**Online Survey Platform**

**Group 9**

**Course: Advanced Database System Design**

**Instructor: Neha**

**Team Members:**

* Member 1 - Priyanka Jonnala
* Member 2 - Venkata Naga Sai Goud Perika

Github link: <https://github.com/saigowd/finalProject/>

**Abstract**

The Online Survey Platform is aimed at devising an effective, user-friendly, easily maintainable application for creating, sharing, and analyzing a survey. This tool would bridge the chasm between such fully featured survey tools and create a solution which, though simple but powerful, meets most user requirements. This web platform used Flask for backend building because of its lightweight architecture, which ensures modularity in rapid development with the assurance of clean structuring of codes. It uses Peewee ORM, which allows for easy, fluid interaction with SQLite during development and PostgreSQL for production.

The platform enables the user to create custom-built surveys, which can be shared with others by unique links for that survey, and allow the respondent to receive feedback in real time. It provides data transformation through dynamic and interactive visualization of data using Chart.js. It takes security and data integrity seriously by implementing different measures: user authentication, role-based access control, and protection against cross-site request forgery.

This project is designed to support academic research, business intelligence, and market analysis by providing tools to gather and interpret data in an efficient manner. Real-time responses with immediate visual insights will enhance decision-making processes and simplify often complex data interpretation. The codebase of the platform is modular and scalable; thus, adding new features in the future, such as multi-language support, advanced analytics like sentiment analysis, or integration with third-party platforms, can be done seamlessly. With its solution of the Online Survey Platform, the demands of both beginners and advanced researchers are met for data gathering and analysis.

**Table of Contents**

1. Introduction
2. Methodology
3. Implementation Details
4. Results and Discussions
5. Challenges and Solutions
6. Conclusion and Future Work
7. Team Member Contributions
8. References

**Introduction**

**Background**

Surveys are indispensable in data gathering, research, and analysis in every field. Whether it is academia or small business ventures, surveying plays a major role in knowing customer behavior, and satisfaction, or even proving hypotheses. Despite all the usefulness of these questionnaires, there exist problems in the currently available survey tools. These platforms are often expensive, require technical expertise to use, or have overwhelmingly large feature sets that make using them by individuals and organizations with poor technical resources challenging.

The conception of this project came about because of a perceived gap in the market for an easy, affordable, and feature-rich survey platform. This platform, therefore, puts much emphasis on usability and functionality to provide a service that is accessible in creating, distributing, and analyzing surveys. It shall be a solution designed for small businesses, students, researchers, and educators who want an easy-to-use tool to collect and analyze data.

**Problem Statement**

While there are many survey tools, most of them fail to find a perfect balance between ease of use and feature set. High-end services come with high subscription prices, which are unaffordable for small-scale users. In contrast, free or low-budget options are often limited in scope and do not offer advanced functionalities like dynamic data visualization or the tracking of responses in real time. Moreover, these tools often have steep learning curves, requiring users to spend time and resources understanding their functionality before being able to use them effectively.

Small businesses, students, and researchers often need simple means of creating questionnaires, gathering responses, and visualizing results without requiring deep technical knowledge or training. The absence of a platform that combines affordability, simplicity, and advanced analytics leaves a significant gap that this project aims to fill.

**Objectives**

The Online Survey Platform is targeted to achieve the following objectives:

1. Intuitive Creation and Management of Surveys: Design a platform where users can create, edit, and manage their surveys intuitively with the help of a clean and user-friendly interface.

2. Secure User Authentication: Provide secure login and registration mechanisms for data privacy and role-based access control.

3. Dynamic Data Visualization: Include real-time response tracking and visualization through interactive charts and graphs that transform raw data into actionable insights.

4. Scalability and Flexibility: Design the architecture to be modular and scalable, considering future enhancements that might include sentiment analysis, support for multiple languages, and integrations with third-party platforms.

5. Cost-Effective Solution: Offer a cost-effective solution compared to existing platforms while not sacrificing key functionalities.

6. Accessibility: Ensure the platform is user-friendly and accessible even to those users who have minimal technical background for wider applicability across various domains.

In attaining these objectives, the platform will be able to empower small businesses, educators, students, and researchers with prior preparedness for decision-making and interpretation of survey data.

**Methodology**

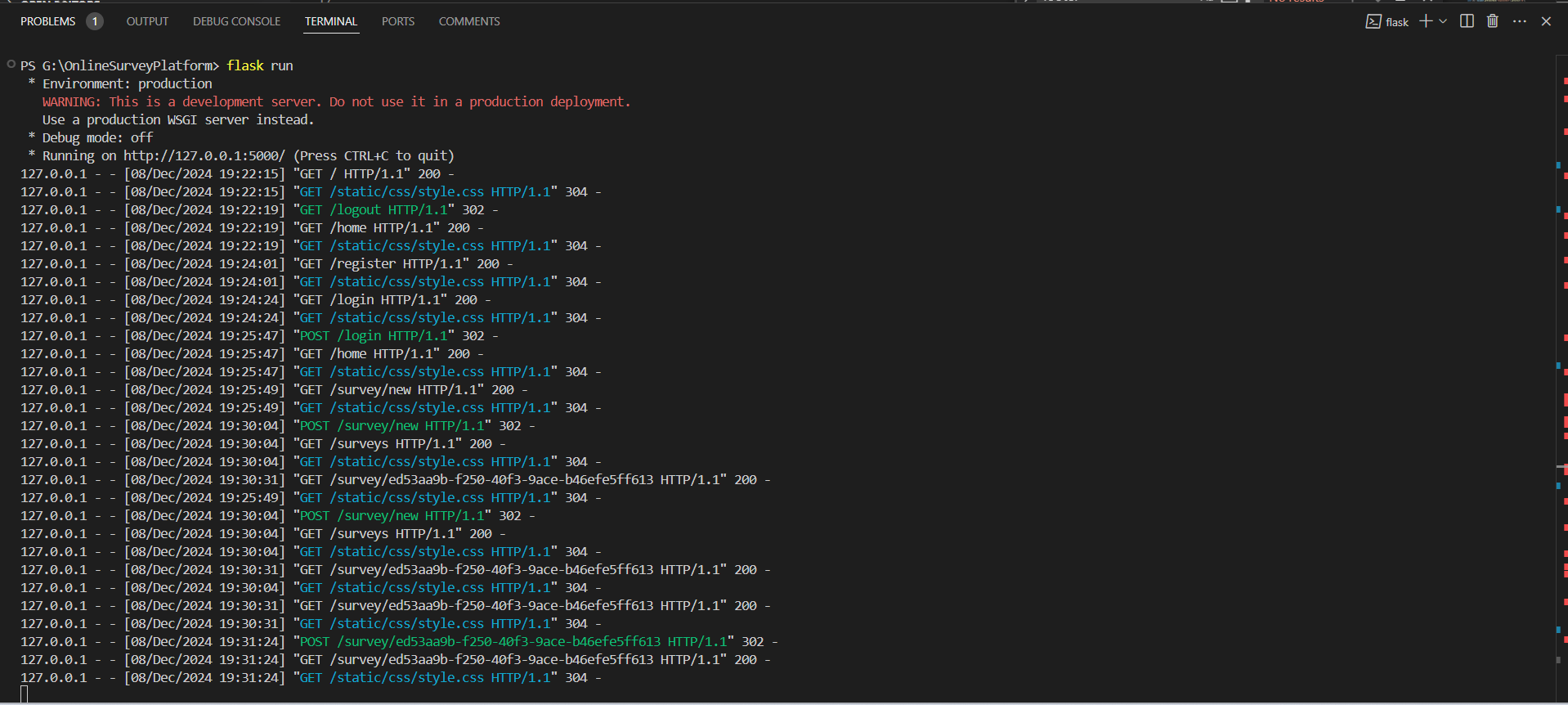
**Technologies Used**

1. **Backend Framework:** Flask  
   Flask’s lightweight and modular architecture supports rapid development and clean code structuring.
2. **Database Management:** Peewee ORM  
   Peewee ORM simplifies interactions between Python and SQL databases, offering compatibility with SQLite for development and PostgreSQL for production.
3. **Frontend Technologies:** HTML, CSS, JavaScript  
   These technologies create responsive and interactive user interfaces.
4. **Visualization Library:** Chart.js  
   Chart.js generates dynamic and interactive graphs to visualize survey responses effectively.

**Workflow**

1. **User Authentication and Management:**  
   Users can register, log in, and manage their accounts securely using Flask-WTF and Flask-Login.
2. **Survey Creation and Listing:**  
   Users can create surveys, which are stored in the database and displayed as lists on their dashboard.
3. **Response Collection:**  
   Surveys generate unique links for users to distribute. Responses are collected and stored dynamically.
4. **Data Visualization:**  
   Interactive bar charts summarize responses, providing visual insights.

**EXECUTION:**



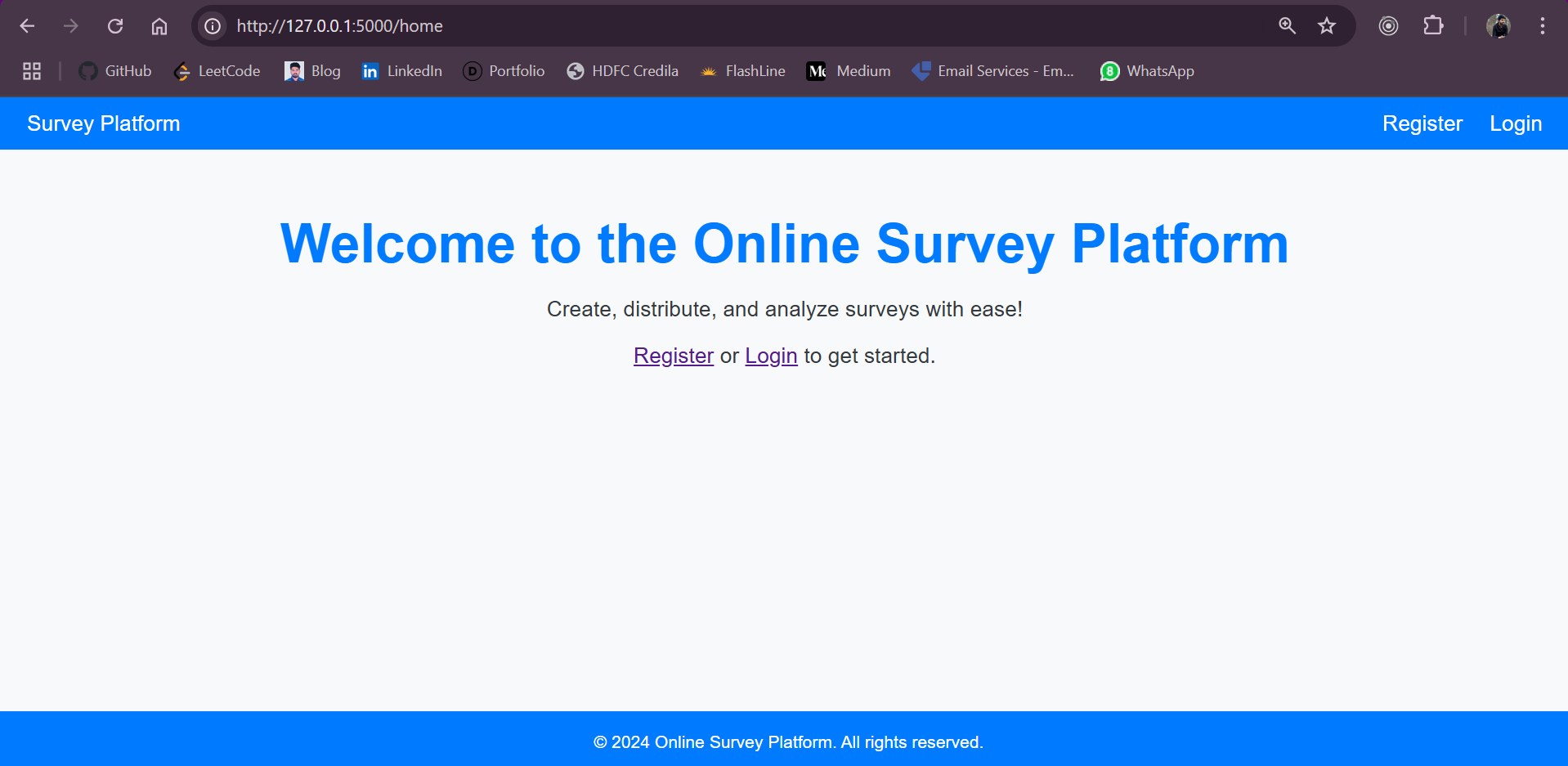
**Implementation Details**

**1. Authentication System**

**Features:**

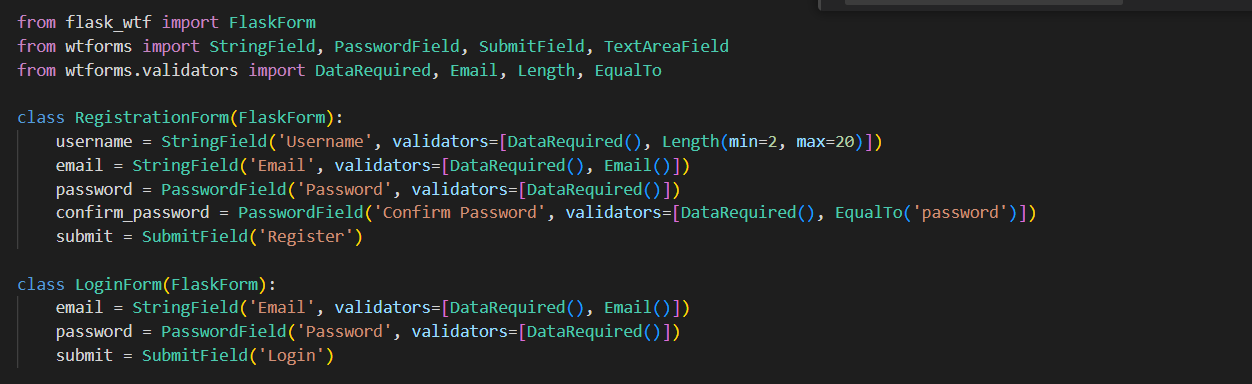
* Users register with a unique username and email.

HomePage-without logging in



* Login functionality verifies credentials securely

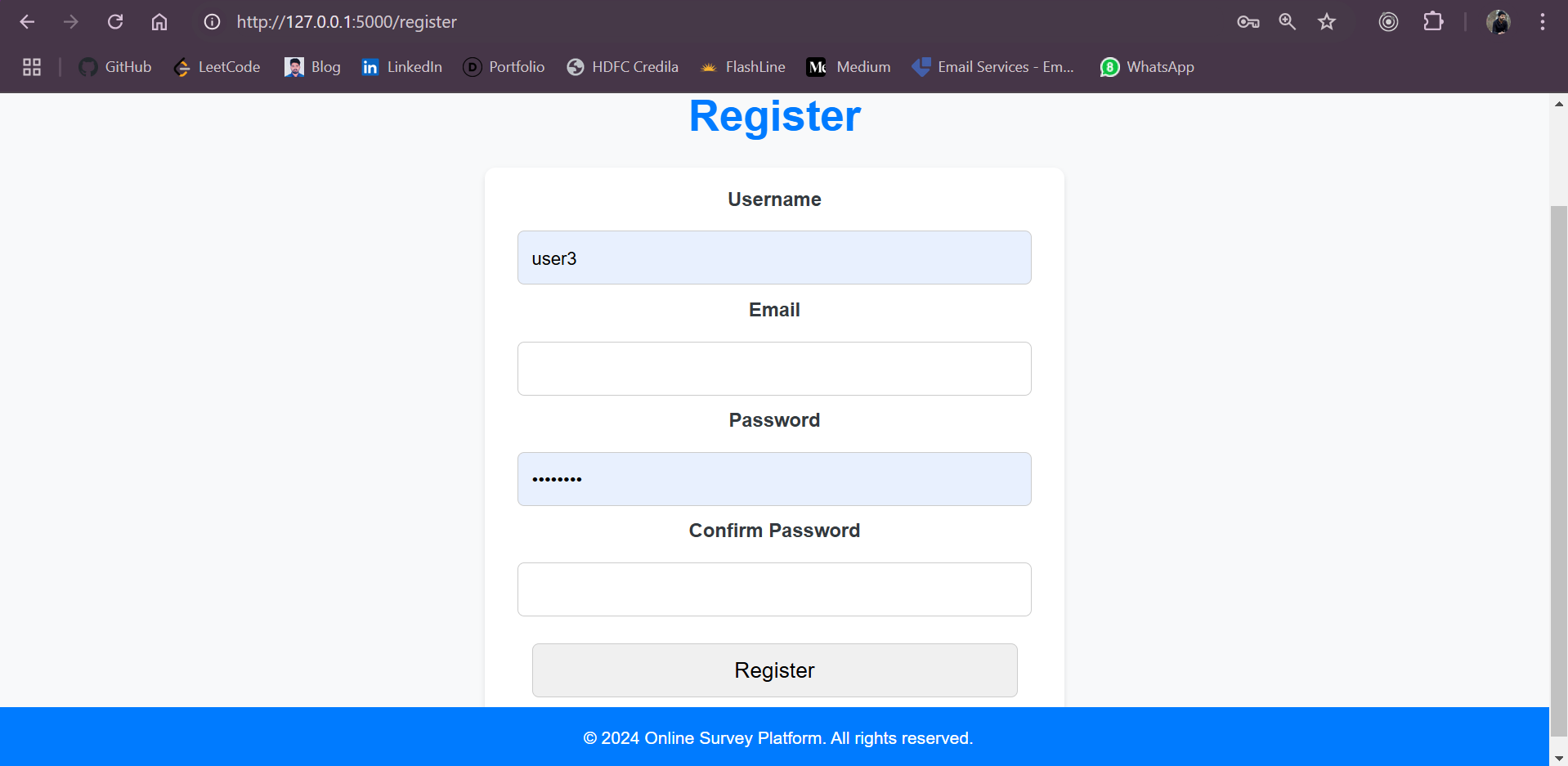
**Code:**

****

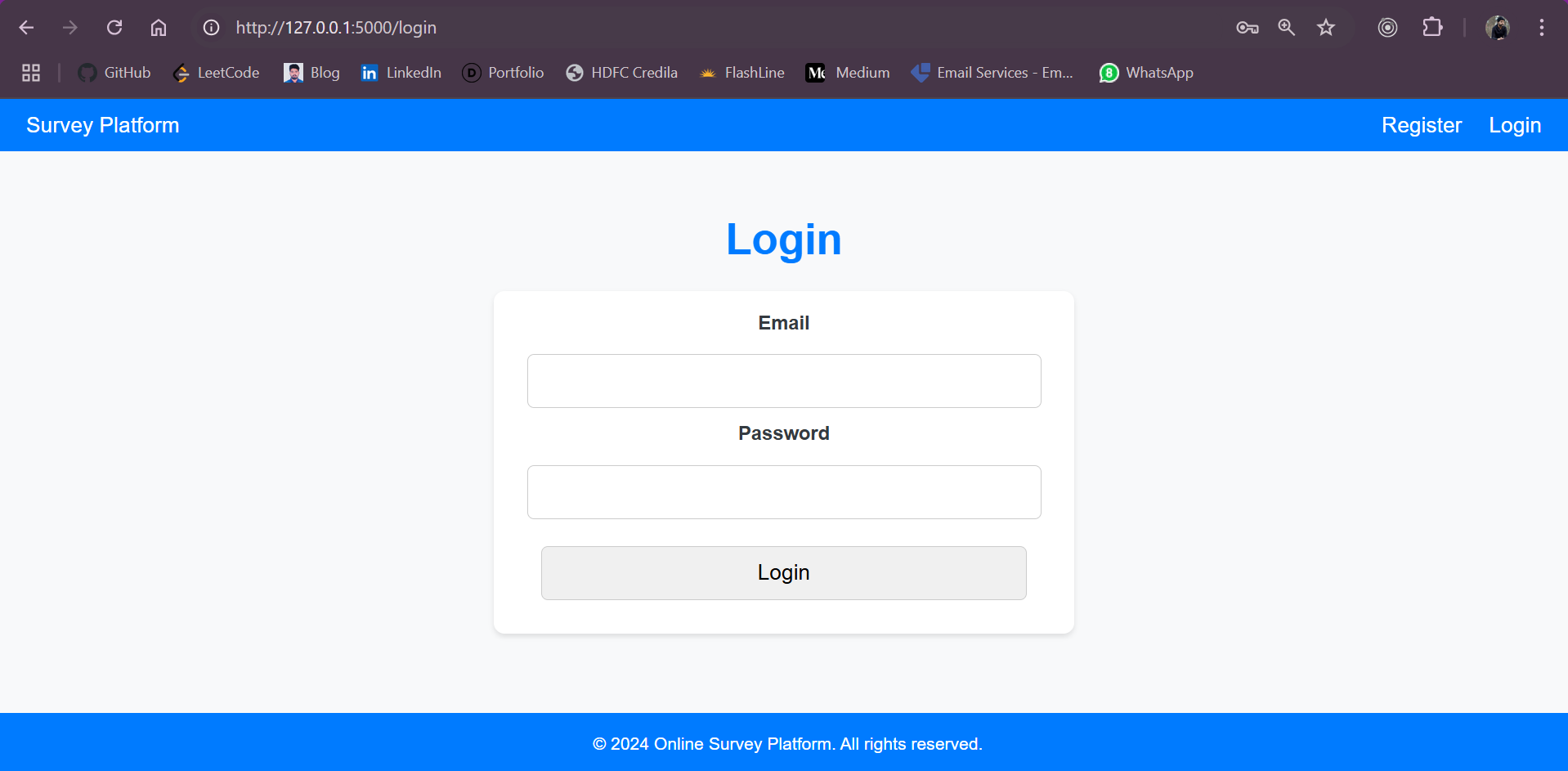
The User model handles user data, while routes for registration and login use Flask-WTF forms for input validation.

**Screenshots:**

* Registration Page



* Login Page

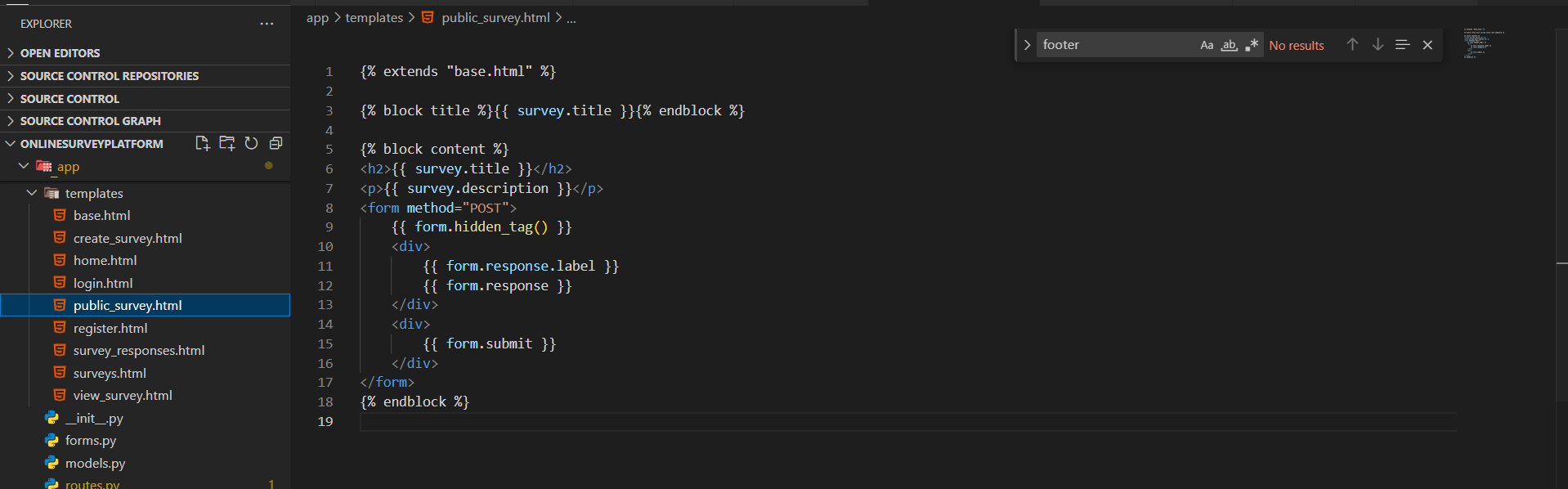


**2. Survey Management**

**Features:**

* Surveys have a title, description, and unique link for distribution.
* Surveys are listed in a user-friendly dashboard.

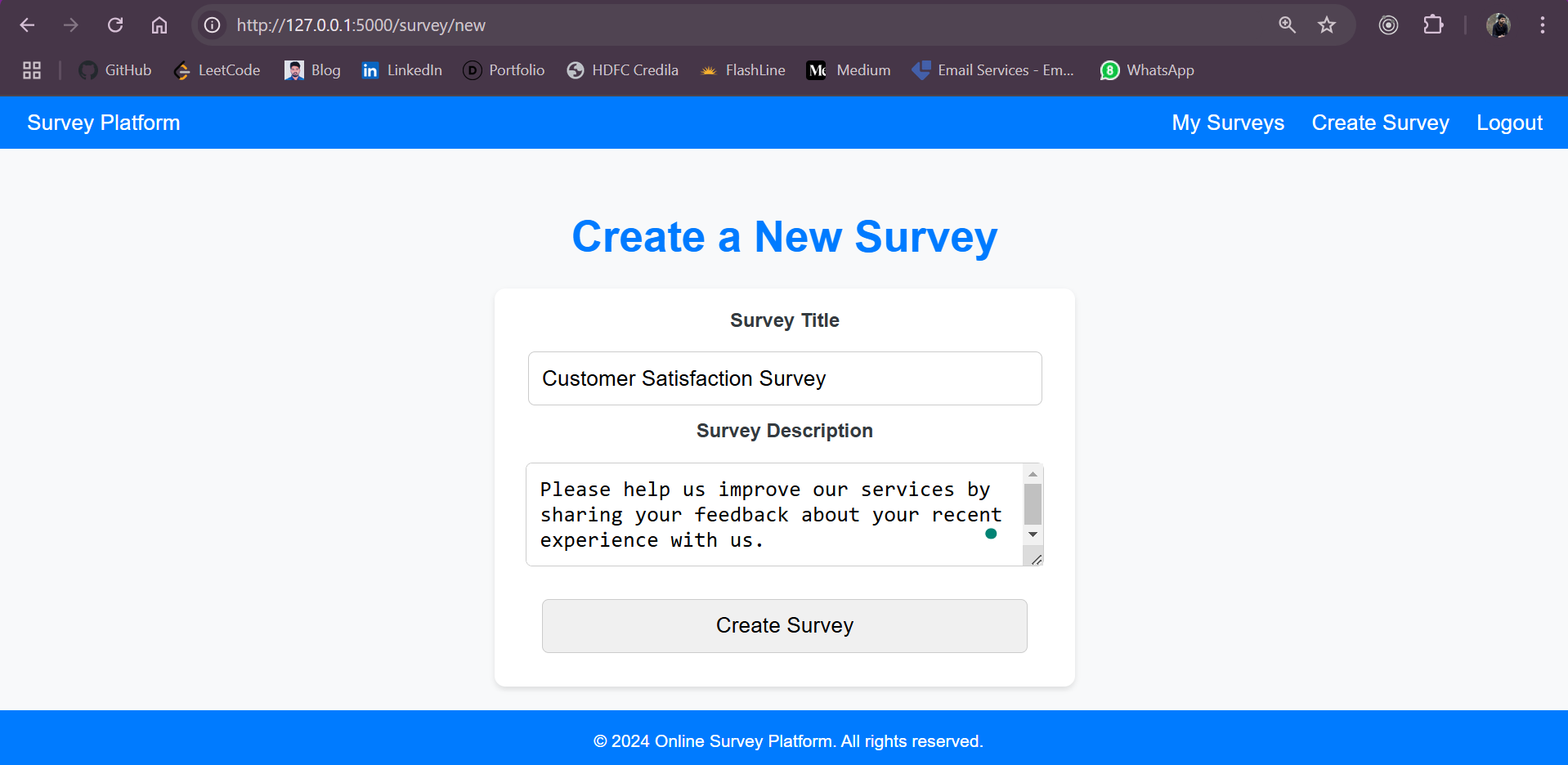
**Code:**

****

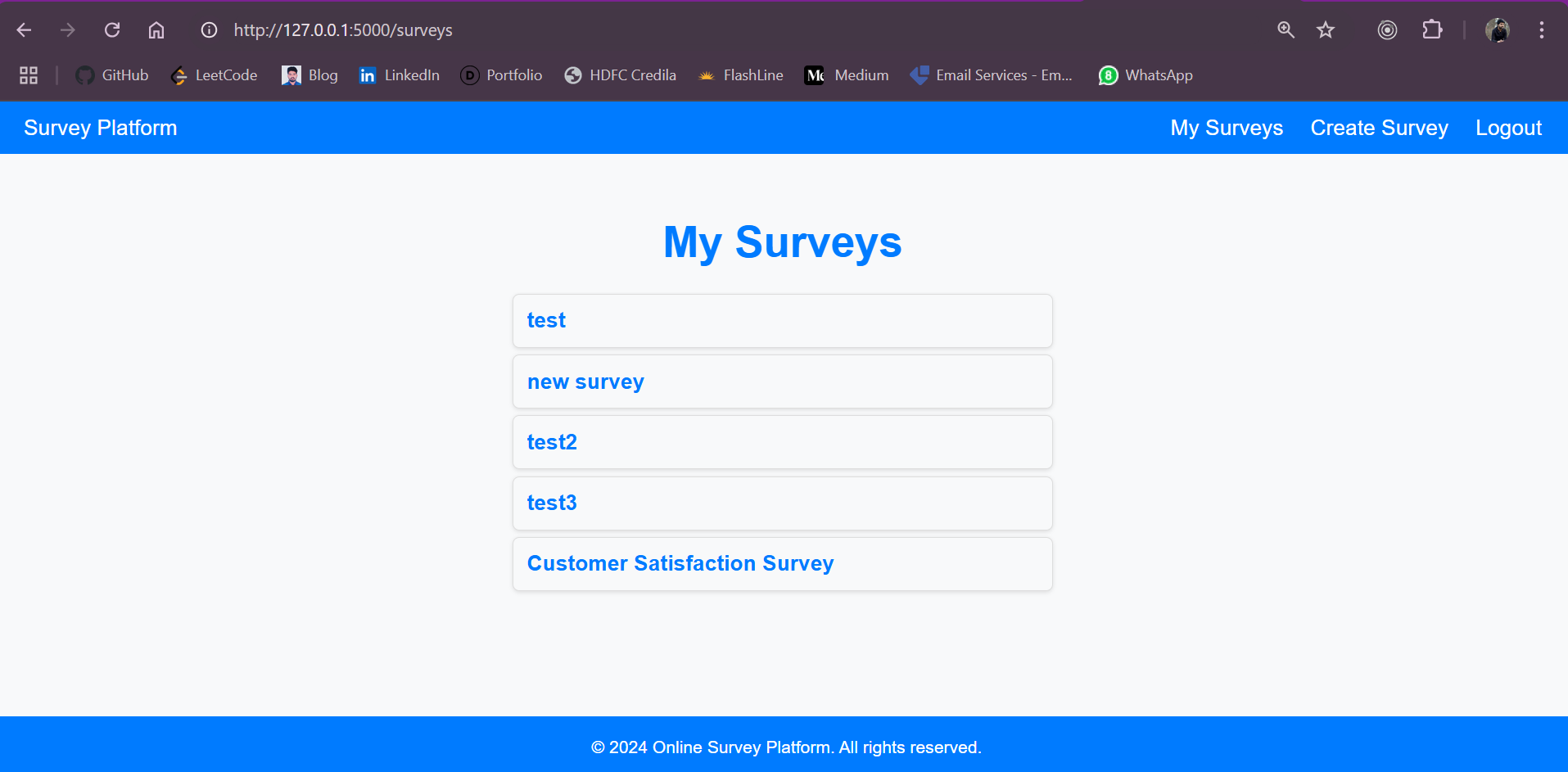
The Survey model stores survey details. Routes for survey creation ensure unique links using UUIDs.

**Screenshots:**

* Create a Survey Form



* My Surveys Page



**3. Response Handling**

**Features:**

* Responses are submitted through survey-specific forms.
* Data is stored in the Response model with timestamps.

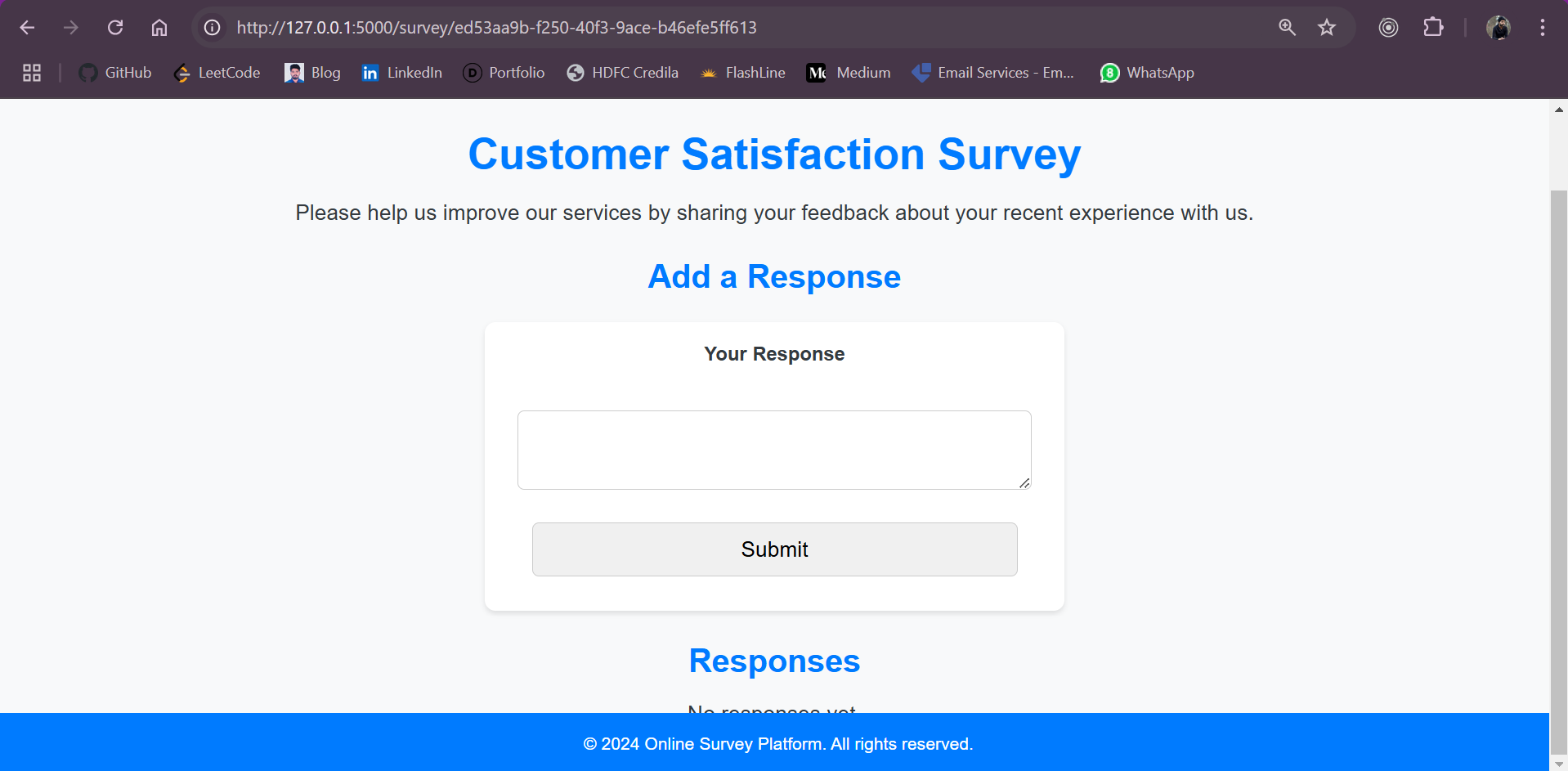
**Code:**

****

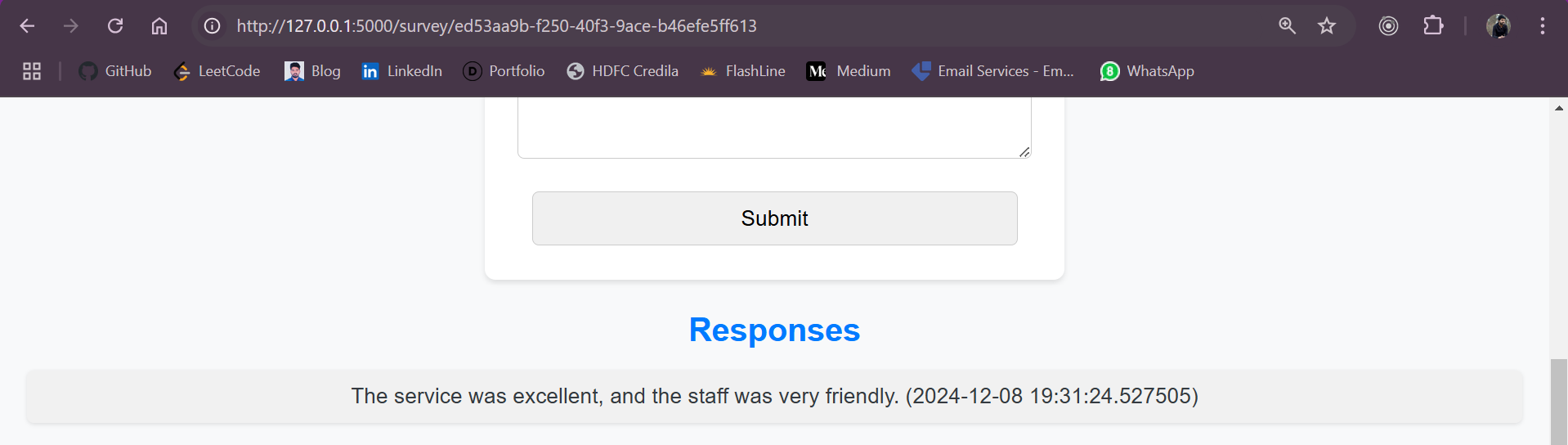
The response submission route validates input and associates it with the correct survey.

**Screenshots:**

* Response Submission Form



* Submitted Responses Page



**4. Data Visualization**

**Features:**

* Survey responses are visualized using bar charts.
* Chart.js dynamically generates charts based on the survey data.

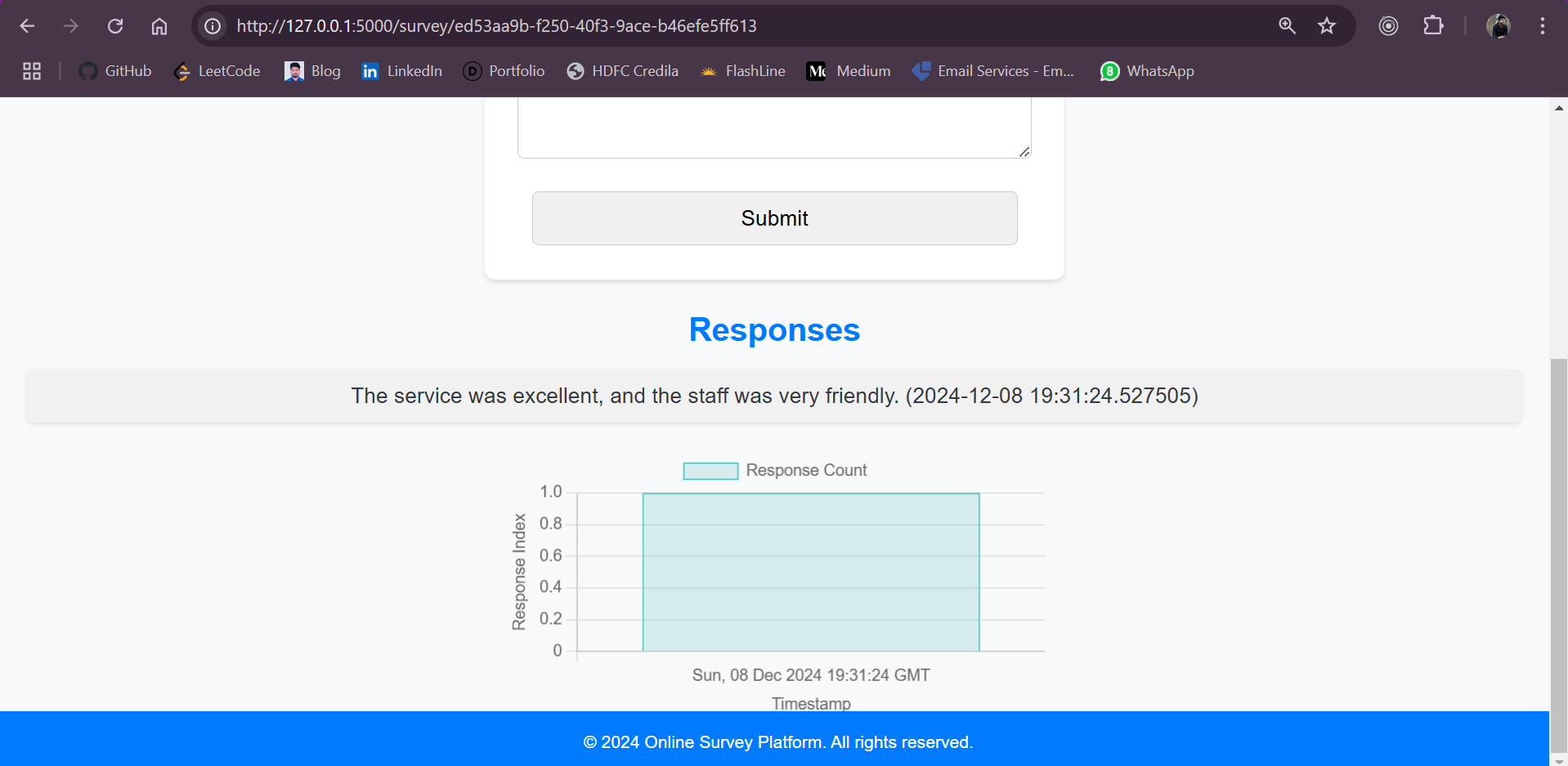
**Code:**

****

JavaScript functions fetch response timestamps and counts, passing them to Chart.js for rendering.

**Screenshots:**

* Response Chart



**Results and Discussions**

**Achievements**

1. Successfully implemented user authentication and survey creation.
2. Response data collection is functional and integrated with the database.
3. Real-time response visualization through interactive charts.

**Insights**

* Charts provide immediate feedback, enhancing decision-making processes.
* Modular design allows seamless integration of new features.

**Challenges and Solutions**

**Challenge 1: Handling dynamic charts with JavaScript.**

**Solution:** Researched Chart.js and debugged asynchronous data handling.

**Challenge 2: Ensuring unique survey links.**

**Solution:** Utilized Python’s uuid library to generate unique identifiers.

**Challenge 3: Integrating multiple libraries with Flask.**

**Solution:** Organized project structure to maintain compatibility and debugged library conflicts.

**Conclusion and Future Work**

**Conclusion**

The project successfully delivers a functional online survey platform with essential features, including authentication, survey management, response collection, and data visualization.

**Future Work**

1. Advanced analytics like sentiment analysis for qualitative data.
2. Integration with third-party services such as Google Forms or Slack.
3. Multi-language support for broader accessibility.
4. Export features for survey results (PDF, CSV).

**Team Member Contributions**

|  |  |  |
| --- | --- | --- |
| **Member 1: Backend Developer (Task Lead)**   * Designed and implemented the core backend functionalities using Flask. * Set up user authentication and session management with Flask-Login. * Developed database models using Peewee ORM for managing surveys, users, and responses. * Configured SQLite for development and planned PostgreSQL integration for production. * Implemented routes for creating surveys, managing responses, and visualizing data. * Handled debugging and resolved issues related to server functionality and database operations.   **Member 2: Frontend Developer and Documentation Lead**   * Created user-friendly HTML templates using Jinja2, ensuring a clean and intuitive interface. * Styled the application using CSS and ensured responsiveness across devices. * Integrated Chart.js for dynamic visualization of survey responses. * Collaborated on implementing CSRF protection for secure form submissions. * Authored detailed project documentation, including the README file and the final report, with an emphasis on methodology, results, and future enhancements. * Conducted user testing to gather feedback and improve usability. |  |  |

**References**

1. Flask Documentation: https://flask.palletsprojects.com
2. Peewee ORM Documentation: https://docs.peewee-orm.com
3. Chart.js Documentation: https://www.chartjs.org/docs
4. Stack Overflow Discussions: <https://stackoverflow.com>