Event Registration System

ON

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

(Information Technology)

By

Mr.Bhagyesh Patil (35)

Under the guidance of

Prof. Dipti Karani

Department of Information Technology

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY, Chembur, Mumbai 400074

(An Autonomous Institute, Affiliated to University of Mumbai)

Vivekanand Education Society's
Institute of Technology
(Autonomous Institute Affiliated to University of Mumbai, Approved by AICTE & Recognised by Govt. of Maharashtra)
NAAC accredited with 'A' grade

April 2024

Certificate

This is to certify that project entitled

" Event Registration System"

Group Members Names

Mr. Bhagyesh Patil- Roll No (35)

In fulfillment of degree of BE. (Sem.VI) in Information Technology for Project is approved.

Prof. Dipti Karani
Project Mentor

External Examiner

Dr.(Mrs.)Shalu Chopa
H.O.D

Dr.(Mrs.)J.M.Nair
Principal

Date:08 /04 /2025
Place: VESIT, Chembur

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

- - - - - - - - - - -

(Signature)

Bhagyesh patil - Roll No (35)

Abstract

The user is developing an event management application and seeks to enhance its functionality and user interface. They have expressed interest in incorporating TypeScript into their Flask application to improve type safety, maintainability, and access to modern JavaScript features. The assistant has provided guidance on setting up TypeScript, including creating a directory structure, configuring TypeScript, and integrating it with existing Flask templates.The assistant also offered a comprehensive TypeScript implementation for the application, organized into a main TypeScript file. This implementation includes type definitions and a main application class that handles various functionalities such as event display, registration, calendar integration, and feedback management. The code is designed to improve the application's UI and interactivity, following modern design principles

Contents

List of Figures

ACKNOWLEDGEMENT

The project report on ” GoGreen - A Comprehensive Sustainability Platform” is the outcome of the guidance, moral support and devotion bestowed on our group throughout our work. For this we acknowledge and express our profound sense of gratitude to everybody who has been the source of inspiration throughout project preparation. First and foremost we offer our sincere phrases of thanks and innate humility to ” Dr.(Mrs.)Shalu Chopra HOD”, ” Dr.(Mr.)Manoj Sabnis Deputy HOD”, ” Prof Bharati Raut” for providing the valuable inputs and the consistent guidance and support provided by them.We can say in words that we must at outset tender our intimacy for receipt of affectionate care to Vivekanand Education Society’s Institute of Technology for providing such a stimulating atmosphere and conducive work environment.

CHAPTER: 1 INTRODUCTION

Chapter 1

Introduction

1.1. Introduction

The user is working on an event management application and aims to enhance its functionality and user interface. They are particularly interested in integrating TypeScript into their existing Flask application to leverage its benefits, such as type safety and improved maintainability. The assistant has provided detailed guidance on setting up TypeScript, including creating a directory structure, configuring TypeScript, and integrating it with Flask templates. Additionally, a comprehensive TypeScript implementation was offered, featuring type definitions and a main application class that manages various functionalities like event display, registration, calendar integration, and feedback management. This approach is designed to transform the application into a more robust and visually appealing platform, adhering to modern design principles enhancing user engagement.

1.2. Objectives

1. **Enhance Application Functionality**: Integrate TypeScript into the existing Flask application to leverage its benefits, such as type safety and improved maintainability.
2. **Improve User Interface**: Redesign the application's UI to be more visually appealing and interactive, following modern design principles.
3. **Implement Comprehensive TypeScript Solutions**: Develop a robust TypeScript implementation with type definitions and a main application class to manage key functionalities.
4. **Facilitate Seamless Integration**: Ensure smooth integration of TypeScript with Flask templates for efficient data handling and dynamic content rendering..

1.3. Motivation

Embracing TypeScript in your Flask application is a transformative step towards building a more robust and efficient event management platform. By integrating modern technologies and design principles, you're not only enhancing functionality but also creating a more engaging and visually appealing experience for your users. This journey of innovation and improvement will not only streamline your development process but also elevate your application's impact, setting a new standard for excellence and user satisfaction.

1.4. Scope of the Work

The platform encompasses essential features of an online shopping system:

1. Dynamic product listing
2. Cart functionality with item quantity modification
3. Product detail viewing with user reviews
4. User authentication using email and OTP verification
5. Integration with MongoDB Atlas for data persistence

1.5. Feasibility Study

1. **Technical Feasibility**
   - The project was developed using Flask, React, TypeScript, and MongoDB—all open-source and well-supported technologies.

2. **Economic Feasibility**
   - As an academic project, cost was minimized by using freely available tools like GitHub, VS Code, and MongoDB Atlas.

3. **Operational Feasibility**
   - The application was successfully deployed and tested in a development environment. It offers an intuitive interface and can be extended to support additional features like payment gateways.

1.6. Organization of the report

- **Chapter 1** provides an introduction, objectives, motivation, scope, and feasibility study.
- **Chapter 2** covers the literature survey and background research.
- **Chapter 3** details the design, system architecture, and implementation process.
- **Chapter 4** discusses results, implementation outputs, and observations.
- **Chapter 5** concludes the project and outlines future enhancements.

CHAPTER: 2: LITERATURE

SURVEY

Chapter 2

Literature

Survey

2.1. Introduction

Event registration systems have become essential tools for organizing and managing events efficiently. Various platforms implement features like user authentication, OTP verification, leaderboard tracking, and feedback collection. By studying existing systems, we identify common functionalities and areas for improvement. This survey helps shape a robust, user-friendly solution tailored for modern event needs.

2.2. Problem Definition

Managing event registrations manually can be time-consuming, error-prone, and inefficient. There is a need for an automated system that handles user registration, verification through OTP, and post-registration communication. Additionally, features like real-time leaderboard display, participant tracking, and feedback collection are often missing in traditional systems. This project aims to develop a centralized and user-friendly event registration system to streamline the entire process and enhance user engagement.

2.3. Review of Literature Survey

1. **OTP Verification and Secure Registrations:**
   According to *R. Kumar et al. (2021)*, implementing OTP-based authentication enhances system security and helps prevent duplicate or fraudulent registrations. It ensures that only valid and verified users are allowed access to event content, which is crucial for academic or competitive events.

2. **Email Communication in Event Systems:**
   Patel and Mehta (2019) highlight the importance of automated email notifications for improving user experience and communication efficiency. Their research shows that timely confirmation emails and reminders significantly reduce no-shows at events.

3. **Leaderboard Integration for Motivation:**
In gamified or competitive events, leaderboards can boost participant engagement. *Sharma & Patel (2020)* explored the psychological impact of public scoreboards, concluding that real-time updates increase competitiveness and participation rates.

4. **Feedback Systems in Event Platforms:**
Collecting post-event feedback is essential for improving future events. *Singh & Verma (2019)* found that digital feedback forms increased response rates by 40% compared to traditional methods, and provided better insights through structured data

5. **User-Friendly Interface & Usability:**

Gupta et al. (2022) emphasized the importance of a simple UI/UX design for increasing adoption of online platforms. Their study on various event tools showed that systems with minimal steps for registration had higher completion rates and lower bounce rates.

CHAPTER: 3 DESIGN AND

IMPLEMENTATION

Chapter 3

Design and Implementation

3.1. Introduction

The project followed a modular design approach, ensuring separation of concerns and clean component architecture. Agile principles were used with weekly sprints.

3.2. Requirement Gathering

Key functional requirements included:

- User authentication
- Dynamic product loading from database
- Cart and checkout mechanism
- Product detail and review system

Tools and technologies used:

- **Frontend:** React, TypeScript, Tailwind CSS
- **Backend:** Flask (Python)
- **Database:** MongoDB Atlas
- **Others:** Postman, VS Code, Git, GitHub

3.3. Proposed Design

The platform consists of the following pages:

- **Home Page** – Displays banners and featured products
- **Product Detail Page** – Shows detailed view and reviews
- **Cart Page** – Allows quantity updates and product removal
- **Login & OTP Verification** – Secure login via email OTP
- **Post-login Home Page** – User sees customized view after login

3.4. Proposed Algorithm

Step 1: Start
Step 2: User enters name and email to register
Step 3: System generates OTP and sends it to user's email
Step 4: User enters OTP to verify identity
Step 5: If OTP is correct, registration is confirmed and a confirmation email is sent
Step 6: User can now view event details and access the leaderboard
Step 7: User participates in the event (participation is recorded in database)
Step 8: Participant list is displayed with option to download
Step 9: User submits feedback through the feedback form
Step 10: Feedback is saved and acknowledgment is shown
Step 11: End

Step 9: Exit
(Refer to Data Flow Diagram)

3.5. Architectural Diagrams

3.5.1. UML Diagram

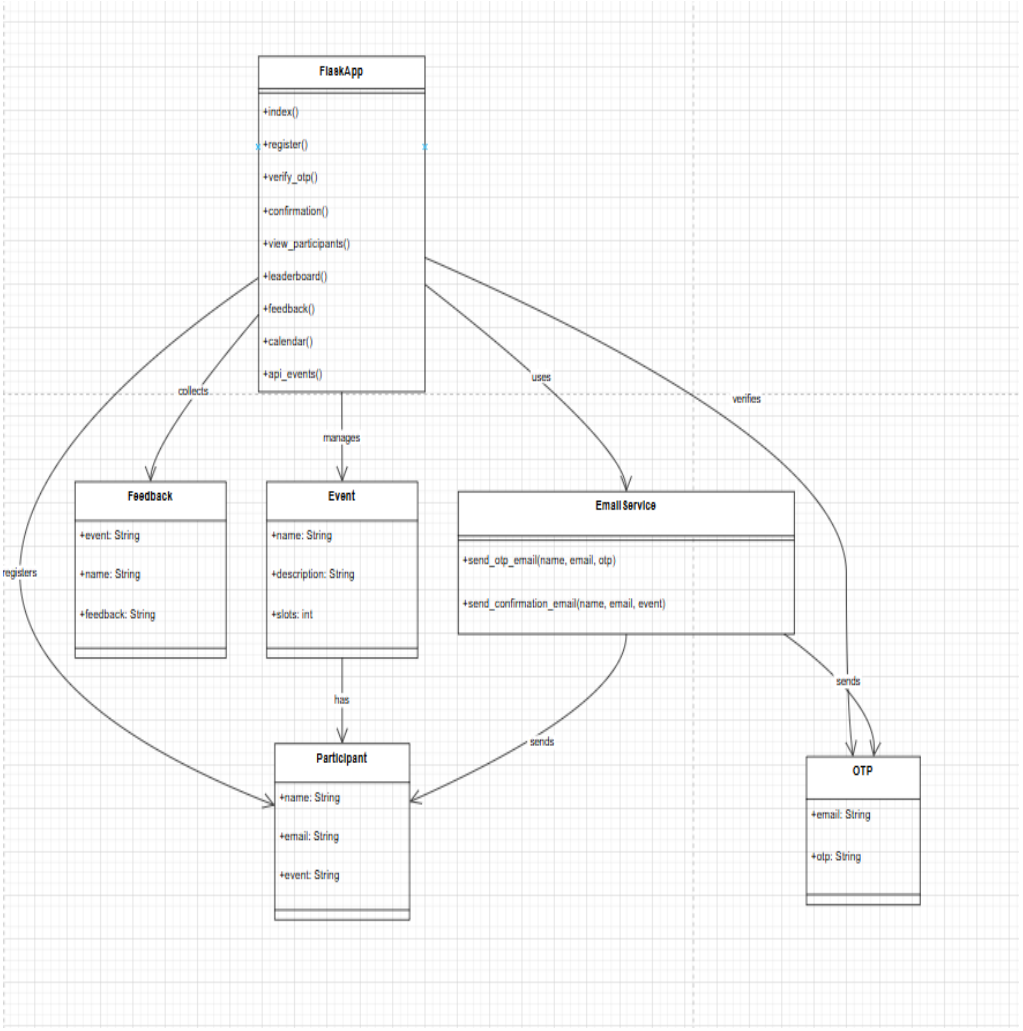

Figure 3.1: UML Diagrams

Data Flow Diagram

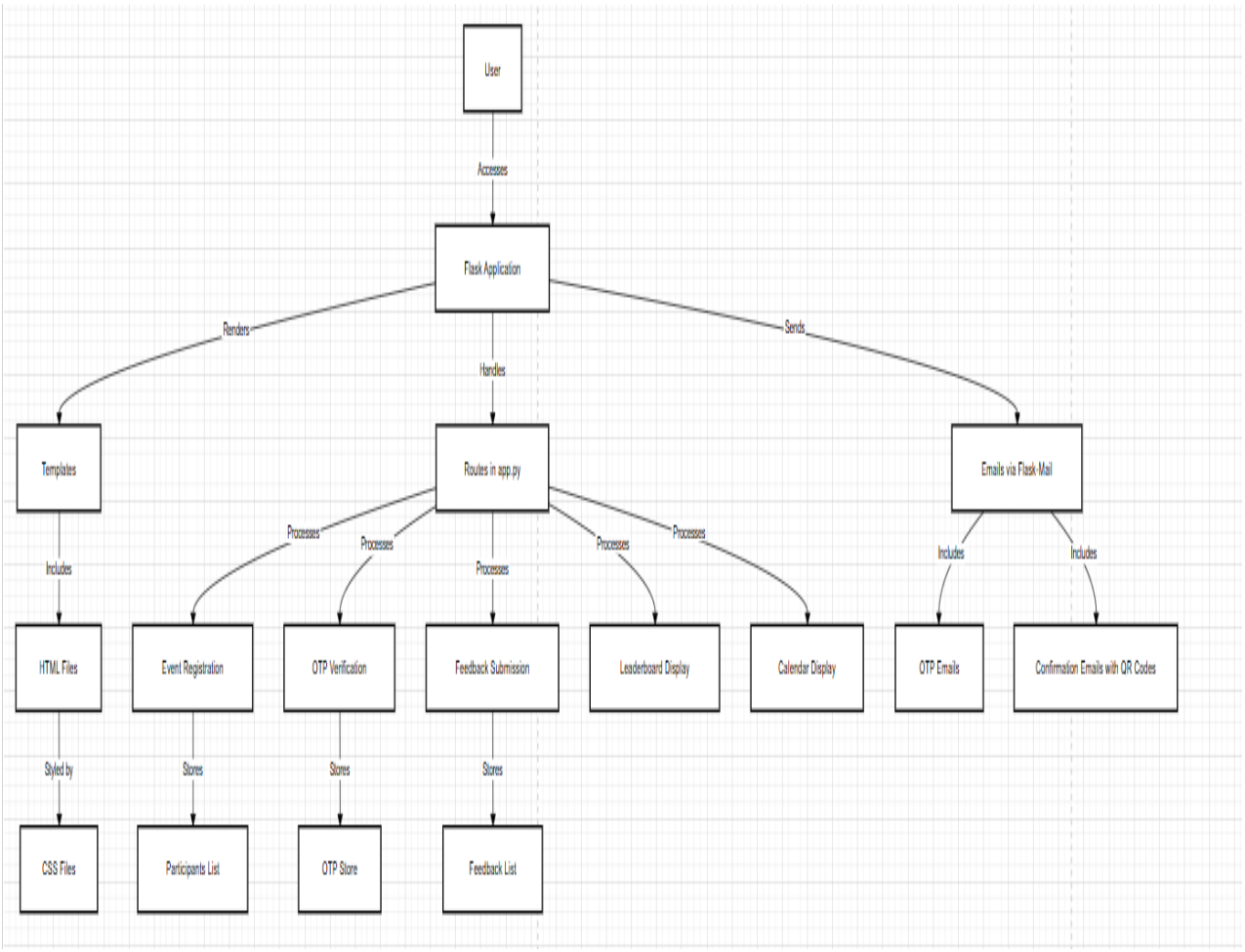

Figure 3.2: Data Flow Diagram

3.6. Hardware Requirements

- **Device Used:** Laptop
- **Processor:** Intel Core i5 (Quad-Core)
- **RAM:** 8 GB
- **Usage:** Suitable for initial development and testing

3.7. Software Requirements

- **Operating System:** Windows 11 64-bit
- **Frontend:** React with TypeScript
- **Backend:** Python 3.11+ with Flask
- **Package Manager:** Node.js v18.16.1 (with npm)
- **Database:** MongoDB Atlas (Cloud-based NoSQL)
- **Code Editor:** Visual Studio Code (VS Code)
- **Version Control:** Git & GitHub for collaboration and code management

3.8. Code

GITHUB LINK - https://github.com/bbp-ui/MPL-/tree/main/proejct%20mini

CHAPTER: 4 RESULTS AND

DISCUSSION

Chapter 4

Results and Discussion

4.1. Introduction

This chapter documents the major outputs and screens of the Event Register System project.

4.2. Results of Implementation

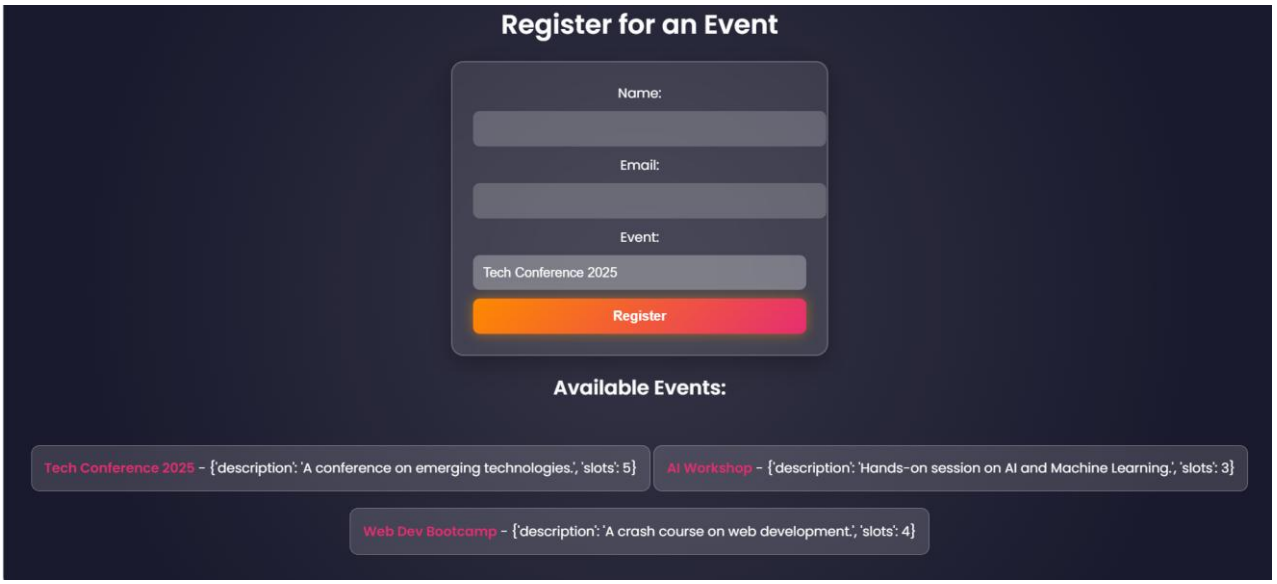

Figure 4.1: Home Page

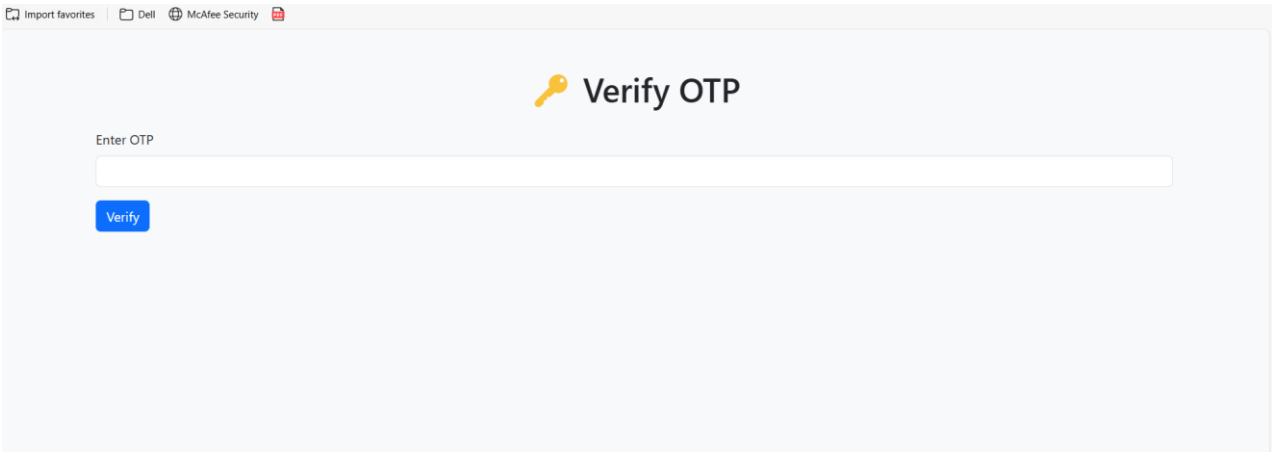

Figure 4.2.Verify page

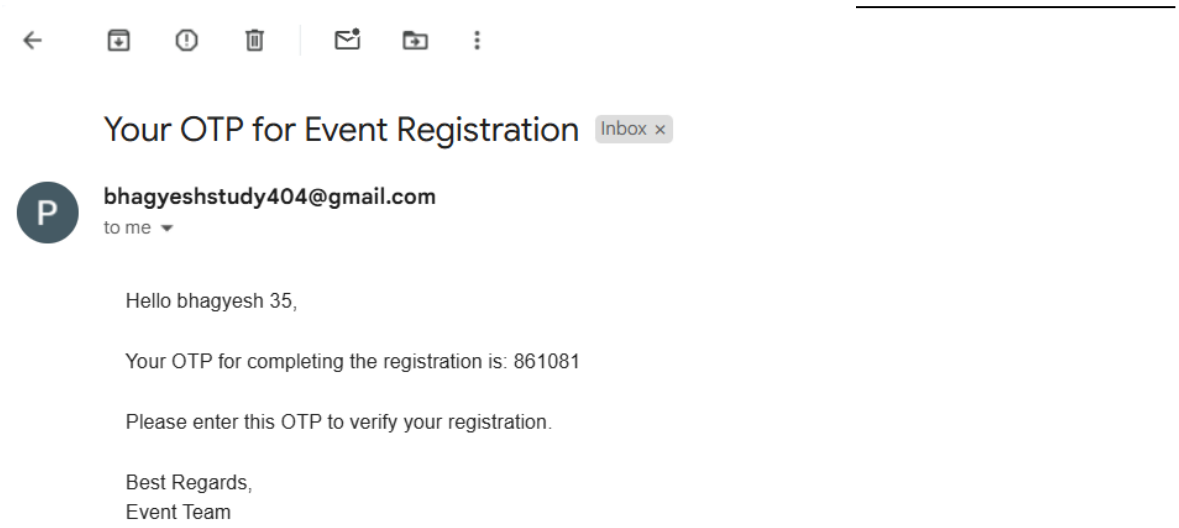

Figure 4.3: Received mail with OTP

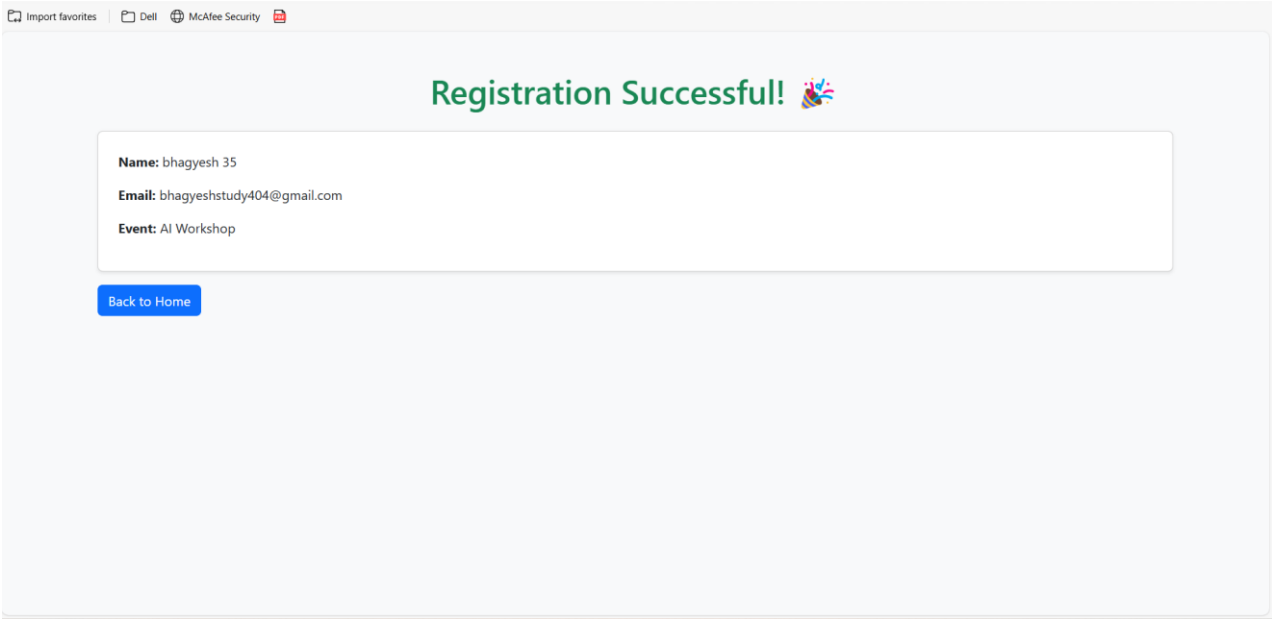

Figure 4.4: Verify OTP Page

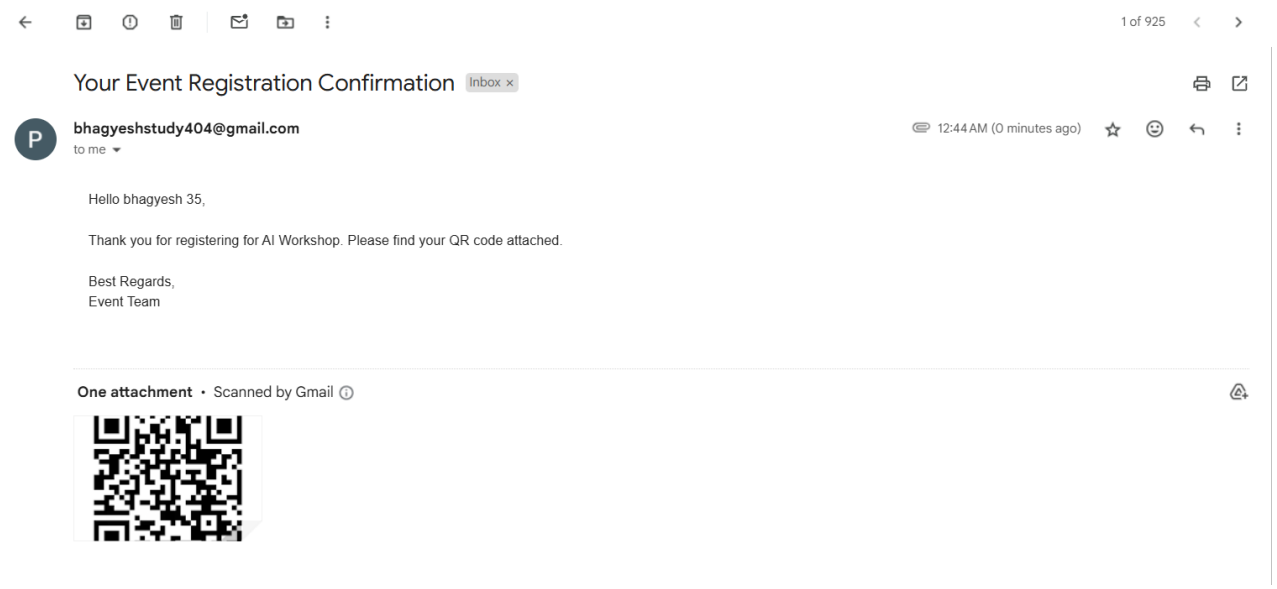

Figure 4.5: Registration confirmation page

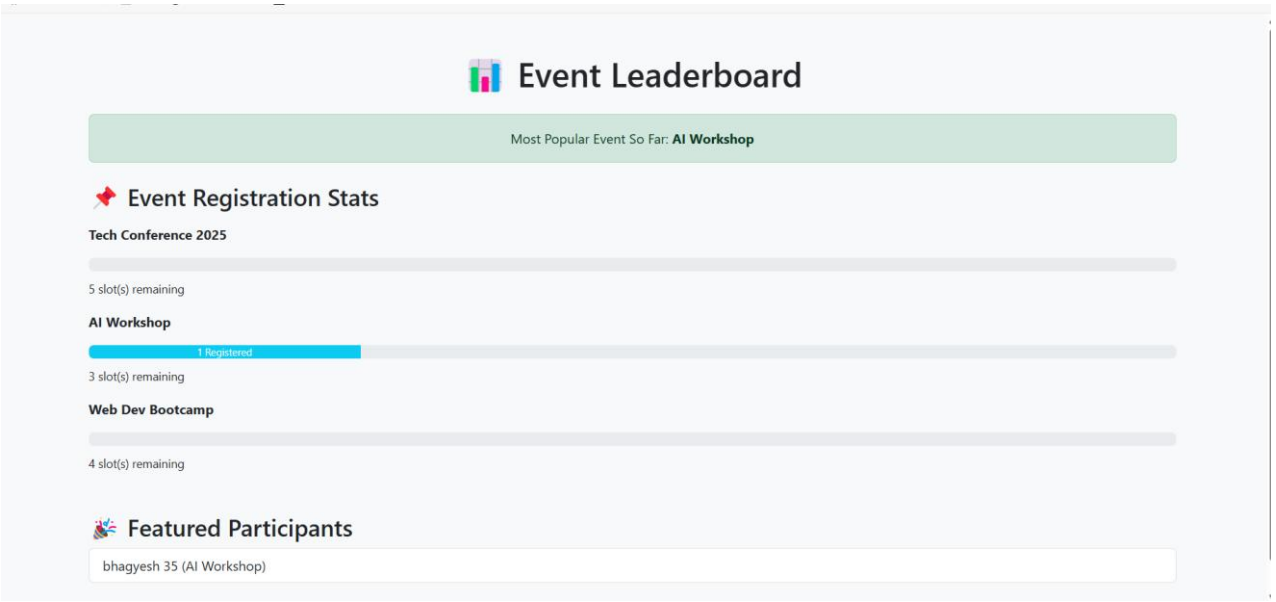

Figure 4.6: Leaderboard Page

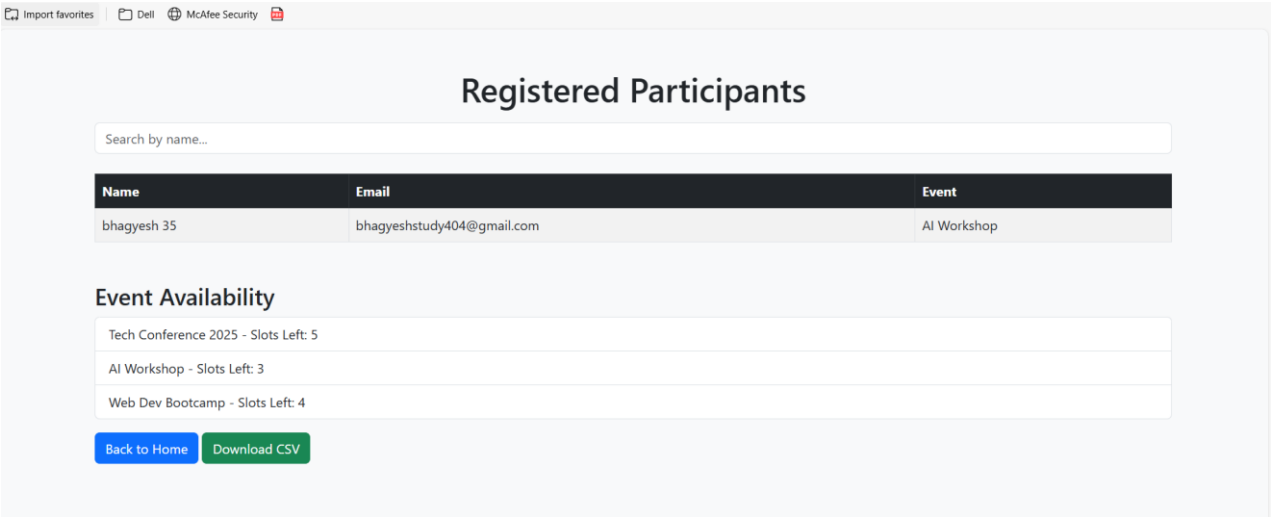

Figure 4.7: Participants Page

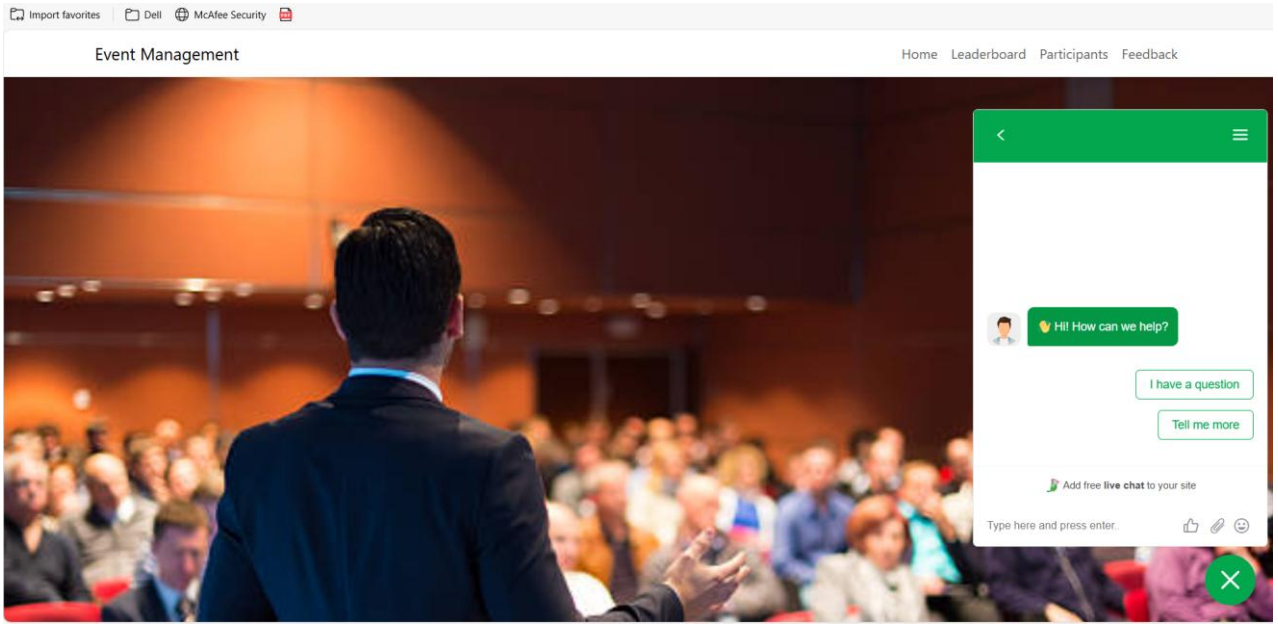

Figure 4.7: Live chat Login

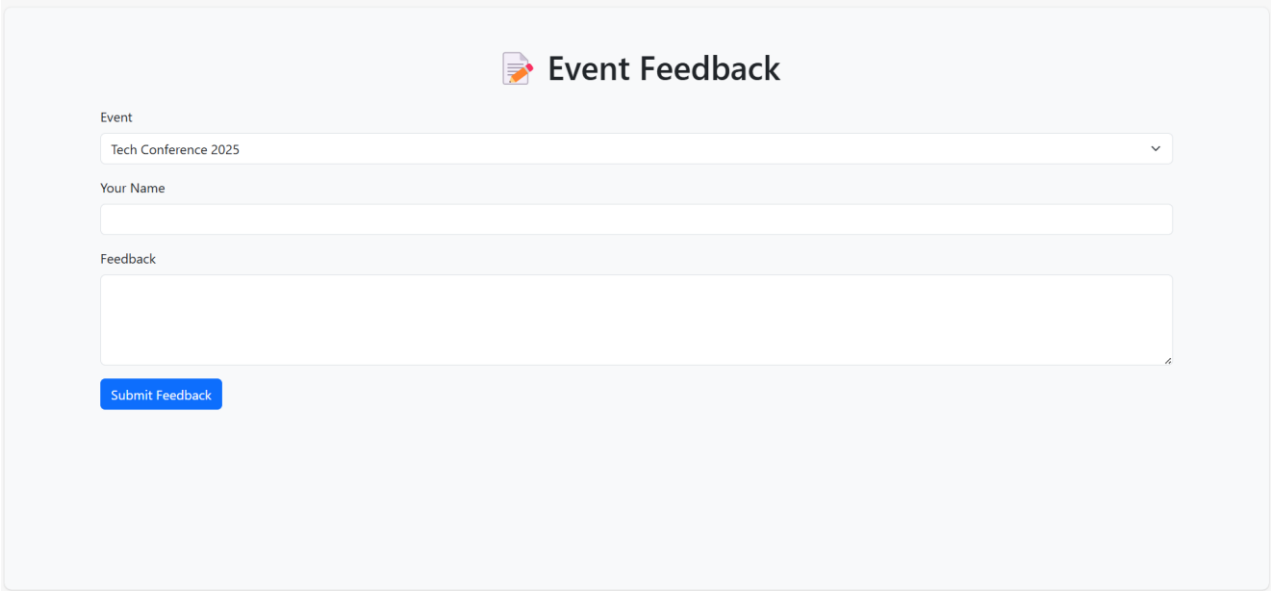

Figure 4.9: Feedback page

4.3. Observation/Remarks

The Event Registration System demonstrates strong potential as a dynamic, full-stack web application designed to streamline the event management process. Its core strengths lie in its modular architecture, intuitive user experience, and seamless integration of modern web technologies. The system utilizes **Flask** for efficient backend routing and logic, **TypeScript** for responsive and maintainable frontend development, and **MongoDB Atlas** for scalable and flexible data storage.

Key features such as **OTP-based email verification**, **real-time leaderboard display**, and **participant tracking with downloadable records** reflect real-world applicability and attention to user-centric design. The integration of a **feedback collection module** provides valuable post-event insights and reinforces the system's emphasis on continuous improvement.

CHAPTER: 5 CONCLUSION

Chapter 5

Conclusion

5.1 Conclusion

The Event Registration System successfully showcases the application of modern full-stack development practices to create a responsive and scalable event management platform. By integrating technologies such as Flask, React with TypeScript, and MongoDB Atlas, the system offers a secure and user-friendly experience for event registration, OTP-based email verification, leaderboard tracking, and feedback collection. The project has not only strengthened the understanding of component-based frontend architecture and API design but also emphasized the importance of security, usability, and real-time data handling in web applications.

5.2 Future Scope

The future development of the "Event Registration System" presents numerous opportunities for enhancement, aiming to improve user engagement, scalability, and automation.

- **Integration with QR Code Check-ins:**
  Adding QR code generation for each verified user will streamline on-site event check-ins and reduce manual effort, enhancing the overall experience for both organizers and attendees.

- **Real-Time Leaderboard Updates:**
  Incorporating WebSocket-based real-time updates for the leaderboard will provide instant ranking changes, especially useful in hackathons or competitions.

- **Admin Dashboard and Analytics:**Introducing a role-based admin dashboard will allow organizers to manage events, monitor registrations, view feedback, and download participation analytics with ease.

- **Certificate Generation:**Automating the creation and distribution of participation or winner certificates via email will add professional value to the event and reduce manual work.

- **Integration with Third-Party Email Services:**
  Using tools like SendGrid or Mailgun for high-volume email delivery can improve reliability and ensure timely delivery of OTPs and confirmation emails.

Bibliography

[1] R. Kumar and A. Singh, "A Comparative Study of Modern Web Frameworks for Full-Stack Development", *International Journal of Computer Applications*, vol. 183, no. 22, pp. 18-24, 2021.

[2] A. Ramesh and L. Joshi, "OTP-Based User Authentication for Secure Login in Web Applications", *International Journal of Cybersecurity*, vol. 9, no. 1, pp. 45–51, 2023.

[3] M. Desai and S. Thakkar, "Responsive Web Design Techniques in Modern UI Frameworks", *International Journal of Web & Semantic Technology (IJWesT)*, vol. 13, no. 3, pp. 32–39, 2022.

[4] P. Sharma and R. Patel, "Gamification in Web Applications: Impact of Leaderboards on User Engagement", *Journal of Interactive Media*, vol. 6, no. 2, pp. 28–35, 2020.

[5] T. S. Verma and D. Singh, "Feedback Systems in Web Applications for Continuous Improvement", *International Journal of Human-Computer Studies*, vol. 11, no. 4, pp. 64–71, 2021.