SOLUTIONS INFINITY

STACK OVERFLOW CLONE

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS(MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

SANOOP PHILIP
REG No:22PMC148

Marian College Kuttikkanam (Autonomous)
Peermade, Kerala – 685 531

2023

SOLUTIONS INFINITY

STACK OVERFLOW CLONE

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS(MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

SANOOP PHILIP
REG No:22PMC148

Marian College Kuttikkanam (Autonomous)
Peermade, Kerala – 685 531

2023

A PROJECT REPORT ON

STACK OVERFLOW CLONE

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT FOR
THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By
SANOOP PHILIP
REG No:22PMC148

UNDER THE GUIDANCE OF

SR ITALIA JOSEPH MARIA

ASSISTANT PROFESSOR

PG Department Of Computer Applications

Marian College Kuttikkanam (Autonomous)

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikkanam (Autonomous)

Peermade, Kerala – 685 531

2023

PG DEPARTMENT OF COMPUTER APPLICATIONS
Marian College Kuttikkanam Autonomous

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

SOLUTIONS INFINITY

is a bonafide record of work done by

SANOOP PHILIP
REG No:22PMC148

In partial fulfilment of the requirements for the award of Degree of

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2022-2024

SR ITALIA JOSEPH MARIA

Assistant Professor

PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

MR. WIN MATHEW JOHN

Head of the Department

PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

Examiner

Examiner

ACKNOWLEDGEMENT

First of all, I thank the "God Almighty" for His immense grace and blessings in my life and at each stage of my project work.

I express my sincere gratitude to Dr. Ajimon George, Principal, Marian College Kuttikkanam(Autonomous), and Dr. Mendus Jacob, Director, PG Department of Computer Applications for the support given throughout the project work.

I extend my gratitude to Mr. Win Mathew John, HoD, PG Department of Computer Applications, who is a constant source of inspiration and whose advice helped me to complete this project work successfully.

I express my deep sense of gratitude to my project guide, Sr Italia Joseph Maria, Assistant Professor, PG Department of Computer Applications, for his profound guidance for the successful completion of this project work.

With great enthusiasm, I express my gratitude to all the faculty members of the PG Department of Computer Applications for their timely help and support.

Finally, I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project work successfully.

SANOOP PHILIP

ABSTRACT

Solutions Infinity is a web application that aims to replicate the core functionality and features of the popular question-and-answer platform, Stack Overflow. The features include signup, login, ask question, post answer, use tags, users list, users profile, delete questions, delete answers etc. Users can post their queries and also answer other's queries. It also includes search capabilities to enable users to quickly find relevant questions and answers, thereby facilitating efficient knowledge discovery. The Existing System was Developers who relied on mailing lists, forums, and online communities specific to programming languages or technologies to ask questions and get assistance.

This Stack Overflow clone offers an accessible and approachable platform for individuals, students, and professionals seeking knowledge exchange and technical assistance. It provides a valuable resource for both beginners and experienced practitioners to grow their skills, find solutions to complex problems.

TABLE OF CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 PROBLEM STATEMENT

The process of seeking programming assistance and exchanging knowledge within the developer community can be cumbersome and time-consuming. Current methods, such as searching through online forums or consulting colleagues, lack efficiency and flexibility, resulting in delays in finding accurate solutions to coding challenges. Additionally, there is a lack of centralized platforms that effectively manage and organize programming-related information, leading to scattered and fragmented resources. These challenges hinder developers' productivity, limit knowledge sharing, and create barriers to efficient problem-solving in the programming community. Therefore, there is a need for a comprehensive and user-friendly platform, similar to Stack Overflow, that facilitates streamlined knowledge exchange, offers reliable solutions to programming queries, and promotes collaboration among developers.

1.2 PROPOSED SYSTEM

The proposed project aims to develop a Stack Overflow clone, a web-based platform designed to facilitate efficient knowledge exchange and support within the developer community. The platform will provide a user-friendly interface for developers and programmers to ask questions, provide answers, and engage in discussions on various programming topics.

Questions are categorized according to tags, so it gets easy for the seekers to get relevant information. User also gets reputation for every question they asked, solutions they give, and for every upvote they get. By creating a centralized hub for programming-related queries and solutions, the project aims to enhance productivity, foster collaboration, and streamline problem-solving processes.

1.3 FEATURES OF THE PROJECT

- Users can ask new questions and tag them with different tags.
- Users can search for or select a question and view all the solutions related to that question. They can also add comments if they have any doubts.
- Users can upvote solutions, and reputation system is in placed based on upvotes, questions asked, and solutions provided.
- Users can utilize the solutions generated by the GPT-based chat API if they are unable to find a solution from the system.

FUNCTIONAL REQUIREMENTS

2. FUNCTIONAL REQUIREMENTS

- **User Registration and Authentication:** Users can create accounts, login, and authenticate themselves to access the features of the platform. This includes managing user profiles and personal information.
- **Question and Answer System:** Users can ask questions on various topics and provide detailed descriptions. Other users can browse and search for questions, post answers, and engage in discussions. The system support features such as upvoting and comment facility.
- **Voting:** Users can vote on the quality and useful answers.
- **Tagging:** Questions and answers can be organized using tags to improve the searchability and discoverability of content. Users can assign relevant tags to their questions and explore questions based on specific tags.
- **Reputation System:** The system can incorporate a reputation system where users earn points based on their contributions, such as asking questions, providing answers, and receiving upvotes.
- **Notifications:** Users can receive notifications about updates, and responses to their questions.
- **Search Functionality:** The platform should have a robust search feature that allows users to easily find relevant questions and answers based on keywords.
- **User Profile:** Users have profiles showcasing their activity, questions, answers, and reputation. They should be able to customize their profile and add or remove their watched tags.

NON-FUNCTIONAL REQUIREMENTS

3. NON-FUNCTIONAL REQUIREMENTS

RELIABILITY

The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes, also the system will be functioning inside a container. Thus, the overall stability of the system depends on the stability of the container and its underlying operating system.

AVAILABILITY

The system should be always available, meaning the user can access it using a web browser, only restricted by the downtime of the server on which the system runs. A customer-friendly system which is accessible of people around the world should work 24 hours. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, a backup of the database should be retrieved from the server and saved by the Organizer. Then the services will be restarted. It means 24 X 7 availability.

MAINTAINABILITY

A commercial database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the project will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.

SUPPORTABILITY

The code and supporting modules of the system will be well-documented and easy to understand.

FEATURES AND HIGHLIGHTS

4.1 FEATURES

- Register as a user with tags selected.
- User get questions related to those tags, but user can filter questions according to tags they have selected.
- User can ask new questions.
- User can view all solutions related to question.
- User can add comment on questions if they have any doubt.
- User can add solution.
- User can search question.
- User can see all tags in system and see questions related to it.
- User can edit profile.
- Admin can view bar chart visualization of all users.
- Admin can view line graph of questions asked and solution given.

4.2 Highlights

- User get notified if they get solution for their question.
- User can use Solutions GPT, if user doesn’t get solution from system.
- User can watch tags or remove them according to their comfort.
- User get reputation on the basis of upvotes, questions, and solutions added.

TECHNICAL ASPECTS

5.1 ARCHITECTURE OF THE PROJECT

Framework: The project uses Django framework for the project. Django is a high-level Python web framework that follows the Model-View-Controller (MVC) architectural pattern.

Model-View-Controller (MVC) : Django follows the MVC architectural pattern, where models represent the data structures of the application, views handle the logic and data processing, and templates are responsible for rendering the user interface.

Database: Django provides an Object-Relational Mapping (ORM) layer that allows to interact with a database. The code snippets suggest that there is defined models (e.g. Question, Solutions, Notification, Upvotes etc.) that represent the application's data and relationships.

Templates: The code includes Django templates (HTML files) that define the structure and layout of your application's user interface. Templates are used to generate dynamic HTML content by rendering data from views.

Views: Views in Django handle the business logic of the application. They receive requests from the user, interact with the models and database, and render the appropriate templates to generate a response.

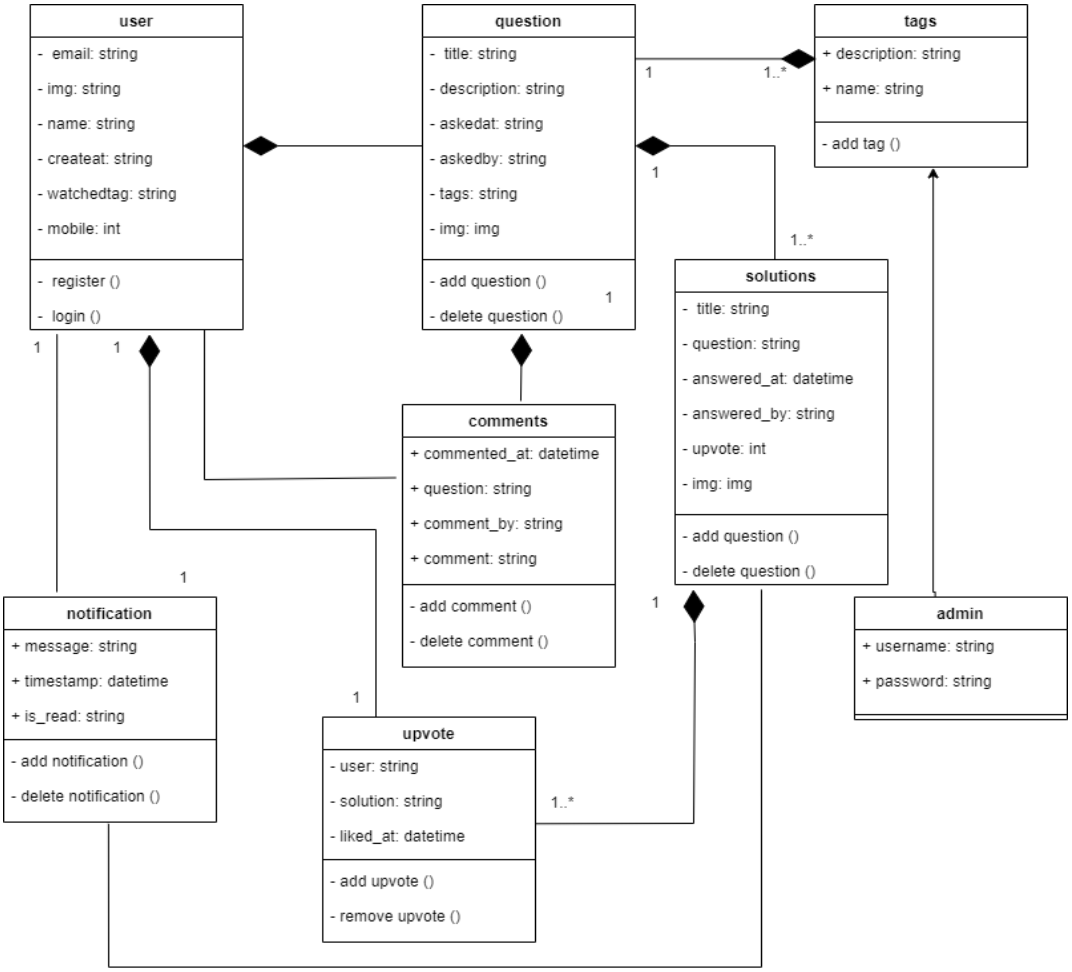
URLs and Routing : There are defined URL patterns using Django's URL routing mechanism. The URLs are mapped to specific views, allowing users to access different pages and functionalities of your application.

Authentication and User Management: Have implemented user authentication using Django's built-in authentication system. Users can register, log in, and access certain features based on their authentication status.

5.2 THIRD PARTY LIBRARIES

- **Django-Jazzmin**: It provides a customizable and modern administration interface for Django applications.
- **Matplotlib**: it provides a wide range of tools for creating various types of plots, charts, and graphs.
- **Open AI** An API for accessing new AI models developed by Open AI.
- **Charts JS**: Chart.js is a popular open-source JavaScript library used for data visualization.

5.3 CLASS DIAGRAM



CHALLENGES

6. CHALLENGES FACED

- It was difficult to implement a notification system.
- Integrating Chat GPT API to my solutions infinity.
- Getting Admin Visualization of Bar chart of users according to their reputation needed some research work.
- It required more time for providing line graph of questions asked and solution.

FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENT

- Chat option between users.
- Edit question and answer option.
- Email Alerts - Receiving email alerts for getting replies and solutions.
- Social Media Integration: Enable social media sharing buttons to allow users to easily share questions, answers, or interesting discussions on popular social platforms.
- Machine Learning and AI Integration: Utilize machine learning algorithms to improve question categorization and recommendation systems.

CONCLUSION

8. CONCLUSION

Solutions Infinity has the potential to foster a vibrant and collaborative community of developers, where knowledge is shared, problems are solved, and valuable connections are created. Constructing a Stack Overflow clone can be a great way to build a healthy community of knowledge-sharing and problem-solving. You can create a platform where users can ask questions, receive answers, and engage in discussions about programming, software development, and other technical issues by reproducing the basic operations and features of Stack Overflow.

Github repository of the project:
https://github.com/Sanoop369/Solutions-Infinity-Stack-Overflow-Clone-Django/tree/master

REFERENCE

9. REFERENCE

Templates- https://github.com/RugvedB/Stackoverflow-Clone/tree/master/stackoverflow/main

And https://mdbootstrap.com/

Chat Gpt- https://chat.openai.com/

Stack Overflow - https://stackoverflow.com/

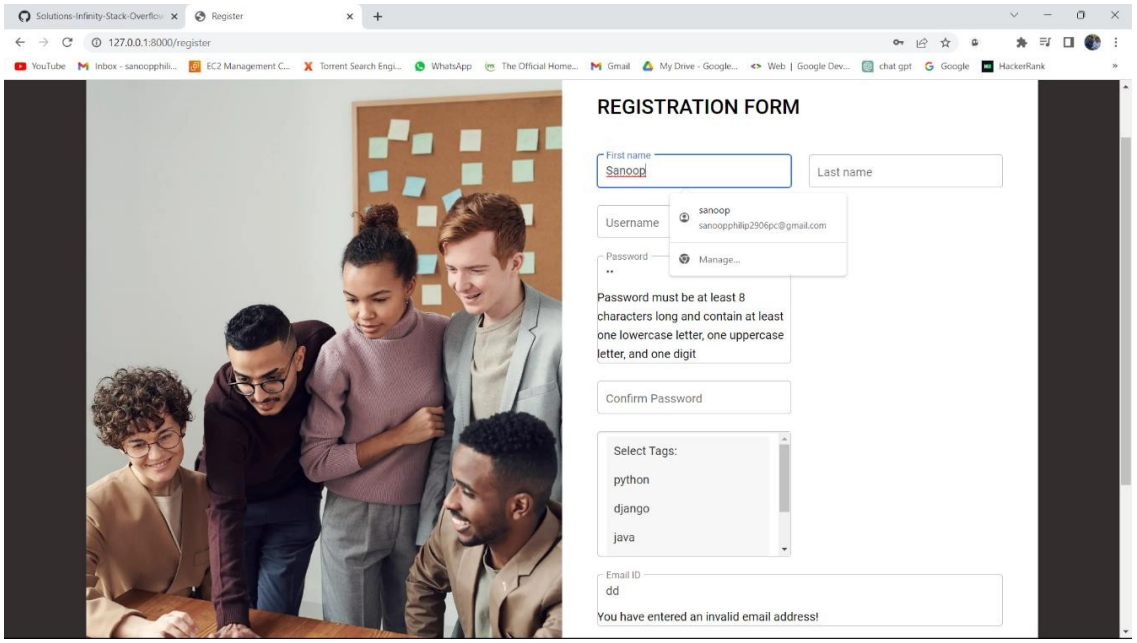
Jazzmin Documentation - https://django-jazzmin.readthedocs.io/configuration/

Chat GPT API documentation - https://platform.openai.com/docs/api-reference/introduction

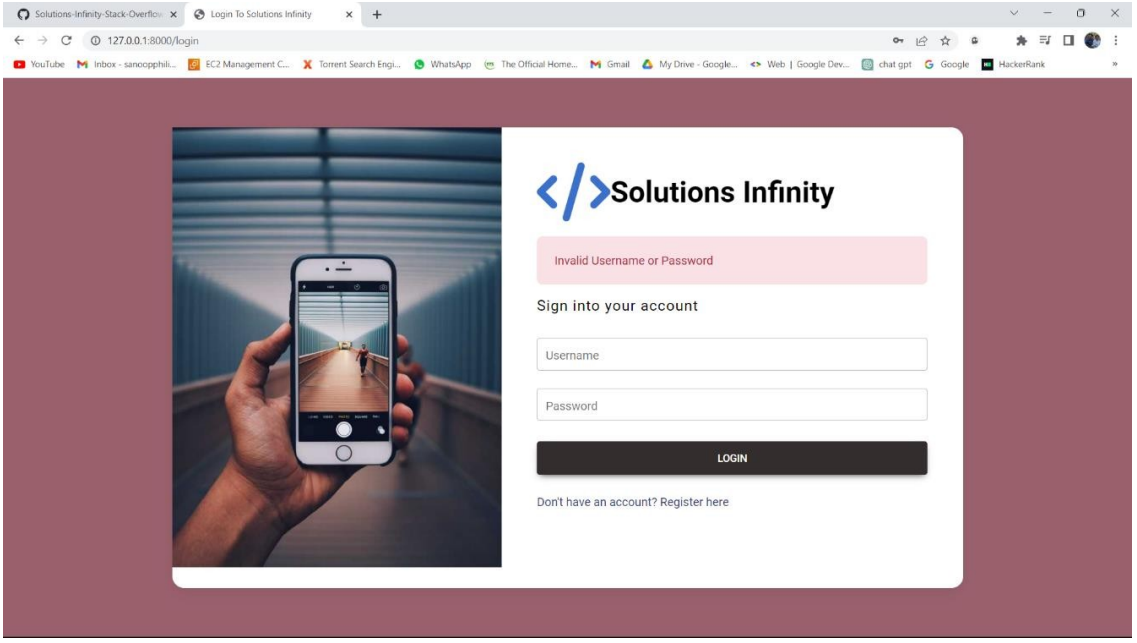
Chartjs - https://www.chartjs.org/docs/latest/charts/bar.html

ANNEXURE

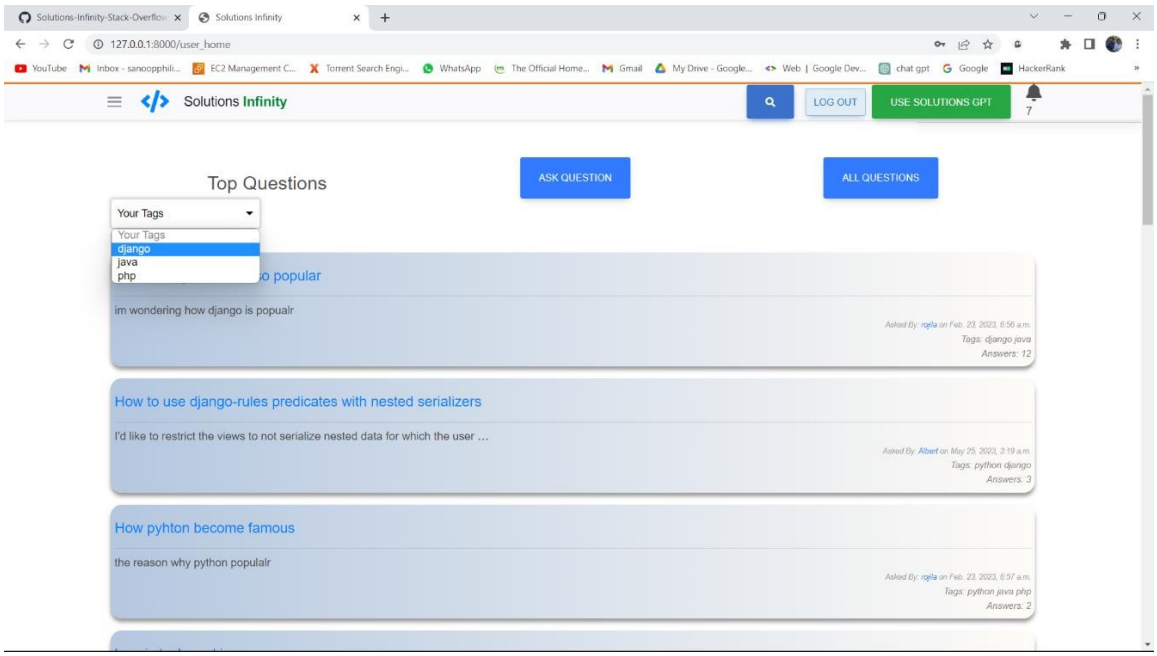
10.1 REGISTRATION



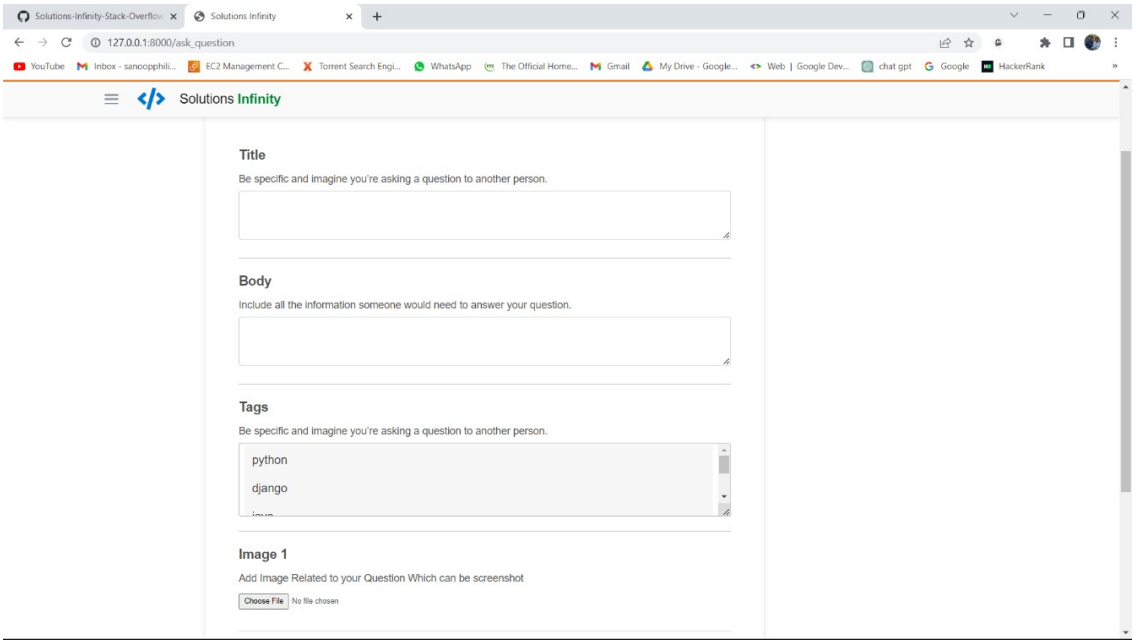
10.2 LOGIN



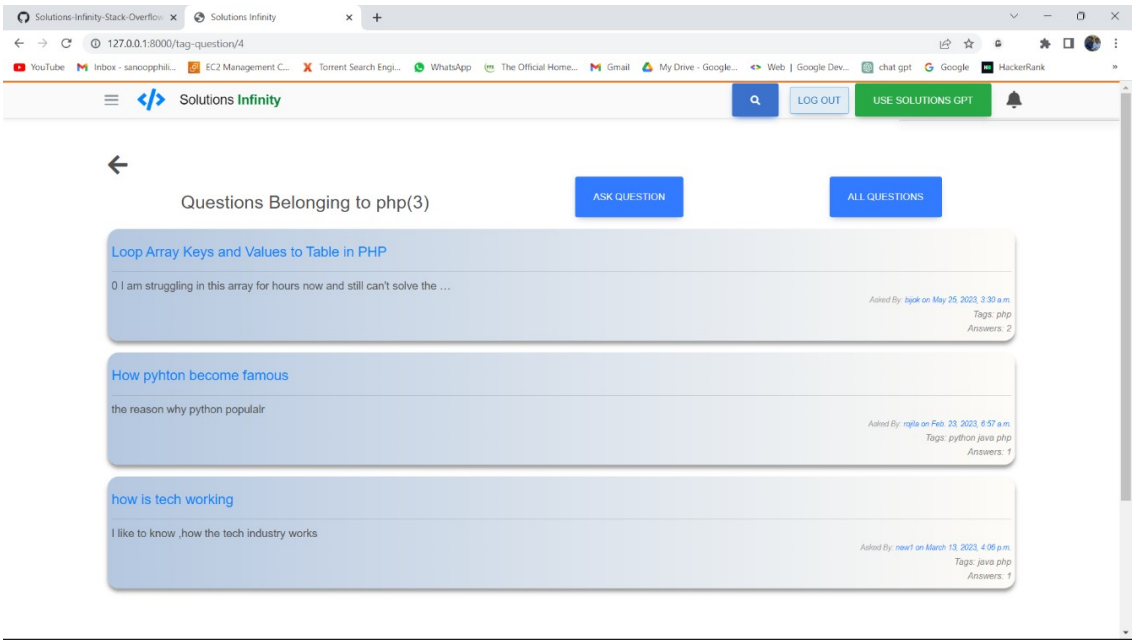
10.3 USER HOME PAGE



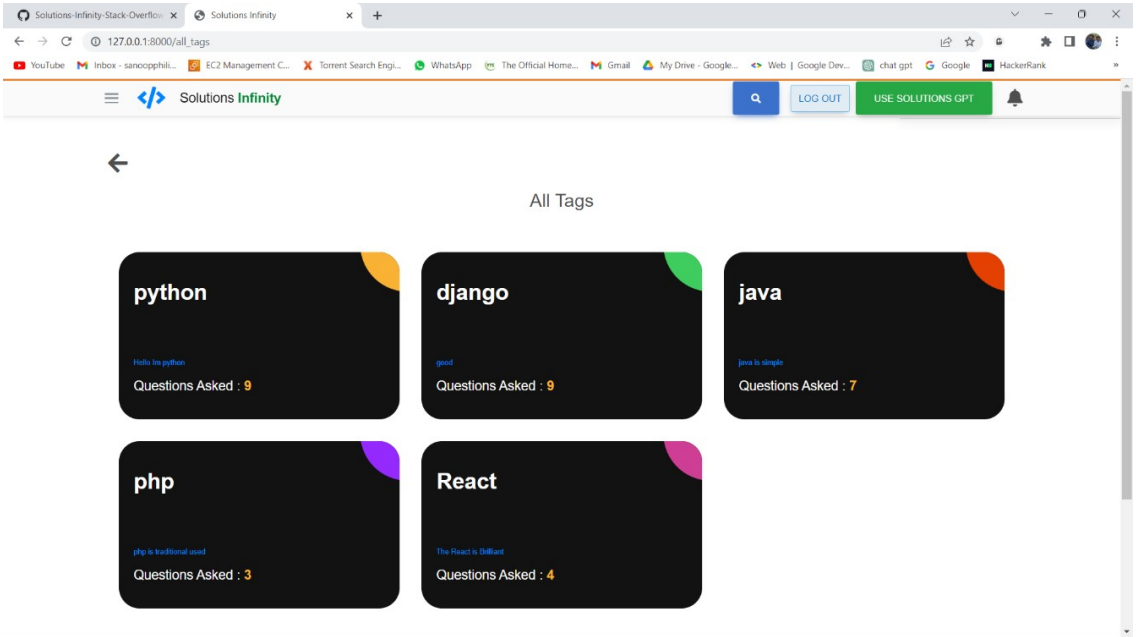
10.4 ASK QUESTION PAGE



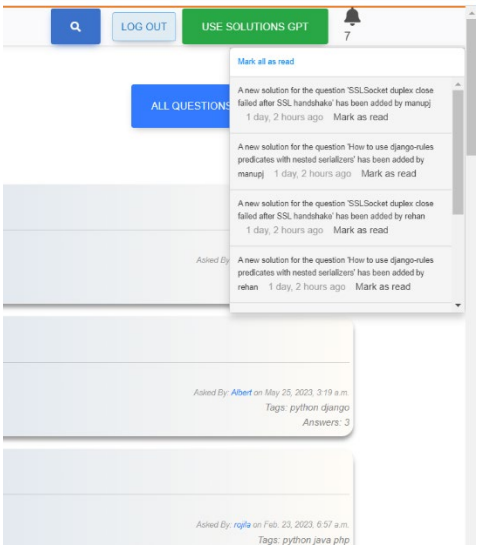
10.5 TAG - QUESTION PAGE



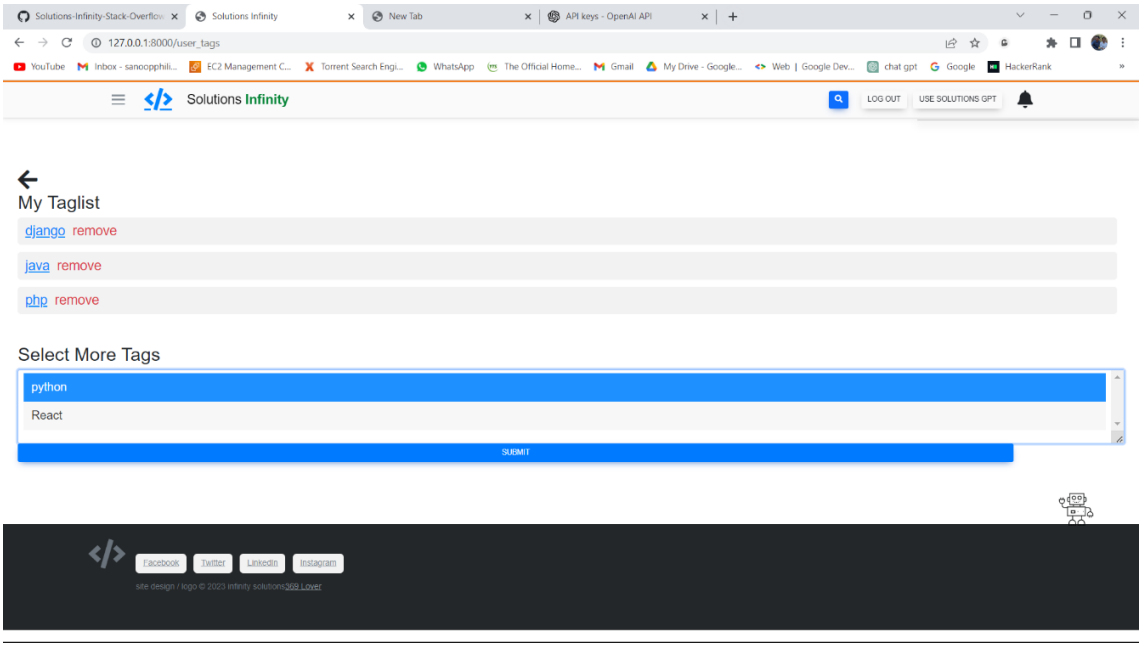
10.6 ALL TAG PAGE



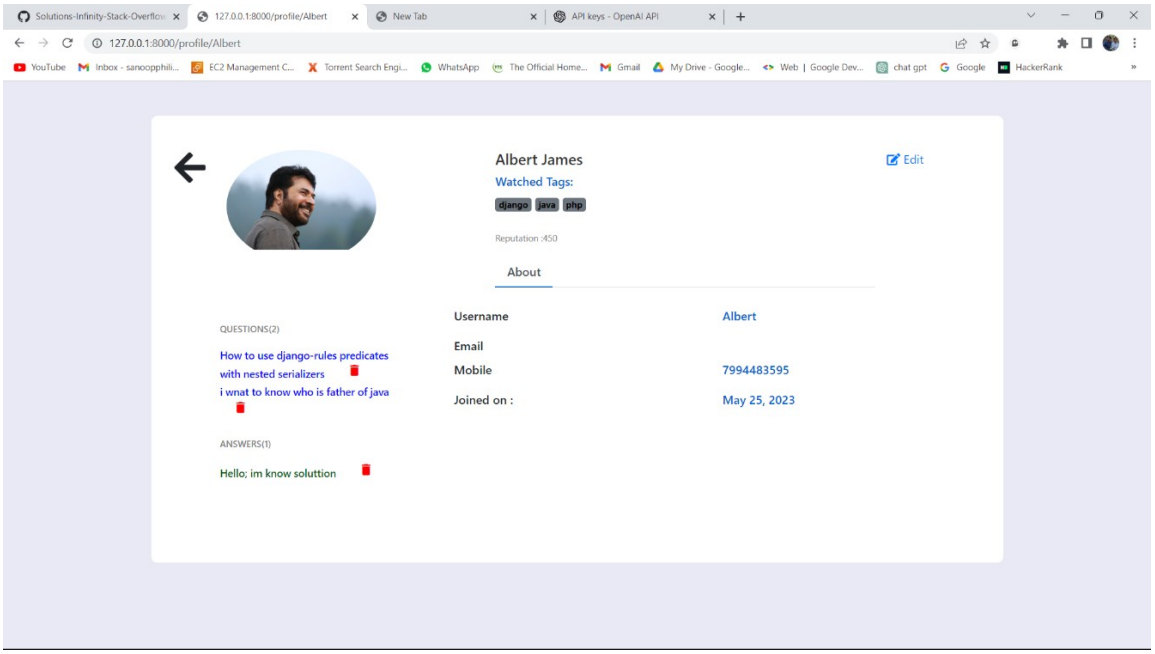
10.7 NOTIFICATION



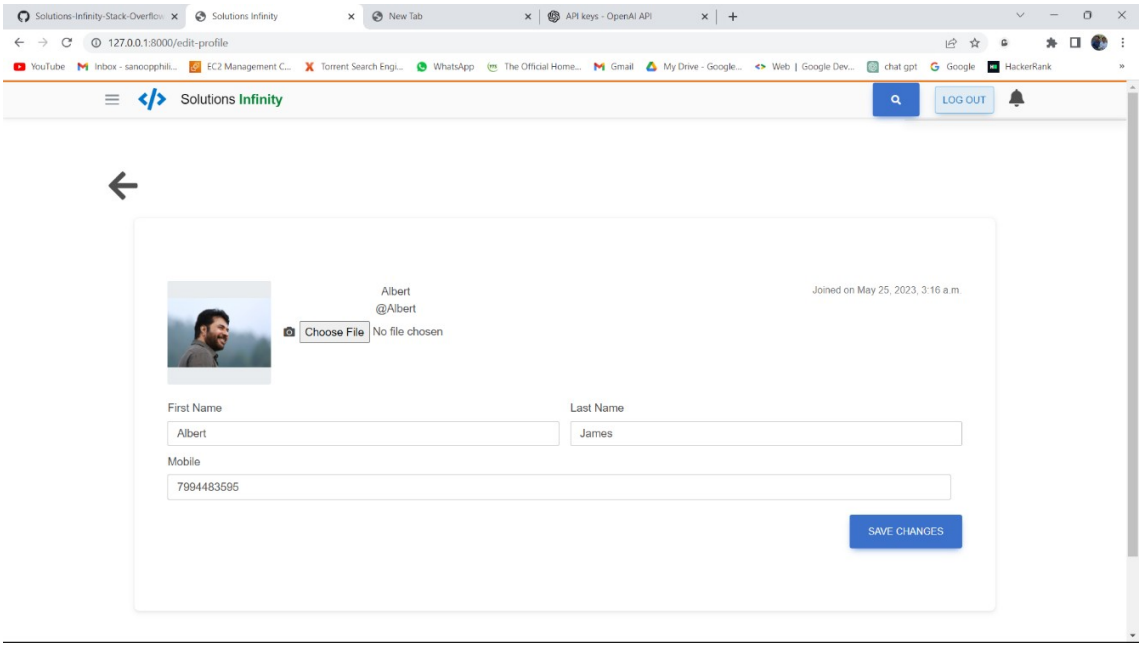
10.8 ADD/REMOVE TAG PAGE



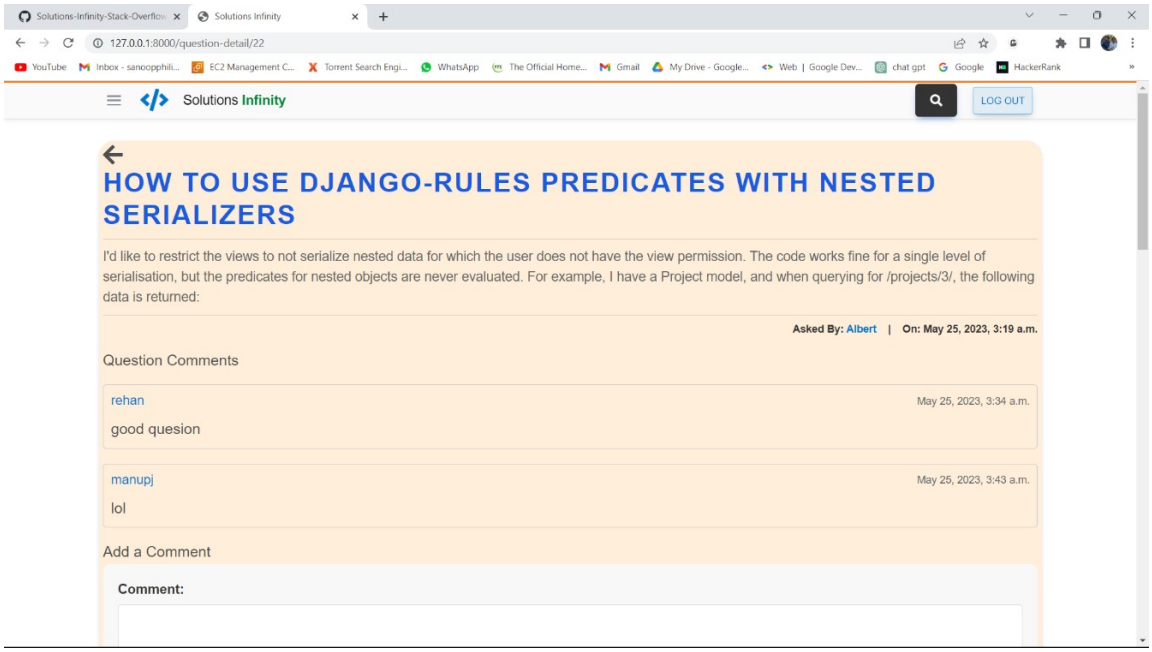
10.9 PROFILE VIEW PAGE

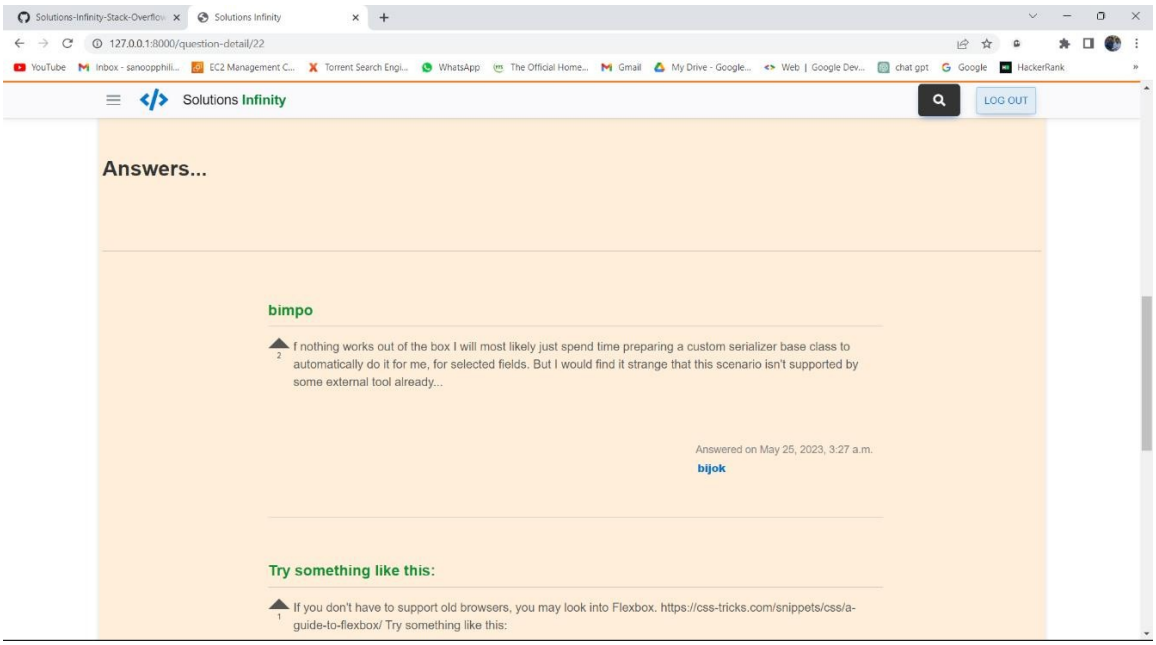


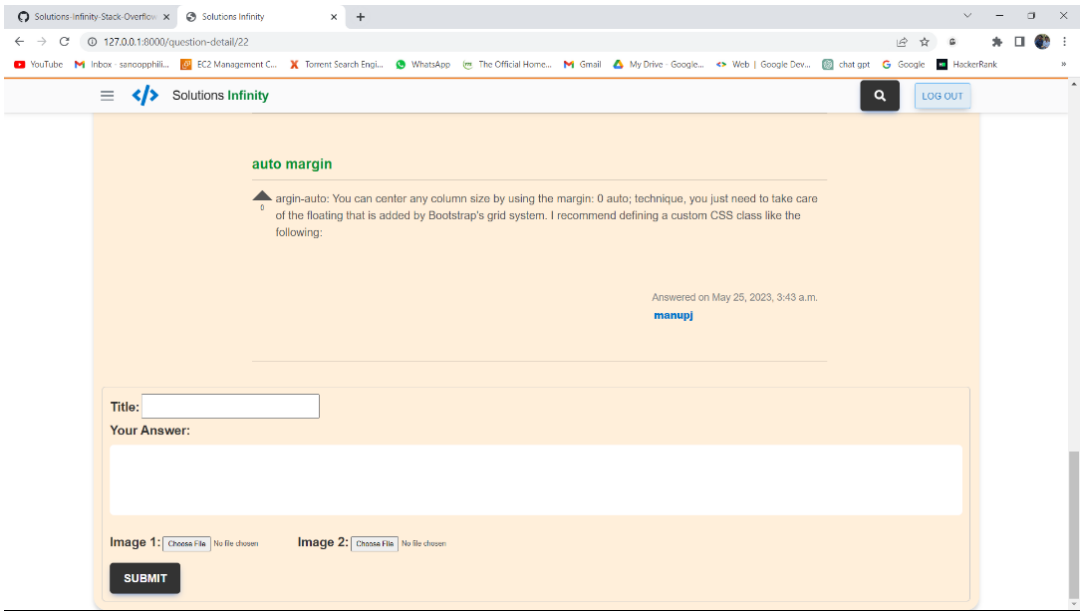
10.10 PROFILE EDIT PAGE



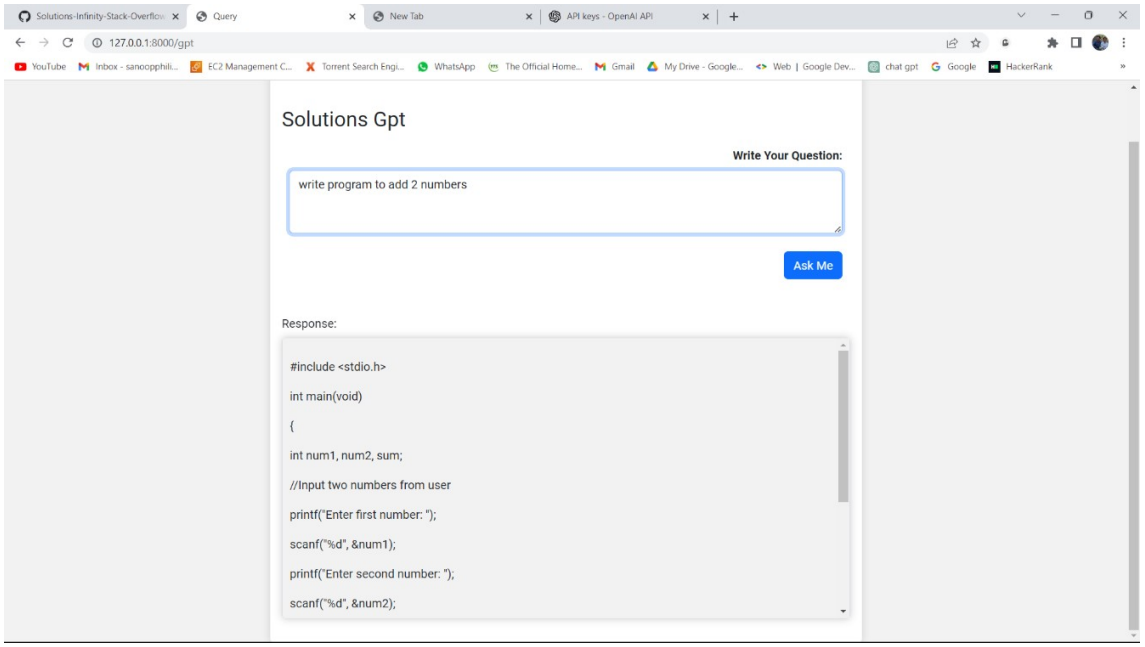
10.11 QUESTION DETAIL PAGE







10.12 SOLUTIONS GPT PAGE



10.13 ADMIN VISULAIZATION ABOUT REPUTATION OF USERS

10.14 ADMIN VISUALIZATION OF SOLUTIONS INFINITY PERFORMANCE