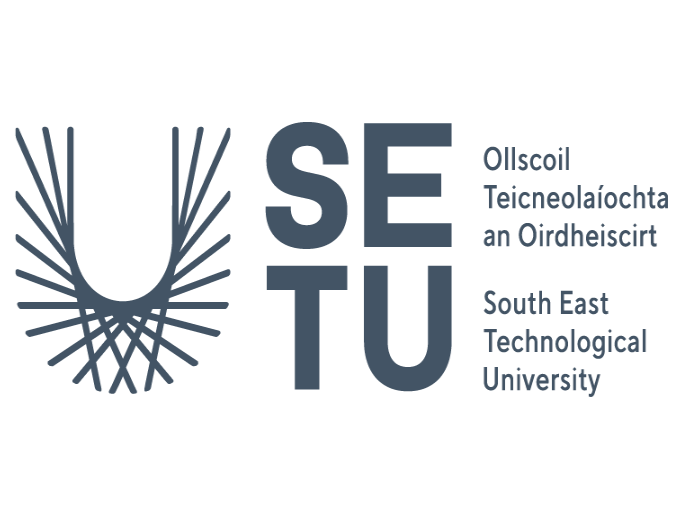
**Project Report Document**



**Name**: Dawid Pionk

**Supervisor:** Jamal Tauseef

**Institution**: SETU, Carlow

**Date**: 28/04/2025

# Table of contents

[Introduction 2](#_Toc1427414861)

[Abstract 3](#_Toc1601091498)

[Problems encountered 4](#_Toc1673955485)

[Twitter issues 4](#_Toc740102703)

[Deciding sentiment analysis model 5](#_Toc1030408476)

[Styling issues 5](#_Toc235626827)

[What I achieved 6](#_Toc2086285293)

[Sentiment Analysis of posts 6](#_Toc1126772226)

[Extraction of posts 7](#_Toc61827392)

[Data visualization 8](#_Toc688687992)

[Insight into individual Posts/Comments 9](#_Toc80736166)

[What I did not achieve 10](#_Toc497448206)

[Website design for mobile 11](#_Toc2104437392)

[Compatibility with other social media platforms 11](#_Toc142468116)

[Sentiment classification for images 11](#_Toc1342448866)

[What I learned 11](#_Toc1300648535)

[Website design 11](#_Toc549545206)

[Interacting with social media API 11](#_Toc1764712625)

[Agile Development 11](#_Toc1220340529)

[What I would do differently If I had to start again 12](#_Toc655695437)

[Differences from initial designs 12](#_Toc1658859186)

[Module descriptions 13](#_Toc1892719587)

[Frontend Module 13](#_Toc1417600080)

[Backend Module 13](#_Toc125091021)

[Sentiment Analysis module 14](#_Toc2036778795)

[Data Handling module 14](#_Toc2141197242)

[Praw module 15](#_Toc212822554)

[Data structures including any files and/or databases and their record layouts 15](#_Toc1203603907)

[Conclusion 16](#_Toc2030910246)

[Glossary 17](#_Toc2014951452)

[Bibliography 18](#_Toc233929291)

Table of tables and figures

Table 1. dataJSON.json structure..................................................................................18

Table 2. redditor.json structure.....................................................................................18

Figure 1. A screenshot of the form in which the user enters to extract posts....................10

Figure 2. A screenshot of a piece of the data visualization section..................................11

Figure 3. A screenshot of the view posts section............................................................12

Figure 4. A screenshot of a post clicked in the view posts section...................................12

# Introduction

This project aimed to create a website in which a user provides certain topics they want analyzed, the website extracts posts from Reddit\*, classifies their sentiment for the user, and then displays it back to them in an easy-to-understand format. The goal of this tool is to provide a useful research tool for companies/individuals who have an interest in how the public perceives them or their products.

A use case example is a YouTuber whose majority of his audience uses Reddit, could search for posts mentioning him, while he is receiving a lot of criticism and then find out what are some common complaints. Another example would be someone who is writing an article on a certain celebrity who has a Reddit account. They would be able to research how they behaved and what things they were saying.

Due to this project, I was able to develop my skills such as web design, data management, machine learning, data visualization, and interaction with APIs\*. The technologies I have used in this project include:

* **HTML, CSS, and JavaScript** -> basic frontend
* **Bootstrap** -> framework for frontend development\*
* **Chart.js** and **d3.js** -> data visualization library for JavaScript
* **Python** -> backend\*
* **Praw** -> Reddit API interaction
* **Hugging face** and **NLTK** -> libraries for machine learning\* models
* And others

This report will examine important aspects of my development over the past 8 months. In this report, I will address some of the general and technical issues I have faced. What were the biggest challenges I had to resolve? I will show what features I implemented and what features had to be dropped. Overall, I have significantly improved my skills and come to understand the software development lifecycle throughout this project. I have become a better programmer by developing skills in several areas that will be useful in my career.

# Abstract

This project was focused on creating a sentiment analysis\* tool for Reddit hosted on the web. A user enters certain topics with additional parameters to they wish to have analyzed. Then those posts/comments are extracted and classified based on their sentiment, the results then are displayed back to the user. The project utilized HTML, CSS, JS, bootstrap and a few other libraries for the frontend. As well as python, NLTK, hugging face and praw for the backend. The goal of this tool is to allow users to gain powerful insights into certain persons/brands/topic opinion on reddit. This project overall has proven itself to be a great exercise in developing my skills in full-stack-development\*, data analysis and machine learning.

# Problems encountered

In this section, I will be talking about the challenges I faced and how I resolved them. Some of the problems that had a major impact on my project were:

* Twitter Issues
* Deciding Sentiment analysis model
* Styling issues

### **Twitter issues**

When I first began my project, I wanted to analyze sentiment on Twitter. I decided this because the discourse on Twitter tends to be more opinionated and has stronger sentiments based on my use, but I ran into couple of issues because of this choice.

The first problem I ran with Twitter was its API changes. While I was researching my project, I didn’t realize that changes have been made to its API. These changes introduced a lot of obstacles in developing programs that rely on access to it. The access has been reduced to just 1500 tweets per month which essentially is useless for any project (X Corp., 2024). Since my project requires each request by a user to extract around 100 posts, the limit essentially makes the free tier useless.

The solution I thought of at the time since I wanted to use Twitter was web scraping\* the posts. There were a couple of issues I ran with this approach. Firstly, when I tried to use a dedicated library to scrape Twitter called snscrape it was unfortunately out of date. The next approach I decided to take was creating a program to web scrape and set up safeguards against my IP getting banned. The problem I ran with this approach was that Twitter implemented changes to not being able to search without having an account. Due to these reasons as well as being afraid of breaking terms of service, I decided to stop going down this path.

The solution for this issue was that I decided to abandon using twitter and do more research on which websites are more friendly towards student developers. I compared the free tiers of several social media sites and eventually decided to stick with reddit due to it having an API that seemed easy to use and had high request rates.

I learned a lot from this entire process. It though me how I should interact with APIs, how different social media APIs differ. It made me understand the importance of doing research with more current sources.

## **Deciding sentiment analysis model**

I had a lot of issues trying to decide what model to use. The major issue that I ran into was deciding between polarity vs classification. I couldn’t get a pre-trained model that did both. I needed my model to classify posts based on whether they are positive or negative and determine how negative or positive a post is. Also, I wanted to be able to determine the sentiment of images posted along with a post.

I came up with 2 solutions for this problem. The first solution was that I could use an AI to classify posts. I would simply run each request through an AI model that would classify and give me a polarity score. The other solution was using a pre-trained classification and polarity modal as well as combining it and training my image classification model.

I decided to settle on the latter option. Due to several reasons, firstly I would need to pay to have each post analyzed by an AI by buying tokens\* for each request, this would quickly add up while extracting several hundred posts, especially with images. As well as that I thought that the second option would be a much greater learning opportunity since setting up a model is quite easy, and I didn’t feel I would learn much from it.

As I was applying the second solution I learned that the posts could mainly be classified essentially by the title, meaning that image classification would take a lot of time with minimal rewards, so I decided to move down my priority list and eventually decide to abandon the idea and focus more other modules.

## **Styling issues**

A major issue I faced was the design of user interfaces. I wanted my website to be appealing since the majority of the time nice and clear designs are the biggest factor in users staying and interacting with the website. According to (Evaluate Media, 2022) users take 50 milliseconds to decide whether to stay or leave. Therefore, I spent the majority of my 3rd iteration focusing on design improvements.

My biggest problems were:

* Making the hero page grab users attention
* Making the main task which is sentiment analysis stick out
* Making content resizable for different screens.

I had an issue in deciding what the hero page was going to be, it had to have a nice design since it was the first thing a user would see. How I decided to resolve this issue was by combining common elements of websites that had hero pages that caught my eye. Some of these elements include large fonts, bright colors, and animations. I wanted the page to stick out from competing services whose initial pages are mostly plain, so I decided by making my hero page flashy it would be easier to catch someone's attention.

At first, I had a problem deciding where I could place my sentiment analysis section on my page. I had the idea of placing it in a different section on the home page, having an entirely different page dedicated just to it. Eventually, I resolved this problem by deciding to place it on the left side of my hero page. I decided to place it here since the hero page is the first page a user sees and on the left side since most users look at content from left to right. I wanted it here since I wanted the main task and purpose of the website to be the first thing a user looks at.

A major problem that I needed to learn how to deal with was making websites fit on different screen sizes and making sure elements don’t break or overflow. To solve this issue, I did research on different libraries and techniques web developers use for this purpose. How I resolved this issue was by deciding to settle on using a free open-source front-end framework called Bootstrap. I decided to convert and redesign a lot of my website to use their elements and then I could specify easily options for screen sizes. Some of my original elements I decided to keep and make my own styling rules for sizes. Although it made resizing for different screen sizes easier, I would have needed to learn how to design websites for mobile which would have taken a lot of my time. I decided to just specify rules for monitors since the majority of usage for my program would be used by desktops or laptops.

# What I achieved

I achieved a lot of useful features that a user can take advantage of to fulfill their research tasks.

Following are the goals I set out to complete at the start of the project that I completed:

1. Analyze Sentiment (core)
2. Search by user, topic brand (core) -> Extraction of posts
3. Display Sentiment statistics (core) -> Data visualization features

## **Sentiment Analysis of posts**

I created a feature which predicts the sentiment within a post. This feature allows the program to pass any post or comment and then receive a classification and polarity score.

Through the development of this feature, I was able to develop my skills and learn how to use technologies. I was able to learn more about how machine learning works, how different systems such as BERT transformers work like. I was able to understand how to use prebuild model through hugging face transformers and NLTK libraries

## **Extraction of posts**

I was able to achieve a system that efficiently allows a user to extract a lot of different information from reddit. The user is able to interact with a form and extract posts/comments based on certain criteria. Below I show each category they can use:

* Topic – Looks for posts containing the provided topic
* User – A specific account on reddit
* Subreddit – A community page on reddit
* Comments – Comments under a post
* Domain – Posts linking to specific domain like bbc.com

Through the development of this feature, I developed my skills and learned how to use several different new technologies. I learned how to interact with a social media API how to handle such data, and how to sort it and extract useful information out of it. I learned how to interact with Reddit's API specifically and how to use the Praw library.

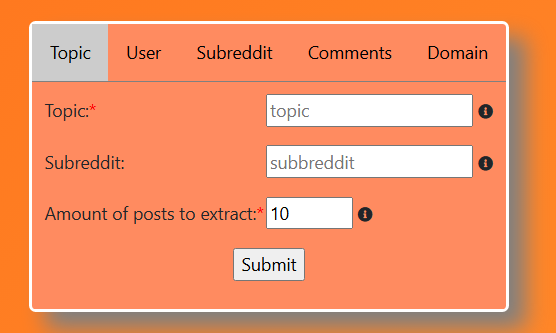


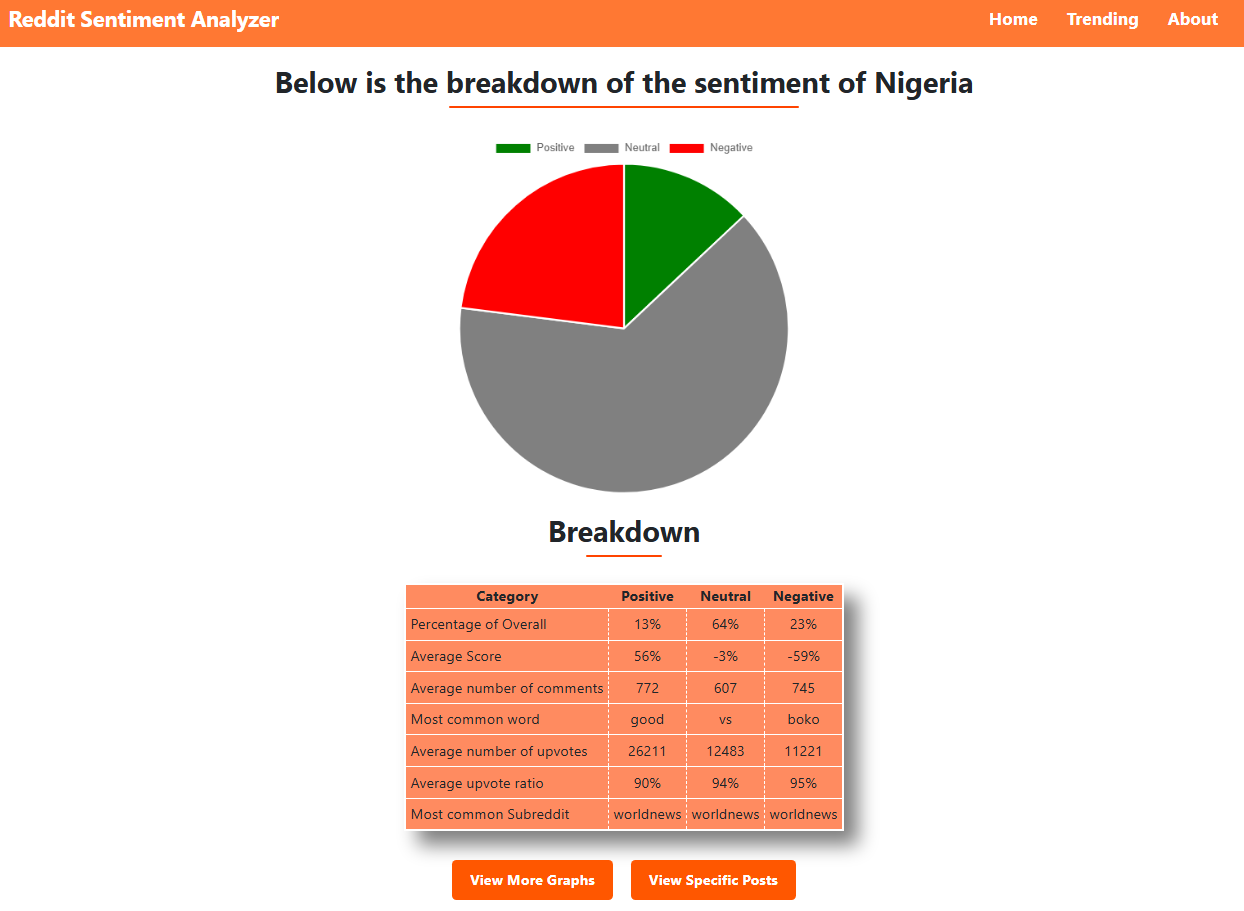
Figure 1 – A screenshot of the form in which the user enters to extract posts

### Data visualization

I managed to achieve a feature in which the user can more easily understand the data processed by my program. Here the user can look at several tables, charts and graphs to get better insight on the sentiment of their topics.

I have the following data visualization sections added into my project:

* **Pie chart of breakdown:** Simple pie chart showing the amount of posts that were positive/neutral/negative
* **Breakdown table:** Table showing breakdown of certain stats per classification category, such as average score, average number of comments, etc.
* **Posts per subreddit:** A stacked bar chart that shows how many posts were made in each subreddit and with what sentiment
* **Sentiment over time**: A simple line graph showing the average sentiment over time
* **Word Cloud:** A simple word cloud showing the most common words in each category.
* **Scatter Graph for sentiment vs the number of comments:** A scatter graph showing the polarity of a post on the y-axis and the number of comments it received on the x-axis.
* **Scatter graph for sentiment vs upvote count:** A scatter graph showing the polarity of a post on the y-axis and the number of upvotes it received on the x-axis.

Figure 2 - A screenshot of a piece of the data visualization section

## **Insight into individual Posts/Comments**

For this feature, I was able to achieve a section in which a user is able to view each of the posts that was extracted in their query. They are placed under whichever classification category they belong to. The user can click and view additional information about each post/comment. This is useful for the user if they want to gain insight into specific criticisms or praise, they received.



Figure 3 – A screenshot of the view posts section



Figure 4 – A screenshot of a post clicked in the view posts section

# What I did not achieve

## Website design for mobile

**Reason for lack of features:** I had to focus on more relevant features. Spend a long time trying to learn how to design desktop websites to make them appealing, I would have need to spend just as much time trying to figure out how to design mobile.

**Impact on the overall project:** The design of the website isn’t suitable for smaller screens, which results in the website layout breaking.

## Compatibility with other social media platforms

**Reason for lack of features:** I had to focus on more relevant features. Focusing on creating keys on different APIs and learning how they work and adjusting rate limits would have taken too much time.

**Impact on the overall project:** Lack of ability for users to assess sentiment on different platforms

## Sentiment classification for images

**Reason for lack of feature:** It would have taken too long to train a model, as well as being too intensive to host.

**Impact on the overall project:** Only title and text of body would have been analyzed.

# What I learned

## **Website design**

During the course of this project, I learned how to design frontends for my programs. I learned several design principles that help my designs stand out. I gained additional insight into technologies like CSS, bootstrap and other frameworks. I also learned how to combine elements I liked from different sources to create a unique and interesting design. This will aid me in the feature whenever I need to make user interfaces.

## **Interacting with social media API**

For this project I needed to learn how to research and interact with APIs. I had to create API keys, interact with 3rd party websites and figure out how rate limits work. I also learned how to handle data from such requests. The specific technologies I learned were the Reddit's API, praw and pandas. The skills I gained will be useful in the future whenever I need to interact with an API that's foreign to me.

## **Agile Development**

During the development of this project, I needed to learn how to better manage my development cycle. I had to create priority lists, focus on planning out development and learn how to research technologies. I also had to learn when to abandon certain features or scale back on my project's scope. These are general skills that will improve how I develop and plan the development of programs.

# What I would do differently If I had to start again

During my project I made a lot of mistakes that ended up slowing me down and negatively impacting the final product. Below are the 2 most major ones that impacted my development the most.

I would make sure my research is as current as possible and make sure the technologies that I am using are not obsolete and up to date. This was especially an issue at the start due to my poor research. I slowed down for couple of weeks trying to figure out how to make twitter compatible with my project. If I had to start again, I would look at more competing products and keep my options open.

I would lower my scale if I had to start again. As the project progressed, I kept finding myself starting on features without properly finishing core components. This became an issue in the 3rd iteration when I had to rush to finish the core features and removing a lot of fluffer that didn’t need to be there. This taught me that next time I tackle a large project, I should follow more strictly to the iterative approach and finish each feature before tackling the next one. This taught me to be more focused in my future approach.

# Differences from initial designs

My initial plan for the architecture was as follows:

1. Tweets would be extracted from X.
2. Would be analyzed using a BERT model
3. Then displayed on a windows desktop app

But this eventually evolved into the following structure:

1. Posts/Comments extracted from reddit
2. Analyzed using a combination of models
3. Displayed on a desktop website

I decided to switch from X to reddit because the limitations from using the free tier of X were making my program difficult to debug. The limitation for X is 50 requests per 24 hours. While for reddit they are much more charitable with 100 requests per minute. This required me to research how to differently handle a new API.

Initially I wanted to get a single model to classify a post's sentiment, but as scope grew, I needed a larger model to also give me a polarity score since that would provide more insight to the user. So eventually instead of settling down on a single model I combined the two.

Initially I wanted to create a desktop app using different python libraries. Eventually I decided to switch to a website. I made this decision for several reasons: This way the tool would work on any device that is able to access the internet, the tool wouldn’t need to be installed also most competing tools are websites based so the user would most likely choose an option that requires the least amount of work to access. This required a lot of research and learning for me to figure out how to correctly design and display my website.

# Module descriptions

In this section I will explain the main modules that are present in my code, i split them into 5 major sections.

## **Frontend Module**

This module allows the user to interact with my application and view information. Here the user is able to interact with forms that they can use to enter what topic they want analyzed. They can view information here to see the breakdown of their analysis.

Key Technologies:

* HTML, CSS, JavaScript -> basic frontend technologies
* Bootstrap -> framework for frontend development
* Chart.js and d3.js for data visualization

Key Features:

* Provide input into a from for analysis
* Select parameter for analysis
* View other information such as contact, use cases, etc.
* Displays sentiment results
* Displays account information
* Displays individual posts
* Gives ability to download all the posts

## **Backend Module**

This is the main module for the program. It manages what pages are loaded in the front end. Manages communication between modules.

Key Technologies:

* Python
* Flask -> web framework for python

Key Features:

* Loads pages
* Takes input
* Transfers input to other modules
* Transfers output from processing modules and transfers to frontend

## **Sentiment Analysis module**

This module is responsible for classifying sentiment

Key Technologies:

* Python
* Transformers -> hugging face library for prebuild models
* NLTK -> natural language processing library

Key Features:

* Takes in posts
* Classifies sentiment
* Determines polarity of that sentiment
* Returns result to data handling module

## **Data Handling module**

This module is responsible for processing data and preparing it to be send to the frontend

Key Technologies:

* Python
* Pandas -> library for data handling

Key Features:

* Takes in data from praw module and sentiment analysis module
* It extracts and sorts of useful information based on their categories
* Puts the data into a Json\* format for the use by the frontend

## **Praw module**

This module is responsible for extracting data from reddit to be used for later analysis.

Key Technologies:

* Python
* Praw -> library to interact with Reddit's API

Key Features:

* Connects to my reddit API Key
* Searches for posts containing certain phrases/words
* Searches for posts or comments made by a specific account
* Searches comments made on a post
* Searches posts made on a subreddit
* Searches posts linking to specific domain

# Data structures including any files and/or databases and their record layouts

For this project I didn’t store any information in databases, but I passed information through a json file from my python backend to my frontend for my JavaScript to deal with.

Table 1 – dataJSON.json structure

|  |  |  |
| --- | --- | --- |
| **dataJSON.json** | | |
| **Field Name** | **Data type** | **Description** |
| title | string | The title of a post |
| subreddit | string | The subreddit a post came from |
| author | string | Author of a post |
| Created\_utc | string | The time the post was created |
| Num\_comments | string | Number of comments on a post |
| Over\_18 | string | Was the post marked over 18 |
| permalink | string | The link to the post |
| upvotes | string | The amount of upvotes on a post |
| upvote\_ratio | string | The upvotes ratio |
| selftext | string | Contains the text contained in a body of a post |
| type | string | Whether |
| url | string | Url linked in a post |
| content | string | Not currently used, for future implementation for image classification |
| contentType | string | Not currently used, for future implementation for image classification |

Table 2 – redditor.json structure

|  |  |  |
| --- | --- | --- |
| **redditor.json** | | |
| **Field Name** | **Data type** | **Description** |
| Comment\_karma | string | How much karma from commenting |
| Created\_utc | string | When was the account created |
| id | string | Account id |
| Is\_employee | string | Is this an employees account |
| Is\_mod | string | Is this account a moderator on any mod |
| Is\_gold | string | Does this account have premius |
| Is\_suspended | string | Is this account suspended |
| Link\_karma | string | Karma from posts |
| name | string | Name of redditor |

# Conclusion

In conclusion the project was successful in creating a sentiment analysis tool for reddit. This tool allows users to research users, topics or communities on reddit. It provides useful insights into public opinion and current trends. This was accomplished by combining several technologies such as python, flask, JavaScript and bootstrap.

Key features of this system include data extraction, data classification and data visualization. These features have been successfully implemented with a clean and easy to understand design. The user is able to clearly distinguish between what posts belong to what category and what criticism or praises each post contains. Although many features were achieved some weren’t such as mobile version and image classification.

There were several major challenges that were faced and resolved during the development of this program. There were issues with the original design of the project. There were difficulties involving the sentiment analysis model. Finding the right way to extract public opinion to analyze. As well as issues involving the styling of the pages. Eventually those issues were resolved, leading to a more cohesive product.

Overall, this project was a great learning experience. I have improved many of my skills as well as picked up new ones. I learned how to better design websites and how to better plan a larger scale project. It was my first experience interacting with social media APIs. My study greatly benefited in the fields of full-stack-development and machine learning. This tool also has the potential for further development, for example the addition of more social media platforms.

# Glossary

*API*

Application program interface, is a set of protocols that allow different software to communicate

*Backend Development*

The development of software that deals with what the user can’t directly see

*Frontend Development*

The development of software that deals with what the user can directly see

*Full-Stack-Development*

The development combining frontend and backend development

*JSON*

JavaScript Object Notation, is an easy lightweight way to exchange data

*Machine learning*

The field in artificial intelligence that deals with machines learning based on provided data

*Reddit*

A social media platform

*Sentiment analysis*

The technique that tries to predict what human emotion is associated with a text

*Token*

A basic unit of processing used by AI

*Web scraping*

The process of extracting data from a webpage

# Bibliography

1. X Corp. (2024) *X API Rate Limits*. Available at: <https://docs.x.com/x-api/fundamentals/rate-limits> (Accessed: 26 April 2025).
2. Evaluate Media (2022) 'The Role of Visual Appeal in Web Design'. Available at: <https://evaluate.ng/the-role-of-visual-appeal-in-web-design/> (Accessed: 26 April 2025).