**Project Report: Instagram Spam Detection System for Enhanced User Experience**

****

**1. Introduction**

Social media platforms, especially Instagram, are prone to spam content, which negatively impacts user experience and security. Spam comments, found under both posts and reels, are particularly problematic, as they can contain fake promotions, phishing attempts, or irrelevant content. The **Instagram Spam Detection System** was developed to automatically detect and categorize spam comments using AI-based techniques. This system focuses on identifying spam comments under Instagram posts and reels, improving user experience by making it easier to manage and remove spam content.

**2. Problem Statement**

Instagram users frequently encounter spam comments under posts and reels, including:

* **Spam Comments**: Repetitive or irrelevant comments promoting fake offers, external links, or phishing scams.

The sheer volume of user interactions makes manual detection of spam difficult. Existing systems lack the capability for automatic detection of spam comments. There is a clear need for an automated system that can detect and categorize spam comments effectively.

**3. Objectives**

The main objectives of this project are:

* **AI-Powered Spam Detection**: Automatically detect spam comments under Instagram posts and reels.
* **Categorization**: Classify detected comments into high, medium, or low-risk categories based on predefined criteria.
* **Real-Time Detection**: Alert users when new spam comments are detected.
* **User Control**: Allow users to review and manage flagged comments.

**4. System Design**

The **Instagram Spam Detection System** is divided into two main components: the **frontend** (user interface) and the **backend** (spam detection engine).

**Frontend (User Interface):**

* **Spam Management**: Users are alerted to flagged comments and can review, approve, or dismiss them.

**Backend (AI and Detection Engine):**

* **Spam Detection Engine**: This engine uses keyword-based classification to identify spam comments.
* **API Integration**: The system fetches comments from Instagram posts and reels using the **Instagram Scraper API**.

**5. Methodology**

**Step 1: Setting Up the Environment**

The project utilizes **Selenium WebDriver** for browser automation. Selenium enables the system to interact with Instagram, retrieve posts and reels, and scrape comments using an external API.

**Step 2: Spam Detection**

Spam comments are detected by checking them for specific keywords associated with spam activities, such as:

* **High-Risk Keywords**: Words like "scam," "fake," and "offer" that signal harmful or deceptive content.
* **Medium-Risk Keywords**: Words like "free," "deal," and "discount" that suggest suspicious content.

**Step 3: Categorization**

Detected comments are categorized into the following levels:

* **High Risk**: Comments that contain harmful or deceptive content, such as scams or phishing attempts.
* **Medium Risk**: Suspicious comments that could potentially be spam but are less harmful.
* **Low Risk**: Normal comments that do not indicate spam behavior.

**Step 4: User Interaction**

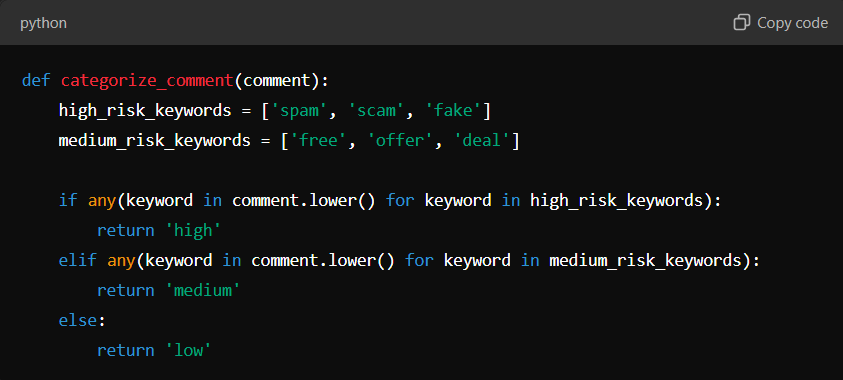
Once spam comments are flagged, users are notified in real-time. They can then review the flagged comments and decide whether to approve or dismiss them.

**6. Code Overview**

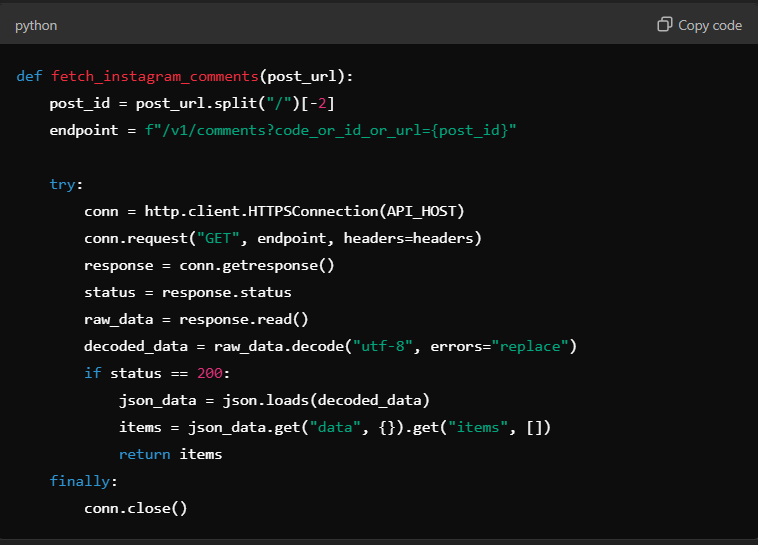
The main functions and key technologies used in this project are outlined below.

**Main Functions:**

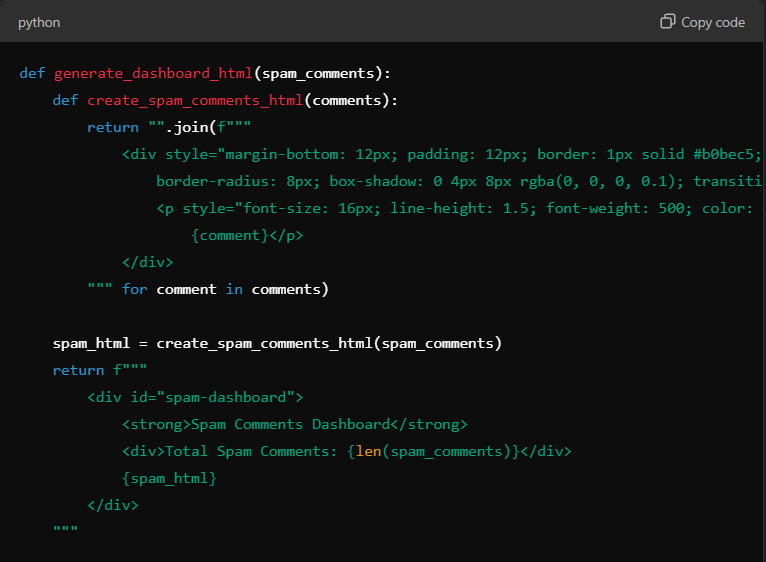
1. **Categorizing Comments**: The function categorize\_comment(comment) categorizes each comment based on the presence of spam-related keywords.



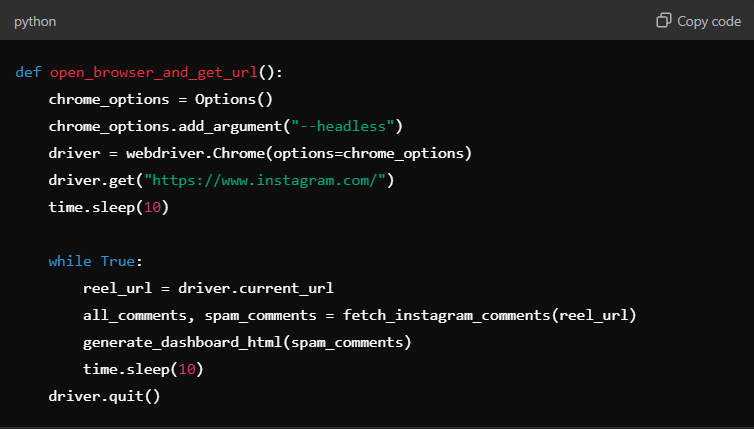
1. **Fetching Comments**: The fetch\_instagram\_comments(post\_url) function retrieves comments from Instagram posts and reels using the **Instagram Scraper AP**



1. **Handling Spam Content**: The function generate\_dashboard\_html(spam\_comments) generates HTML content for displaying flagged spam comments. The HTML is injected into the Instagram interface for user interaction.



1. **Browser Automation**: The function open\_browser\_and\_get\_url() utilizes **Selenium WebDriver** to open Instagram, fetch comments, and display flagged spam comments.



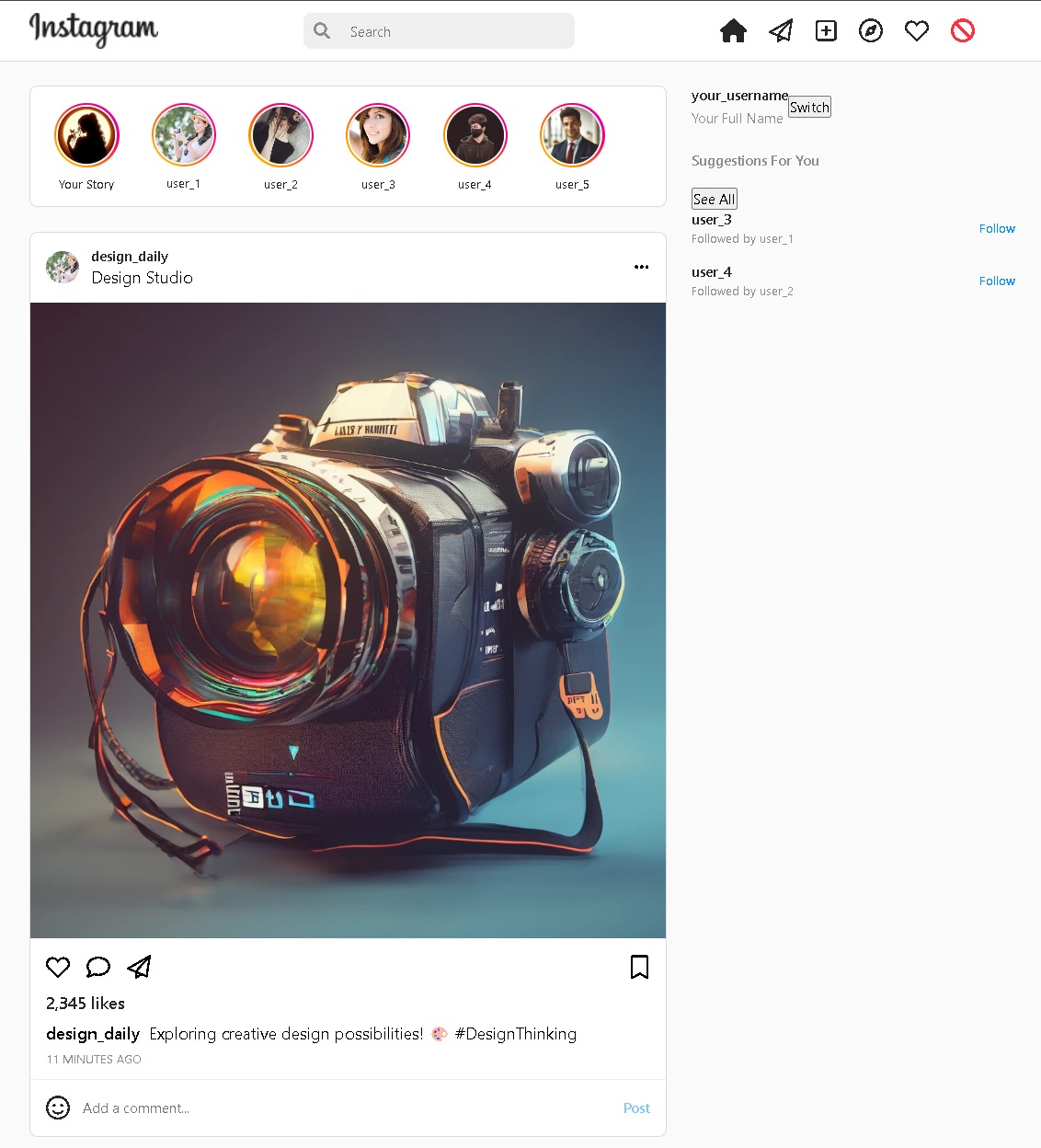
**7. Technologies and Modules Used**

* **Selenium WebDriver**:
  + Selenium is used for automating browser interactions, which allows the system to fetch posts, reels, and comments from Instagram in real-time.
* **RapidAPI Instagram Scraper API**:
  + The API is used to retrieve comments from Instagram posts and reels. It allows the system to scrape the comments without the need for complex web scraping techniques.
* **Python Libraries**:
  + **http.client**: Used to send HTTP requests to the Instagram Scraper API.
  + **json**: Handles the parsing and formatting of data fetched from the API.
  + **time**: Adds delays to ensure that content loads properly and prevents rapid repeated requests.
* **HTML/CSS**:
  + HTML is used to dynamically display flagged comments in the browser.

**8. Results and Testing**

**Testing:**

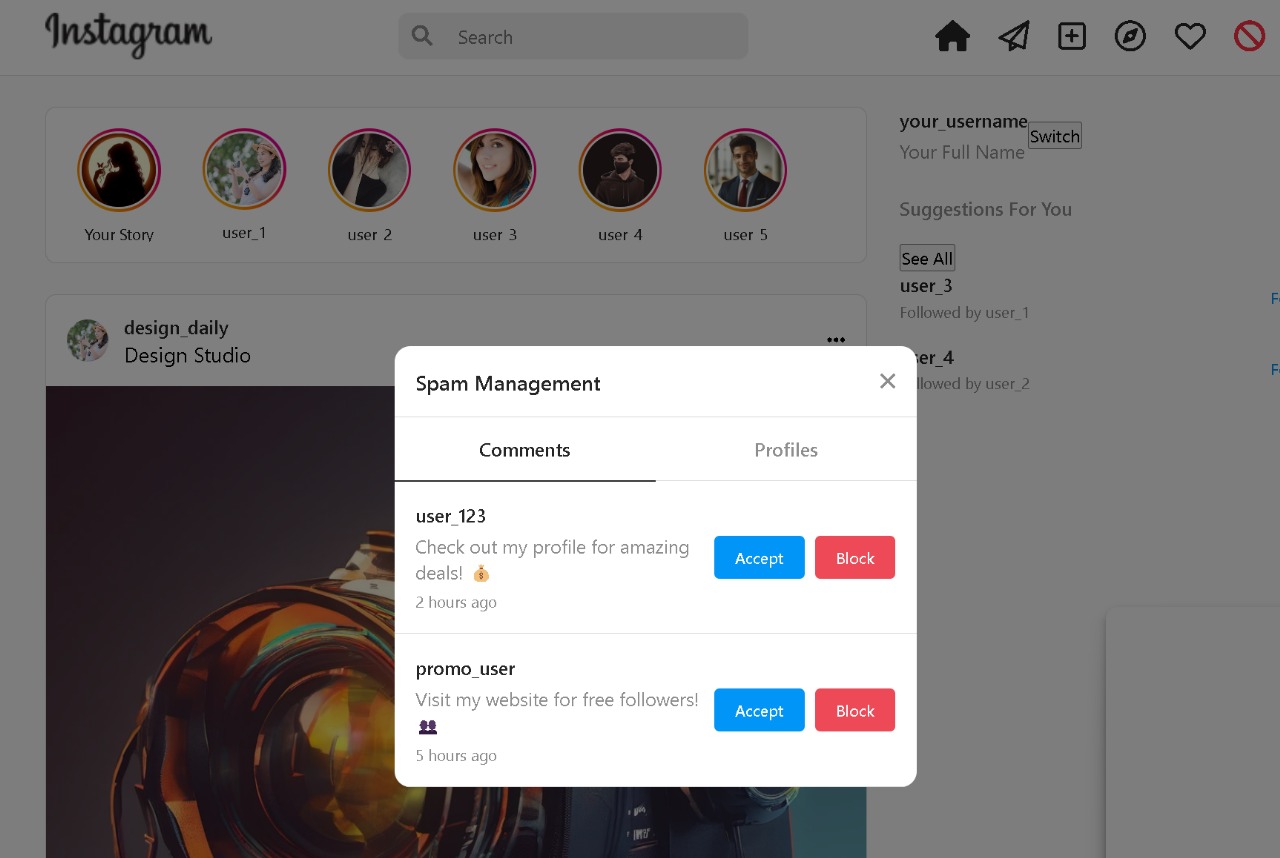
The system was tested using several Instagram posts and reels. It successfully detected and categorized spam comments into high, medium, and low-risk categories. The flagged spam comments were displayed for user review.



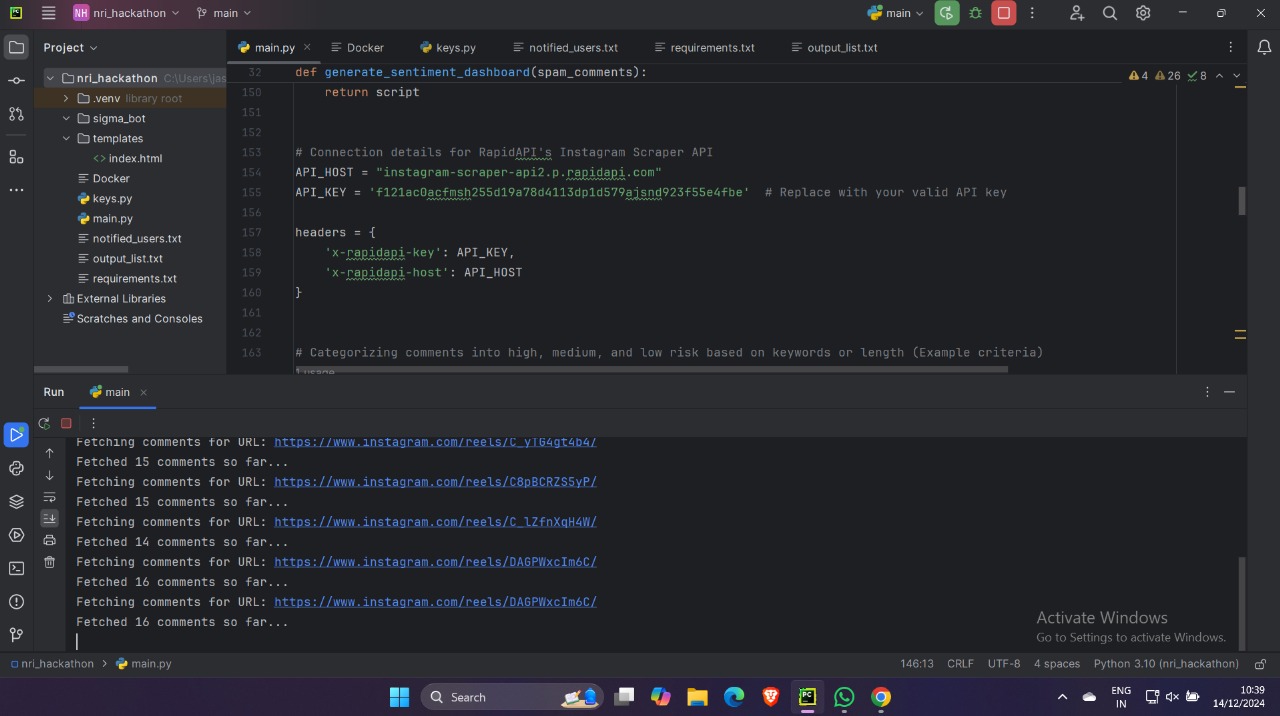
**Figure : Basic Prototype of our Proposed Solution**

**Challenges:**

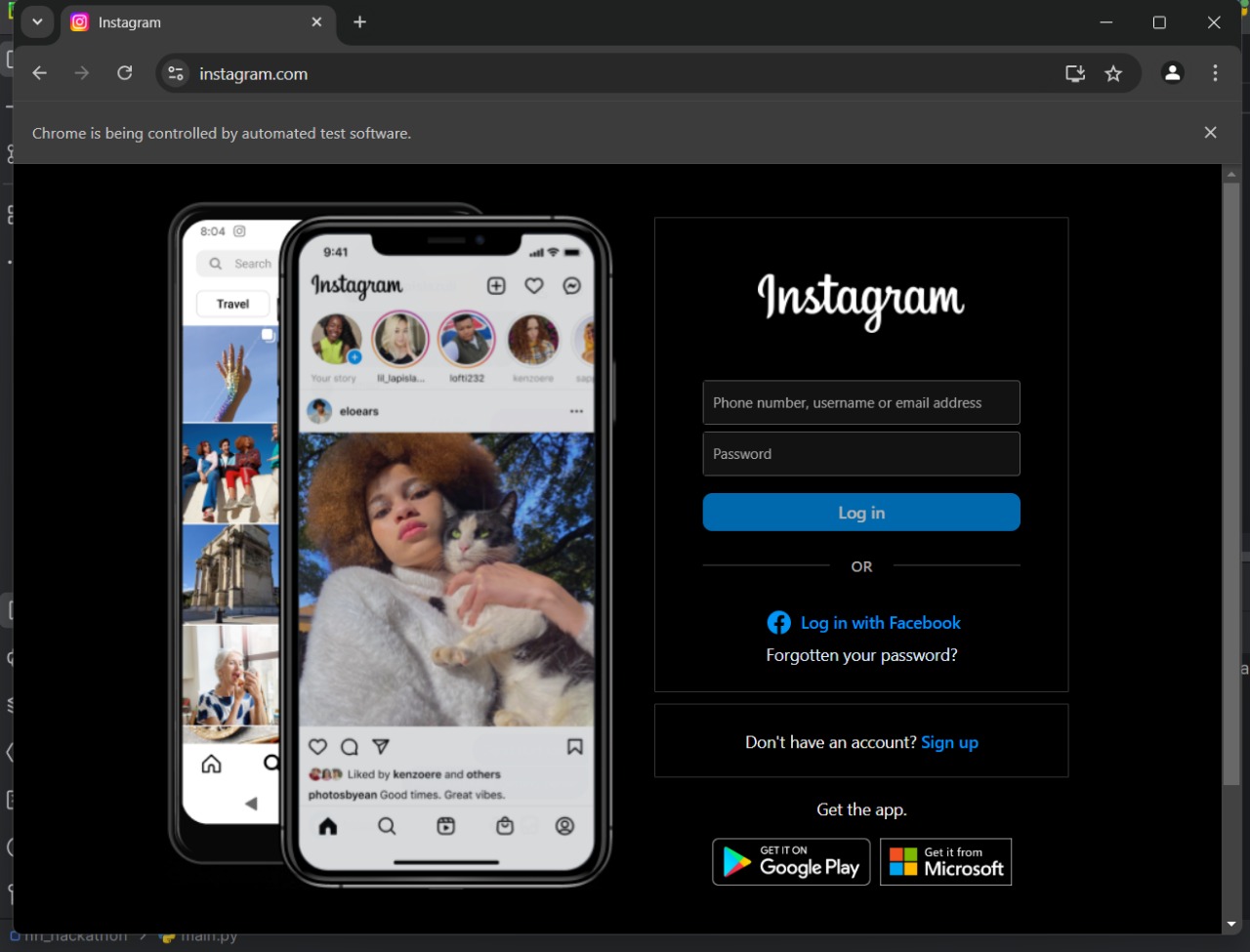
* **Instagram's Dynamic Loading of Comments**: Instagram’s dynamic content loading required Selenium to be used for simulating user interactions and loading all comments.
* **Rate Limits**: The external API has rate limits, which required the system to handle delays effectively.



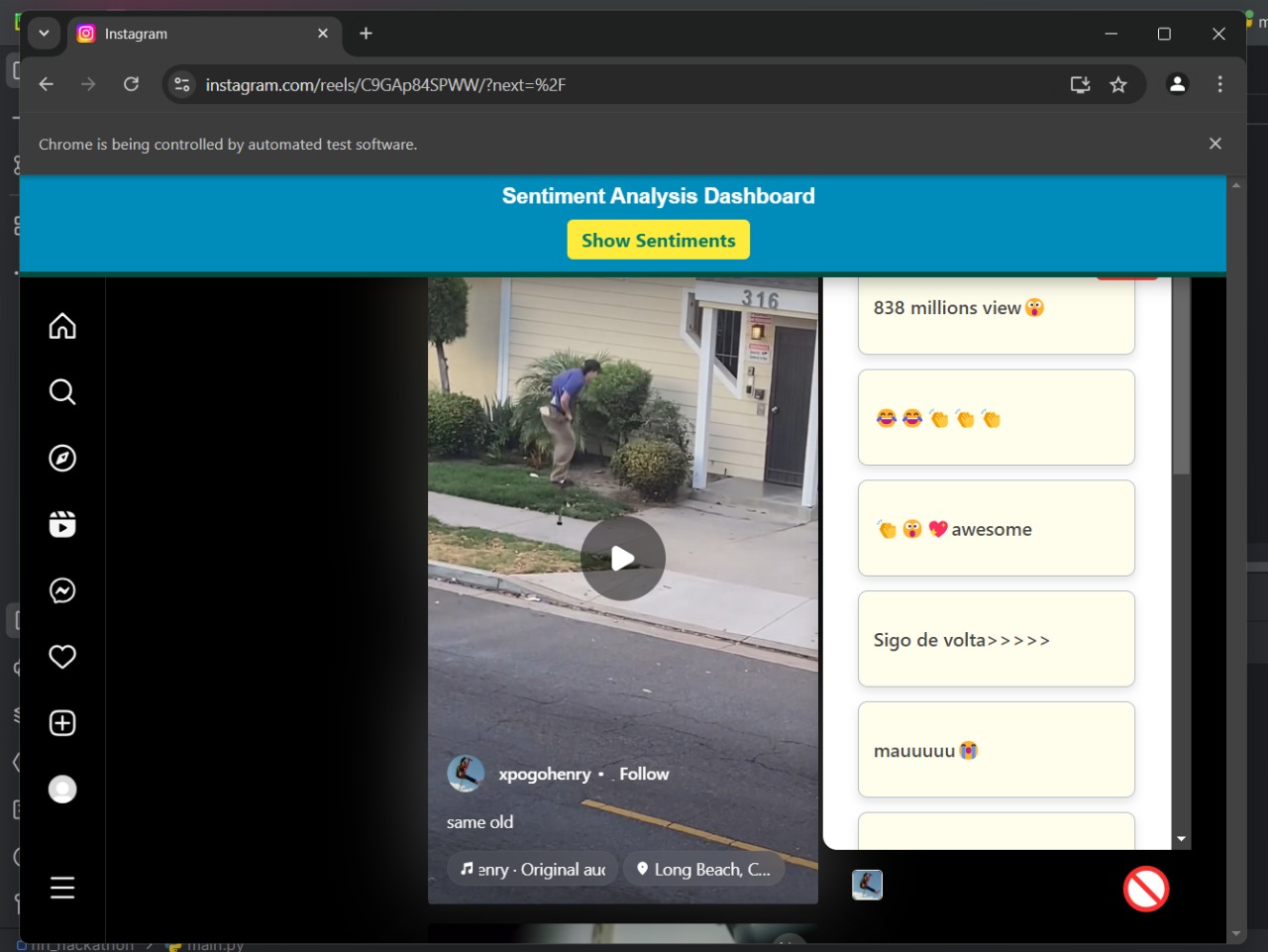
**Figure : Prototype of Spam Management**



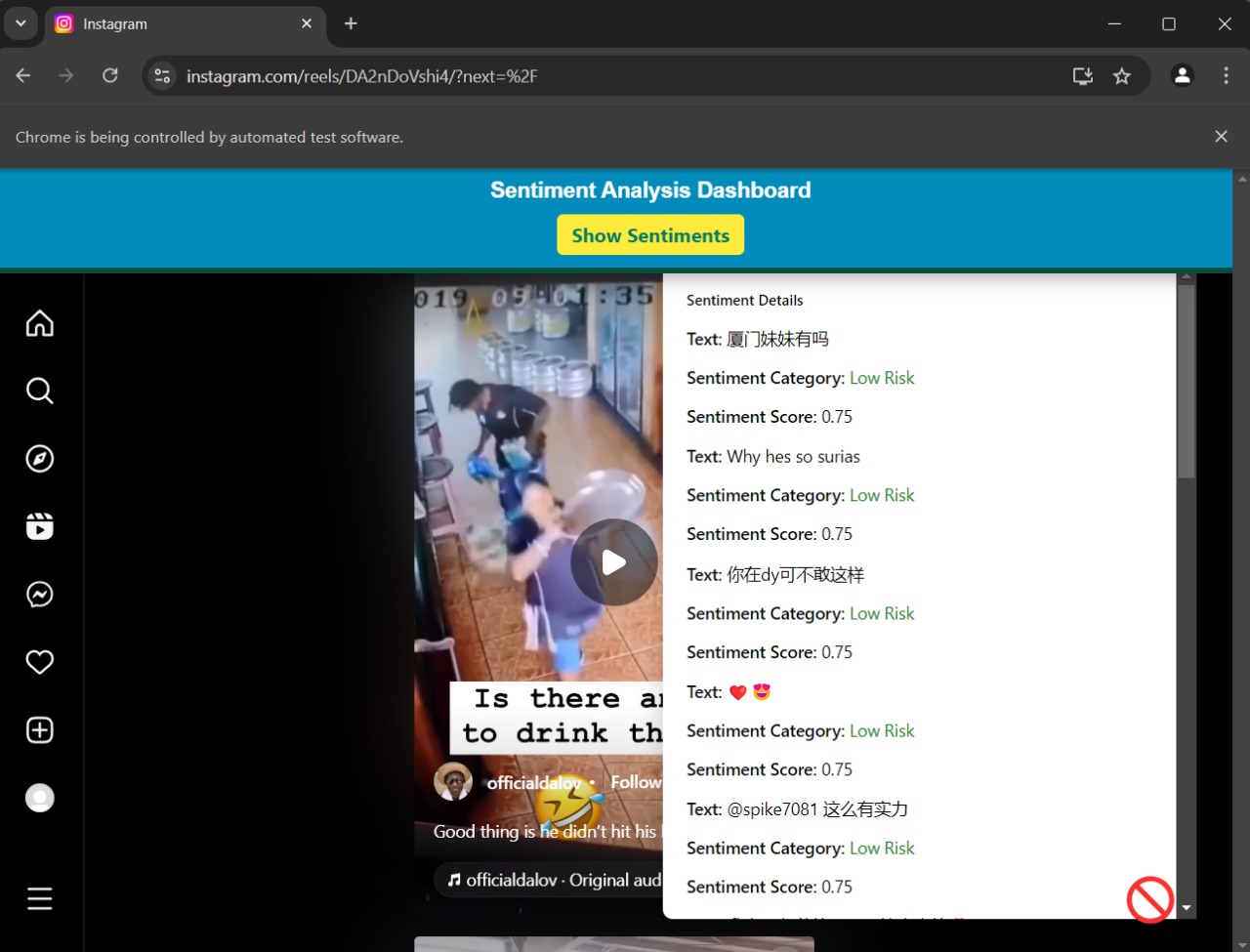
**Figure : Backend of the Real-time Integrated Instagram Platform**



**Figure : Real-Time Instagram Integrated Page**



**Figure : Spam Messages detected in a Reel**



**Figure : Sentiment Priority for according to spam messages**

**9. Conclusion**

The **Instagram Spam Detection System** provides an automated solution to detect and manage spam comments under Instagram posts and reels. By leveraging **Selenium** for browser automation and **RapidAPI’s Instagram Scraper API** for scraping data, the system efficiently flags spam comments based on predefined criteria. Users can easily review and manage flagged comments, improving the overall Instagram experience.

Future improvements may involve **machine learning models** for more accurate spam detection and extending the solution to other social media platforms.

**10. Future Work**

Future enhancements could include:

* **Advanced Machine Learning Models**: Implementing models such as **BERT** for better contextual spam detection.
* **Cross-Platform Integration**: Extending the solution to work on other platforms like Facebook and Twitter.
* **Behavioral Analysis**: Detecting bot-like behavior or fake accounts based on comment patterns.

**11. References**

* Selenium WebDriver Documentation: https://www.selenium.dev/documentation
* RapidAPI Instagram Scraper API Documentation: <https://rapidapi.com>
* "Spam Detection with NLP" – Research Papers on Text-based Spam Detection.