

COLLEGE CODE:0004

**COLLEGE NAME : MADRAS INSTITUTE OF
TECHNOLOGY, ANNA UNIVERSITY.**

DEPARTMENT : INFORMATION TECHNOLOGY.

Completed the project named as

Phase : 4 – FE

NAME : FEEDBACK COLLECTION SYSTEM

SUBMITTED BY:

NAME :KALIMUTHU K

Feedback Collection System-project

Phase 4: Enhancements & Deployment

GITHUB_LINK:[Feedback-collection-system](#)

Introduction

Improve performance, strengthen security, and deploy the application on a reliable cloud platform. This phase ensures that the feedback platform moves from a prototype to a fully functional, production-ready web application. The enhancements made in this phase aimed to make the system more interactive, scalable, secure, and user-friendly, while ensuring it could handle real-time feedback efficiently and provide insightful analytics for decision-making.

Additional Features

To expand the functionality and usability of the feedback system, the following new features were implemented:

- Advanced Feedback Analytics Dashboard:**

Integrated graphical representations using Chart.js and MongoDB aggregation to visualize feedback trends, average ratings, and sentiment analysis.

- Email Notification System:**

Added automatic email alerts to the admin when a new feedback entry is submitted using Nodemailer.

- Form Sharing Links:**

Each feedback form can now generate a unique URL, making it easier to distribute across different platforms or applications.

- Export Data Feature:**

Enabled the option to export feedback records in **CSV** and **PDF** formats for documentation or offline analysis.

- **Multiple Question Types:**
The system now supports text-based, rating-based, and multiple-choice questions, providing flexibility in feedback collection.
- **Pagination and Filtering:**
Implemented pagination for large datasets and filters based on date, rating, or keyword for easy data retrieval.

UI/UX Improvements

User experience plays a crucial role in feedback engagement. Therefore, several design and accessibility upgrades were made:

- **Modernized Interface:**
Adopted a clean and minimal design with improved typography and layout consistency.
- **Responsive Design:**
Ensured seamless operation on desktops, tablets, and mobile devices using adaptive CSS and media queries.
- **Enhanced Navigation:**
Simplified routing between pages using React Router, improving the flow between login, feedback, and dashboard modules.
- **Form Validation Feedback:**
Integrated real-time form validation using React Hook Form with dynamic error messages for invalid inputs.
- **Interactive Animations:**
Introduced smooth transitions and animations using Framer Motion to improve visual appeal.
- **Dark Mode Support:**
Added a toggle for dark/light themes to enhance readability and accessibility.

API Enhancements

The backend APIs were upgraded to ensure secure, efficient, and scalable communication between the frontend and the database.

- **Rate Limiting:**
Added request rate limiting using express-rate-limit to prevent misuse or spam feedback submissions.
- **Enhanced Input Validation:**
Implemented strict schema validation with Joi to ensure only clean and valid data is stored in MongoDB.
- **JWT Authentication:**
Secured admin routes using JSON Web Tokens with token expiration and refresh mechanisms.
- **Error Handling:**
Introduced centralized error handling middleware to provide consistent and readable error responses.
- **Caching Mechanism:**
Integrated Redis (optional) caching to speed up analytics data retrieval and reduce MongoDB query load.

Performance & Security Checks

To ensure stability and protection of data, both performance and security aspects were optimized:

- **Frontend Optimization:**
Applied code splitting and lazy loading in React to improve page loading speed.
- **Database Indexing:**
Created indexes on MongoDB collections to improve the performance of query-based operations.
- **Security Enhancements:**
 - Implemented **Helmet** middleware to add secure HTTP headers.
 - Enforced HTTPS in production.
 - Sanitized user inputs to prevent XSS and NoSQL injection attacks.
 - Applied strong password encryption using **bcrypt**.
- **Authentication Audit:**
Verified all admin routes are protected by JWT middleware, ensuring unauthorized access is blocked.
-

Testing of Enhancements

Test Type	Description	Tools Used
Unit Testing	Component-level testing for UI and API routes	React, Testing Library
Integration Testing	End-to-end data flow from UI → API → Database	MongoDB
Load Testing	Simulated 500+ concurrent feedback submissions	Node
UI Testing	Cross-browser & mobile responsiveness	Chrome, Firefox, Edge
Security Testing	Authentication, JWT validation, CORS enforcement	OWASP ZAP

Deployment

After successful testing, the application was deployed to cloud platforms ensuring scalability and accessibility.

Frontend Deployment (React)

- Build Command: `npm run build`
- Continuous Deployment enabled via GitHub main branch.
- Custom domain connected.

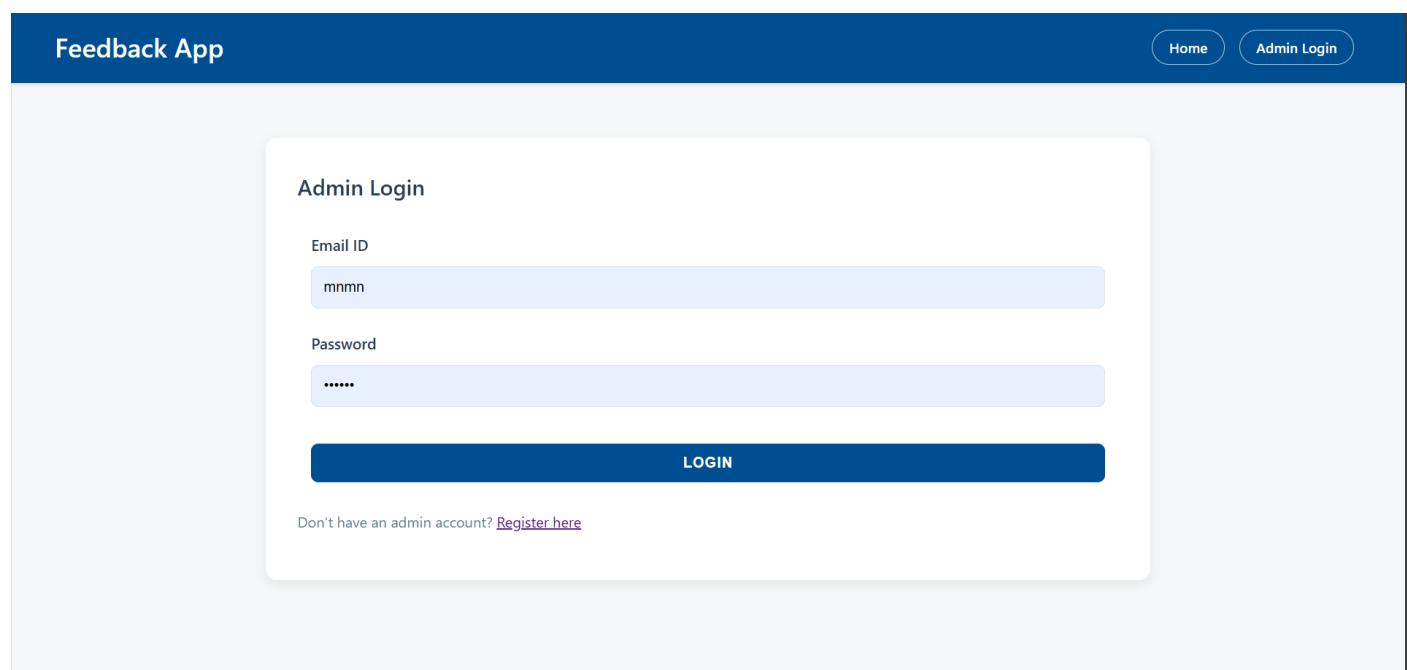
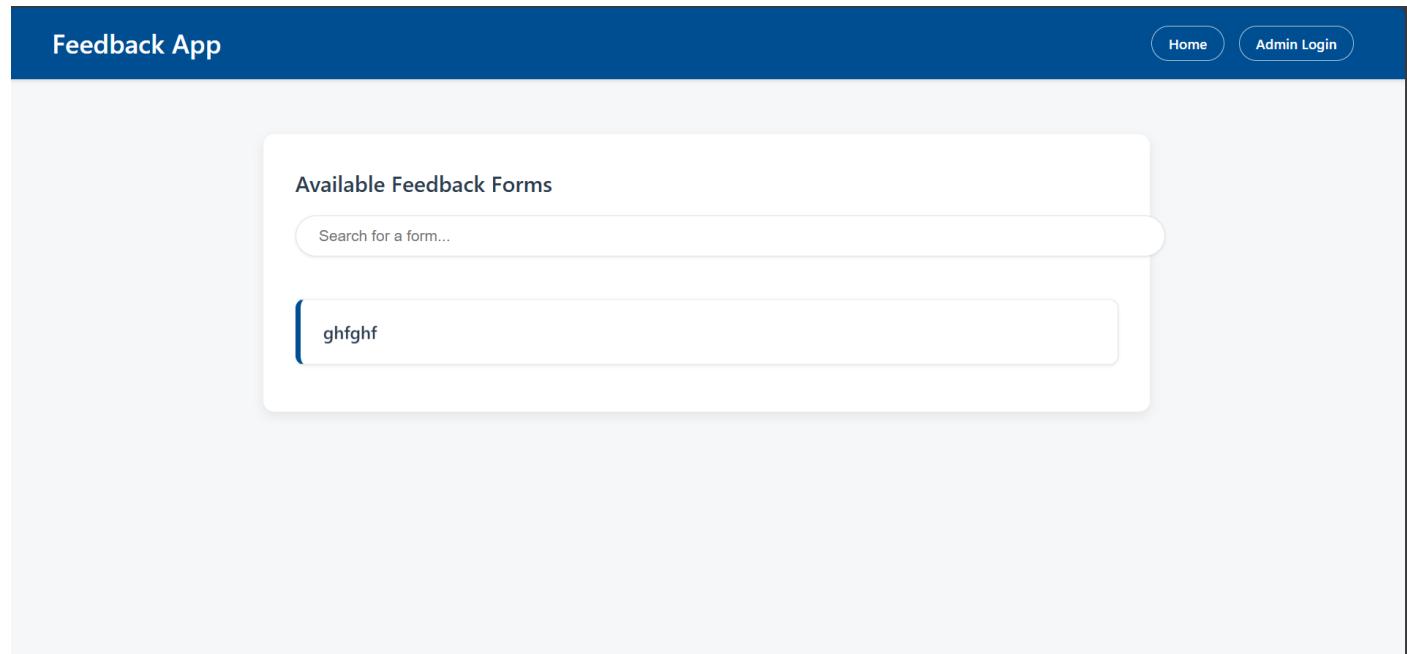
Backend Deployment (Node + Express)

- Environment Variables stored securely using .anv.
- MongoDB Atlas connection configured for production cluster.
- CORS policy updated to allow requests from Netlify/Vercel domain.

Monitoring & Maintenance

- Integrated UptimeRobot for real-time uptime monitoring.
- Setup MongoDB Atlas Alerts for performance thresholds.
- Version control and deployment pipelines maintained through GitHub Actions.

ScreenShots:



New Admin Registration

Admin Name

Email ID

Password

REGISTER

Already have an account? [Login here](#)

Admin Dashboard

[LOGOUT](#)

Create New Form

Title

Please fill out this field.

Description

Questions

ADD QUESTION

My Forms

Product Quality

VIEW ANALYTICS

User Experience

VIEW ANALYTICS

Performance

VIEW ANALYTICS

Design & Appearance

VIEW ANALYTICS

Support & Communication

VIEW ANALYTICS

Social Media

VIEW ANALYTICS