

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” 2024 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

на тему: Плагін для Chromium для фільтрації контенту

Виконала: студентка 4 курсу, групи ІО-01
(шифр групи)

Бібік Міла Сергіївна

(прізвище, ім’я, по батькові)

(підпис)

Керівник асистент кафедри ОТ, Гайдай А. Р.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) ст. викладач Виноградов Ю. М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент доцент кафедри ІСТ, к.т.н., Шимкович В. М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

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

Студент _____
(підпис)

КИЇВ 2024 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ____ ” _____ 2024 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт

студенту Бібік Мілі Сергіївни

1. Тема проєкту Плагін для Chromium для фільтрації контенту

керівник проєкту Гайдай Анатолій Русланович, асистент кафедри ОТ,

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «27» травня 2024 р. №2112-с

2. Термін здачі студентом закінченого проєкту: 3 червня 2024 р.

3. Вихідні дані до проєкту: технічна документація, теоретичні дані з проєктування плагінів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)

Аналіз предметної області та рішень.

Аналіз та вибір засобів реалізації.

Проектування та реалізація системи.

Огляд та тестування системи.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) структурна схема, функціональна схема, принципова схема.

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультація	Підпис, дата	
		Завдання видав	Завдання Прийняв
Нормоконтроль	Виноградов Ю.М.		

7. Дата видачі завдання: «31» жовтня 2023 р.

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	Затвердження теми проєкту	22.12.2023	
2.	Вивчення та аналіз завдання	23.12.2023-12.03.2024	
3.	Розробка архітектури та загальної структури системи	12.03.2024-29.03.2024	
4.	Розробка структур підсистем	29.03.2024-10.04.2024	
5.	Програмна реалізація системи	10.04.2024-22.04.2024	
6.	Оформлення пояснювальної записки	22.04.2024-17.05.2024	
7.	Захист програмного продукту	02.05.2024	
8.	Передзахист	07.06.2024	
9.	Захист	21.06.2024	

Студент-дипломник _____ Міла БІБІК
(підпис)

Керівник проєкту _____ Анатолій ГАЙДАЙ
(підпис)

АНОТАЦІЯ

Даний дипломний проєкт присвячений розробці програми для фільтрації контенту на основі моделі машинного навчання. Було проаналізовано існуючі рішення, що мають подібний функціонал, враховані їхні сильні та слабкі сторони. Обрано відповідні технології для розробки системи. Робота складається з чотирьох етапів: аналіз, проєктування, розробка та тестування системи.

Систему реалізовано у вигляді браузерного розширення. Створена система націлена на створення безпечного середовища для українських користувачів в мережі інтернет.

Ключові слова: браузерне розширення, машинне навчання, фільтрація контенту, алгоритми класифікації.

ANNOTATION

This diploma project is devoted to developing a content filtering program based on a machine learning model. Existing solutions with similar functionality were analyzed, and their strengths and weaknesses were considered. The appropriate technologies for system development have been chosen. The work consists of four stages: analysis, design, development, and testing of the system.

The system is implemented as a browser extension. The system is designed to provide a safe environment for Ukrainian internet users.

Keywords: browser extension, machine learning, content filtering, classification algorithms.

ТЕХНІЧНЕ ЗАВДАННЯ ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «Плагін для Chromium для фільтрації контенту»

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.467200.002 ТЗ			
Зм.	Арк.	№ докум	Підпис	Дата	Плагін для Chromium для фільтрації контенту Технічне завдання	Літ.	Аркуш	Аркушів
Розробила	Бібік М.С.							
Перевірив	Гайдай А.Р.						1	4
						КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО-01		
Н. контр.	Виноградов Ю.М.							
Затверд.	Стіренко С.Г.							

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання передбачає розробку програмного модуля для фільтрації текстового онлайн-контенту, а також його подальшу підтримку та вдосконалення.

Область застосування цього модуля охоплює створення безпечного інтернет-простору для користувачів, зокрема для дітей та підлітків.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даної системи є завдання для виконання роботи кваліфікаційно-освітнього рівня «бакалавр комп'ютерної інженерії», який був затверджений факультетом “Інформатики та обчислювальної техніки” кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою та призначенням даної роботи є розробка плагіна для вебоглядачів, який дозволить користувачам інтернету створювати та налаштовувати комфортний та безпечний доступ до інформації.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки даного дипломного проєкту є офіційна документація, публікації та статті в мережі інтернет на тему дипломного проєкту, науково-технічна література.

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Арк.	№ докум	Підпис	Дата		

5. ТЕХНІЧНІ ВИМОГИ

5.1 Розроблений програмний модуль має відповідати таким вимогам:

- Простий та інтуїтивно зрозумілий користувацький інтерфейс.
- Автоматична цензура неприйнятої лексики в режимі реального часу.
- Можливість користувача налаштовувати рівень фільтрації.
- Можливість користувача додавати власні небажані слова та фрази.

5.2 Вимоги до програмного забезпечення:

- Операційна система Windows 10.
- Наявність веббраузера, основанийого на Chromium.

5.3 Вимоги до апаратної частини:

- Наявність інтернет-з'єднання.
- 8 або більше ГБ оперативної пам'яті.

					ІАЛЦ.467200.002 ТЗ	Арк.
						3
Зм.	Арк.	№ докум	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

Назва етапів виконання	Термін виконання
Затвердження теми роботи	22.12.2023
Вивчення та аналіз завдання	23.12.2023-12.03.2024
Розробка архітектури та загальної структури системи	12.03.2024-29.03.2024
Розробка структур окремих частин системи	29.03.2024-10.04.2024
Програмна реалізація системи	10.04.2024-22.04.2024
Виправлення помилок	22.04.2024-30.04.2024
Оформлення пояснювальної записки	30.04.2024-02.05.2024

					ІАЛЦ.467200.002 ТЗ	Арк.
						4
Зм.	Арк.	№ докум	Підпис	Дата		

ПОЯСНЮВАЛЬНА ЗАПИСКА ДО ДИПЛОМНОГО ПРОЄКТУ

на тему: «Плагін для Chromium для фільтрації контенту»

КИЇВ 2024

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА РІШЕНЬ.....	4
1.1 Опис та аналіз предметної області.....	4
1.2 Постановка задачі.....	6
1.3 Огляд наявних рішень.....	7
ВИСНОВОК ДО РОЗДІЛУ 1.....	14
РОЗДІЛ 2. АНАЛІЗ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	15
2.1 Вибір та обґрунтування використання мови розробки для плагіна.....	15
2.2 Вибір та обґрунтування використання мови розробки для моделі машинного навчання.....	17
2.3 Вибір та обґрунтування використання платформи.....	18
2.4 Обробка та керування текстовими даними.....	22
2.5 Інструменти розробки.....	23
ВИСНОВОК ДО РОЗДІЛУ 2.....	27
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ.....	28
3.1 Архітектура програмної системи.....	28
3.2 Реалізація програмної системи.....	33
ВИСНОВОК ДО РОЗДІЛУ 3.....	40
РОЗДІЛ 4. ОГЛЯД ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЯ.....	41
ВИСНОВОК ДО РОЗДІЛУ 4.....	52

					ІАЛЦ.467200.003 ПЗ			
Зм.	Арк.	№ докум	Підпис	Дата				
Розробила		Бібік М.С.			Плагін для Chromium для фільтрації контенту Пояснювальна записка	Літ.	Аркуш	Аркушів
Перевірив		Гайдай А.Р.					1	55
						КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО-01		
Н. контр.		Виноградов Ю.М.						
Затверд.		Стіренко С.Г.						

ВИСНОВКИ.....53

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....55

					ІАЛЦ.467200.003 ПЗ	Арк.
						2
Зм.	Арк.	№ докум	Підпис	Дата		

ВСТУП

Близько 80% українців використовують мережу інтернет щоденно. Цей показник постійно зростає. Інтернет надає безліч можливостей. Однак разом з цим він також приносить ризики, пов'язані з небажаним контентом. Особливо це стосується молодих користувачів, які можуть стати жертвами цього негативного впливу.

Мета даного проекту — розробка плагіна для браузерів, який буде фільтрувати виявляє та блокує нецензурну лексику, тим самим забезпечуючи безпечне та позитивне середовище для онлайн-користувачів.

Цей плагін буде використовувати список слів для фільтрації, складений моделлю машинного навчання. Він буде аналізувати текстовий контент вебсторінок в режимі реального часу та блокувати небажані слова або фрази. Крім того, він буде надавати користувачам можливість налаштовувати рівень фільтрації відповідно до їхніх потреб.

У цій роботі я буду створювати та тренувати модель машинного навчання для виявлення небажаного контенту, а також розроблю плагін для веббраузера. Я також проведу тестування та оціню ефективність плагіна.

					ІАЛЦ.467200.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум	Підпис	Дата		

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА РІШЕНЬ

1.1 Опис та аналіз предметної області

Інтернет постає перед нами як потужний інструмент, що відкриває доступ до невичерпного джерела інформації. Однією з його ключових особливостей є можливість для кожного користувача публікувати власний контент. Завдяки цьому люди мають змогу вести особисті блоги, активно використовувати соціальні мережі, брати участь у форумах та дискусійних групах, створювати власні вебсайти, а також генерувати або редагувати інформацію на платформах спільного редагування. Через простоту публікації контенту в інтернеті, мережа постійно заповнюється небажаною лексикою, що несе негативні наслідки для більшості користувачів. Вплив небажаного контенту може варіюватись залежно від віку, емоційного стану, особистого досвіду людини та інших факторів, проте найбільш вразливими категоріями залишаються діти та підлітки.

Серед проблем, які виникають, можна виділити:

- поширення ненависті та дискримінаційних висловлювань, що призводить до підризу довіри до інституцій та розпалювання ворожнечі;
- зростання кібербулінгу та онлайн-переслідувань, що має негативний вплив на психічне та емоційне здоров'я людини;
- порушення норм поведінки, що може вплинути на формування звички використання негативної культури спілкування та загальних норм поведінки;
- розповсюдження нелегальної діяльності, торгівля наркотиками, насильство та інші злочини;

					ІАЛЦ.467200.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум	Підпис	Дата		

— погіршення когнітивних здібностей, особливо концентрації та пам'яті;

Описані проблеми призводять до негативного впливу на формування світогляду, ціннісних орієнтирів, емоційної стійкості, погіршують психоемоційний стан людини, розвиваючи сильну тривогу, агресію, депресію, різноманітні страхи, зниження самооцінки [1]. Постає гостра проблема у створенні інструментів, що допоможуть зменшити, а в ідеальному варіанті зупинити, поширення такого впливу для кожного користувача. Таким чином одним із ключових аспектів безпечного та відповідального користування інтернетом виступає фільтрація неприйнятної лексики.

Фільтрація мовного контенту — це взаємопов'язані процеси розпізнавання й приховування неприйнятого або небажаного контенту з тексту. Цей механізм може використовуватися для боротьби з описаними проблемами, а також для підтримки таких цілей, як створення безпечного середовища та захист психічного й емоційного здоров'я користувача.

Існує багато різних типів небажаного текстового контенту, який можна фільтрувати, зокрема ненормативна лексика, мова ненависті, жорстокість, сексуальний контент, матеріали, що пропагують насильство або тероризм. Багато організацій уже впровадили програмні рішення для фільтрації небажаного контенту. У школах піклуються про безпечне навчальне середовище, а уряди мають на меті захистити користувачів від контенту, який становитиме загрозу для національної безпеки. До таких програм належать Websense, Net Nanny, Deep Packet Inspection та CyberPatrol.

Ці програми дозволяють блокувати потенційно небезпечні запити та сайти, які вважаються неприйнятними. Проте вони не позбавлені недоліків, тому що іноді вони блокують цілком нормальні сайти, а також не завжди можуть впоратися з контентом, що генерується користувачами. Деякі сайти з відгуками, сайти форумів та соціальних мереж, а також сайти з новинними статтями, дозволяють своїм користувачам вільно висловлюватися, навіть якщо ці вираження містять ненормативну лексику. Приклади

					ІАЛЦ.467200.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум	Підпис	Дата		

таких сайтів: Yelp, TripAdvisor, Reddit, Gab та інші. Користувачі мають бути обережними при відвідуванні таких сайтів і критично оцінювати контент, який вони там знаходять.

Розробка нових та вдосконалених методів фільтрації текстового контенту може бути здійснена за допомогою штучного інтелекту та машинного навчання для більш точного виявлення та видалення або цензурування небажаного контенту.

Необхідно зазначити, що цензура тексту є достатньо складною темою, а тому існує ризик, що надмірна цензура може обмежувати загальноприйняте право на свободу слова. Важливо розробляти та використовувати плагіни для цензури тексту свідомо та відповідально, щоб збалансувати захист від шкідливого контенту, залишаючи доступ до інформації.

1.2 Постановка задачі

Завданням проєкту є створення програмного модуля, або плагіна, на основі даних, отриманих за допомогою моделі машинного навчання, для вебоглядачів, який використовуватиметься для автоматизації цензури текстового контенту в режимі реального часу, що допоможе в підсумку гарантувати безпечне онлайн-середовище для користувача мережі. Цільовою аудиторією проєкту є українські користувачі.

Основні можливості програми будуть зосереджені на таких функціях:

- Ідентифікація та блокування мови ненависті, образливих слів, спаму та іншого небажаного контенту.
- Налаштування рівня фільтрації відповідно до індивідуальних потреб користувача.
- Підтримка української мови й адаптація до українського контексту

Важливими аспектами системи є відкритість коду інструментів, що будуть задіяні, а також те, що система є безкоштовною у використанні. Це сприятиметь швидкому розвитку та оптимізації програмного модуля.

					ІАЛЦ.467200.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум	Підпис	Дата		

1.3 Огляд наявних рішень

Плагін програмного забезпечення — це програмний модуль, який розширює функціональність основної програми. Він дозволяє користувачам налаштовувати або доповнювати програмне забезпечення відповідно до їхніх потреб без необхідності змінювати основний код програми або встановлювати окреме програмне забезпечення.

Концепція плагінів для веббраузерів з'явилися в 1995 році, коли компанія Netscape Communications випустила Netscape Navigator 2.0 з підтримкою Netscape Plugin API [2]. Цей API дозволив розробникам створювати модулі, які могли розширювати браузер, додаючи до них нові функції, такі як відтворення мультимедіа або підтримка нових вебстандартів. Плагіни швидко стали популярними, і до кінця 1990-х років існували сотні плагінів для різноманітних цілей, від відтворення мультимедійних ігор до електронної комерції.

Плагіни для веббраузерів пропонують ряд переваг порівняно з використанням окремих бібліотек або програмного забезпечення. Насамперед це забезпечення зручності у використанні, тому що розширення достатньо легко встановлюються, а також вони налаштовуються безпосередньо з веббраузера й, як правило, мають простий та зручний інтерфейс. Окрім того, плагіни зазвичай регулярно оновлюються, щоб гарантувати їхню безпеку та сумісність з найновішою версією веббраузера. Зазвичай плагіни займають менше місця, ніж бібліотеки та програмне забезпечення, оскільки вони встановлюються та використовуються безпосередньо з веббраузера.

Вебоглядачі — це програмні додатки для взаємодії користувача з гіпертекстовими вебсторінками. З моменту свого народження в 1990 році веббраузери пройшли довгий шлях, еволюціонуючи від простих текстових інтерфейсів до потужних інструментів, які ми використовуємо сьогодні.

Наприкінці 1980-х років з'явилися перші програми з гіперпосиланнями, які

					ІАЛЦ.467200.003 ПЗ	Арк.
						7
Зм.	Арк.	№ докум	Підпис	Дата		

стали попередниками сучасних веббраузерів. Ці ранні програми, насамперед такі як Gopher і Lynx, дозволяли користувачам переходити між вебсторінками, але не мали багатьох функцій, які користувачі звикли бачити у сучасних браузерах.

Lynx було розроблено в 1992 році Кевіном Кларком [3]. Користувачі взаємодіяли з ним за допомогою командного рядка. Вони дізнавались команди з вбудованої довідки Lynx, яка містить опис всіх доступних команд та їхнього використання.

```
Lynx Information

Lynx
Lynx is a text browser for the World Wide Web.  Lynx 2.8.8 runs on
Un*x, MacOS, VMS, Windows 95/98/NT, DOS386+ (but not 3.1, 3.11), as
well as OS/2 EMX. The current developmental version (2.8.9) is also
available for testing.

* Many user questions are answered in the online help provided with
  Lynx. Press the '?' key to find this help.
* If you are encountering difficulty with Lynx you may write to
  lynx-dev@nongnu.org. Be as detailed as you can about the URL where
  you were on the Web when you had trouble, what you did, what Lynx
  version you have (try '=' key), and what OS you have. If you are
  using an older version, you may well need to upgrade.

-----

Maintained by web@lynx.browser.org.

http://lynx.invisible-island.net/release/
```

Рисунок 1.1 – Командний рядок браузера Lynx

У 1990 році британський спеціаліст і автор видатних розробок в галузі інформаційних технологій на ім'я Тім Бернерс-Лі, а також один із творців Всесвітньої павутини, під час роботи в науково-дослідницькому інституті в Швейцарії розробив перший веббраузер під назвою WorldWideWeb. Цей браузер

					ІАЛЦ.467200.003 ПЗ	Арк.
						8
Зм.	Арк.	№ докум	Підпис	Дата		

був текстовим і не мав графічного інтерфейсу користувача. Задля того, щоб уникнути плутанини між програмою та інформаційним простором, WorldWideWeb було перейменовано на Nexus.

1990-ті та 2000-ті роки ознаменувалися жорсткою конкуренцією між Netscape Navigator та Internet Explorer. Netscape Navigator, який спочатку домінував на ринку, врешті-решт поступився місцем Internet Explorer, що здобув перемогу завдяки своїй інтеграції з Windows. Internet Explorer був попередньо встановлений на всіх комп'ютерах з операційною системою Windows, що давало йому значну перевагу в досяжності користувачів. Окрім цього, повільне реагування на нові технології та функції та несвоєчасне впровадження їх у Netscape Navigator дозволило Internet Explorer перевершити його.

2008 рік став визначним в історії розвитку вебглядачів. Google випускає Chrome, який стрімко здобув популярність завдяки своїй швидкості, безпеці та простоті використання, зрештою ставши найпопулярнішим веббраузером у світі в 2011 році.

Chrome ґрунтується на Chromium, браузері з відкритим кодом, який став основою для багатьох популярних браузерів, таких як Microsoft Edge, Opera та Brave. Chromium пропонує швидкість, безпеку та широкий спектр функцій, що робить його чудовим вибором для користувачів, які прагнуть до зручного та продуктивного вебперегляду. Браузери на основі Chromium мають багато спільних рис, зокрема інтерфейс користувача, базові функції та підтримку вебстандартів. Це робить їх зручними для користувачів, при переході з одного браузера на інший, їм не потрібно заново звикати до нового інтерфейсу чи шукати знайомі функції.

Однією з найважливіших переваг браузерів на основі Chromium є їхня загальнодоступна розширюваність. Завдяки величезному магазину розширень багатьох веббраузерів користувачі можуть легко знайти безліч безкоштовних та платних розширень та плагінів, які додають новий функціонал до веббраузера, змінюють інтерфейс веббраузера або блокують докучливу рекламу [4]. Це дозволяє

					ІАЛЦ.467200.003 ПЗ	Арк.
						9
Зм.	Арк.	№ докум	Підпис	Дата		

налаштувати браузер під свої індивідуальні потреби та вподобання. Плагіни, розроблені для одного браузера, часто сумісні з іншим браузером цієї категорії.

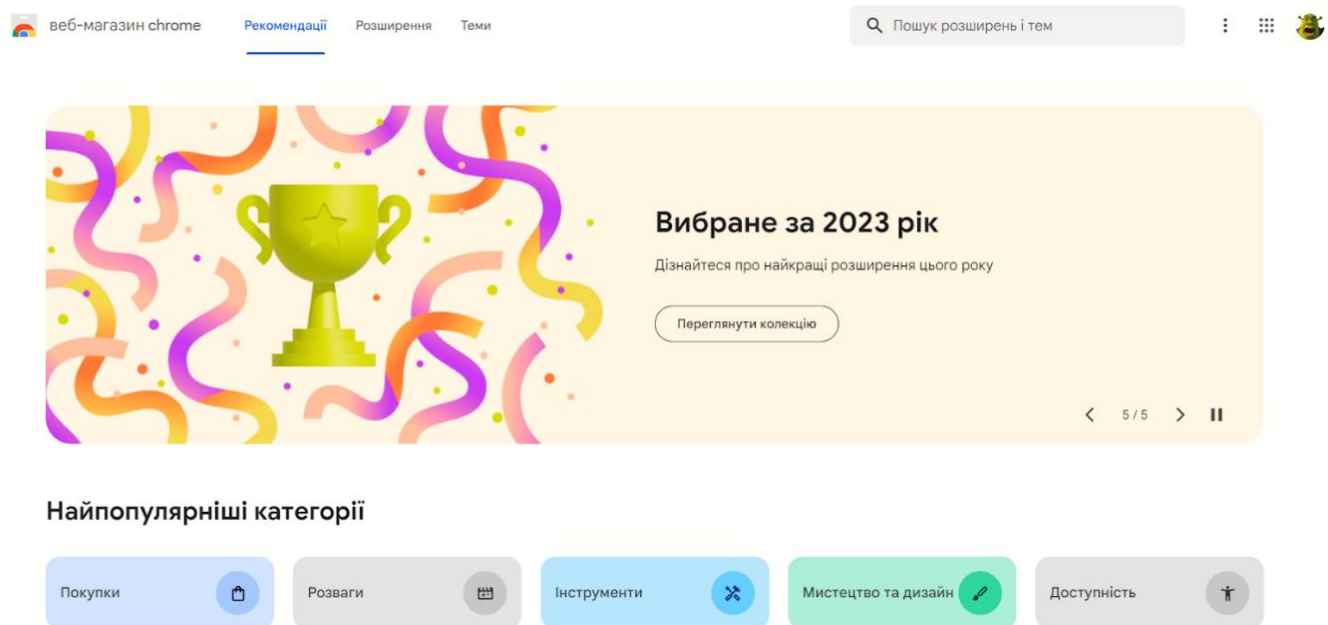


Рисунок 1.2 – Інтерфейс вебмагазину Chrome

Однак не всі браузери на основі Chromium є однаковими. Деякі з них, такі як Microsoft Edge, мають додаткові функції та інтеграцію з іншими продуктами Microsoft. Інші, наприклад Brave, особливо зосереджуються на конфіденційності та захисті даних користувачів, пропонуючи специфічні функції, як-от вбудований блокувальник реклами, спеціальну пошукову систему, яка за замовчуванням не відстежує історію пошуку користувача, а також автоматичний захист даних від перехоплення.

Користувачам, які потребують специфічних функцій, в залежності від потреб може бути краще використовувати альтернативний браузер, який не ґрунтується на Chromium. Наприклад, веббраузер Mozilla Firefox відомий своїми налаштуваннями конфіденційності та сильним фокусом на захисті користувачів, а веббраузер Safari пропонує глибоку й достатньо зручну для багатьох користувачів інтеграцію з екосистемою Apple. При виборі браузера користувачу обов’язково необхідно

					ІАЛЦ.467200.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум	Підпис	Дата		

враховувати різні фактори, зокрема визначитись з функціями, які є найбільш важливими для користувача.

Найпоширеніші оглядачі для вебплагінів:

- Google Chrome є найпопулярнішим браузером у світі з часткою ринку понад 65%. Він підтримує широкий спектр плагінів.
- Mozilla Firefox є другим за популярністю браузером, його обирають близько 9% користувачів.
- Safari є браузером за замовчуванням для операційних систем macOS та iOS. Він має трохи меншу частку ринку, ніж Chrome та Firefox, але все ж таки є популярним вибором для користувачів Apple. Safari підтримує плагіни, але їхня кількість обмежена порівняно з Chrome та Firefox.
- Opera має вбудований VPN, який забезпечує конфіденційне під'єднання до інтернету. Opera використовують близько 2,4% користувачів. Цей оглядач має власні вбудовані плагіни, але підтримує й інші плагіни.
- Brave не є домінуючим гравцем на ринку, його обирають 0,5% користувачів. У Brave доступні плагіни можна знайти у магазині плагінів та розширень веббраузера.
- Microsoft Edge є веббраузером за замовчуванням для Windows 10. Він має подібну частку ринку, як Safari, і підтримує плагіни, але їхня кількість також обмежена.

Серед доступних плагінів є велика кількість плагінів для фільтрації тексту, однак, як правило, всі вони призначені для англомовних користувачів і англійських текстів. Це пов'язано з тим, що англійська мова є найбільш поширеною на просторах інтернету, відповідно саме для неї розроблено більше алгоритмів та інструментів.

Англійська та українська мови мають суттєві відмінності в граматичних структурах та наборах слів, що унеможливорює використання наявних плагінів для фільтрування текстів, написаних українською мовою. Додатковою проблемою є використання українськими користувачами запозиченої неприйнятої лексики з

					ІАЛЦ.467200.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум	Підпис	Дата		

інших мов, переважно з англійської мови та російської мови. Виникає гостра необхідність створити базу даних неприйнятої лексики, яка буде містити поширені слова та вирази, які вважаються образливими й шкідливими в українському суспільстві [5].

Фільтрація контенту в Україні має також враховувати культурні особливості та цінності українського суспільства. Це означає, що деякі слова або висловлювання, які можуть вважатися неприйнятими в інших країнах, можуть бути цілком прийнятими в українському суспільстві й навпаки. Не існує єдиного підходу до фільтрації контенту, який би підходив для всіх культур.

Серед прикладів того, як культурні особливості можуть впливати на фільтрацію контенту можна виділити такі категорії:

- гумор;
- релігія;
- сексуальність.

Наприклад, те, що вважається смішним у одній культурі, може бути образливим або неприйнятним в іншій. Чорний гумор може бути поширеним для однієї культури або суспільства, але бути табуйованим в інших.

Релігійні переконання можуть мати сильний вплив на те, що люди вважають прийнятним та неприйнятним. Наприклад, деякі релігії забороняють зображення певних релігійних фігур або символів.

Рівень сексуальної відкритості може значно варіюватися в різних культурах. Те, що вважається прийнятним у одній культурі, може бути табуйованим в іншій.

Для того, щоб зробити фільтрацію контенту в Україні більш чутливою залучити до процесу прийняття рішень представників різних верств суспільства. Це допоможе гарантувати, що різні точки зору будуть враховані.

Доречно буде окреслити та використовувати чіткі та прозорі критерії для визначення того, який контент вважається неприйнятним. Ці критерії можуть ґрунтуватися на українських цінностях та законах.

					ІАЛЦ.467200.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум	Підпис	Дата		

Також необхідно обов'язково забезпечити можливість оскарження рішень про видалення контенту. Це допоможе гарантувати, що жоден контент не буде видалено несправедливо.

					ІАЛЦ.467200.003 ПЗ	Арк.
						13
Зм.	Арк.	№ докум	Підпис	Дата		

ВИСНОВОК ДО РОЗДІЛУ 1

Небажаний контент у мережі несе значну небезпеку, особливо для юних користувачів. Його негативний вплив на дітей та підлітків може бути руйнівним, тому важливо вживати заходів для його мінімізації. Фільтрація текстового контенту може стати дієвим інструментом у цій боротьбі. Однак наразі бракує плагінів, які б ефективно фільтрували небажаний контент українською мовою та з урахуванням українського контексту. Розробка такого плагіна могла би суттєво покращити безпеку українських користувачів в інтернеті.

При цьому важливо чітко окреслити етичні аспекти цензури тексту, щоб не ущемляти свободу слова. Необхідно знайти баланс між захистом користувачів від шкідливого контенту та збереженням права на вільне висловлення думок. Це складне завдання, яке потребує ретельного аналізу та обговорення.

Створення плагіна для фільтрації текстового контенту українською мовою може стати значним кроком уперед на шляху до безпечного та відповідального користування інтернетом в Україні.

					ІАЛЦ.467200.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум	Підпис	Дата		

РОЗДІЛ 2

АНАЛІЗ ТА ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

Створення плагіна для фільтрації контенту в браузерях ставить перед розробником унікальні завдання, пов'язані з обробкою текстової інформації, забезпеченням гнучкості налаштування та високою швидкістю роботи. Створення моделі машинного навчання для класифікації тексту для браузерного плагіна фільтрації контенту також ставить перед розробником низку унікальних завдань, таких як забезпечення ефективного розуміння контексту та уникнення упередженості. У цьому розділі буде проведено аналіз й опис технологій та інструментів, які можуть бути використані для успішної реалізації вимог проєкту.

2.1 Вибір та обґрунтування використання мови розробки для плагіна

Мови програмування для браузерних розширень мають відповідати декільком критеріям. Оскільки браузері виконують код без попередньої компіляції, мова програмування повинна мати інтерпретатор, який може безпосередньо читати та виконувати інструкції. Розширенням браузера необхідно мати можливість взаємодіяти з вебсторінками та змінювати їхню поведінку. Для цього мова програмування в обов'язковому порядку повинна мати доступ до об'єктної моделі документа та API браузера. Також важливо, щоб мова програмування забезпечувала безпечне виконання коду розширення в ізольованому середовищі. Це запобігає шкідливим розширенням від доступу до конфіденційних даних або системних ресурсів користувача.

JavaScript є однією з найпоширеніших мов програмування у світі. Згідно з опитуванням на платформі Stack Overflow у 2023 році, 67.9% розробників

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум	Підпис	Дата		

програмного забезпечення використовують JavaScript частіше, ніж будь-яку іншу мову [6]. Він підтримується всіма сучасними браузерами, що робить його ідеальним для розробки вебсайтів та плагінів, які повинні працювати на різних платформах і пристроях.

JavaScript, створений у 1995 році компанією Netscape Communications, спочатку мав назву Mocha. Він був задуманий як спосіб зробити вебсторінки більш динамічними та інтерактивними, додаючи їм нові можливості, яких не вистачало статичним HTML-документам того часу.

JavaScript дозволив розробникам обробляти дії користувачів, оновлювати вміст сторінки без попереднього перезавантаження та створювати загалом більш захоплюючий користувацький досвід.

Хоча спочатку JavaScript не підтримувався іншими браузерами, його популярність швидко зросла завдяки простоті використання та гнучкості. У 1997 році мова була стандартизована європейська асоціація виробників комп'ютерів під назвою ECMAScript. Це призвело до більшої узгодженості та сумісності між браузерами, що ще більше стимулювало прийняття JavaScript розробниками.

З часом мова програмування JavaScript значно еволюціонувала, ставши потужною мовою програмування з широким спектром можливостей. Завдяки цьому сьогодні він використовується не лише для створення динамічних вебсторінок, але й для розробки серверних застосунків, як-от Node.js, мобільних додатків, як-от React Native або Flutter, ігор та багато іншого.

JavaScript — це дуже гнучка мова, яка може використовуватися для вирішення широкого кола завдань. Вона підтримує різні парадигми програмування, такі як процедурне програмування, об'єктно-орієнтоване програмування та функціональне програмування. Підтримка різних парадигм програмування надає можливість вибору і дозволяє розробникам використовувати найефективніший підхід для вирішення конкретної задачі.

Серед переваг використання JavaScript можна виділити велику та активну

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум	Підпис	Дата		

спільноту розробників, широкий спектр бібліотек, як-от jQuery, Lodash та фреймворків, як-от React, Angular, Vue.js, постійний розвиток та появу нових функцій.

Недоліками використання JavaScript для розробки браузерних розширень є вразливість та проблеми зі швидкістю виконання коду. Помилки та недоліки коду, який виконується на клієнтському боці, може бути експлуатовано для шкідливих цілей. Інтерпретована природа JavaScript може призвести до значних проблем з продуктивністю в ресурсомістких додатках.

TypeScript, розроблений Microsoft, є надбудовою над JavaScript, що додає статичну типізацію, класи, модулі та інтерфейси. Такі можливості дають розробникам писати більш структурований код, який легше читати та підтримувати, а також допомагають виявляти помилки одразу на етапі компіляції.

Dart є ще однією мовою програмування, яку можна використовувати для створення браузерних розширень. Dart поділяє багато подібностей з TypeScript, пропонуючи статичну типізацію, об'єктно-орієнтовне програмування за допомогою класів і модулі, які сприяють написанню структурованого, надійного та читабельного коду. Однак на відміну від TypeScript, Dart не компілюється безпосередньо в JavaScript. Замість цього він компілюється в машинний код, який може бути виконаний спеціальною віртуальною машиною Dart, яка називається Dart VM. Так як це відносно нові мова програмування, деякі функції та бібліотеки Dart ще перебувають у стадії розробки.

Зважаючи на специфіку програмної системи цього проєкту, для реалізації браузерного розширення було обрано мову програмування JavaScript через її вбудовані можливості для маніпулювання елементами на вебсторінках та широкій підтримці на всіх сучасних браузерах.

					ІАЛЦ.467200.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум	Підпис	Дата		

2.2 Вибір та обґрунтування використання мови розробки для моделі машинного навчання

Вибір мови програмування є критичним етапом у розробці моделі машинного навчання. Мова програмування має бути потужною та гнучкою, щоб ефективно підтримувати всі етапи процесу машинного навчання, від збору та підготовки даних до тренування моделі.

Мова програмування Python є популярною високорівневою мовою програмування, яка відома своєю лаконічністю, простотою вивчення та широким спектром застосувань. Вона широко використовується в галузі наукових обчислень, аналізу даних та машинного навчання. Python має велику кількість бібліотек, які надають готові до використання функції для попередньої обробки даних, тренування моделей, валідації та тестування. Прикладами цих бібліотек є NLTK, spaCy, scikit-learn та pandas. Серед недоліків Python можна виділити проблематичність інтеграції з іншими мовами, оскільки це вимагає додаткових знань та зусиль від розробника й збільшує складність проєкту. Окрім цього, візуалізація великих обсягів даних може бути повільною та потребувати додаткових оптимізацій.

R є мовою програмування, яка спеціалізується на статистичному аналізі та візуалізації даних. До основних галузей використання R входять біоінформатика, економіка, медицина та охорона здоров'я, а також наукові дослідження. R також знаходить достатньо поширене застосування в маркетингових дослідженнях, спортивній аналітиці та освіті. Вона має багато бібліотек, таких як tm, text2vec, quanteda, які спрощують обробку та аналіз текстових даних. Наприклад, дані бібліотеки мають функції для очистки тексту, лематизації, токенізації тощо.

R відома своїми можливостями для візуалізації даних завдяки пакету ggplot2, що дозволяє легко створювати графіки та діаграми для аналізу текстових даних та результатів моделей. Продуктивність даної мови програмування може бути

					ІАЛЦ.467200.003 ПЗ	Арк.
						18
Зм.	Арк.	№ докум	Підпис	Дата		

набагато повільнішою у порівнянні з іншими мовами, наприклад такими як Python, особливо при обробці великих текстових даних або при тренуванні великих моделей машинного навчання. Хоча R має багато бібліотек для машинного навчання, деякі нові або вузько спеціалізовані бібліотеки зазвичай з'являються в Python, а потім інтегруються в R.

Julia є відносно новою мовою програмування, яка була створена спеціально для високопродуктивних обчислень та наукових досліджень. Вона має вбудовану підтримку паралельних та розподілених обчислень, що дозволяє масштабувати обробку даних та тренування моделей на кількох процесорах або навіть кластерах, що є особливо корисним для роботи з великими наборами даних. Хоча Julia активно розвивається, вона все ще має менше бібліотек та інших інструментів у порівнянні, наприклад, з Python. Спільнота користувачів достатньо невелика, що може призвести до обмеженої підтримки та меншої кількості навчальних матеріалів, підручників та прикладів коду.

Проаналізувавши можливі варіанти, було обрано Python, оскільки її переваги переважають недоліки для даного проєкту.

2.3 Вибір та обґрунтування використання платформи

Chromium, проєкт з відкритим кодом, який лежить в основі таких популярних браузерів, як Chrome, Edge, Opera та Brave, пропонує розробникам плагінів для фільтрації контенту безліч переваг, роблячи його вигідним вибором для створення плагінів, які будуть доступними, функціональними, ефективними та працюватимуть на різних платформах.

У 2019 році Chromium використовувався як основа для приблизно 55% відсотків браузерів. Згідно з даними ірландського інструменту для аналізу вебтрафіку StatCounter станом на травень 2024 року платформа Chromium використовується як основа для більш ніж 60% браузерів у світі, що гарантує, що плагіни, розроблені для

					ІАЛЦ.467200.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум	Підпис	Дата		

цієї платформи, будуть доступні понад 2,6 мільярдам активних користувачів. Це значно розширює потенційну аудиторію плагіну та збільшує шанси на його успіх.

Chromium використовує технологію віртуалізації під назвою «пісочниця» для ізоляції плагінів один від одного та від основного процесу браузера. Це означає, що кожен плагін працює у своєму власному середовищі, яке має власні ресурси, такі як пам'ять та процесорний час. Використання даної технології підвищує безпеку та стабільність всієї системи. Технологія «пісочниці» переважно запобігає ситуаціям, коли помилковий або шкідливий код одного плагіну може вплинути на інші плагіни або сам браузер.

До Chromium входять DevTools, які є набором інструментів для відладки та тестування плагінів. DevTools дозволяє розробникам покроково відстежувати виконання коду, виявляти та усувати помилки, записувати та відтворювати сценарії тестування, а також мати змогу оптимізувати продуктивність своїх плагінів. Цей набір інструментів робить процес розробки більш ефективним та зменшує час виходу плагіну на ринок.

Chromium підтримує надзвичайно широкий спектр Web API, які дозволяють розробникам плагінів взаємодіяти з браузером та вебсторінками. Ці API охоплюють різні аспекти веб-розробки, надаючи розробникам гнучкість та інструменти, необхідні для створення потужних та розширюваних вебдодатків.

Web APIs для об'єктної моделі документа дозволяють розробникам плагінів маніпулювати деревом HTML-елементів, що представляє вебсторінку. Розробники мають можливості за потреби створювати, модифікувати або видаляти елементи об'єктної моделі документа, оброблювати події користувача, такі як кліки, наведення миші або введення тексту, маніпулювати стилями та атрибутами елементів, додавати динамічний контент до вебсторінок.

Web API для мережі надають розробникам плагінів можливість здійснювати HTTP-запити, завантажувати дані з вебсерверів та надсилати дані на вебсервери, інтегруватися з вебсервісами та API, створювати вебдодатки, що залежать від даних.

					ІАЛЦ.467200.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум	Підпис	Дата		

Web API для графіки дозволяють розробникам плагінів створювати та маніпулювати візуальними елементами на веб-сторінках. Ці API охоплюють 2D та 3D графіку, зображення, відео та SVG, візуальні ефекти та анімації, інтерактивні графічні елементи.

Web API для мультимедіа надають розробникам плагінів можливість вбудовувати та відтворювати аудіо та відео на веб-сторінках. Такі APIs дозволяють розробникам розширень відтворювати аудіо- та відеофайли з різних форматів, створювати елементи керування відтворенням, інтегрувати мультимедійні послуги, розробляти інтерактивні медіа-додатки.

Web API для зберігання даних дозволяють розробникам плагінів зберігати інформацію на локальному пристрої користувача. Дані API охоплюють локальне сховище HTML5, файлову систему, механізми синхронізації даних.

Chromium пропонує для використання достатньо широкий спектр інших Web API, які можуть надавати доступ до різноманітних функцій, таких як геолокація, веб-камера та мікрофон, Bluetooth, розширені можливості JavaScript, інструменти для тестування та налагодження.

Важливо зазначити, що не всі Web API доступні у всіх браузерах. Деякі API можуть бути доступні лише у Chromium або інших браузерах на основі Chromium, а інші API можуть бути підтримувані більшістю сучасних браузерів. Перед використанням будь-якого Web API хорошою практикою вважається рекомендація перевірити його сумісність з цільовим браузером.

Chromium постійно оновлюється новими функціями та виправленнями помилок, що гарантує, що плагіни, розроблені для цієї платформи, будуть сумісні з найновішими версіями браузерів та використовуватимуть найсучасніші технології. Chromium оновлюється кожні шість тижнів у середньому з семистами змінами в кожному оновленні.

Платформа Chromium доступна для різних операційних систем, зокрема Windows, macOS, Linux, Android та iOS. Це робить його зручним для розробників,

					ІАЛЦ.467200.003 ПЗ	Арк.
						21
Зм.	Арк.	№ докум	Підпис	Дата		

які хочуть, щоб їхні плагіни працювали на різних пристроях та були доступні для широкого кола користувачів веббраузерів.

2.4 Обробка та керування текстовими даними

Плагін має початковий список негативних слів, які використовуються для аналізу тексту. Цей список можна зберігати в базі даних або в локальному сховищі браузера.

Розглянемо перший варіант. Підключена база даних може добре масштабуватися при необхідності зберігання великих обсягів даних, що робить її кращим вибором для розширення з великою кількістю користувачів або даних. Даними можна керувати централізовано, що спрощує адміністрування розширення. Основним недоліком такого підходу є забезпечення швидкості доступу до даних та конфіденційності користувачів. Доступ до даних з під'єднаної бази даних може виявитись набагато повільнішим, ніж доступ до локального сховища, а зберігання даних користувачів у підключеній базі даних може викликати проблеми з конфіденційністю, якщо не вжито належних заходів.

Зберігання списку локально забезпечує декілька переваг:

- Зберігання даних користувачів локально гарантує, що їхні слова не надсилаються на сторонні сервери. Це важливо, адже деякі користувачі можуть не хотіти, щоб їхні слова відстежували або зберігали.
- Локальне зберігання даних зменшує ризик витоку або злому даних користувачів. Дані доступні лише браузеру або розширенню, яке їх створило.
- Доступ до локального сховища значно швидший, ніж до віддаленої бази даних, що покращує швидкий старт роботи програми.
- Зберігання та оновлення списку слів не потребує складних запитів до бази даних.

					ІАЛЦ.467200.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум	Підпис	Дата		

Користувачі також можуть додавати свої власні негативні слова до списку. Ці додані слова також зберігаються в локальному сховищі.

Існують також недоліки такого підходу. Зміни, внесені до списку слів на одному пристрої, не будуть синхронізовані з іншими пристроями користувача. Окрім того, слід враховувати, що зберігання великих списків слів локально може призвести до проблем з продуктивністю на деяких пристроях.

Для даного проєкту пріоритетами є швидкість завантаження та опрацювання даних, а також забезпечення конфіденційності користувачів, тому обираємо локальне сховище як найкращий варіант для зберігання даних.

2.5 Інструменти розробки

Для розробки розширення обираємо браузер Google Chrome. Розглянемо детальніше Chrome DevTools, набір інструментів, вбудованих у браузер Chrome, які дозволяють розробникам вебсторінок та плагінів відлагоджувати, тестувати та оптимізувати свій код.

Таблиця 1.1 - Панелі Chrome DevTools

Панель	Призначення	Детальний опис
Elements	Перегляд та редагування вебсторінки.	Перегляд та редагування структури вебсторінки, включаючи теги, атрибути та текстовий контент. Перегляд та редагування стилів CSS.

					ІАЛЦ.467200.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум	Підпис	Дата		

Продовження таблиці 1.1

Панель	Призначення	Детальний опис
		Перегляд та редагування коду, який додає динамічні можливості до вебсторінки. Перевірка доступності вебсторінки.
Sources	Перегляд та редагування файлів та інших ресурсів, які використовуються вебсторінкою.	Перегляд та редагування JavaScript-файлів, які завантажуються вебсторінкою. Перегляд та редагування CSS-файлів, які завантажуються вебсторінкою. Перегляд та редагування зображень, які використовуються на вебсторінці.
Network	Перегляд мережевих запитів, які робить вебсторінка, та відповідей.	Перегляд списку всіх мережевих запитів, зроблених вебсторінкою, включаючи URL-адреси, методи HTTP та заголовки запитів.
Performance	Аналіз продуктивності вебсторінки та виявлення недоліків.	Вимірювання часу, необхідного для завантаження вебсторінки. Вимірювання використання ресурсів, таких як CPU, пам'ять

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		24

Кінець таблиці 1.1

Панель	Призначення	Детальний опис
		та мережа. Виявлення вузьких місць у продуктивності вебсторінки, які можуть призводити до уповільнення роботи.
Console	Виведення повідомлень журналу та виконання JavaScript-коду у контексті вебсторінки.	Виведення повідомлень журналу з вебсторінки, включаючи помилки, попередження та інформаційні повідомлення. Виконання JavaScript-коду в контексті вебсторінки для тестування та налагодження.
Debugger	Покрокове відлагодження JavaScript-коду.	Виконання JavaScript-коду покроково, зупиняючись на певних рядках коду.

Таблиця описує лише деякі з основних функцій панелей інструментів розробника. Кожна панель має багато інших функцій, які можна активно використовувати для розробки та тестування програм.

WebStorm — це інтегроване середовище розробки, створене компанією-розробником програмного забезпечення JetBrains, яка спеціально призначена для розробки додатків, написаних на JavaScript або TypeScript. В екосистемі розробки плагінів, де якість та надійність є ключовими факторами успіху, інтегроване середовище розробки WebStorm від JetBrains виступає як незамінний інструмент. Це середовище розробки, яке налічує понад десять мільйонів користувачів по всьому світу, пропонує безліч функцій, які допомагають зробити повний процес розробки

					ІАЛЦ.467200.003 ПЗ	Арк.
						25
Зм.	Арк.	№ докум	Підпис	Дата		

плагінів більш ефективним та приємним.

Потужний редактор коду надає розробникам все необхідне для написання чіткого, лаконічного та читабельного коду.

Автозаповнення, рефакторинг, форматування та інші інтелектуальні функції економлять час, мінімізують помилки. Вбудовані інструменти для відлагодження та тестування дозволяють розробникам крок за кроком відстежувати виконання коду, писати та запускати unit-тести та end-to-end-тести. Webstorm також інтегрується з системами контролю версій, такими як Git та SVN.

Згідно з опитуванням розробників, 90% респондентів сказали, що вони рекомендують WebStorm іншим.

Отже, Chrome DevTools та WebStorm можна використовувати разом для розробки плагінів Chromium. DevTools можна використовувати для відлагодження та тестування коду плагіна, а WebStorm можна використовувати для редагування та рефакторингу коду плагіна.

					ІАЛЦ.467200.003 ПЗ	Арк.
						26
Зм.	Арк.	№ докум	Підпис	Дата		

ВИСНОВОК ДО РОЗДІЛУ 2

Розробка плагіна для фільтрації текстового контенту потребує ретельного підходу до вибору технологій та інструментів, які забезпечать його ефективність, надійність та зручність використання.

JavaScript було обрано мовою програмування завдяки її гнучкості, поширеності та підтримці всіма сучасними браузерами. Chromium став оптимальним вибором платформи для розробки плагіна. Цей відкритий код має широке поширення, що гарантує доступність плагіну для мільйонів користувачів. Для відлагодження та тестування коду плагіна буде використовуватися Chrome DevTools. Цей набір інструментів дозволить розробникам виявляти та виправляти помилки, оптимізувати продуктивність плагіна та тестувати його роботу в різних сценаріях. Редагування та рефакторинг коду плагіна буде здійснюватися за допомогою WebStorm.

					ІАЛЦ.467200.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум	Підпис	Дата		

РОЗДІЛ 3

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі описується архітектура розробленого браузерного розширення, а також архітектура моделі для класифікації тексту на основі категорій.

3.1 Архітектура програмної системи

Браузерне розширення складається з декількох компонентів, які працюють разом, щоб додати нову функціональність до браузера. Розроблене розширення складається з наступних компонентів: маніфест, контент-скрипт, спливаюче вікно, файл стилів. Розглянемо детальніше кожен компонент.

Файл маніфесту описує метадані розширення в JSON-форматі. Він містить інформацію про розширення, таку як його назва, опис, дозволи та ресурси, які використовуються розширенням. Браузер використовує дану інформацію для завантаження, встановлення та запуску розширення. Назва «маніфест» точно відображає роль та функцію файлу у системі розширень браузера. Цей файл є обов'язковим компонентом для всіх браузерних розширень.

					ІАЛЦ.467200.003 ПЗ	Арк.
						28
Зм.	Арк.	№ докум	Підпис	Дата		

```

1  {
2      "manifest_version": 3,
3      "name": "Вартівник",
4      "version": "1.0",
5      "description": "Розширення для цензури певних слів на вебсторінках.",
6      "permissions": [
7          "activeTab",
8          "storage"
9      ],
10     "content_scripts": [
11         {
12             "matches": ["<all_urls>"],
13             "js": ["contentScript.js"],
14             "exclude_matches": [
15                 "https://docs.google.com/forms/*"
16             ]
17         }
18     ],
19     "action": {
20         "default_popup": "popup.html"
21     }
22 }

```

Рисунок 3.1 – Параметри конфігурації manifest.json

Параметр «manifest_version» описує рівень сумісності розширення з браузером. Станом на сьогодні необхідно використовувати третю версію. Ця версія зменшує ризик того, що розширення зловживатимуть даними користувачів або відстежуватимуть їхню активність. У зв'язку з таким покращенням безпеки та приватності користувачів Chrome прийняв рішення зупинити прийняття розширень, які мають другу версію, у січні 2022 року. Розширення, які використовували другу версію, мали оновитись до червня 2023 року, щоб забезпечити свою сумісність з Chrome та іншими браузерами.

Параметр «name» містить заголовок, який користувачі бачитимуть у магазині розширень та в інтерфейсі браузера. Заголовок має бути чітким, лаконічним та описувати суть розширення.

					ІАЛЦ.467200.003 ПЗ	Арк.
						29
Зм.	Арк.	№ докум	Підпис	Дата		

Параметр «version» вказує номер версії розширення. Він використовується для відстеження оновлень розробниками та користувачам, а також для виправлення помилок.

Параметр «description» відповідає за короткий опис функціоналу розширення, який відображається користувачам під час встановлення. Він має бути інформативним, зрозумілим та спонукати користувачів до встановлення.

Параметр «permissions» описує перелік дозволів, які необхідні для роботи розширення. Надаємо лише ті дозволи, які дійсно необхідні, щоб мінімізувати вплив на приватність та безпеку користувачів. Розглянемо кожен дозвіл детальніше. Параметр «activeTab» дозволяє розширенню взаємодіяти з активною вкладкою браузера. Параметр «storage» дозволяє розширенню зберігати та отримувати дані у сховищі браузера.

Параметр «content_scripts» визначає, коли та де буде завантажено та виконано скрипт. Для ключа «matches» вказуємо масив URL-адрес або шаблонів, що визначають, на яких сторінках повинен виконуватися скрипт. Для ключа «js» вказуємо масив шляхів до JavaScript-файлів, які потрібно завантажити та виконати на відповідних сторінках. Для ключа «exclude_matches» вказуємо масив URL-адрес або шаблонів, що визначають, на яких сторінках скрипт не буде завантажено. У розробленому розширенні виключено завантаження скрипту на сторінках Google Forms, щоб уникнути можливих конфліктів.

Параметр «action» визначає назву файлу, який містить код для спливаючого вікна розширення. Для ключа «default_popup» необхідно вказати шлях до HTML-файлу, який буде використовуватись для відображення спливаючого вікна розширення.

Контент-скрипт — це JavaScript-код, який виконується на вебсторінках, що завантажуються користувачем. У даному розширенні контент-скрипт надає простий та ефективний спосіб цензурування заданих слів на вебсторінках. Варто зазначити, що контент-скрипт має власне середовище виконання, або контекст. Це означає, що

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум	Підпис	Дата		

контент-скрипт ізольований від інших контекстів виконання JavaScript, зокрема від контексту вебсторінки, а тому не має прямого доступу до змінних, функцій та об'єктів інших контекстів. Однак контент-скрипт має доступ до дерева об'єктної моделі документа вебсторінки, на якій виконується. Це дозволяє йому змінювати вміст сторінки, додавати нові елементи та реагувати на дії користувача. Контент-скрипт має доступ лише до деяких API браузера. Таку обмеженість було введено з міркувань безпеки, щоб запобігти розширенням виконувати шкідливі дії. Також контент-скрипт не має доступу до файлів на комп'ютері користувача або до мережі. Для доступу до цих ресурсів йому потрібно взаємодіяти з фоновим скриптом розширення.

Спливаюче вікно — це HTML-інтерфейс, який з'являється при натисканні на значок розширення. Воно використовується для надання користувачеві можливості взаємодіяти з розширенням та відображає певну інформацію. У розробленому розширенні спливаюче вікно містить два поля для вводу для задання або змінювання персоналізованих параметрів, статистику щодо кількості відфільтрованих слів на поточній вебсторінці та можливість зворотнього зв'язку, яка безпосередньо представлена посиланням на Google-форму.

Файл стилів використовується для стилізації спливаючого вікна та інших елементів розширення. Він містить правила CSS, які визначають шрифти, кольори, макет та інші візуальні аспекти.

Після того, як користувач встановив розширення з магазину розширень браузера, браузер починає читати файл `manifest.json`, щоб отримати основну необхідну інформацію для роботи з розширенням. Коли користувач відвідує вебсторінку, браузер інтегрує файл `contentScript.js` зі сторінкою. Тепер файл `contentScript.js` може взаємодіяти з поточною сторінкою, а саме збирати інформацію про весь текстовий контент і змінювати вміст сторінки, замінюючи певні слова або фрази.

Також файл `contentScript.js` взаємодіє з локальним сховищем користувача, надсилаючи туди оновлені дані щодо поточного списку слів для фільтрації,

					ІАЛЦ.467200.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум	Підпис	Дата		

максимальної кількості дозволених слів та кількості поточних цензурованих слів.

З метою створення списку слів для фільтрації, було розроблено модель машинного навчання, яка навчалась на даних, отриманих з аналізу списку спеціальних вебсайтів.

Алгоритмом машинного навчання для цього проєкту було обрано найвний баєсовий класифікатор для мультиноміальної моделі [7]. Цей алгоритм широко використовується для класифікації текстів завдяки своїй простоті та високій точності. Він ґрунтується на теоремі Баєса та припускає незалежність ознак при визначенні ймовірності приналежності елемента до одного з класів.

Наївний баєсовий класифікатор використовує наступну формулу для класифікації тексту до класу C:

$$P(C | X) = \frac{P(X | C) * P(C)}{P(X)} \quad (3.1)$$

де $P(C | X)$ — апостеріорна ймовірність того, що текст належить до класу C за умови наявності в тексті ознак X;

$P(X | C)$ — ймовірність того, що в тексті є ознаки X за умови, що він належить до класу C;

$P(C)$ — ймовірність того, що в текст належить до класу C;

$P(X)$ — ймовірність того, що в тексті є ознаки X незалежно від класу;

Для класифікації нового слова класифікатор обчислюватиме $P(C | X)$ для усіх категорій. Після цього класифікатор обиратиме категорію з більшою апостеріорною ймовірністю.

Апостеріорна ймовірність завжди є умовною. Вона залежить від даних, які використовуються для її оновлення. Апостеріорна ймовірність використовується для порівняння декількох конкуруючих гіпотез. Гіпотеза з найвищою апостеріорною ймовірністю є найбільш ймовірною, враховуючи дані.

					ІАЛЦ.467200.003 ПЗ	Арк.
						32
Зм.	Арк.	№ докум	Підпис	Дата		

Однак це припущення може бути не завжди справедливим, що може призвести до зниження точності класифікації. Наприклад, якщо два слова часто з'являються в тексті разом, алгоритм може не врахувати цей зв'язок через припущення незалежності. Тому ефективне використання даного алгоритму потребує правильної підготовки даних.

Підготовка даних складається з таких етапів, як чітке розмежування категорій, очищення та збалансування даних. Дані повинні бути перевірені на відсутність орфографічних помилок та інших можливих неточностей. Якщо деякі категорії мають значно більше даних, ніж інші, алгоритм наївного баєсового класифікатора може бути упередженим щодо цих категорій.

3.2 Реалізація програмної системи

Для створення та тренування моделі було використано такі бібліотеки мови Python:

- Бібліотека pandas;
- Бібліотека rymorphy2;
- Бібліотека NLTK;
- Бібліотека scikit-learn.

Бібліотека pandas — це відкрите програмне забезпечення, розроблене для роботи з багатовимірними структурованими наборами даних, як-от з науковими даними або текстовими даними. Вона пропонує широкий спектр інструментів та функцій для маніпулювання, аналізу та візуалізації даних. Бібліотека дозволяє створювати складні статистичні розрахунки, зокрема обчислення середніх значень, стандартних відхилень, кореляцій та розподілів ймовірностей. Окрім широкого застосування в галузі машинного навчання, модуль буде також достатньо корисним для розробки та тестування проєктів у сферах фінансового аналізу, бізнес-аналітики та наукових досліджень. У даному проєкті pandas використовується для зручного

					ІАЛЦ.467200.003 ПЗ	Арк.
						33
Зм.	Арк.	№ докум	Підпис	Дата		

представлення даних. Наприклад, після перетворення початкових даних в об'єкт DataFrame призначимо змістовні назви стовпцям для зручного звертання до цього коду програми в подальшому.

Бібліотека `rumorphy2` — це також бібліотека з відкритим кодом, призначена для морфологічного аналізу та обробки текстів. Вона пропонує широкий спектр функцій, зокрема нормалізацію та лематизацію. Щоб використовувати `rumorphy2` для роботи з текстами, написаними українською мовою, необхідно виконати додаткові дії. Спочатку в командному рядку треба ввести команду «`pip install rumorphy2-dicts-uk`» [8]. При успішному виконанні команда завантажить українські словники. Окрім цього, необхідно створити змінну, що буде містити інструмент під назвою `MorphAnalyzer` з параметром `lang`, що має значення `'uk'`. `MorphAnalyzer` може використовуватися для виконання широкого спектру завдань, пов'язаних з морфологічним аналізом.

Бібліотека `NLTK` пропонує набір інструментів для обробки природної мови. Завдяки вбудованим функціям `NLTK` можна робити синтаксичний, семантичний та прагматичний аналізи тексту, виявляти та витягувати відповідну інформацію з тексту, а також виконувати машинний переклад тексту. Вона може допомогти проаналізувати контекст та намір тексту, дозволити розбивати текст на його синтаксичні компоненти, такі як частини мови, фрази та речення, надати інструменти для виявлення та видалення конкретної інформації з тексту, такої як імена людей, місця та дати. Бібліотека пропонує інтерфейси до інших популярних бібліотек для машинного навчання, таких як `scikit-learn` та `TensorFlow`. У даному проєкті використано функцію `WordNetLemmatizer()` для лематизації слів у текстах. Лематизація — це процес зведення слів до їхньої лексичної основи. Вона потрібна для більш точної обробки та аналізу текстових даних.

Бібліотека `scikit-learn` призначена для виконання завдань машинного навчання. Вона надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації. `Scikit-learn` побудована на основі

					ІАЛЦ.467200.003 ПЗ	Арк.
						34
Зм.	Арк.	№ докум	Підпис	Дата		

двох інших бібліотек - NumPy та ScyPy. Вона має зручний інтерфейс та чітку документацію. У даному проєкті використано три модулі даної бібліотеки: `sklearn.feature_extraction.text`, `sklearn.naive_bayes` та `sklearn.model_selection`.

Перший модуль використовується для векторизації тексту, тобто для перетворення тексту в числовий формат, який може бути оброблений алгоритмами машинного навчання. Перетворення відбувається за допомогою методу TF-IDF. TF-IDF — це статистична методика, яка використовується для оцінки важливості слова в документі відносно колекції документів.

Розглянемо концепцію детальніше, розділивши її на дві частини. Частота терміна - це вимір того, як часто слово з'являється в конкретному документі. Чим частіше слово зустрічається, тим більшу вагу воно отримує. Зворотня частота документа оцінює, наскільки рідкісним є слово в усій колекції документів. Якщо слово зустрічається в багатьох документах, його зворотня частота буде низькою, тому що дане слово не допомагає відрізнити документи одне від одного. Добуток TF на IDF дозволяє отримати вагу для окремого слова в документі. Слова з високими балами TF-IDF матимуть більший вплив на класифікацію, оскільки вони вважаються більш репрезентативними для категорії документа.

Другий модуль застосовано для навчання моделі класифікації на основі алгоритму наївного баєсового класифікатора.

Третій модуль використовує функцію під назвою `train_test_split` для розбиття початкових даних на навчальний та тестовий набори. Навчальний набір використовується для навчання моделі, а тестовий для оцінки її продуктивності. Функція має три заданих параметри.

Розглянемо детальніше кожен з параметрів. Перші два параметри представляють два основних набори даних, які використовуються для тренування, а в подальшому і безпосереднього тестування моделі машинного навчання. Розглянемо перший набір даних. Перший набір даних є матрицею, в якій кожен рядок відповідає зразку даних, а кожен стовпець відповідає певній характеристиці або атрибуту

					ІАЛЦ.467200.003 ПЗ	Арк.
						35
Зм.	Арк.	№ докум	Підпис	Дата		

заданого зразка. Другий набір даних є вектором, де кожен елемент відповідає мітці або цільовому значенню відповідного зразка даних. Наступний параметр називається `test_size`. Він визначає частку даних, яка буде виділена для тестового набору. Також існує параметр `train_size`, який не використовується явно в даному коді. Він відповідає за частку даних, яка буде виділена для тренувального набору. За замовчуванням його значення буде дорівнювати одиниці. Параметр `random_state` заданий неявно і використовується для контролю випадкового розподілу даних на тренувальний та тестовий набори. За замовчуванням використовується випадкове значення.

Дані для навчання завантажуються з модуля під назвою `training_data.py`. Цей модуль представляє список словників. Кожен словник має однакову структуру, складається з ключів «category» та «words». Ключ «category» містить значення однієї з шести можливих категорій: «Мова ненависті», «Ненормативна лексика», «Насильство», «Тероризм», «Жорстокість» або «Сексуальний контент». Ключ «words» має список з близько п'ятдесяти слів, що відповідають зазначеній категорії.

Після завантаження категорій і списків слів додаємо ще одну категорію, яка буде називатися «Інше». Вона використовуватиметься для слів, які не належать до жодної іншої категорії. Відповідно до списку слів додаємо порожній рядок. Створюємо функцію з назвою `clean_text` для попереднього очищення тексту. Вона зводитиметь вхідний текст до нижнього регістру, видалятиметь символи, які не є літерами, а далі проводитиметь лематизацію слів за допомогою `py morphology2`.

Створюється спеціальний об'єкт під назвою `TfidfVectorizer`, який перетворює текст у числові вектори. Метод `fit_transform` застосовується до списку очищених текстів `words` для отримання матриці TF-IDF векторів `word_vectors`. Навчальні та тестові дані розділяються за допомогою `train_test_split`. Це робиться для того, щоб оцінити продуктивність моделі на невідомих раніше даних. Об'єкт класифікатора `MultinomialNB` створюється та навчається на навчальних векторах `x_train` та навчальних категоріях `y_train`.

					ІАЛЦ.467200.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум	Підпис	Дата		

Функція `classify_text` отримує новий текст, попередньо зібраний з вибірки вебсайтів, як вхідні дані. Текст очищується за допомогою функції `clean_text`. Вектор нового тексту створюється за допомогою `vectorizer.transform`. Модель класифікатора `classifier` використовується для прогнозування категорії нового тексту з його вектора. Найвища ймовірність у векторі `predicted_probabilities` відповідає категорії, яку модель вважає найімовірнішою для заданого тексту. Чим вища ймовірність, тим більше модель упевнена в своєму прогнозі. Якщо найвища має значення ймовірність менше, ніж 0.2, тоді текст буде відноситися до категорії «Інше».

Прогнозована категорія повертається функцією. Новий текст записується до списку слів разом з прогнозованою категорією.

На початку роботи розширення скрипт отримує список слів для фільтрації, попередньо сформований моделлю машинного навчання, та максимальну кількість дозволених цензурованих слів для користувача з локального сховища браузера. За замовчуванням максимальна кількість таких слів дорівнює п'яти. Якщо кількість відфільтрованих слів на поточному вебсайті перевищує максимальну кількість дозволених слів, увесь контент вебсторінки розмивається на десять секунд, надаючи користувачу можливість покинути вебсайт. Якщо користувач залишається на вебсайті, він отримує червоне спливаюче вікно з попередженням про небезпеку поточного вебсайту. Користувач може змінити даний параметр за допомогою відповідного поля для вводу в спливаючому вікні розширення. Далі викликається функція, яка перевіряє типи всіх вузлів об'єктної моделі документа. Якщо поточний вузол є текстовим вузлом, функція шукає в його текстовому контенті слова зі списку слів для фільтрації. У разі успішного виявлення таких слів, вони замінюються в режимі реального часу на текст такої ж довжини, який складається з символів «*». При цьому щоразу збільшується лічильник цензурованих слів, якій буде доступний користувачеві на спливаючому вікні розширення.. Якщо поточний вузол є елементним вузлом, функція рекурсивно викликає себе для кожного дочірнього вузла.

					ІАЛЦ.467200.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум	Підпис	Дата		

Наступний фрагмент коду налаштовує події натискання для двох кнопок, розташованих на спливаючому вікні, які відповідають за персоналізацію параметрів, за допомогою прослуховувачів подій. Перевірка наявності цих кнопок необхідна, оскільки на момент запуску розширення вони не існують в об'єктній моделі документа. Ці кнопки зможуть з'явитися в об'єктній моделі документа лише після того, як користувач відкриє спливаюче вікно. Натискання на кнопки призводить до виклику відповідних функцій, які реалізують оновлення параметрів та їхнє зберігання в локальному сховищі браузера.

У розробленому коді у ролі локального сховища використовується механізм зберігання даних під назвою Chrome Storage Sync. Він є частиною Chrome API. Це сховище дозволяє розширенням Chrome зберігати дані локально на пристрої користувача та синхронізувати їх між усіма його пристроями, де він увійшов до Chrome з тим самим обліковим записом Google. Це надає користувачам зручний спосіб синхронізації налаштувань та даних розширень між своїми пристроями.

Продуктивність сховища залежить від декількох факторів. Чим швидше інтернет-з'єднання в користувача, тим швидше будуть збережені та синхронізовані дані. Тип операції теж має вплив на продуктивність роботи сховища. Наприклад, операції читання зазвичай виконуються швидше, ніж операції запису.

Chrome Storage Sync намагається синхронізовувати дані кожні п'ять хвилин. Якщо користувач вноситиме багато змін до своїх даних, синхронізація автоматично відбуватиметься швидше. Також сховище дозволяє користувачам отримати в режимі офлайн. Зміни, внесені до даних офлайн, буде синхронізовано при підключенні до мережі інтернет.

Для взаємодії з локальним сховищем використовуються функції `set` та `get`. Перша функція приймає об'єкт як аргумент, де ключі – це назви властивостей для зберігання даних, а значення – дані, які потрібно зберегти. Друга функція використовується для отримання даних із локального сховища. Вона приймає один аргумент, який містить назви властивостей, які потрібно отримати. Функція повертає

					ІАЛЦ.467200.003 ПЗ	Арк.
						38
Зм.	Арк.	№ докум	Підпис	Дата		

спеціальний об'єкт, де ключі – це назви властивостей, а значення є відповідними даними, збереженими в сховищі.

					ІАЛЦ.467200.003 ПЗ	Арк.
						39
Зм.	Арк.	№ докум	Підпис	Дата		

ВИСНОВОК ДО РОЗДІЛУ 3

У даному розділі дипломної роботи була виконана реалізація двох програмних модулів з використанням різних технологій, зокрема мови програмування Python та мови програмування JavaScript. Спочатку було описано архітектуру браузерного розширення та архітектура моделі для класифікації тексту. Далі детально було розглянуто реалізацію браузерного розширення, зокрема проаналізовано роботу кожного компоненту, та моделі для класифікації тексту, зокрема бібліотеки pandas для зручного представлення даних, rymorphy2 для морфологічного аналізу українських слів, NLTK для лематизації, scikit-learn для імплементації алгоритму наївного баєсового класифікатора та функцій для тренування моделі.

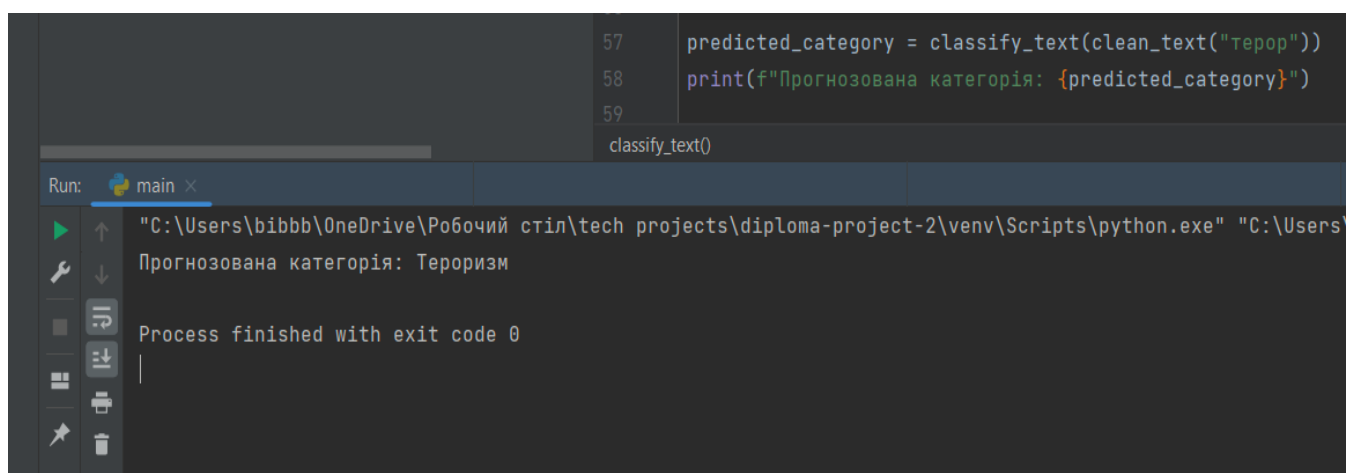
Використання описаних рішень та технологій наділяє програмні модулі властивостями розширюваності, масштабованості та стабільності, що робить їх незамінними компонентами системи.

					ІАЛЦ.467200.003 ПЗ	Арк.
						40
Зм.	Арк.	№ докум	Підпис	Дата		

РОЗДІЛ 4

ОГЛЯД ТА ТЕСТУВАННЯ ПРОГРАМНОГО МОДУЛЯ

Розглянемо сценарій роботи з розробленим програмним модулем. Для початку протестуємо модель машинного навчання і перевіримо, як вона справляється з класифікуванням тексту. Перевіримо модель, вводючи окремі слова. Вводимо перше слово. Отримуємо очікуваний результат.



The screenshot shows a Python IDE with a code editor and a console. The code in the editor is:

```
57 predicted_category = classify_text(clean_text("терор"))
58 print(f"Прогнозована категорія: {predicted_category}")
59
```

The console output shows the execution of the code:

```
Run: main x
"C:\Users\bibbb\OneDrive\Робочий стіл\tech projects\diploma-project-2\venv\Scripts\python.exe" "C:\Users\bibbb\OneDrive\Робочий стіл\tech projects\diploma-project-2\venv\Scripts\python.exe"
Прогнозована категорія: Тероризм
Process finished with exit code 0
```

Рисунок 4.1 – Результат тестування моделі

Тепер протестуємо слово з іншою прогнозованою категорією. Отримуємо очікуваний результат.

					ІАЛЦ.467200.003 ПЗ	Арк.
						41
Зм.	Арк.	№ докум	Підпис	Дата		

```

57     predicted_category = classify_text(clean_text("невдаха"))
58     print(f"Прогнозована категорія: {predicted_category}")
59
classify_text()

```

Run: main x

```

"C:\Users\bibbb\OneDrive\Робочий стіл\tech projects\diploma-project-2\venv\Scripts\python.exe" "C:\Users\bibbb\OneDrive\Робочий стіл\tech projects\diploma-project-2\venv\Scripts\python.exe"
Прогнозована категорія: Мова ненависті
Process finished with exit code 0

```

Рисунок 4.2 – Результат тестування моделі

Для тестування розширення необхідно його завантажити до списку встановлених розширень зручного для нас браузера. Нехай це буде Google Chrome.

Відкриваємо обраний браузер і переходимо до сторінки, яка розміщена за посиланням `chrome://extensions/`. У правому верхньому куті необхідно увімкнути режим розробника. Після цього на панелі керування з лівої сторони натискаємо на кнопку «Завантажити розпаковане розширення». Очікуємо на завершення завантаження. Розроблене розширення має бути доступним на вкладці «Усі розширення».

					ІАЛЦ.467200.003 ПЗ	Арк.
						42
Зм.	Арк.	№ докум	Підпис	Дата		

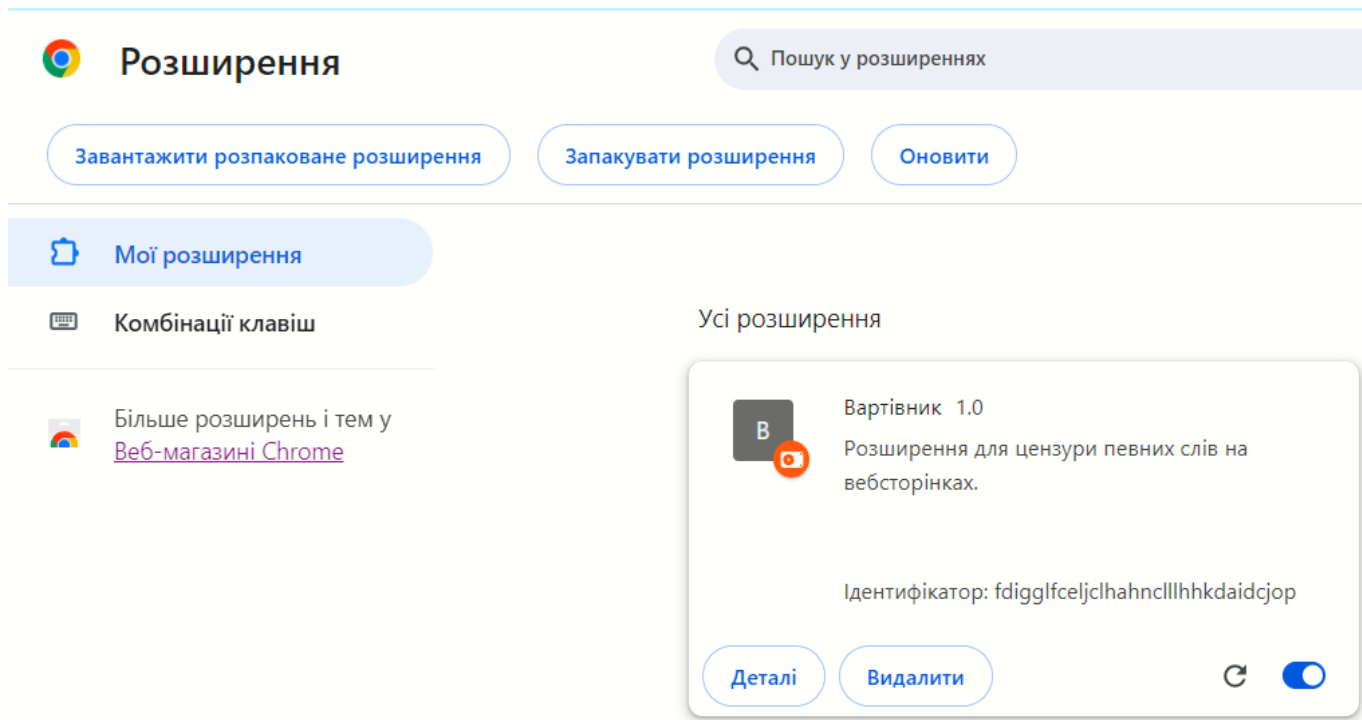


Рисунок 4.3 – Демонстрація встановленого розширення

Натискаємо на кнопку «Деталі». Можемо переглянути інформацію про опис, версію, розмір, ідентифікатор та дозволи розширення. Для зручного тестування увімкнемо опцію «Дозволити в анонімному режимі» та опцію «Збирати повідомлення про помилки». Тепер можна запускати розширення в анонімному вікні, а також миттєво переглядати повідомлення про можливі помилки.

					ІАЛЦ.467200.003 ПЗ	Арк.
						43
Зм.	Арк.	№ докум	Підпис	Дата		

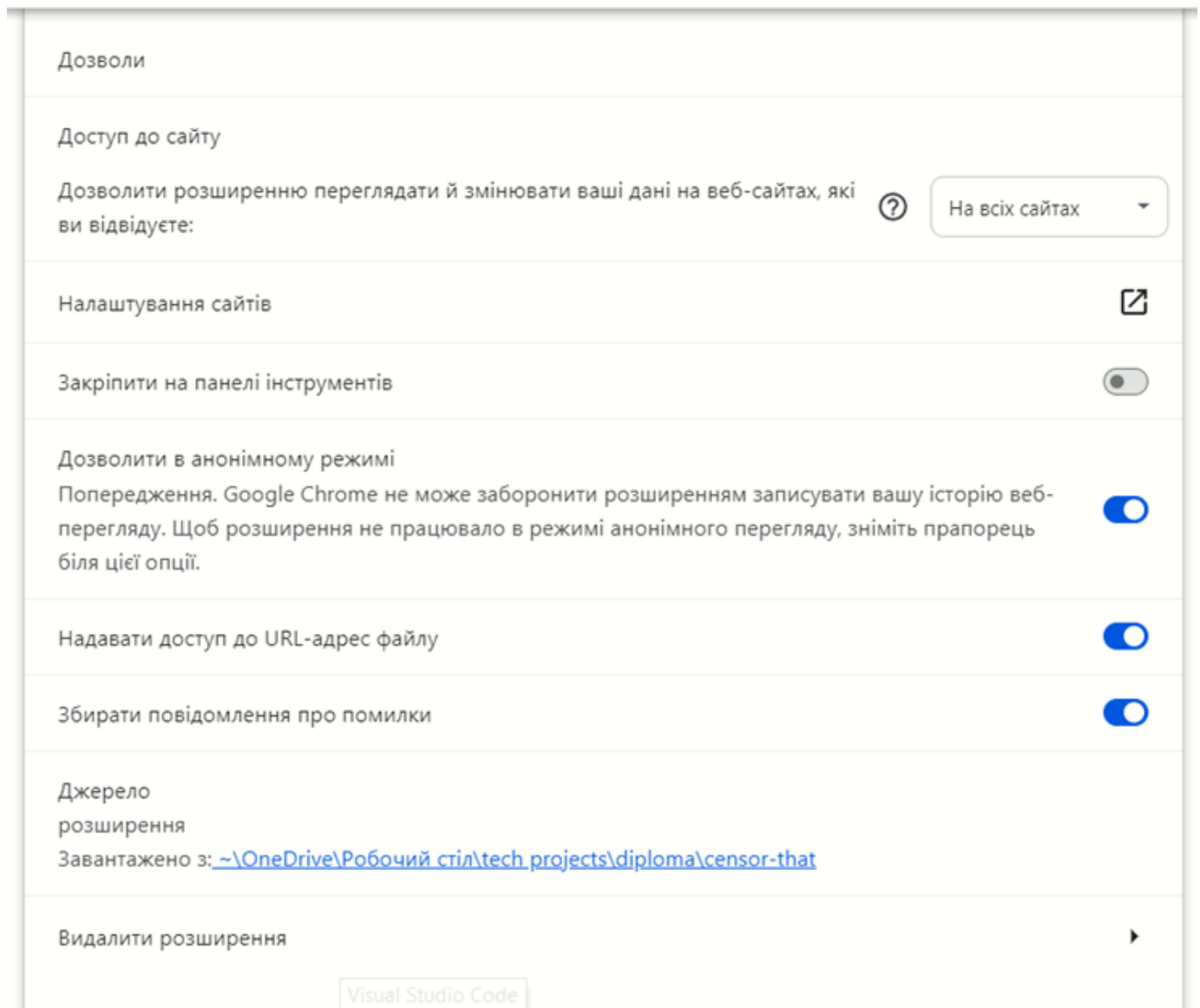


Рисунок 4.4 – Деталі розширення

Оберемо сайт, які містить слова хоча би з однієї з доступних категорій. Переходимо на новинний сайт та обираємо новину, яка стосується категорії про жорстокість.

					ІАЛЦ.467200.003 ПЗ	Арк.
						44
Зм.	Арк.	№ докум	Підпис	Дата		

```
{
  "category": "Жорстокість",
  "words": [
    "знущання", "безжальність", "шантаж", "маніпуляція", "жорстокість",
    "залякування", "тортури", "вбивство", "біль", "злість",
    "травма", "розчленування", "душити", "труп", "різанина",
    "зарізати", "поріз", "розправа", "звірство", "перерізати",
    "страждання", "страждати", "жорстко", "жорстокий", "перелом",
    "переломати", "страта", "стратити", "потерпілий"
  ]
},
```

Рисунок 4.5 – Список слів для категорії «Жорстокість»

Одразу після переходу на цю вебсторінку увесь контент стає недоступним на десять секунд через застосування фільтру розмиття. Це означає, що поточний вебсайт містить більше п'яти слів зі списку слів для фільтрації й є потенційно небезпечним для відвідування.



Рисунок 4.6 – Результат роботи розширення на вебсайті, де кількість слів, що підпадають під фільтрацію, перевищує дозволену

					ІАЛЦ.467200.003 ПЗ	Арк.
						45
Зм.	Арк.	№ докум	Підпис	Дата		

Через десять секунд зверху сторінки побачимо попередження з яскравим червоним кольором, яке буде на сторінці протягом усього часу відвідування сторінки користувачем.

Увага! Цей вебсайт містить неприйнятий контент.

Вчені провели експеримент, запросивши групу молодих людей. Вони виявили довгострокове зниження співчуття у студентів, що часто грають у ***** відеоігри, і короточасне зменшення емоційної реакції на хворобливі зображення після 40 хвилин гри у тих, хто раніше не грав у ***** ігри.

Проте вплив ***** відеоігор на здатність гравця співпереживати *****м іншої людини є неоднозначним.

Спостереження за *****ми віртуальних героїв у грі зазвичай викликає у гравця автоматичну емпатичну реакцію. Але згодом попередній ігровий досвід та усвідомлення того, що це нереально, можуть покращити когнітивну оцінку гравця та знизити його початкову реакцію на **** через десенсибілізацію, що впливає як на когнітивні, так і на афективні процеси.

Рисунок 4.7 – Результат роботи розширення після розмиття контенту

Як можна побачити з рисунку 4.7, слова зі списку успішно відфільтровані та приховані від користувача.

За результатами дослідження американської компанії SOASTA у 2017 році [9] було доведено, що 53% відсотки відвідувачів мобільних сайтів залишають сторінку, яка недоступна протягом трьох секунд. Можна припустити, що очікування користувачів десктопних сайтів не набагато довші. Якщо контент не завантажується протягом десяти секунд, дорослі користувачі здатні досягнути попередження про небезпеку вебсайту, в той час як діти та підлітки з ймовірністю 90% покинуть сторінку, не чекаючи повного завантаження контенту вебсторінки.

Тепер спробуємо додати власне слово для фільтрації. Для виконання цієї дії необхідно перейти до спливаючого вікна розширення. Відкривши спливаюче

					ІАЛЦ.467200.003 ПЗ	Арк.
						46
Зм.	Арк.	№ докум	Підпис	Дата		

вікно, обираємо слово й вводимо його у відповідне поле. Після закінчення вводу натискаємо кнопку «Додати».

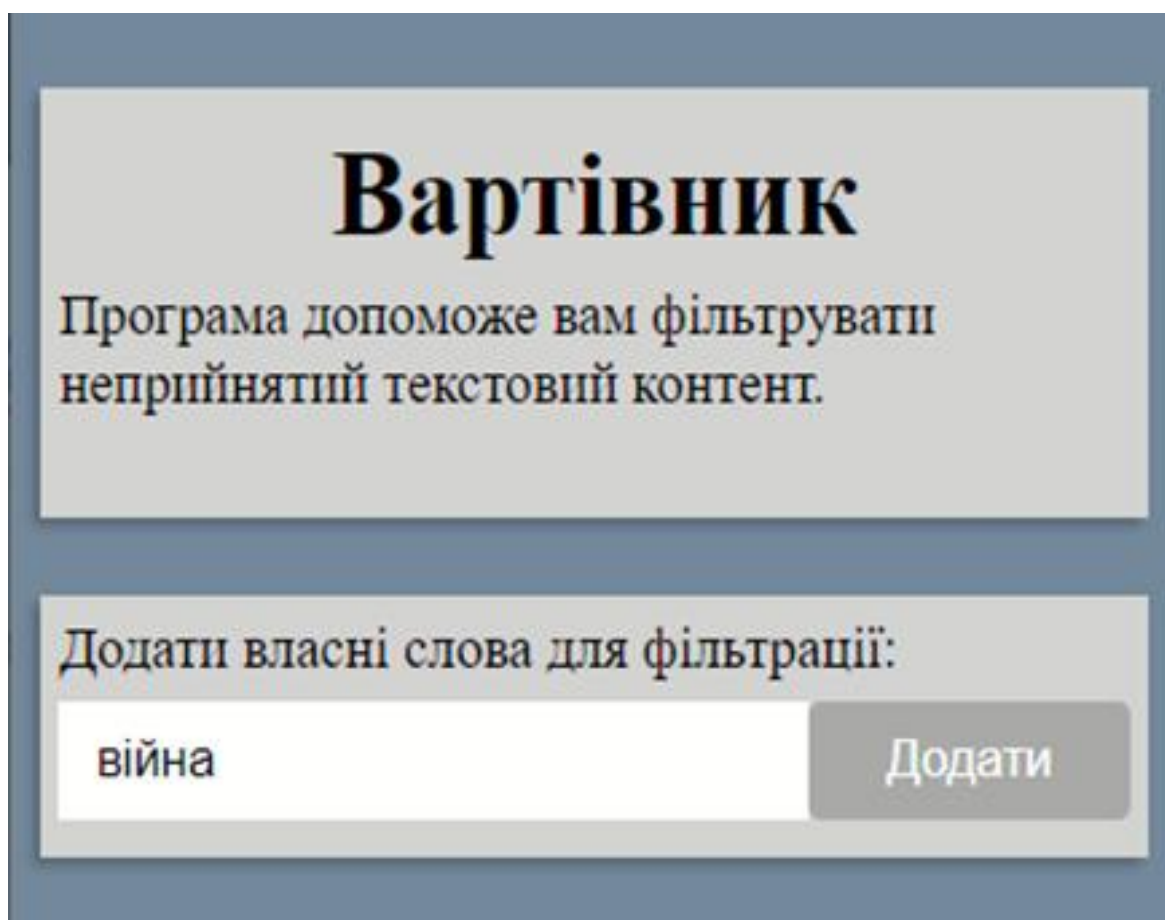


Рисунок 4.8 – Додавання власного слова для фільтрації

Переходимо на сайт, який гарантовано міститиме задане слово. Отримуємо очікуваний результат.

					ІАЛЦ.467200.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум	Підпис	Дата		

Увага! Цей вебсайт містить неприйнятний контент.

ПРИВІТАННЯ

АКТУАЛЬНІ ЗАПИТИ

ЯКИЙ СЬОГОДНІ ДЕНЬ

"Росія намагатиметься наступати, але нові території не зможе захопити. ЗСУ здобудуть перемогу. ***** може закінчитися швидко після складного періоду. Все може змінитися в один день, ***** завершиться швидко, настане мир", – розповів Феодул.

Рисунок 4.9 – Результат роботи розширення з власним словом для фільтрації

Кожного разу при переході користувача на нову вебсторінку знову запускається функція для фільтрації слів. У середньому процес займає від однієї до восьми секунд у залежності від кількості вузлів об'єктної моделі документа, що визначається загальним обсягом контенту. Це є хорошим результатом, однак наявний код містить деякі обмеження, які можна буде покращити при подальшій розробці системи.

Розроблене розширення також показує кількість відфільтрованих слів на поточній сторінці. Таким чином користувач може мати уявлення про потенційну небезпеку вебсайту.

					ІАЛЦ.467200.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум	Підпис	Дата		

Вартівник

Програма допоможе вам фільтрувати неприйнятний текстовий контент.

Додати власні слова для фільтрації:

Додати максимальну кількість відфільтрованих слів до попередження:

Кількість неприйнятих слів на поточному вебсайті: 94

[Залишити відгук](#)

Рисунок 4.10 – Демонстрація статистики, доступної через спливаюче вікно розширення

З метою отримання зворотного зв'язку від користувачів та покращення програми до графічного інтерфейсу розробленого розширення було додано посилання на Google-форму, де користувачам можна залишити відгуки про помилки або запропонувати нові функції.

					ІАЛЦ.467200.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум	Підпис	Дата		

Вартівник - Зворотній зв'язок

Привіт!

Дякуємо за інтерес до нашого продукту. У цій формі ви можете запропонувати ідею для покращення продукту або описати проблему, з якою зіштовхнулися.

bibbbik@gmail.com [Змінити обліковий запис](#)



Спільно не використовується

Зірочка (*) указує, що запитання обов'язкове

Рисунок 4.11 – Демонстрація форми для зворотнього зв'язку

Форма містить три запитання. Обов'язковим є оцінка продукту за десятибальною шкалою, де десять – це максимальна оцінка, яка вказує на високий рівень задоволення продуктом, а один – це найнижчий бал, який вказує на низький рівень задоволення продуктом.

					ІАЛЦ.467200.003 ПЗ	Арк.
						50
Зм.	Арк.	№ докум	Підпис	Дата		

Наскільки ви задоволені продуктом? *

12345678910

Мінімально
☐
☐
☐
☐
☐
☐
☐
☐
☐
☐
Максимально

Опишіть ідею, що покращить продукт.

Ваша відповідь

Опишіть проблему, яка у вас виникла.

Ваша відповідь

Надіслати

Очистити форму

Рисунок 4.11 – Демонстрація форми для зворотнього зв’язку

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		51

ВИСНОВОК ДО РОЗДІЛУ 4

Розроблений програмний модуль пройшов тестування та огляд, і результати показали його успішну роботу та виконання всіх поставлених завдань. Тестування не виявило жодних помилок чи проблем у роботі модуля, що підтверджує його високу якість та надійність.

Для подальшого розвитку системи пропонується розглянути такі можливі покращення: зменшення часу фільтрації, розширення списку категорій, щоб зробити модуль більш універсальним та покращення алгоритму класифікації, щоб підвищити точність моделі.

Важливо зазначити, що навіть без цих вдосконалень розроблений програмний модуль є ефективним інструментом для фільтрації контенту та захисту користувачів від шкідливих вебсайтів. Враховуючи успішне тестування та огляд системи, можна зробити висновок, що розроблена програмний модуль відповідає всім вимогам, працездатний та повністю готовий до експлуатації. Його можна використовувати як самостійний продукт або інтегрувати з іншими системами.

					ІАЛЦ.467200.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум	Підпис	Дата		

ВИСНОВКИ

Дипломна робота присвячена розробці браузерного розширення, яке використовує результати класифікації тексту моделлю машинного навчання для створення списку слів, що застосовуються для фільтрації текстового контенту в мережі інтернет.

У першому розділі обґрунтовується актуальність досліджуваної теми, аналізуються існуючі рішення та їхні недоліки. На основі детального вивчення потреб українських користувачів чітко сформульовано очікувані характеристики та тип продукту.

У другому розділі було проаналізовано програмні засоби та інструменти, які застосовуються на різних етапах створення та тренування моделі машинного навчання, а також розробки браузерного розширення. Для розробки моделі машинного навчання було обрано мову програмування Python. Для створення браузерного розширення було обрано JavaScript. Платформою для розробки розширення було обрано Chromium. Для обробки та керування даними було обрано локальне сховище. Інструментами розробки було обрано Chrome DevTools.

Третій розділ присвячено опису архітектури браузерного розширення та моделі для класифікації тексту. Було детально розкрито процес реалізації розширення, з акцентом на аналізі роботи кожного його компонента, а також моделі класифікації. Вона використовує бібліотеки pandas для зручного представлення даних, rymorphy2 для морфологічного аналізу українських слів, NLTK для лематизації, scikit-learn для імплементації наївного баєсового класифікатора та функцій тренування моделі.

У четвертому розділі було проведено тестування розроблених програмних модулів. Також було розглянуто декілька можливих сценаріїв використання програми, після чого було підтверджено відповідність програми поставленим вимогам.

					ІАЛЦ.467200.003 ПЗ	Арк.
						53
Зм.	Арк.	№ докум	Підпис	Дата		

Отже, програмні модулі, розроблені в межах дипломної роботи, є ефективним інструментом для безпечного користування мережею інтернет.

Технічне завдання на дипломний бакалаврський проєкт виконано повністю.

					ІАЛЦ.467200.003 ПЗ	Арк.
						54
Зм.	Арк.	№ докум	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. An Exploratory Analysis of the Relation Between Offensive Language and Mental Health [Електронний ресурс] – Режим доступу до ресурсу: <https://arxiv.org/pdf/2105.14888>
2. Історія веббраузера [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/History_of_the_web_browser
3. Lynx [Електронний ресурс] – Режим доступу до ресурсу: <https://lynx.browser.org/>
4. Вебмагазин Chrome [Електронний ресурс] – Режим доступу до ресурсу: <https://chromewebstore.google.com/>
5. Українська електронна бібліотека [Електронний ресурс] – Режим доступу до ресурсу: <https://ukrayinska.libretexts.org/>
6. Опитування Stack Overflow 2023 [Електронний ресурс] – Режим доступу до ресурсу: <https://survey.stackoverflow.co/2023/>
7. Наївний баєсовий класифікатор [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
8. Документація pymorphy2-dicts-uk [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/pymorphy2-dicts-uk/>
9. Звіт Akamai про стан онлайн-роздрібної торгівлі навесні 2017 року [Електронний ресурс] – Режим доступу до ресурсу: <https://www.akamai.com/newsroom/press-release/akamai-releases-spring-2017-state-of-online-retail-performance-report>
10. Machine Learning Classification Algorithms [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/top-6-machine-learning-algorithms-for-classification/>

					ІАЛЦ.467200.003 ПЗ	Арк.
						55
Зм.	Арк.	№ докум	Підпис	Дата		

ДОДАТОК 1

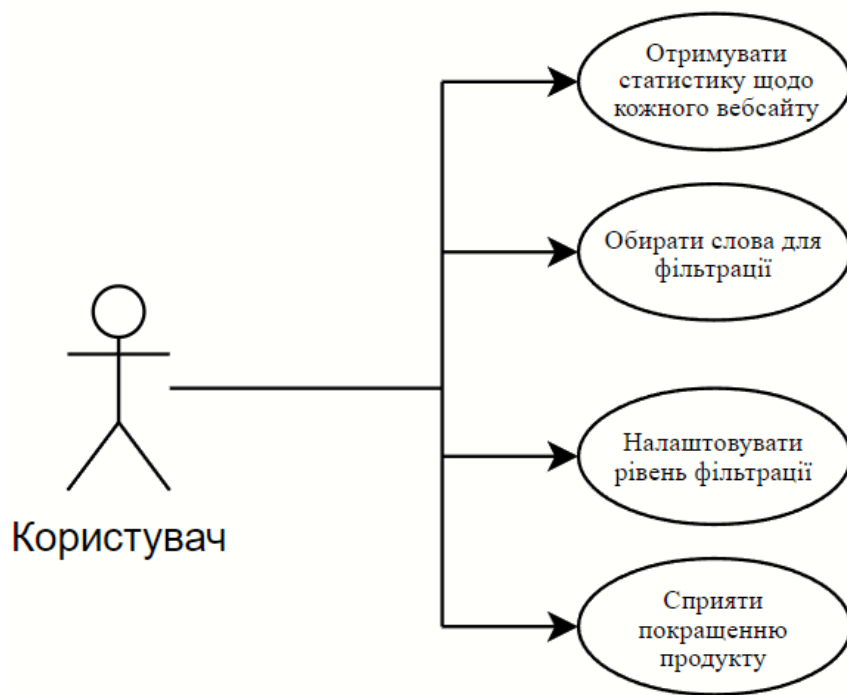
Плагін для Chromium для фільтрації контенту

Схема системи (структурна схема)

ІАЛЦ.467200.004 Д1

Аркушів 1

Київ 2024 р.



					ІАЛЦ.467200.004 Д1			
Зм.	Арк.	№ докум	Підпис	Дата	Плагін для Chromium для фільтрації контенту Схема системи Структурна схема	Літ.	Аркуш	Аркушів
Розробила		Бібік М.С.					1	1
Перевірив		Гайдай А.Р.						
Н. контр.		Виноградов Ю.М.				КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО-01		
Затверд.		Стіренко С.Г.						

ДОДАТОК 2

Плагін для Chromium для фільтрації контенту

Взаємодія підсистем (функціональна схема)

ІАЛЦ.467200.005 Д2

Аркушів 1

Київ 2024 р.



					ІАЛЦ.467200.005 Д2			
Зм.	Арк.	№ докум	Підпис	Дата	Плагін для Chromium для фільтрації контенту Взаємодія підсистем Функціональна схема	Літ.	Аркуш	Аркушів
Розробила	Бібік М.С.						1	1
Перевірив	Гайдай А.Р.							
Н. контр.	Виноградов Ю.М.					КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО-01		
Затверд.	Стіренко С.Г.							

ДОДАТОК 3

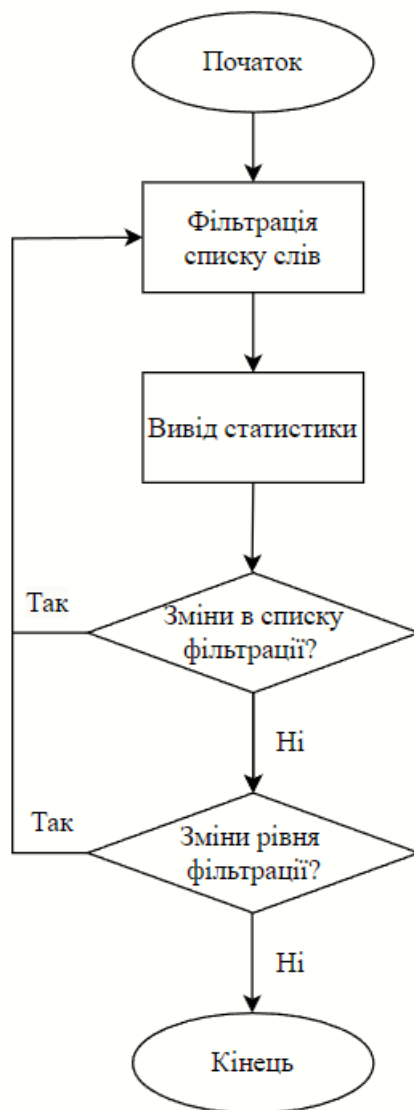
Плагін для Chromium для фільтрації контенту

Алгоритм роботи системи (принципова схема)

ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ 2024 р.



					ІАЛЦ.467200.006 ДЗ			
Зм.	Арк.	№ докум	Підпис	Дата	Плагін для Chromium для фільтрації контенту Алгоритм роботи системи Принципова схема	Літ.	Аркуш	Аркушів
Розробила	Бібік М.С.						1	1
Перевірив	Гайдай А.Р.							
Н. контр.	Виноградов Ю.М.					КПІ ім. Ігоря Сікорського, ФІОТ, Група ІО-01		
Затверд.	Стіренко С.Г.							