`. За допомогою циклу `for (let ... in ...)` для елемента призначаються атрибути з масиву `attributes`, після чого повертається даний елемент.

На рис. 2.14 показано створення елемента DOM:



```
const menu = new DOMElement({
  tag: "div",
  attributes: {
    class: "yt-polyglot-menu",
  },
});
```

Рис. 2.14. Використання бібліотеки

2.5.3. SVG іконки

При розробці меню в якості додаткового графічного оформлення було використано іконки. Іконка – це невелике зображення, що графічно описує певний елемент. Вони стають в нагоді, коли для розміщення текстового опису недостатньо місця, а також в якості додаткового оформлення. Одним із

найпоширеніших форматів є «png», оскільки даний формат може зберігати зображення із прозорим фоном. Проте недоліком зображень у даному форматі є низька якість зображення при збільшенні, тому що «png» зображення є растровими.

Існує й інший формат – «svg». Зображення у даному форматі не втрачають якості при збільшенні та не виглядатимуть розмитими на екранах із високою щільністю пікселів («High DPI»).

У проєкті використано декілька svg іконок, тому для їх створення використано методи JavaScript. Код для створення іконок прописаний у файлі «svg.js». У даному файлі містяться функції, при виклику яких буде створено та повернуто svg об'єкт. Даний об'єкт можна змінювати та додавати до сторінки. Даний підхід дозволяє організувати код. Таким чином, при вбудовуванні шаблону меню для вставлення svg іконки необхідно буде лише викликати необхідну функцію.

На рис. 2.15 зображено іконки, що були використані для розробки інтерфейсу користувача.



Рис. 2.15. Іконки для інтерфейсу плагіна

2.5.4. Вбудовування меню налаштувань у програвач YouTube

Для доступу до елементів, їх атрибутів і їх вмісту використовується інтерфейс DOM (з англ. «Document Object Model» - Модель Об'єктів Документу). Він представляє структуру HTML-документа або XML-документа у вигляді дерева об'єктів.

DOM надає набір методів і властивостей, які можна використовувати для взаємодії з елементами документа. Завдяки DOM можна змінювати структуру документа, отримувати доступ до елементів, зчитувати і

змінювати їх атрибути, вміст і стилі, а також реагувати на події, такі як кліки миші або натискання клавіші.

Перед вбудовуванням меню у програвач, його необхідно створити за допомогою вбудованих методів JavaScript. Код, що відповідає за створення меню знаходиться в файлі «menu.js» у папці «js». З використанням класів власноруч написаної бібліотеки створено основні елементи меню.

Елементи вибору опцій меню структурно представляють собою блоки, які поділені на декілька частин. Вони містять в собі іконки, назву опції, додаткові підписи, поля для введення даних (зміна гучності). Тому для зручного створення таких елементів було розроблено функцію, що приймає параметри налаштування і в результаті повертає готовий елемент із потрібними вкладеними іконками та надписами.

В залежності від переданих даних у функцію, для елемента буде вказано унікальний підпис, власна іконка а також додаткові підписи, або поля введення даних. Дана функція значно зменшує кількість коду та полегшує розробку інтерфейсу меню.

Після створення, елементи інтерфейсу потрібно вкласти у батьківські блоки. Для цього було використано метод *appendChild*. Синтаксис використання методу виглядає так:

```
parentElement.appendChild(childElement);
```

де *parentElement* - це елемент, до якого необхідно додати дочірній елемент, а *childElement* -це елемент, який буде доданий як дочірній. Коли метод *appendChild* викликається, дочірній елемент буде приєднаний до кінця списку елементів батьківського елемента. Це означає, що дочірній елемент з'явиться в документі після всіх інших вже існуючих дочірніх елементів.

Додавання дочірніх елементів відбувається починаючи з нижчої ланки в дереві DOM поступово рухаючись до верхньої. Таким чином на кінцевому етапі залишиться лише одна змінна, що буде містити в собі всю структуру меню. Далі це меню за допомогою методу *appendChild* вбудовується у відео програвач (рис. 2.16).

```

190 const menuWrap = new DOMElement({
191   tag: "div",
192   attributes: {
193     class: "yt-polyglot-menu-wrap",
194   },
195 });
196 menuWrap.appendChild(menu);
197 document.getElementsByClassName("html5-video-player")[0].appendChild(menuWrap);

```

Рис. 2.16. Вбудовування меню на сторінку перегляду відео

Також для елементів меню необхідно додати прослуховувач подій. За допомогою методів *querySelector*, *querySelectorAll* отримується доступ до елементів меню. Дані методи в якості параметрів приймають CSS селектори, за якими необхідно вибрати елементи DOM. Далі за допомогою методу *addEventListener* додається прослуховувач подій. В якості параметрів, у даний метод передаються назва події (наприклад, «click») та функція, що повинна виконатись.

Таким чином, при натисканні на опцію «Voice Over Subtitle» буде відбуватись зміна вигляду іконки перемикача а також даний прослуховувач запускатиме функцію відтворення субтитрів. При виборі опції «Translate into» з'являтиметься додаткове меню для вибору мови.

Для відкриття меню необхідно додати відповідну кнопку поруч із кнопками налаштувань перегляду в YouTube плеєрі. Для цього спершу необхідно отримати елементи, в яких знаходяться дані кнопки. Використовуючи інструменти розробника в браузері було визначено, що налаштування знаходяться в елементі з класом «ytp-right-controls». За допомогою бібліотеки було створено елемент кнопки та добавлено в нього іконку розширення (рис. 2.17). З використанням методу *insertBefore* даний елемент вбудовано на сторінку перегляду відео.

```

const ytToolbar = document.querySelector(".ytp-right-controls");
const ytSettingsButton = document.querySelector(".ytp-settings-button");

const polyglotSettingsButton = new DOMElement({
  tag: "button",
  attributes: {
    class: "ytp-button ytp-polyglot-settings-button",
  },
});
polyglotSettingsButton.appendChild(polyglotToolbarSVG());

```

Рис. 2.17. Створення кнопки для відкриття налаштувань розширення

Важливо, що даний скрипт (так як і «library.js» та «svg.js») повинен виконуватись у контексті веб сторінки. Тому відповідне налаштування необхідно вказати у «manifest.json». Для цього необхідно прописати ключ «content_scripts» та в якості значення передати масив із об'єктом, що містить в собі файли, які мають виконуватись у контексті сторінки. На рис. 2.18 наведено приклад налаштування.

```

"content_scripts": [
  {
    "matches": ["https://*.youtube.com/*"],
    "js": ["js/library.js", "js/svg.js", "js/menu.js", "contentScript.js"],
    "css": ["css/style.css"]
  }
],

```

Рис. 2.18. Налаштування контексту виконання у manifest.json

У даному випадку скрипти у файлах «library.js», «svg.js», «menu.js», «content.js» у заданій послідовності будуть виконання при вході на сторінку домену «youtube.com». Також підвантажується файл «style.css».

2.5.5. Відслідковування переходів у background.js

Фоновий скрипт, що прописаний у файлі «background.js» виконується в контексті браузера. Це означає, що він має доступ до керування вкладками. Завдяки цьому є можливість відслідковування оновлення сторінки, а також зміни посилання.

Для отримання посилання та ідентифікатора (унікального коду) створено функцію *getVideoId*, яка приймає параметр *tab* – об'єкт класу *tabs* для доступу до даних відкритої вкладки (рис. 2.19).

```
const getVideoId = (tab) => {  
  const queryParameters = tab.url.split("?")[1];  
  const urlParameters = new URLSearchParams(queryParameters);  
  return urlParameters.get("v");  
};
```

Рис. 2.19. Функція *getVideoId()*

За допомогою методу *split* посилання розділяється на дві частини, у константу *queryParameters* записується друга частина. Далі створюється об'єкт типу *URLSearchParams*. Із даного об'єкту повертається значення параметра «v».

Для зв'язку та передачі даних між фоновим та контентним скриптами використовується *runtime.connect*. Для події *chrome.runtime.onConnect* вказується прослуховувач, що реєструє функцію зворотного виклику (callback), яка буде викликана під час спрацювання події *onConnect*. Функція зворотного виклику приймає аргумент *port*, який представляє з'єднання між скриптами.

У функції зворотного виклику відбувається прослуховування події оновлення вкладки. У випадку оновлення виконується функція, в яку передаються параметри *tabId*, *changeInfo*, *tab*. Виконується перевірка посилання: якщо посилання містить в собі рядок «youtube.com/watch», то за допомогою *port* надсилається повідомлення до контентного скрипту із вказаним типом повідомлення та ідентифікатором відео. Тип повідомлення вказується з метою вказання функції, що повинна обробляти повідомлення. На рис. 2.20 представлено загальний код надсилання повідомлення.

```

chrome.tabs.onUpdated.addListener((tabId, changeInfo, tab) => {
  if (tab.url && tab.url.includes("youtube.com/watch") && changeInfo.status === "complete") {
    port.postMessage({
      type: "ID",
      videoId: getVideoId(tab),
    });
  }
});

```

Рис. 2.20. Отримання коду відео та надсилання повідомлення

2.5.6. Озвучування субтитрів

Код для озвучення субтитрів реалізовано у файлі «contentScript.js». Озвучення поділяється на декілька етапів. Спершу необхідно отримати ідентифікатор відео. Далі слід виконати запит до серверу, передавши у посиланні наступні параметри: ідентифікатор відео, мова субтитрів та мова перекладу. Далі субтитри необхідно озвучити за допомогою Web Speech API.

Для того, що отримати ідентифікатор необхідно виконати команду `chrome.runtime.connect`. Вона повертає порт, за допомогою якого можна передавати та отримувати повідомлення між контекстами виконання скриптів. Для події `onMessage` налаштовується прослуховувач подій. При отриманні повідомлення виконається функція, в яку в якості параметра буде передано повідомлення.

При отриманні повідомлення виконується перевірка типу. Якщо тип «ID», то в глобальний об'єкт `settings` буде записано ідентифікатор відео. Також виконається запит до серверу для того, щоб отримати список доступних субтитрів і відповідно оновити цей список в меню налаштувань плагіна. Для цього буде викликано функцію `appendSubtitleOption` та передано результат запиту.

У функції `appendSubtitleOption` виконується видалення вмісту додаткових меню, та додавання нових варіантів вибору. Кожному варіанту присвоюється атрибут «lang-code». Даний атрибут містить в собі код мови, який використовується для визначення мови субтитрів та перекладу. Для створення елементів меню використовується раніше створена бібліотека.

Для кожного пункту також додається прослуховувач подій. У разі натискання на пункт меню в глобальний об'єкт *settings* записується значення атрибуту «lang-code».

Після налаштування плагіна виконується запит до сервера. В посиланні запиту передаються параметри у форматі «ключ=значення». Зокрема, передається ідентифікатор відео, мова субтитрів та мова перекладу. Результат запиту записується у глобальний об'єкт *settings*.

При натисканні на опцію «Voice Over Subtitle» відбувається озвучення субтитрів. Для цього на даний елемент меню призначено прослуховувач подій. Субтитри, що надсилає сервер, надходять у форматі JSON об'єкту, в якому кожен рядок містить такі дані, як текст, час початку та тривалість фрази. Ці дані використовуються для синхронізації та налаштування темпу промови.

Проте перед синтезом тексту необхідно відредагувати субтитри. Для цього використовується функція *formatTranscript*. У циклі «for (let ... in ...)» по чергово із кожного рядка символи розриву рядка замінюються на пробіл, слова що знаходяться поміж квадратних та круглих дужок видаляються, а також видаляються символи тире і дефісу, що знаходяться на початку репліки. Це виконується за допомогою методу *replaceAll* а також регулярних виразів у мові JavaScript. На рис 2.21 продемонстровано функцію *formatTranscript*.

```
const formatTranscript = (transcript) => {
  for (let i in transcript) {
    transcript[i].text = transcript[i].text.replaceAll("\n", " ");
    transcript[i].text = transcript[i].text.replaceAll(/\[.*\]|\(.*\)|- |- /g, "");
  }
  // transcript = transcript.filter((item) => item.text !== "");
  return transcript;
};
```

Рис. 2.21. Функція *formatTranscript*

Після форматування субтитрів для кожної репліки визначається швидкість відтворення. З метою побудови формули розрахунку було

проведено декілька дослідів у ході яких було визначено, що середня швидкість вимови текстових повідомлень становить 1 символ за 70 мс. Тобто для вимови повідомлення довжиною 40 символів знадобиться в середньому 2,8 с. Дане число порівнюється із тим, що задано за замовчуванням. Якщо обрахована тривалість більша, то швидкість відтворення збільшується. На рис. 2.22 наведено реалізацію даного алгоритму.

```
window.speechSynthesis.cancel();  
let speed = (line.text.length / (line.duration * 1000)) * 70;  
speed = speed < 1.0 ? 1.0 : speed;  
console.log("line : ", line.text, line.text.length, speed, line.duration);  
const tts = new SpeechSynthesisUtterance(line.text);  
tts.lang = "uk-UA";  
tts.rate = speed * 3;
```

Рис. 2.22. Формула розрахунку швидкості вимови

Далі створюється об'єкт *tts* типу *SpeechSynthesisUtterance*, в який передаються текст, швидкість відтворення та мова. Команда *window.speechSynthesis.speak()* виконується не відразу. За допомогою методу *setTimeout* виконання даної команди відкладається на час, що вказаний у параметрі *start*.

Важливим є і вимкнення відтворення. Для цього при натисканні на опцію «Voice Over Subtitle» перевіряється атрибут *status*. Якщо даний атрибут має значення *disable* то виконується команда *window.speechSynthesis.cancel()*, що зупиняє відтворення. Також у циклі «for (let ... of ...)» виконується скасування таймерів відтворення рядків, що призначалися раніше.

ВИСНОВОК ДО РОЗДІЛУ 2

У даному розділ розглянуто процес створення розширення. Дане завдання поділяється на декілька етапів: вибір засобів для реалізації, ознайомлення із методами створення розширень для браузерів, створення дизайну та шаблону інтерфейсу користувача, програмної реалізації та ін.

Для розробки було обрано редактор коду Visual Studio Code, що має ряд корисних функцій. Розширення створене за допомогою мови програмування JavaScript, оскільки це є основною мовою при розробці веб додатків. Для стилізації інтерфейсу користувача було використано мову CSS. Пошук та переклад відбувається на стороні серверу, що написаний на мові програмування Python з використанням фреймворку Flask, а також бібліотеки YouTubeTranscriptApi.

Розширення поділяється на дві частини: контентну та фонову. До контентної відноситься код, що прописаний у файлах «library.js», «svg.js», «menu.js», «contentScript.js». До фонові відноситься «background.js».

У файлі background.js відбувається слідкування за оновленням вкладки та отримання ідентифікатора відео.

У файлах «library.js», «svg.js», «menu.js» прописано код для створення та вставлення меню у відео плеєр. У «contentScript.js» відбувається отримання субтитрів та їх озвучення.

РОЗДІЛ 3

ТЕСТУВАННЯ РОЗШИРЕННЯ

3.1. Встановлення та запуск розширення

Для роботи розширення його необхідно встановити в браузер. Зазвичай розширення встановлюються із магазину додатків. Проте перед цим його необхідно завантажити. Це може зайняти певний час та витрату коштів, оскільки для публікації додатків у магазині додатків необхідно придбати відповідну ліцензію. Проте є ще один спосіб. Для цього необхідно відкрити налаштування розширень (рис. 3.1).

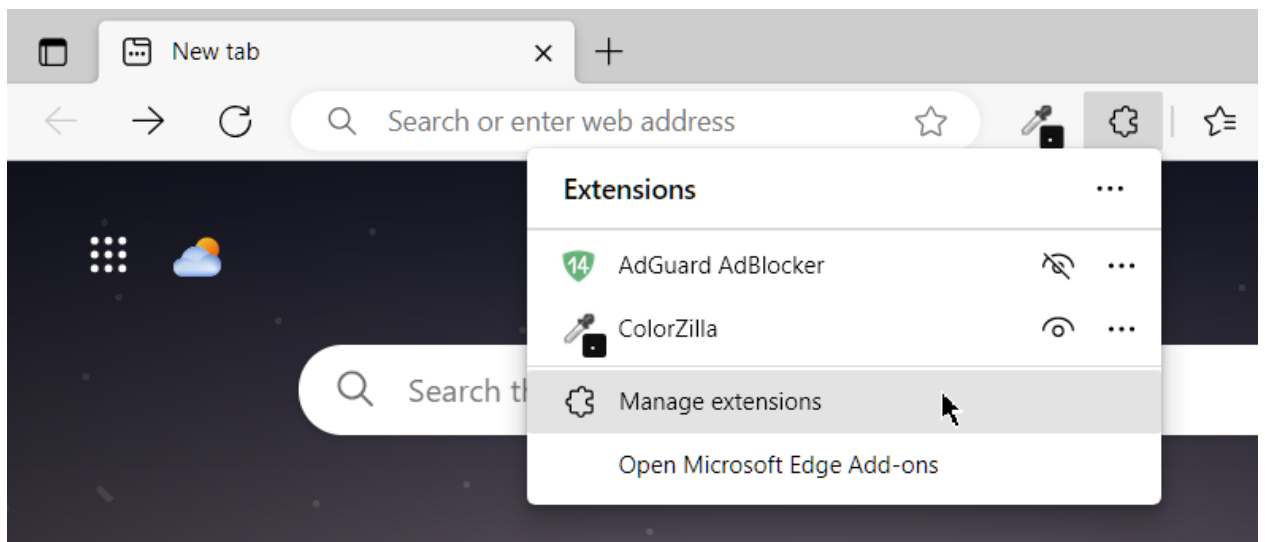


Рис. 3.1. Налаштування розширень

Після цього необхідно натиснути на піктограму «Load Unpacked». З'явиться вікно, в якому необхідно вказати папку з кодом розширення. Після цього розширення добавиться у браузері та автоматично увімкнеться.

Для роботи програми також необхідно запустити сервер. Це робиться за допомогою командного рядка, слід виконати наступну команду:

Кафедра КІТ				НАУ 23 10 14 000 ПЗ								
				РОЗДІЛ 3. ТЕСТУВАННЯ РОЗШИРЕННЯ			<i>Літ.</i>		<i>Аркуш</i>		<i>Аркушів</i>	
									53		7	
							ТП-415Б – 122					
		<i>ПІБ</i>										
<i>Розроб.</i>		Зелінський В.О.										
<i>Керівник</i>		Зудов О. М.										
<i>Н. Контр.</i>		Толстікова О.В.										

flask --app server.py run,
після виконання даної команди запуститься сервер, у консолі з'явиться відповідне повідомлення (рис. 3.2).

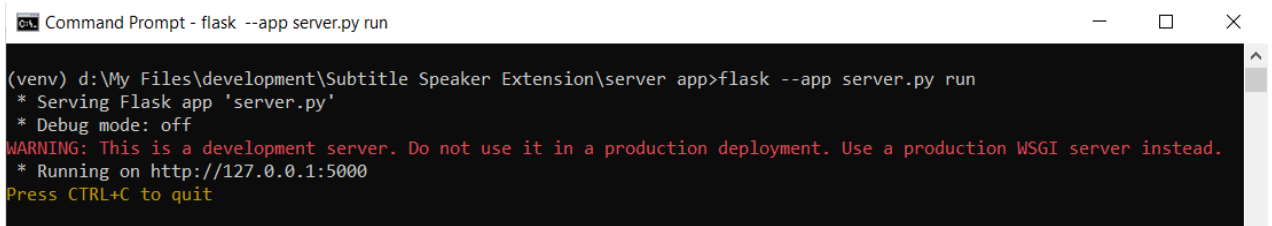


Рис. 3.2. Запуск сервера

3.2. Тестування та перевірка

На даному етапі додаток повністю готовий до роботи. У новій вкладці браузера заходимо на сайт www.youtube.com та відкриваємо англomовне відео. У цілях демонстрації обрано відео із YouTube каналу «freeCodeCamp.org». На даному каналі розміщено велика кількість навчальних відео по програмуванню.

При відкриванні плеєра поряд із кнопкою налаштувань з'явилася ще одна кнопка. При натисканні відкривається меню налаштувань плагіна (Рис.3.3).

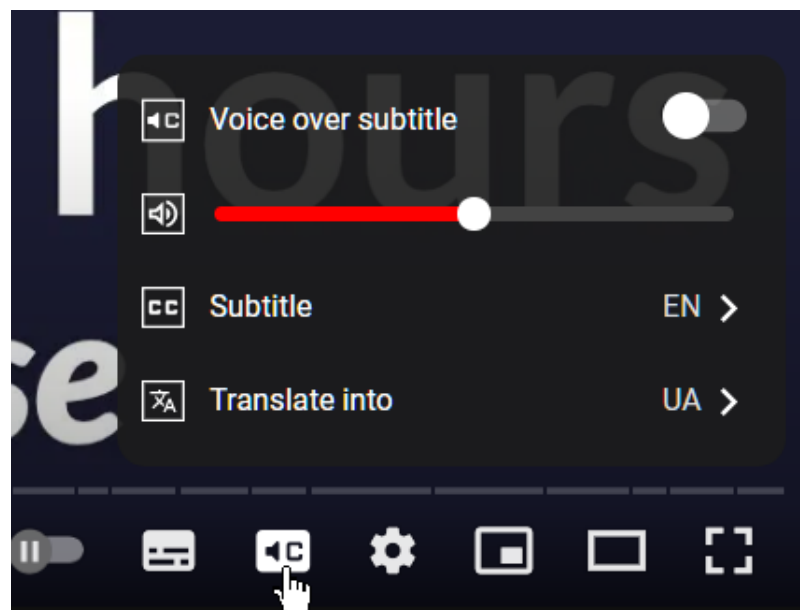


Рис. 3.3. Налаштування розширення

Опція «Subtitle» дає змогу відкрити меню вибору субтитрів. У даному вікні в списку представлені усі субтитри, що доступні для відео. Після вибору субтитрів налаштування запам'ятовується та додаткове меню закривається (рис. 3.4).

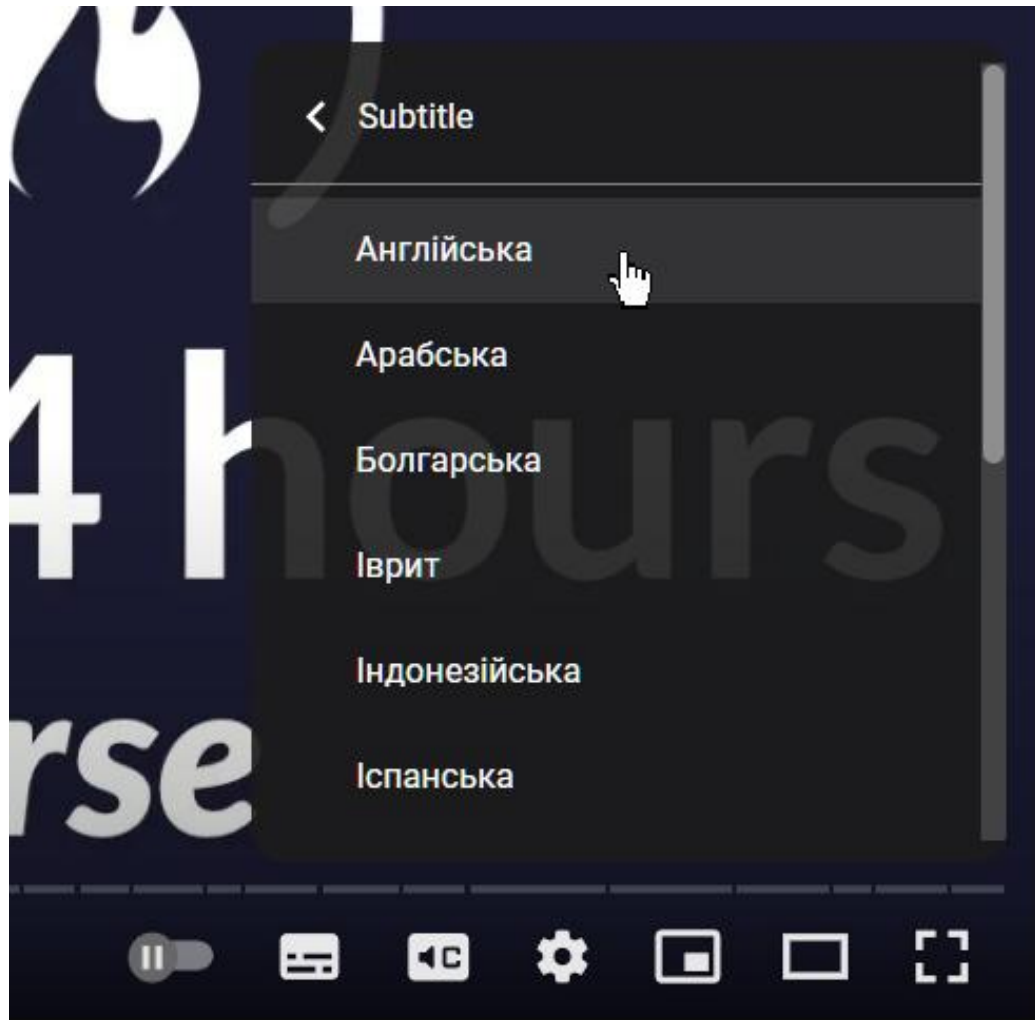


Рис. 3.4. Вибір субтитрів

Аналогічно вибирається і мова перекладу (рис. 3.5). Список доступних мов автоматично оновлюється. Для цього робиться запит до сервера, який для вказує можливі мови перекладу для доступних субтитрів.

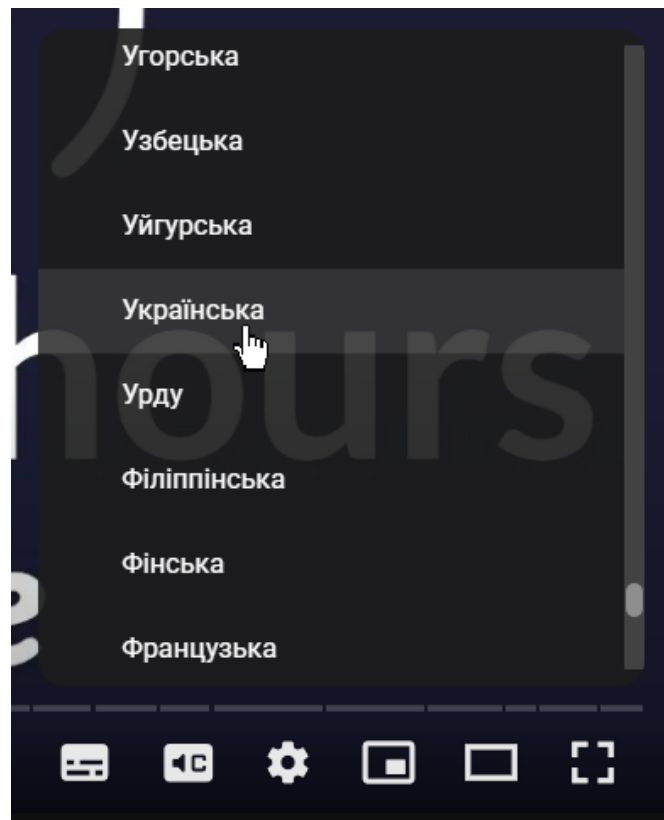


Рис. 3.5. Вибір мови перекладу

Після вибору мови з'являється можливість увімкнути відтворення. Для цього необхідно натиснути на кнопку «Voice over subtitle» та переключити перемикач у положення «on». Перекладені субтитри почнуть відтворюватися із заданою гучністю.

Для перевірки роботи розширення та якості перекладу було проведено кілька експериментів, у ході яких виявилось, що точність перекладу не завжди 100%. В залежності від тематики певні терміни можуть бути перекладені і озвучені не зовсім вірно. Наприклад назву мови програмування Python перекладено і озвучено як «пайсон» (правильно «пайтон»). Також деякі ідіоми та фрази можуть мати не пряме значення, з чим машинний переклад на сьогодні справляється погано.

Також фактором, що впливає на якість перекладу є джерело субтитрів. Якщо це субтитри, що згенеровані автоматично, в такому разі в оригінальній мові можуть міститися помилки при розпізнаванні тексту. Точність

розпізнавання залежить від багатьох факторів, серед яких: швидкість вимови тексту, наскільки чітко текст вимовляється та різний шум. Якість перекладу в такому разі ще гірша, проте загальну думку та ідею зрозуміти можна.

3.3. Можливі покращення

Незважаючи на те, що розширення повністю справляється із поставленим завданням є ряд покращень, що можливі у майбутньому. Такі покращення сприятимуть удосконаленню програмного застосунку та додаванню нових функцій.

Наприклад, підтримка декількох голосів. Завдяки можливості вибору різних голосів для озвучення субтитрів користувачі зможуть вибрати голос, який найкраще відповідає їхнім потребам або вподобанням.

Також можна задіяти методи штучного інтелекту. За допомогою методів машинного навчання можна покращити якість перекладу, виправляти помилки в тексті.

Серед можливих функцій також є редагування субтитрів. Для реалізації функції редагування субтитрів можна додати інтерактивний інтерфейс на YouTube, де користувачі зможуть вносити зміни до субтитрів та їх перекладу. Результат виправлення зберігається в базі даних. Це може включати виправлення помилок перекладу, зміну розміщення субтитрів на екрані та редагування часових міток. Користувачі зможуть надати свої внески та вдосконалити якість перекладу та озвучення.

Ще одним покращенням можуть бути користувацькі налаштування. Для цього необхідно розробити функцію, що буде зберігати налаштування за замовчуванням. Таким чином користувачу не потрібно постійно відкривати налаштування, а достатньо лише увімкнути переклад і озвучення.

Також можливо додати спеціальний словник сленгових виразів. В такому словнику будуть міститись слова та фрази, що перекладаються не зовсім вірно машинним перекладом.

Для продовження розробки застосунку, а також залучення інших розробників можна зробити відкритим вихідний код за допомогою сервісу GitHub. Завдяки цьому будь-хто зможе переглянути програмний код, завантажити та відредагувати його. Таким чином різні програмісти зможуть запропонувати свої рішення та покращення для застосунку.

У такого підходу також є свої переваги:

1. Перевірка безпеки та надійності. Завдяки відкритому вихідному коду, багато експертів може аналізувати код на предмет потенційних вразливостей та помилок безпеки. Це дозволяє швидко виявляти та виправляти проблеми безпеки, що забезпечує більшу надійність та захищеність розширення.

2. Гнучкість та адаптованість. З відкритим вихідним кодом розширення можна змінювати та адаптувати до власних потреб. Користувачі можуть вносити зміни, додавати нові функції або інтегрувати з іншими інструментами. Це дозволяє кожному користувачеві налаштувати розширення під свої індивідуальні потреби та використовувати його у своїх проектах.

3. Вільна ліцензія та розповсюдження. Завдяки вільній ліцензії, розширення може бути розповсюджуваним та використовуваним безкоштовно. Це зробить його доступним для широкої аудиторії та сприятиме його поширенню серед користувачів.

4. Взаємодія з іншими проектами: Відкритий вихідний код розширення дає можливість інтегрувати його з іншими проектами або інструментами. Це сприяє співпраці та обміну даними з іншими розробниками та сприяє створенню екосистеми зі спільними цілями та завданнями.

ВИСНОВОК ДО РОЗДІЛУ 3

У даному розділі було проведено тестування та перевірка роботи розробленого розширення. Для функціонування додатку необхідно встановити розширення в браузер, а також необхідно запустити сервер.

Після запуску розширення готове до роботи. Скрипти виконуються лише при вході на сторінку перегляду відео.

У меню YouTube програвача додається додаткова опція поруч із налаштуваннями. При натисканні з'являється меню, в якому можна налаштувати роботу плагіна: вибрати субтитри, мову, вказати гучність та ввімкнути відтворення. Швидкість відтворення субтитрів підбирається автоматично.

Точність перекладу при цьому не є 100%, проте достатня для того, щоб зрозуміти про що говорить автор. В даному випадку якість перекладу залежить від складності термінів та мовних конструкцій. При великій кількості ідіом та фразових дієслів, а також неформального стилю переклад видається дивним, проте суть відео залишається зрозумілою.

На даному етапі розширення виконує необхідні функції з перекладу та озвучення субтитрів.

ВИСНОВКИ

Із приходом Інтернету постала проблема мовного бар'єру при пошуку інформації. Викласти відео, завантажити фото, написати статтю зараз можливо знаходячись у будь якій частині світу. Відповідно інформація в мережі представлена різними мовами. Це створює перешкоди для тих, хто не володіє іноземними мовами.

У даній кваліфікаційній роботі розглянуто проблему локалізації іноземних відео на онлайн платформі YouTube. Обрано засоби проектування, описано їх переваги, а також наведено огляд кожного засобу. Детально описано процес розробки розширення для браузера та проведено тестування.

Для розробки розширення обрано редактор коду VS Code від компанії Microsoft. Даний редактор зручний та інтуїтивний у використанні, має корисні функції, такі як: підсвічування коду, автоматичне форматування, можливість встановлювати плагіни, одночасний перегляд кількох документів та ін. Основними мовами програмування є JavaScript та Python. Також використано мови CSS та HTML, оскільки це основні інструменти при розробці веб додатків.

У процесі розробки плагіну було вивчено методи створення розширення для браузера. Використано основні інструменти для розробки.

Інтерфейс користувача повністю повторює дизайн меню налаштувань програвача на YouTube. Це зроблено для того, щоб розширення не виділялося на фоні інших елементів.

Програмний застосунок умовно поділяється на дві частини: розширення та сервер. На сервері виконується пошук та переклад субтитрів. Розширення вбудовує меню у програвач відео та озвучує субтитри. Зв'язок між розширенням та сервером відбувається за допомогою HTTP запитів типу GET.

На стороні клієнта виконується послідовно кілька задач: створюється об'єкт меню, який вбудовується на сторінку, поруч із кнопкою налаштувань вставляється кнопка відкриття меню плагіна. При переході між відео списки

доступних субтитрів та мов перекладу автоматично оновлюються. При виборі певних опцій налаштування зберігаються та застосовуються. При перемиканні в режим «on» відбувається отримання від сервера субтитрів, їх форматування. За спеціальною формулою розраховується швидкість відтворення.

Серед можливих покращень може бути використання штучного інтелекту для покращення якості перекладу, створення словника сленгів, можливість редагування субтитрів, підтримка вибору різних голосів. На GitHub можна опублікувати програмний код застосунку, таким чином залучити інших розробників до покращення функціоналу. Також такий підхід сприяє покращенню безпеки програмного коду.

В результаті було отримано програмний застосунок, інтегрований у програвач відео YouTube, що перекладає та озвучує субтитри. Якість перекладу достатня для того, щоб розуміти загальну тему та основні пункти відео. Проте є помилки та неточності перекладу.

Результати кваліфікаційної роботи можуть бути використанні для ознайомлення із методами розробки розширення для браузера. Саме розширення є корисним для людей, що не розуміють мовою оригіналу відео. Воно може бути використане в навчальних, пізнавальних та розважальних цілях. Завдяки перекладеним субтитрам, користувачі можуть краще розуміти та сприймати вміст відео.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бородкіна І. Л., Бородкін Г. О. WEB-Технології та WEB-дизайн: Застосування для створення електронних ресурсів: навчальний посібник / Бородкіна І.Л., Бородкін Г.О. – Київ: Видавництво Ліра-К, 2020. – 209 с.
2. Ерік Меєр. CSS Pocket Reference: Visual Presentation for the Web 5th Edition / Ерік Меєр. – США: Видавництво O'Reilly Media, 2018. – 250 с.
3. David Flanagan. JavaScript. The Definitive Guide. Master the World's Most-Used Programming Language 7th Edition / David Flanagan. – США: Видавництво O'Reilly Media, 2020. – 706 с.
4. Luciano Ramalho. Fluent Python. Clear, Concise, and Effective Programming. 2nd Edition / Luciano Ramalho. – США: Видавництво O'Reilly Media, 2022. – 234 с.
5. Рейтинг мов програмування 2023. JavaScript/TypeScript завойовують світ, Python увійшов у топ-3, Salesforce Apex випередив 1С | DOU [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/language-rating-2023/>
6. YouTube — Вікіпедія (wikipedia.org) [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/YouTube>
7. YouTube | History, Founders, & Facts | Britannica [Електронний ресурс] – Режим доступу до ресурсу: <https://www.britannica.com/topic/YouTube>
8. Top Websites ranking - Most Visited Websites in the world [April 2023] | Semrush [Електронний ресурс] – Режим доступу до ресурсу: <https://www.semrush.com/website/top/>
9. EF EPI 2022 – EF English Proficiency Index [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ef.com/wwen/epi/>
10. API reference - Chrome Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.chrome.com/docs/extensions/reference/>
11. Welcome to the Chrome Extension Manifest V3 - Chrome Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.chrome.com/docs/extensions/mv3/intro/>

12. REST - MDN Web Docs Glossary: Definitions of Web-related terms |
MDN (mozilla.org) [Электронный ресурс] – Режим доступа до ресурсу:
<https://developer.mozilla.org/en-US/docs/Glossary/REST>