A PROJECT REPORT ON

(Mystery Message Platform)

BY

Virbhadra Uttam Haibatpure

(SP47230038)

IN PARTIAL FULFILLMENT OF

M.Sc.(Computer Science) Part II, Sem IV

SIR PARASHURAMBHAU COLLEGE, PUNE

(2024 - 2025)

Elite Softwares

TO WHOMSOEVER IT MAY CONCERN

INTERNSHIP CERTIFICATE

This is to certify that **Virbhadra Haibatpure** has successfully completed internship in "**Web Development, Python-DJango Framework Intern**" and the Project title was during internship **"Mystery Message"** under the assistance and guidance in connection with his/her Industrial Internship at Elite Softwares.

Duration of this Internship was from **6th / January / 2025 to 1st / May / 2025**

He/she was found sincere & hard working during this tenure.
We wish him/her all the best for his/her future endeavors.

Elite Softwares
Pune

Best Regards,

Swami Panjala
Founder and CEO
Elite Softwares, Pune

Shikshana Prasaraka Mandali’s

SIR PARASHURAMBHAU COLLEGE
(EMPOWRED AUTONOMOUS)

TILAK ROAD, PUNE – 411 030.

Department of Computer Science

Certificate

This is to certify that the work in the entitled project of

____

has been carried out by Mr./Mrs./Ms. **Virbhadra Uttam Haibatpure**
as the partial fulfillment of the requirements of the M.Sc. (Computer Science)
Part II, MSCOSDSC401 - Industrial training at Elite Softwares

____

for the last semester starting from 2024-2025

Date : / /20

Head
Teacher Incharge *Department of Science Computer*

1. Internal Examiner *2. External Examiner*

INDEX

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who supported and guided me throughout the development of my project, **“Mystery Message”**, a web-based platform for secure and anonymous feedback sharing.

First and foremost, I am deeply thankful to my project guide, **Prof. Mansi Bornare**, for her constant guidance, valuable insights, and encouragement during every phase of this project. Her mentorship, under the academic environment of **Sir Parshurambhau College, Pune**, has been instrumental in shaping the direction and quality of this work.

I am also sincerely grateful to my company mentor, **Mr. Swami Panjala**, at **EliteSoftwares**, for providing me the opportunity to work on this live project as part of my internship. His industry knowledge, continuous feedback, and practical advice helped me bridge the gap between theory and application.

My appreciation also goes to the entire **EliteSoftwares** team for their cooperation and technical support during the internship period. I would like to thank my faculty and technical mentors for imparting hands-on experience in technologies like **Next.js, React, MongoDB, TypeScript, Node.js, Tailwind CSS**, and deployment tools such as **Vercel and Azure Web Apps**, all of which were integral to building and launching this platform.

I extend my thanks to my peers for their constructive feedback and collaborative spirit throughout the development process, and to my family and friends for their constant encouragement and motivation.

Working on *Mystery Message* has been an enriching journey that combined technical learning with real-world application, and I am truly grateful for the opportunity to grow through this experience.

Thank you all.

INTRODUCTION TO PROBLEM

In today’s digital-first world, open and honest communication within teams, classrooms, or communities is more important than ever. However, individuals often hesitate to share critical feedback or personal thoughts due to fear of judgment, social pressure, or workplace hierarchy. This lack of transparent and anonymous communication can hinder growth, trust, and collaborative improvement.

Traditional feedback channels typically involve emails, in-person discussions, or suggestion boxes—methods that can be inefficient, biased, or intimidating. Moreover, most organizations lack a streamlined digital solution that allows users to share genuine thoughts anonymously while ensuring security, usability, and structured feedback management.

Users seeking to share anonymous feedback commonly face several challenges:

- No secure, centralized platform for anonymous communication.
- Lack of accountability or moderation in informal feedback channels.
- Difficulties in sending or receiving feedback without revealing identity.
- Limited customization and automation in existing solutions.

Mystery Message addresses this gap by offering a modern, user-friendly platform that enables users to share feedback or messages anonymously. Built with Next.js, React, MongoDB, and TypeScript, the platform provides a seamless experience where users can send secure messages, receive real-time notifications, and even engage with moderation tools if needed. With features like automated email notifications, responsive UI, and easy deployment through Vercel and Azure Web Apps, *Mystery Message* streamlines the feedback process while ensuring trust, privacy, and ease of use.

EXISTING SYSTEM

The current methods for sharing feedback—whether in educational institutions, workplaces, or community settings—are often fragmented, informal, and inefficient. Most environments rely on manual or traditional channels like verbal communication, suggestion boxes, or general messaging platforms that lack anonymity, structure, and security.

In many cases, individuals are hesitant to provide honest feedback due to fear of repercussions, lack of trust, or the absence of a safe and anonymous space. As a result, valuable insights are often withheld, limiting opportunities for personal and organizational growth. Moreover, without a dedicated digital platform, collecting, managing, and responding to feedback becomes a cumbersome and unorganized task for administrators or moderators.

From the perspective of those receiving feedback (e.g., teachers, managers, or teams), the lack of a structured system also means:

- Difficulty in identifying trends or addressing concerns effectively.
- No automation for managing submissions, responses, or notifications.
- Limited opportunity to engage users or track feedback over time.

In summary, the existing feedback and communication process faces the following key challenges:

- No centralized or secure platform for anonymous messaging.
- Manual and unscalable systems for collecting feedback.
- Poor user experience due to inconvenient or outdated methods.
- Lack of structure, moderation, and data-driven insights.
- Limited adoption due to privacy concerns and technical barriers.

SCOPE OF WORK

The scope of the **Mystery Message** project includes the end-to-end development of a secure, scalable, and user-friendly web-based platform that facilitates **anonymous feedback sharing** in a structured and efficient manner. This platform is designed to address the communication gaps in traditional feedback systems by ensuring privacy, accessibility, and automation.

Key Areas Covered:

1. Frontend Development

- Built responsive, modern UI components using **React**, **Tailwind CSS**, and **Radix UI**.
- Designed intuitive user interfaces for message submission, authentication, and admin moderation.
- Ensured a consistent and accessible design experience across devices.

2. Backend Development

- Developed server-side logic using **Node.js** and **Next.js API routes**.
- Used **MongoDB** (via Mongoose) to manage user accounts, messages, and activity logs.
- Implemented secure form handling, API validation, and feedback processing.

3. Core Feature Implementation

- **Anonymous Messaging Module**: Allows users to send messages securely without revealing their identity.
- **Authentication System**: Integrated with `next-auth` for user management and session handling.
- **Email Notification System**: Used **React Email** and **Resend** to send automated alerts and updates.
- **Admin Controls**: Provided restricted access for admins to review, moderate, or archive messages.

4. Type Safety & Validation

- Integrated **TypeScript** and **Zod** for type safety, input validation, and improved development workflow.
- Ensured strong typing across backend and frontend code to minimize bugs and enhance maintainability.

5. Deployment & Hosting

- Deployed the frontend and backend via **Vercel** for seamless CI/CD integration.
- Configured automated deployments using **GitHub Actions** for testing, building, and release cycles.
- Ensured scalability and performance for handling over 1,000 users post-launch.

FEASIBILITY STUDY

A feasibility study was conducted to assess the practicality, sustainability, and overall success potential of the **Mystery Message** project before its full-scale development and deployment. This evaluation covered key areas such as technical viability, operational fit, cost-efficiency, and scheduling to ensure the project could be executed effectively within the given resources and timeframe.

1. Technical Feasibility

The project is highly feasible from a technical standpoint, as it utilizes modern, widely adopted technologies and tools:

- **Frontend**: Built using **React**, **Tailwind CSS**, and **Radix UI** for a responsive and accessible user experience.
- **Backend**: Developed using **Next.js API routes** and **Node.js**, which support scalable server-side rendering and RESTful endpoints.
- **Database**: **MongoDB**, integrated with **Mongoose**, ensures efficient data modeling and secure storage of messages and user information.

These tools are not only well-supported but also integrate seamlessly with cloud-based platforms like **Vercel** for fast, reliable deployments, making the project manageable with current infrastructure and developer skillsets.

2. Operational Feasibility

The platform was designed with both user experience and maintainability in mind:

- **End users** can anonymously send feedback messages easily through a clean and intuitive interface.
- **Administrators** can monitor, moderate, and respond to messages through protected routes and a potential admin panel.
- The system aligns well with operational needs of educational institutions, teams, or organizations seeking safe and anonymous communication, reducing communication barriers and promoting honest feedback loops.

3. Economic Feasibility

The project is cost-effective and suitable for budget-conscious implementations:

- The technology stack, including **Next.js**, **MongoDB Atlas**, **Tailwind CSS**, and **Vercel**, is primarily open-source or has generous free tiers.
- Development was conducted as part of an academic and internship initiative, minimizing labor costs.
- The platform automates several manual feedback processes, potentially leading to long-term operational savings and improved engagement outcomes.

4. Schedule Feasibility

The development cycle was realistically scoped and completed within the academic/internship timeline:

- The project followed a structured 16-week schedule covering planning, UI/UX design, development, testing, and deployment.
- Agile-style task management ensured steady progress and timely delivery in accordance with academic and organizational milestones.

SYSTEM REQUIREMENT

This section outlines the hardware and software specifications required to develop, deploy, and run the **Mystery Message** platform effectively.

1. Hardware Requirements

- **Processor**: Intel Core i3/i5 or equivalent (AMD Ryzen 3/5)
- **RAM**: Minimum 4 GB (8 GB recommended for smooth development)
- **Storage**: At least 100 GB (to accommodate code, dependencies, and build files)
- **Display**: Minimum 13" screen with 720p resolution (1080p recommended for better UI design experience)

2. Software Requirements

Development Environment

- **Operating System**: Windows 10/11, macOS, or any modern Linux distribution (e.g., Ubuntu)
- **Code Editor**: Visual Studio Code (recommended) or any preferred IDE with JavaScript/TypeScript support
- **Web Browser**: Google Chrome or Mozilla Firefox (for testing and debugging)
- **Version Control**: Git with GitHub for code collaboration and version tracking

Backend

- **Runtime**: Node.js (v18 or higher)
- **Framework**: Next.js (v14+ with API Routes support)
- **Database**: MongoDB (hosted via MongoDB Atlas or local installation)
- **ORM**: Mongoose (for schema modeling and MongoDB interaction)

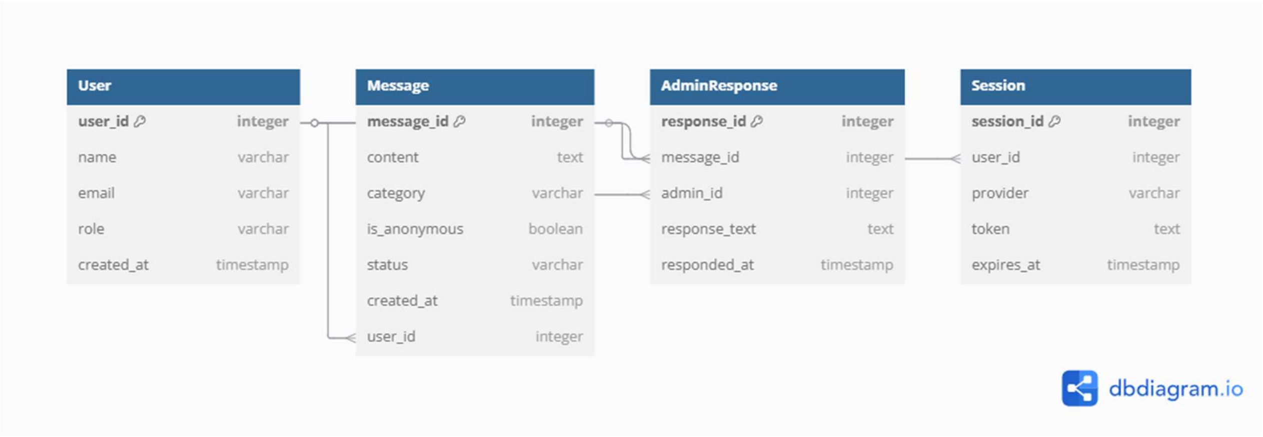
Frontend

- **Libraries & Tools**: React (v18+), Tailwind CSS, Radix UI
- **Type Safety**: TypeScript (v5+)
- **Form Handling**: React Hook Form with Zod for validation
- **Authentication**: NextAuth.js for secure and flexible auth

Deployment & CI/CD

- **Hosting**: Vercel (for frontend and backend via Next.js)
- **CI/CD**: GitHub Actions (for automated deployment, linting, and testing)

ER - DIAGRAM



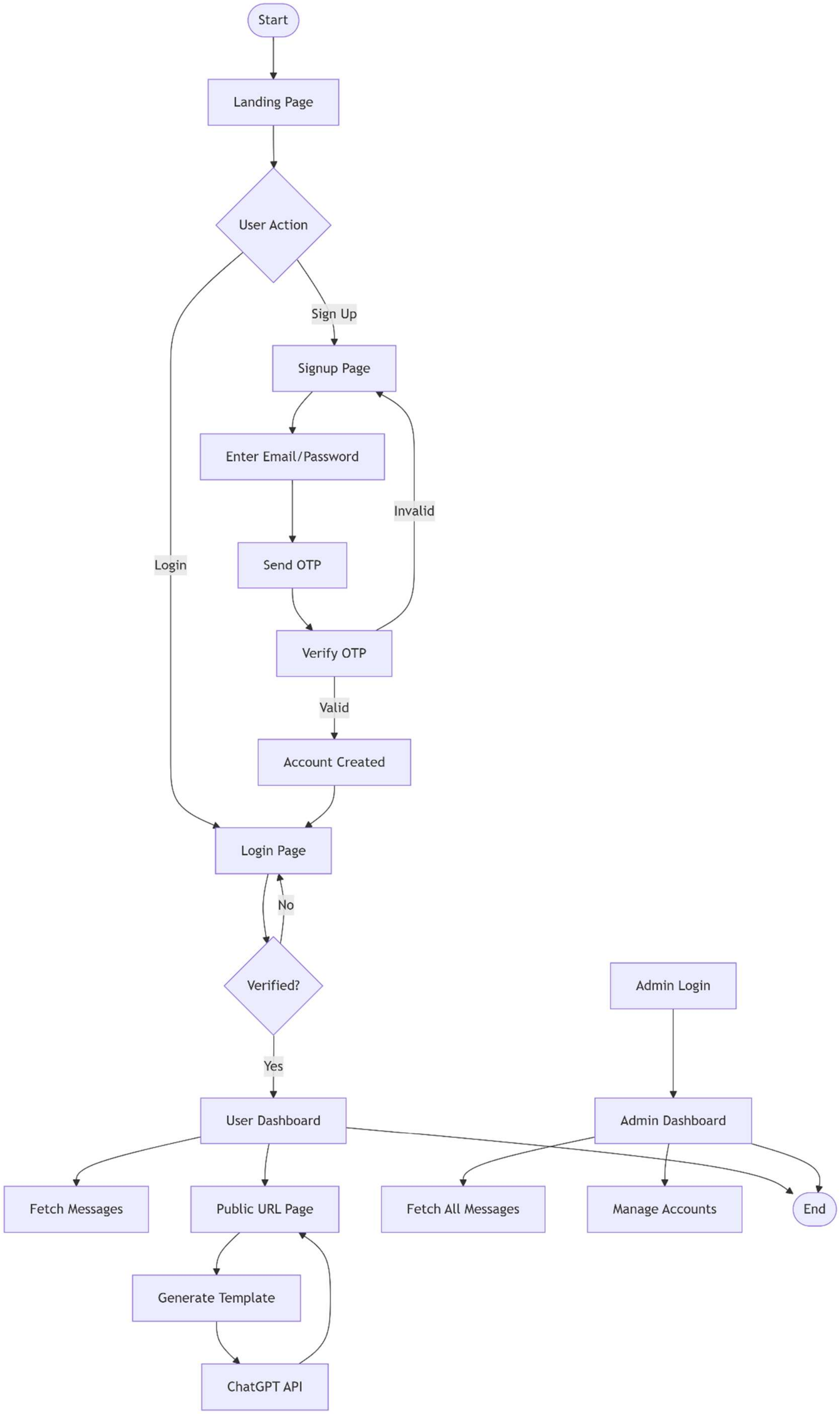
User	
user_id	**integer**
name	varchar
email	varchar
role	varchar
created_at	timestamp

Message	
message_id	**integer**
content	text
category	varchar
is_anonymous	boolean
status	varchar
created_at	timestamp
user_id	integer

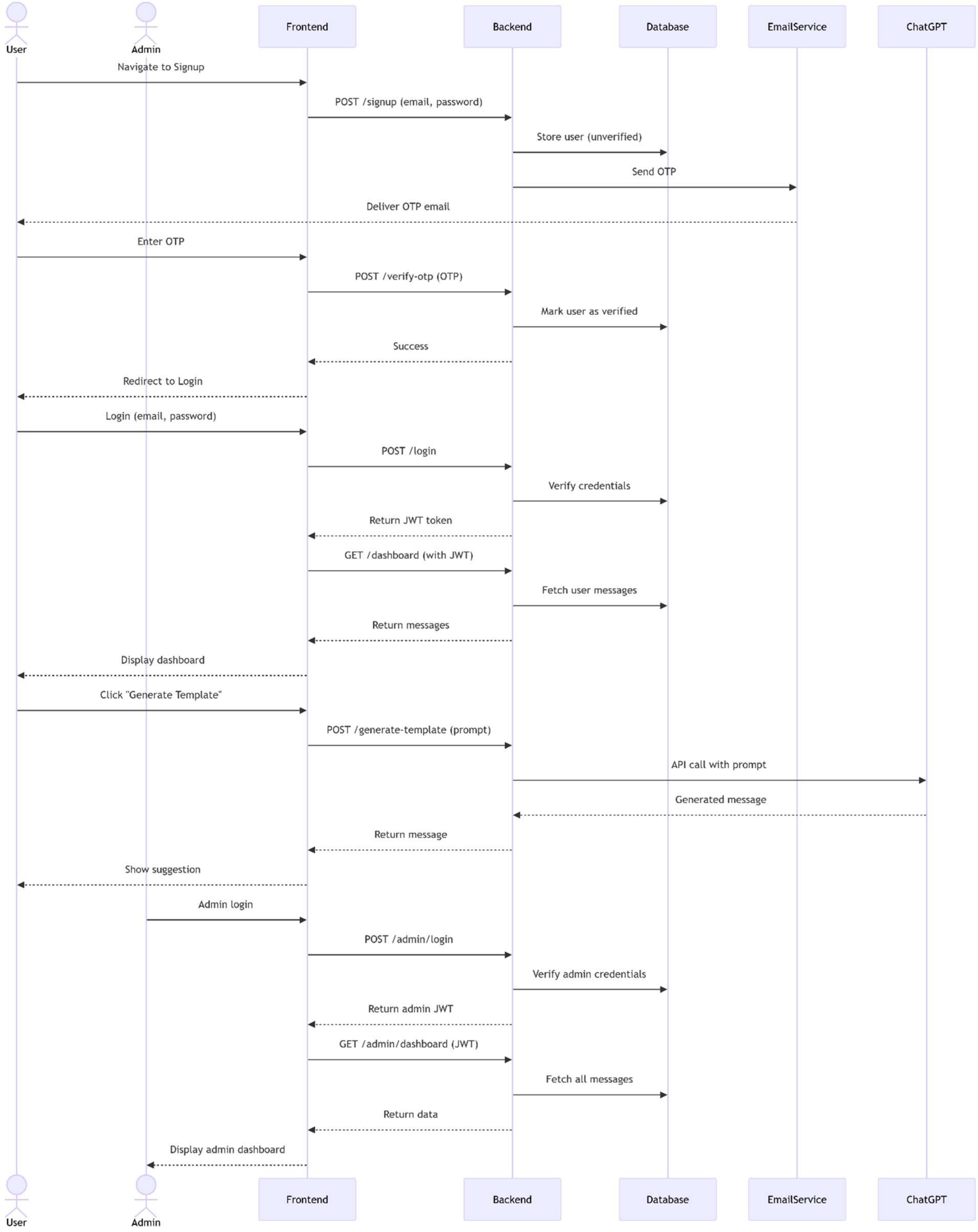
AdminResponse	
response_id	**integer**
message_id	integer
admin_id	integer
response_text	text
responded_at	timestamp

Session	
session_id	**integer**
user_id	integer
provider	varchar
token	text
expires_at	timestamp

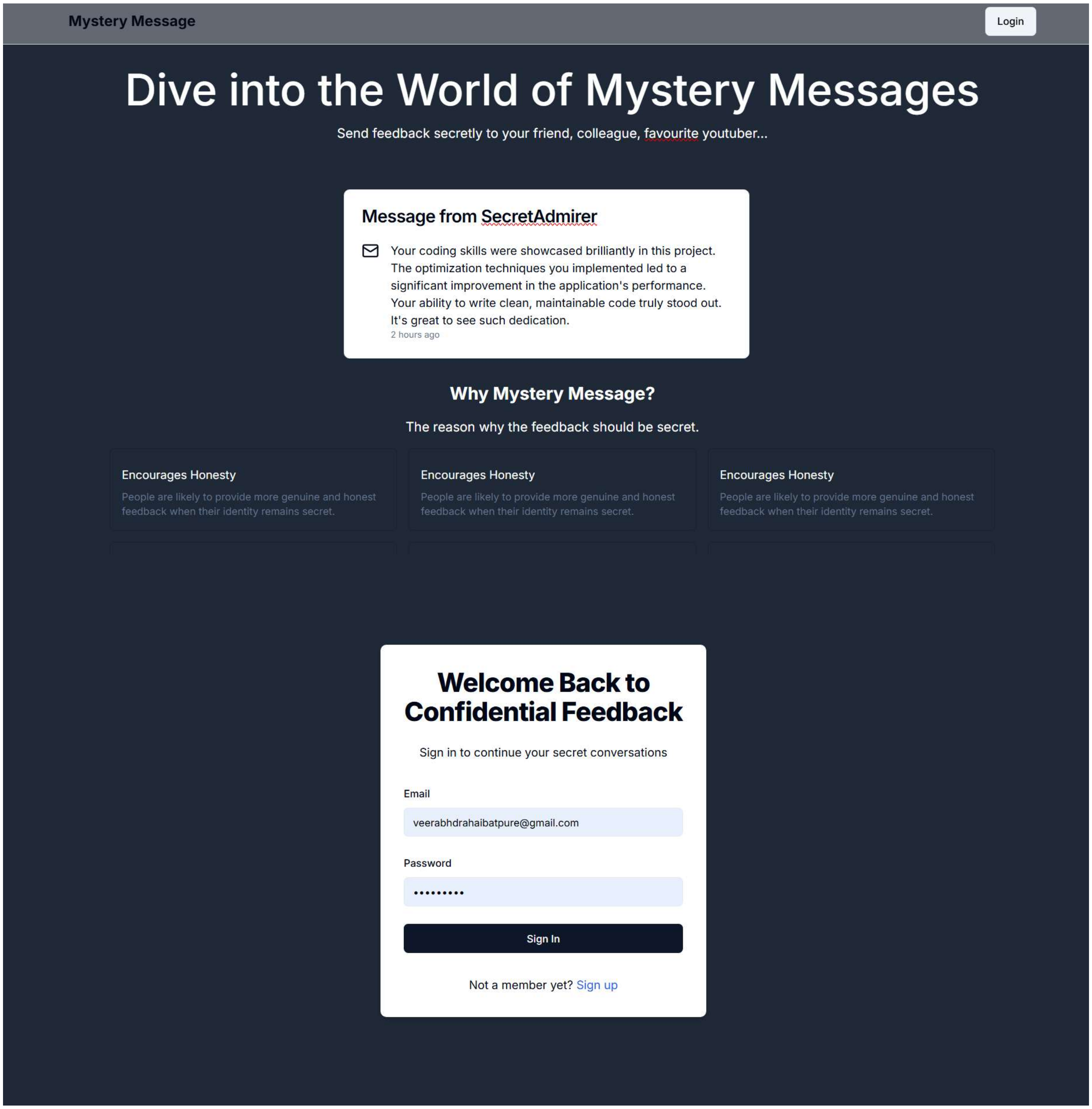
ACTIVITY DIAGRAM



Sequence Interface



USER INTERFACE



Dive into the World of Mystery Messages

Send feedback secretly to your friend, colleague, favourite youtuber...

Message from SecretAdmirer

Your coding skills were showcased brilliantly in this project. The optimization techniques you implemented led to a significant improvement in the application's performance. Your ability to write clean, maintainable code truly stood out. It's great to see such dedication.

2 hours ago

Why Mystery Message?

The reason why the feedback should be secret.

Encouraging Honesty

People are likely to provide more genuine and honest feedback when their identity remains secret.

Encourages Honesty

People are likely to provide more genuine and honest feedback when their identity remains secret.

Encourages Honesty

People are likely to provide more genuine and honest feedback when their identity remains secret.

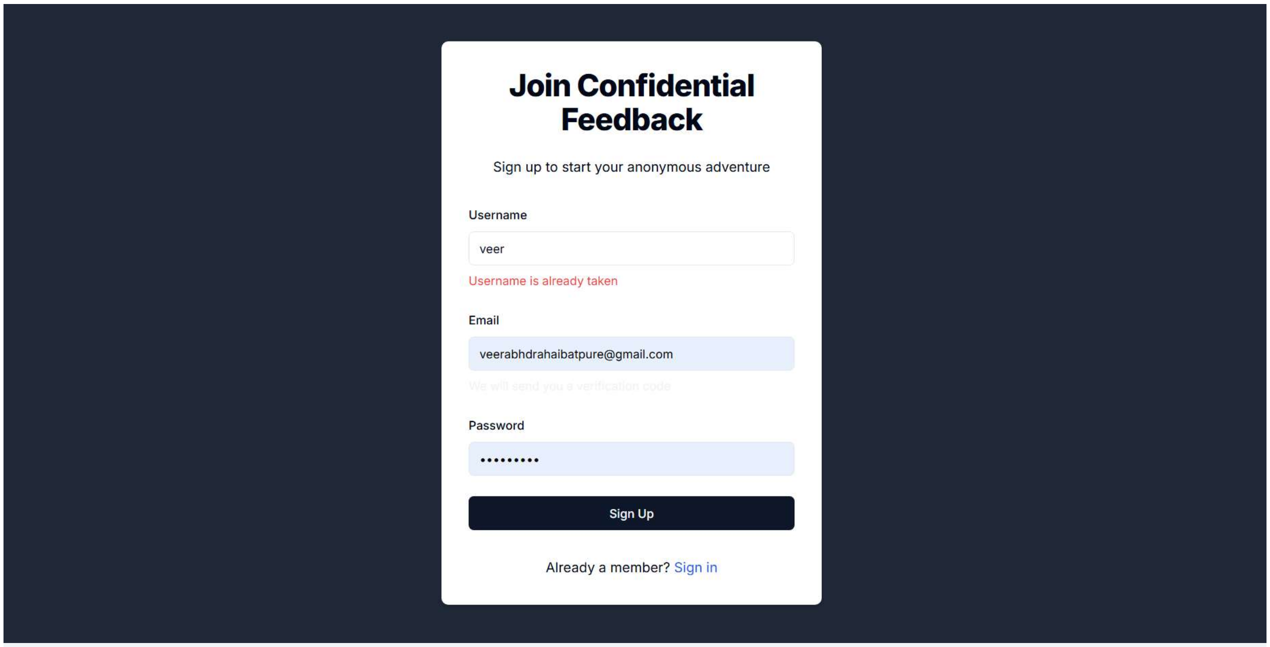
Welcome Back to Confidential Feedback

Sign in to continue your secret conversations

Email: veerabhdrahaibatpure@gmail.com

Password: •••••••••

Not a member yet? Sign up



Join Confidential Feedback

Sign up to start your anonymous adventure

Username

veer

Username is already taken

Email

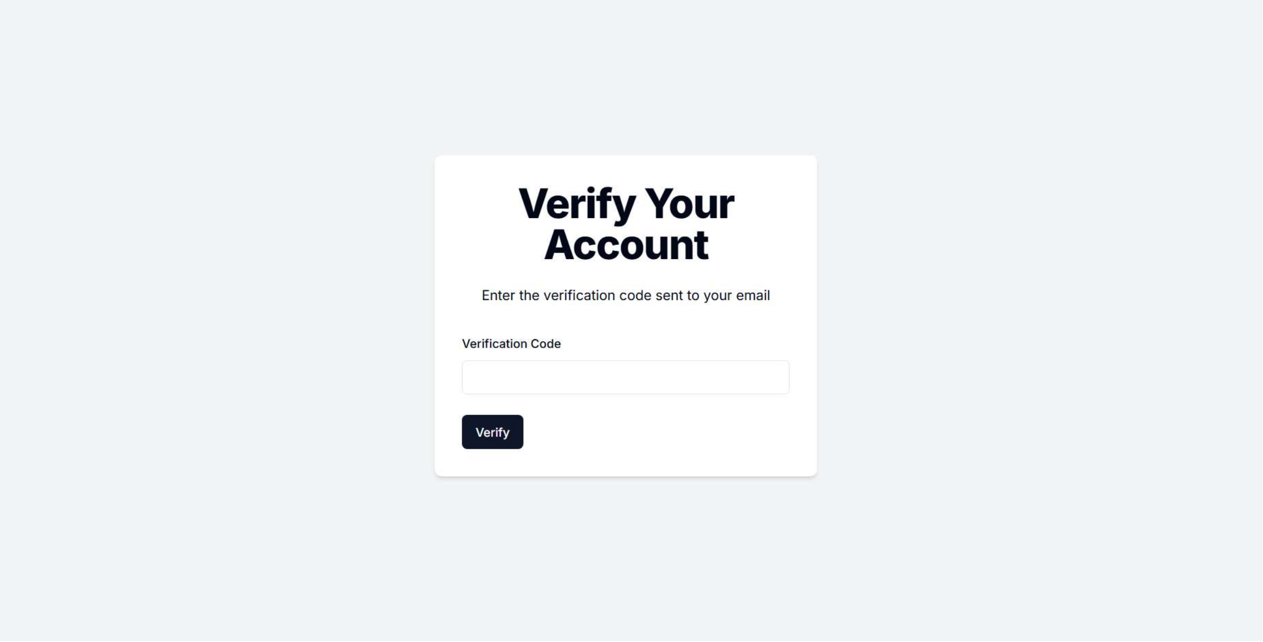
veerabhdrahaibatpure@gmail.com

We will send you a verification code

Password

•••••••••

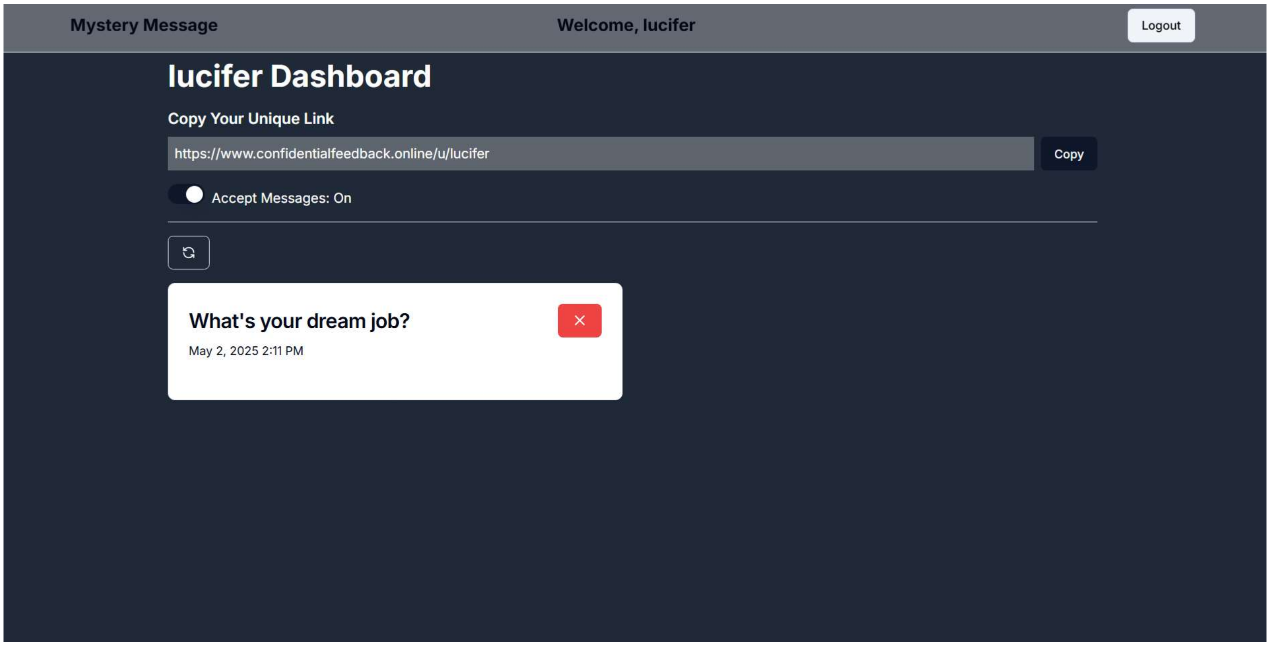
Already a member? Sign in



Verify Your Account

Enter the verification code sent to your email

Verification Code



lucifer Dashboard

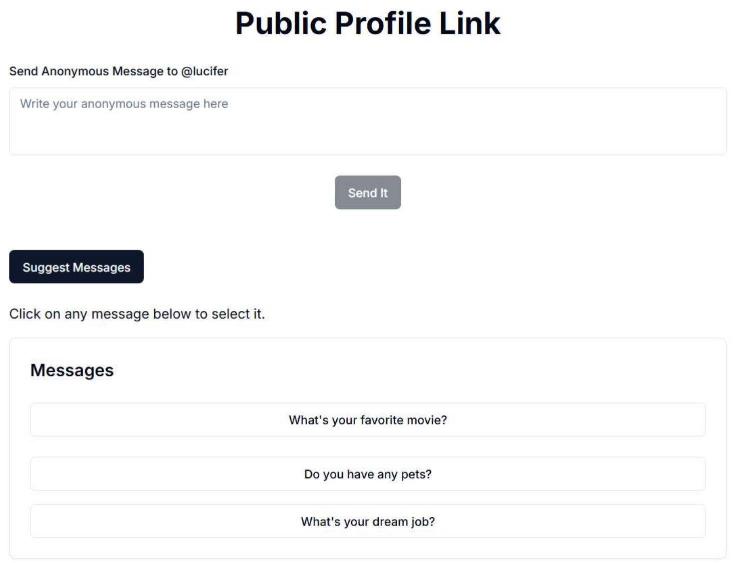
Copy Your Unique Link

https://www.confidentialfeedback.online/u/lucifer

Accept Messages: On

What's your dream job?

May 2, 2025 2:11 PM



Public Profile Link

Send Anonymous Message to @lucifer

Write your anonymous message here

Click on any message below to select it.

Messages

What's your favorite movie?

Do you have any pets?

What's your dream job?

TESTING STRATEGIES

Testing played a crucial role in ensuring the reliability, security, and usability of the *Mystery Message* web application. A combination of manual and automated testing approaches was employed to validate the core functionalities across both frontend and backend components.

1. Unit Testing

- Focused on testing individual functions and components of the **Next.js/React frontend** and **Node.js backend**.
- Ensured that isolated logic such as **form input handling**, **authentication logic**, and **MongoDB operations** behaved as expected.
- **Example:** Verifying that the OTP verification logic correctly validates user-entered tokens and handles incorrect attempts gracefully.

2. Integration Testing

- Validated the interaction between the frontend, backend APIs, and the MongoDB database.
- Ensured that **form submissions (sign-up, message, response)** were correctly processed and persisted.
- **Example:** Testing the end-to-end flow of message reception via a public link and its retrieval in the user dashboard.

3. Functional Testing

- Assessed whether the system met functional requirements like:
  - **User registration and login (NextAuth)**
  - **Email OTP verification**
  - **Viewing received messages**
  - **Public message template display**
  - **ChatGPT-based message suggestion**
- Ensured consistent and expected behavior for all user actions.
- Included validation of both successful and edge-case scenarios.

4. Validation & Error Handling Testing

- Checked that all input fields across the platform (login, signup, public message submission) were properly validated using **React Hook Form + Zod**.
- Ensured accurate error messaging for invalid inputs, duplicate accounts, expired OTPs, or missing fields.
- **Example:** Submitting a form with a missing email or mismatched password results in an appropriate frontend error and blocks submission.

5. System Testing

- Comprehensive end-to-end testing was conducted on the fully integrated system deployed via **Vercel**.
- Tested the platform on different devices and browsers to ensure cross-platform compatibility.
- Validated that the system performs well in real-world scenarios, such as multiple users submitting messages concurrently or accessing dashboards with high message volumes.

LIMITATIONS

- **No Real-Time Updates:** The message dashboard does not auto-refresh; users must manually reload to see new messages.

- **Basic Admin Controls:** Admin capabilities are limited to viewing data; no advanced analytics or user management features yet.

- **Limited Message Filtering:** There is minimal support for searching, filtering, or organizing received messages.

- **OTP Verification Delay:** Occasional email delivery delays may affect the signup experience.

- **Single Language Support:** The platform currently supports only English, limiting broader accessibility.

FUTURE SCOPE

- **Real-Time Messaging:** Integrate WebSockets or server-sent events to enable live updates for message reception.

- **Advanced Admin Panel:** Add tools for managing users, generating usage reports, and monitoring message trends.

- **AI-Powered Features:** Expand ChatGPT integration for smart auto-replies or tone suggestions.

- **Enhanced Message Security:** Add encryption and spam-filtering options for public message submissions.

- **Mobile App Version:** Develop a mobile-friendly version or a dedicated app for improved user experience.

CONCLUSION

The *Mystery Message* platform effectively fulfills its goal of offering users a simple, anonymous, and engaging way to send and receive feedback or supportive messages. It bridges the gap between anonymity and interaction by combining a clean user interface with modern technologies like Next.js, MongoDB, and OpenAI integration.

Throughout development, the project focused on building a secure, scalable, and responsive web application. The use of TypeScript, GitHub Actions, and automated deployment via Vercel ensured a robust development workflow and smooth deployment experience.

Core features such as OTP-based user verification, a personalized dashboard for viewing received messages, and a public URL system with AI-generated message suggestions were successfully implemented and tested. The platform demonstrates both frontend responsiveness and backend reliability, validated through comprehensive testing strategies.

While fully functional, the system offers several opportunities for future enhancements—such as message encryption, real-time updates, advanced admin tools, and multilingual support—to further enrich the user experience.

Overall, *Mystery Message* showcases a practical solution that blends user empathy, anonymous expression, and scalable web technology, with strong potential for future expansion.

BIBLIOGRAPHY & REFERENCES

□ **Django Documentation**
Django Software Foundation. *Django Documentation*. Retrieved from https://docs.djangoproject.com/en/stable/

□ **W3Schools**
W3Schools. *HTML, CSS, and JavaScript Tutorials*. Retrieved from https://www.w3schools.com/

□ **Bootstrap Documentation**

Bootstrap. *Bootstrap 5 Documentation*. Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/

□ **SQLite Documentation** (For database, if applicable)
SQLite Consortium. *SQLite Documentation*. Retrieved from https://www.sqlite.org/docs.html

□ **Google Fonts**
Google. *Google Fonts*. Retrieved from https://fonts.google.com/