

Vignan's Foundation for Science, Technology & Research



Context : Campus Recruitment Training (2025) – Capstone project

Course : API design & Development

Batch : 2nd Year Batch-1

Title of the Project : Monitoring Suspicious Discussions

Teammates:

Name	Registration number
M. Abhilash	231FA04B98
A. Sai Vamsi Krishna	231FA04B55
K. Samuel Ebenezer	231FA04A52
S. Rohit	231FA04C17
P. Varun Sai	231FA04421

0. Abstract :

This project focuses on the development of a content monitoring system using a Flask-based API that evaluates user-generated posts for suspicious content. The system utilizes Natural Language Toolkit (NLTK) for natural language processing tasks such as tokenization, stop-word removal, and text normalization. A predefined database containing a list of suspicious keywords forms the basis for content evaluation. When a post is submitted, the API processes the text and compares it against the stored keywords to determine whether the content is suspicious. If any matches are found, the post is flagged accordingly. This solution offers a lightweight and effective method for automating content moderation, with potential applications in social media platforms, forums, and other user-interactive systems that require real-time monitoring of textual inputs.

1. Introduction

1.1 Problem Statement :

This project focuses on the development of a content monitoring system using a Flask-based API that evaluates user-generated posts for suspicious content. The system utilizes Natural Language Toolkit (NLTK) for natural language processing tasks such as tokenization, stop-word removal, and text normalization. A predefined database containing a list of suspicious keywords forms the basis for content evaluation. When a post is submitted, the API processes the text and compares it against the stored keywords to determine whether the content is suspicious. If any matches are found, the post is flagged accordingly. This solution offers a lightweight and effective method for automating content moderation, with potential applications in social media platforms, forums, and other user-interactive systems that require real-time monitoring of textual inputs.

1.2 Proposed System :

To address the outlined problem, the proposed system integrates a Flask-based API with a structured database and the Natural Language Toolkit (NLTK) for intelligent text analysis. The database is organized into three main tables: Users, Admins, and Posts. Registered users can submit textual posts through the API. These posts are automatically processed using NLTK to remove stopwords, tokenize text, and compare against a predefined list of suspicious keywords. Posts are categorized into three types based on their content: acceptable, suspicious, and banned. If a post contains severely harmful content, it is automatically rejected and its status marked as "banned". If a post is clean, it is immediately published and visible to both users and admins. Posts containing suspicious words but with unclear context are flagged for manual review by an admin. Each post carries a default post score of 100, which is reduced if suspicious content is detected even if contextually the post is not harmful. A user's cumulative behaviour is tracked via a user score, which increases or decreases based on the nature of their posts. If a user's score drops below a critical threshold (e.g., 40), they are removed from the platform, ensuring that persistent offenders are effectively filtered out. This system provides a balanced and automated approach to content moderation by blending machine learning with human oversight where necessary.

2. Key Technologies

- **Python 3.x**

Python is the core programming language used for backend logic and data processing in this project. With its clean syntax and strong support for libraries such as NLTK, Python enables efficient text preprocessing and analysis. It also integrates seamlessly with Flask for web development and MySQL for database operations.

- **Flask**

Flask is a lightweight and flexible Python web framework used to build the RESTful API for the project. It handles HTTP requests, connects with the database, and routes user actions such as submitting and moderating posts. Flask's modularity and simplicity make it ideal for rapid development and deployment of API-based systems.

- **MySQL Server**

MySQL is the relational database management system used to store and manage the project's data, including users, admins, posts, and suspicious words. It provides structured data storage, indexing, and foreign key relationships to ensure data integrity and consistency across the system.

- **Jinja**

Jinja is the templating engine used with Flask to dynamically generate HTML pages. It allows the integration of backend data into frontend views, enabling conditional rendering, loops, and variable substitution within HTML templates. This helps create interactive and user-responsive web interfaces.

- **Code editor (Visual Studio Code (VS Code) Recommended)**

VS Code is a lightweight yet powerful source code editor used for writing and managing the project's codebase. With features like syntax highlighting, terminal integration, extensions for Python and MySQL, and Git support, it streamlines the development workflow.

- **Web Browser (Chrome Recommended)**

A modern web browser like Google Chrome is used to access and test the web interface of the application. It helps in rendering the frontend, debugging with developer tools, and interacting with the Flask API endpoints during development and testing phases.

3. Description

Project Overview: Suspicious Content Monitoring System Using Flask and NLTK

This project is designed to build a lightweight yet powerful web-based system that monitors user-generated text content for suspicious or harmful language using natural language processing (NLP). At its core, the system consists of a Flask-powered API backend, a MySQL relational database, and an intelligent text analysis mechanism using the Natural Language Toolkit (NLTK). The application aims to automate the process of content moderation, which is critical in today's digital platforms where users frequently post content in real-time.

When a user submits a post, the system performs several tasks:

- Preprocesses the text using NLTK (tokenization, stop word removal, etc.)
- Matches words against a predefined list of suspicious terms stored in the words database table
- Calculates a `post_score` based on the severity and frequency of violations
- Determines whether the post is safe, needs admin review, or should be banned
- Adjusts the user's `user_score` accordingly
- Triggers automatic or manual moderation actions based on thresholds

The database supports three key roles:

1. Users: Contains user information such as username, email, etc
2. Admins: Contains admin info such as name, etc.
3. Posts: contains post information such as post content, poster id, etc.

This approach ensures that content moderation is not just binary (good or bad), but graded based on context, frequency, and user history.

Real-World Usage and Applications

1. Social Media Platforms

On platforms like Facebook, Twitter, or Reddit, millions of users submit posts daily. Manual moderation teams are often overwhelmed, leading to delayed action or missed violations. This system can act as a first layer of defence by automatically flagging posts with suspicious content, reducing moderation load and preventing the spread of harmful content in real-time.

2. Discussion Forums and Online Communities

Websites like Stack Overflow, Quora, or niche forums often deal with community-generated content. This system can help in maintaining decorum by flagging off-topic or toxic discussions while still allowing for contextual review by human moderators when needed.

3. Educational Platforms and e-Learning Portals

In systems where students and instructors interact (like Moodle or Edmodo), content filters are important to keep communication professional and respectful. This project can be adapted to detect slang, bullying, or inappropriate comments in discussion threads or chat features.

4. Corporate Intranets and Feedback Systems

Companies often collect feedback from employees through forms or anonymous portals. This system could help HR teams detect offensive language or threats in submissions, ensuring workplace policies are upheld without compromising anonymity.

5. News Websites and Comment Sections

Public comment sections on news platforms can quickly become toxic. By using this moderation system, inappropriate or hate speech-laden comments can be filtered or flagged before they appear, helping preserve a respectful discussion space.

6. Gaming Communities and Chat Systems

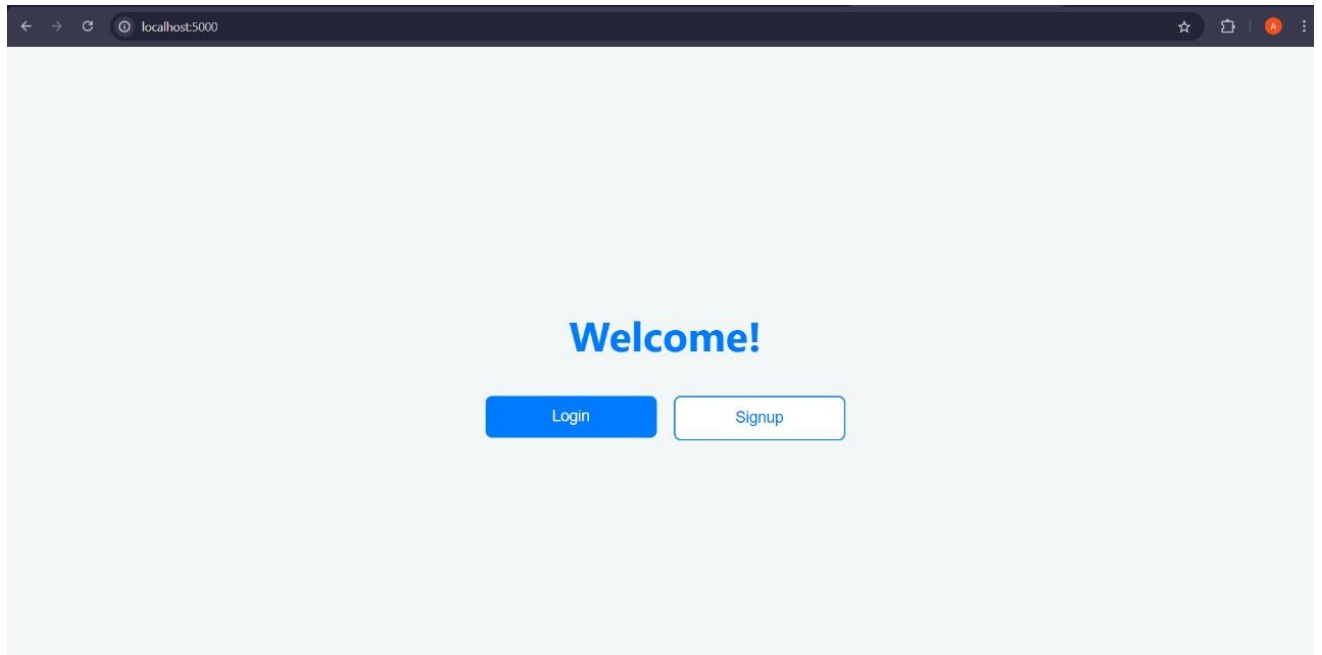
In online multiplayer games, real-time chats often include toxic behavior. Integrating this system with in-game chat logs can help automatically flag users who frequently use abusive language and take corrective action.

Advantages

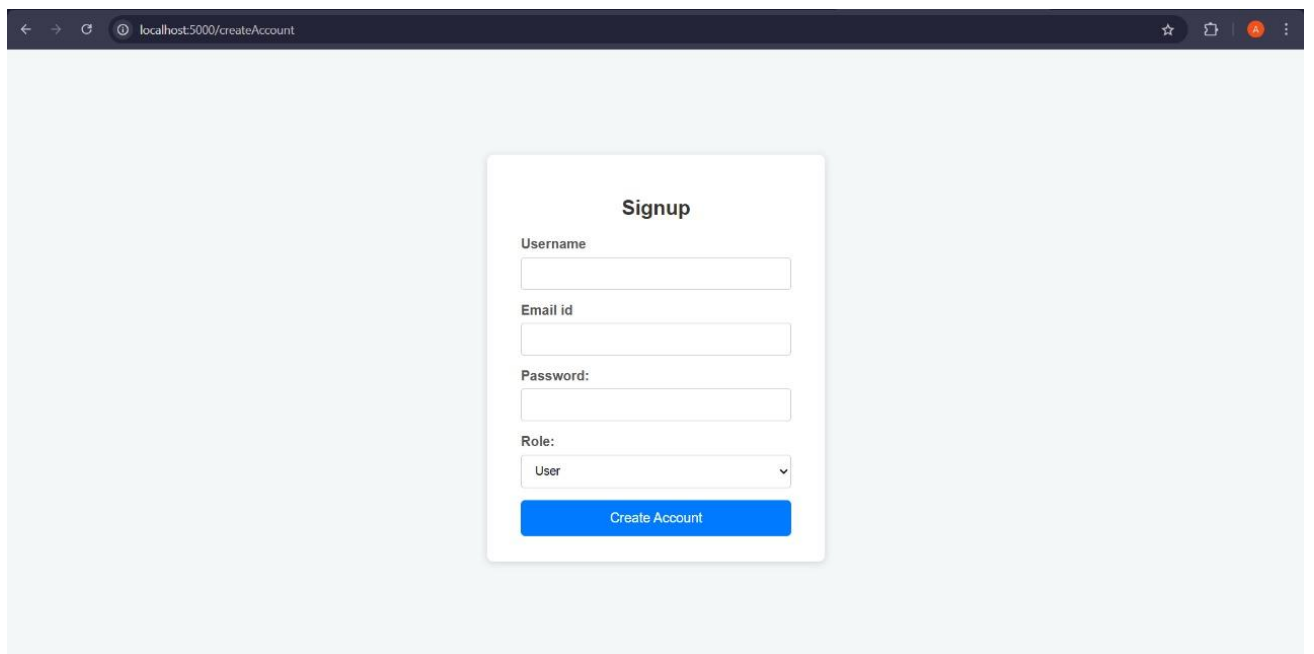
- **Automated, Scalable Moderation:** Reduces the need for constant manual oversight.
- **Context-Aware Analysis:** Uses NLP, not just keyword matching.
- **Adaptive Scoring:** Tracks and adjusts user credibility over time.
- **Database Integration:** Ensures centralized and persistent data management.
- **Admin Control Panel:** Allows human moderation where needed, for accuracy.

Outputs:

1. Home page

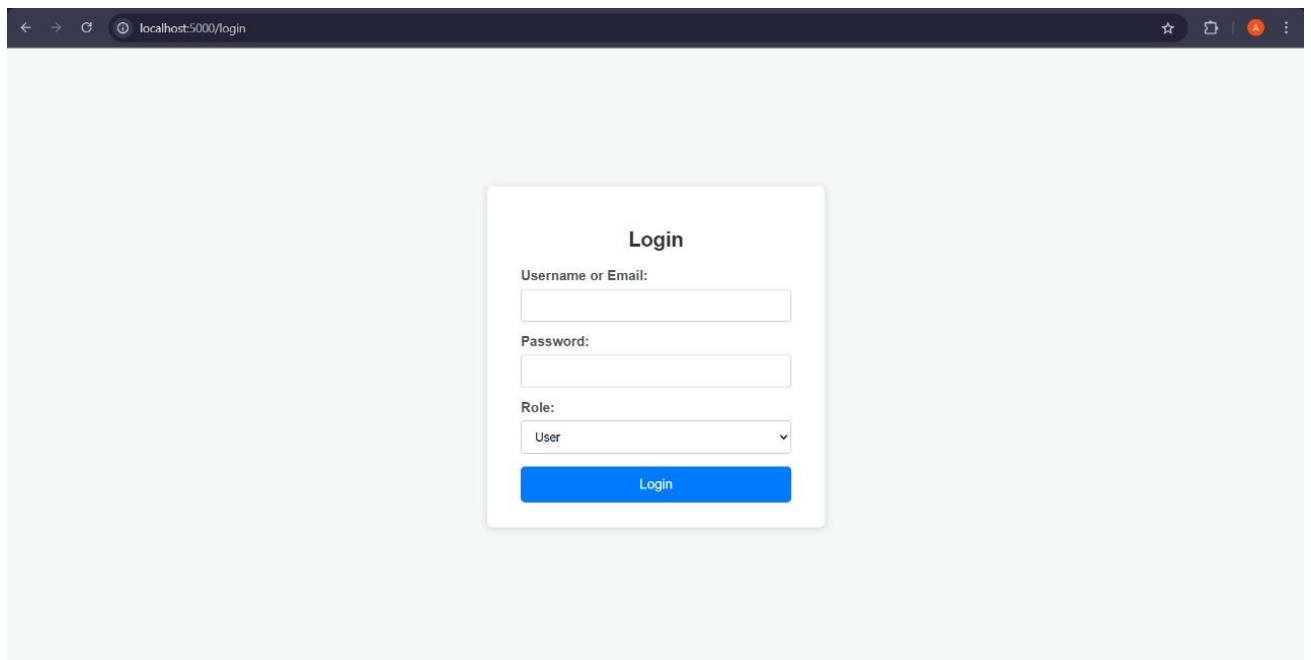


2. Signup page



A screenshot of a web browser displaying the signup page. The browser's address bar shows 'localhost:5000/createAccount'. The page has a light blue background. In the center, there is a white card with a blue shadow. The card has a title 'Signup' in bold. Below the title, there are four input fields: 'Username', 'Email id', 'Password:', and 'Role:'. The 'Role:' field is a dropdown menu with 'User' selected. At the bottom of the card, there is a solid blue button labeled 'Create Account'.

3. Login page



localhost:5000/login

Login

Username or Email:

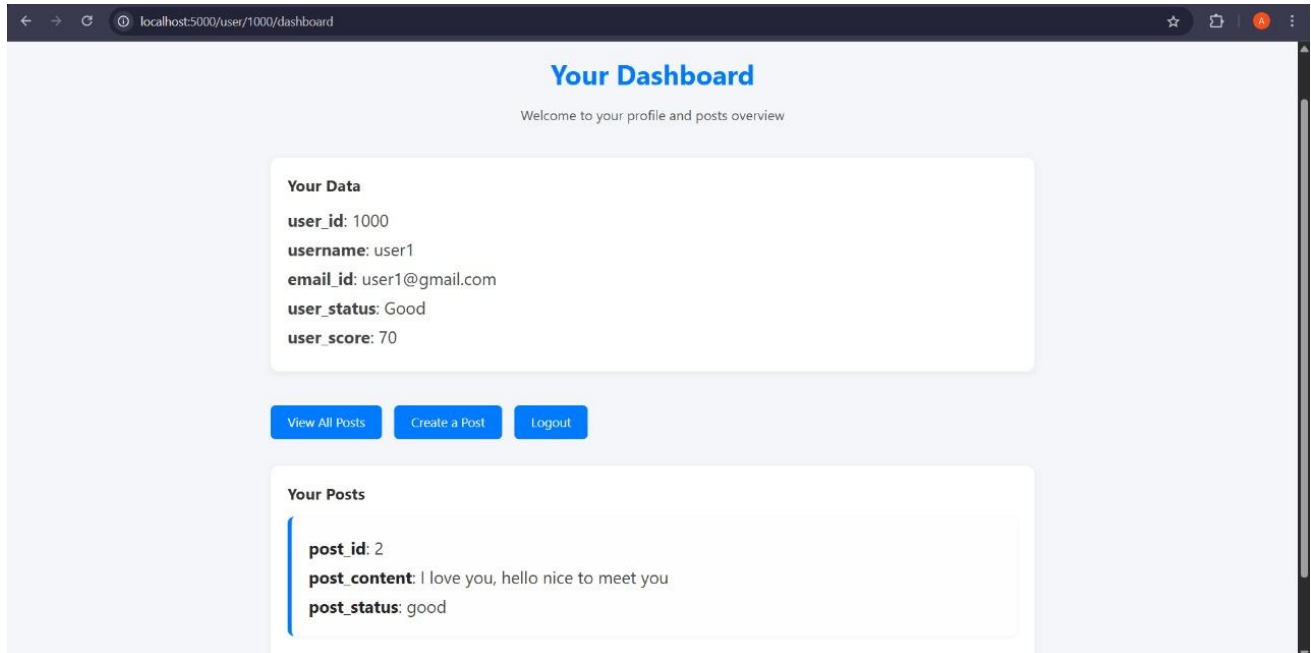
Password:

Role:

User

Login

4. User Dashboard



localhost:5000/user/1000/dashboard

Your Dashboard

Welcome to your profile and posts overview

Your Data

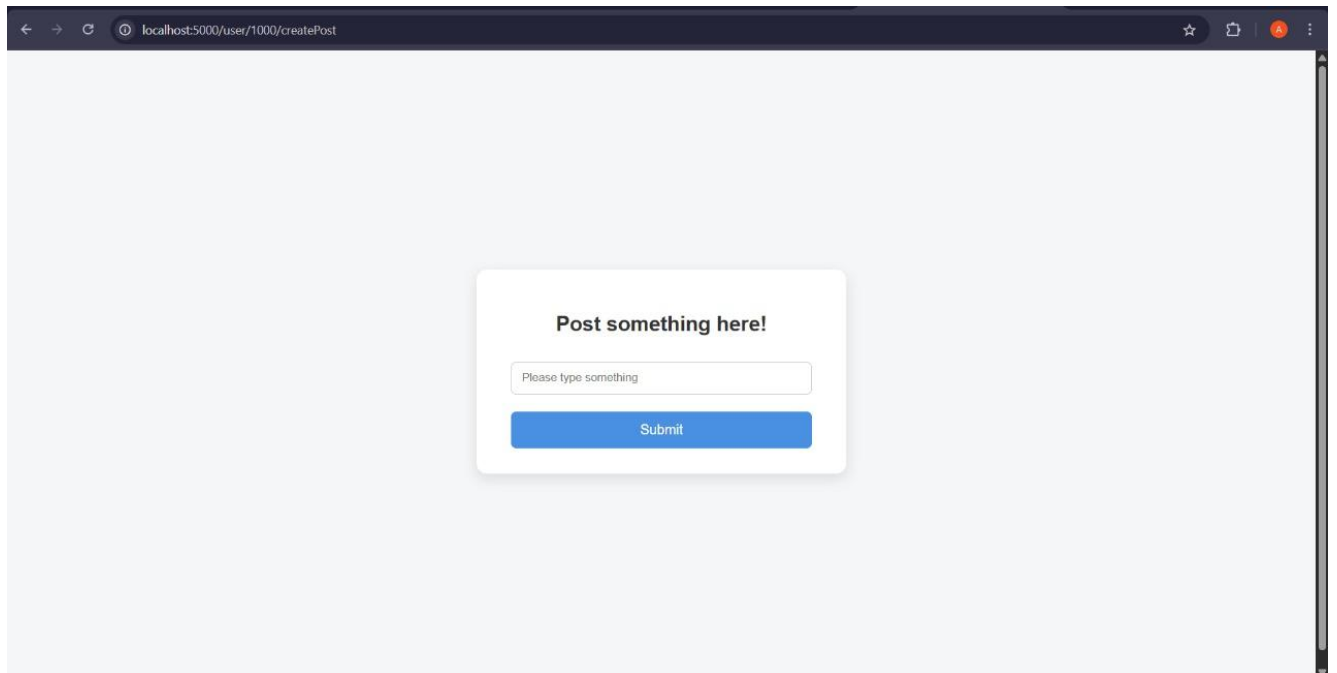
user_id: 1000
username: user1
email_id: user1@gmail.com
user_status: Good
user_score: 70

[View All Posts](#) [Create a Post](#) [Logout](#)

Your Posts

post_id: 2
post_content: I love you, hello nice to meet you
post_status: good

4. Create post



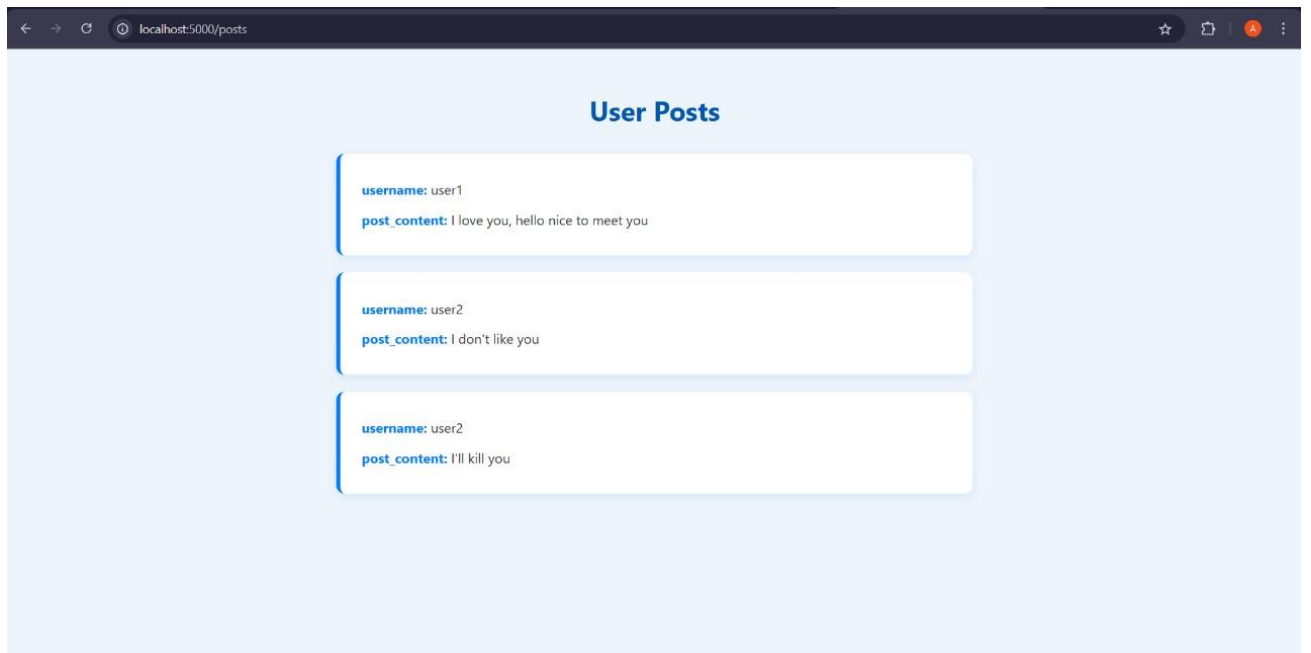
localhost:5000/user/1000/createPost

Post something here!

Please type something

Submit

5. Get All Posts



localhost:5000/posts

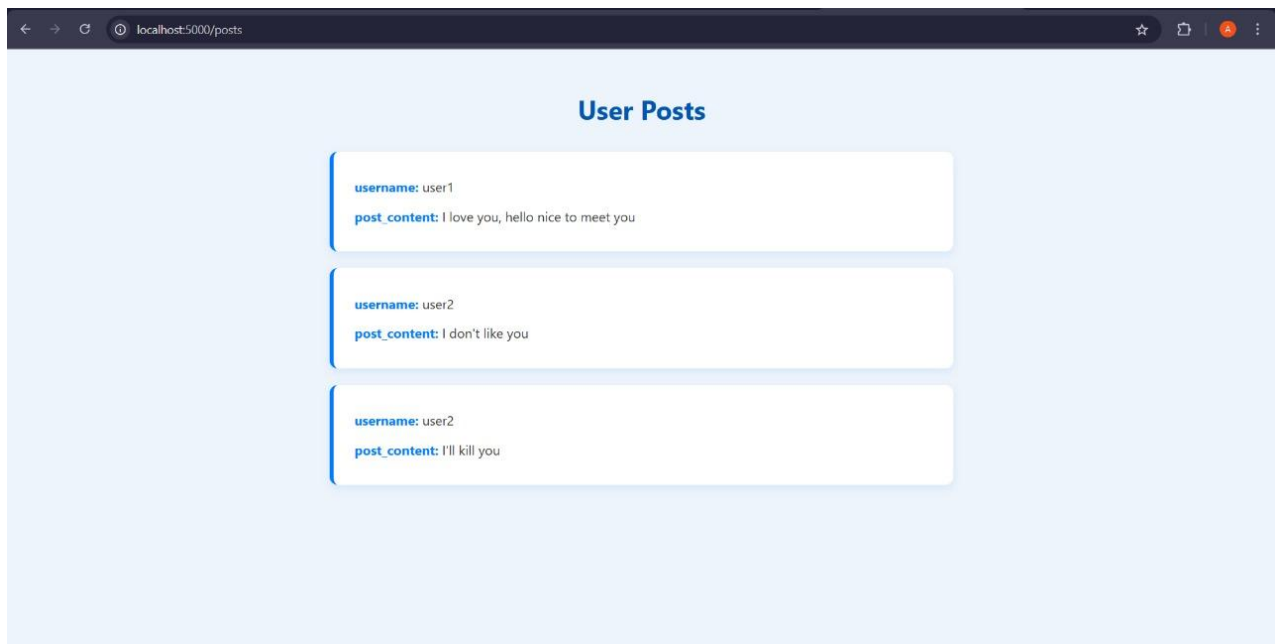
User Posts

username: user1
post_content: I love you, hello nice to meet you

username: user2
post_content: I don't like you

username: user2
post_content: I'll kill you

6. Admin Moderation Page



4. Future Research

Future Scope

1. Add real-time post scanning with Web-Sockets
2. Integrate with BERT or LLMs for deeper context
3. Add JWT-based role login system
4. Dashboard for moderators with charts & stats

Conclusion :

The Suspicious Content Monitoring System developed using Python, Flask, MySQL, and NLTK addresses a critical need for automated and intelligent moderation of user-generated content in digital platforms. By integrating natural language processing with a structured database and rule-based logic, the system effectively analyses textual posts, identifies suspicious or harmful language, and takes appropriate action—either banning, approving, or flagging content for admin review. The user scoring mechanism adds a behavioural layer, encouraging responsible posting and automatically filtering out repeat offenders.

This project demonstrates how lightweight frameworks like Flask, combined with NLP tools and relational databases, can be used to build efficient, real-time moderation systems. It is modular, scalable, and adaptable to various platforms such as social media, forums, educational portals, and corporate feedback systems. Moreover, it lays the groundwork for future enhancements such as machine learning integration, real-time analytics, multilingual support, and a more refined reputation system.

In conclusion, this project offers a practical and technically sound approach to content moderation and serves as a valuable prototype for building safer and more respectful online communities.