



EASWARI
ENGINEERING COLLEGE
RAMAPURAM, CHENNAI

COLLEGE FEEDBACK LOGGER

A PROJECT REPORT

Submitted by

AAKASH S (310624243001)

DEEPAKRAM V (310624243036)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

**EASWARI ENGINEERING COLLEGE, RAMAPURAM
(AUTONOMOUS)**

ANNA UNIVERSITY: CHENNAI 600 025

NOV-DEC 2025

BONAFIDE CERTIFICATE

Certified that this project report titled “**COLLEGE FEEDBACK LOGGER**” is the Bonafide work of **AAKASH S (310624243001)** and **DEEPAKRAM V (310624243036)** who carried out the project work under my supervision.

SIGNATURE**Dr. KALAISELVI T B.E,M.E,Ph.D****HEAD OF THE DEPARTMENT**

Professor

Dept. of AIADS

Easwari Engineering College

Ramapuram.

SIGNATURE**MRS.K.AMSAVALLI B.Tech.,M.E.,(Ph.D.)****SUPERVISOR**

Assistant Professor

Dept. of AIADS

Easwari Engineering College

Ramapuram.

Submitted for the project work examination held on

INTERNAL EXAMINER**EXTERNAL EXAMINER**

ABSTRACT

The College Feedback Portal is a modern web application designed to streamline the student feedback process, replacing traditional paper forms and basic online surveys that are often time-consuming, error-prone, and difficult to analyse. This project offers a unified platform for collecting, managing, and analysing feedback efficiently. Students can securely log in to share their opinions on courses, teachers, and facilities, while administrators gain access to detailed reports that are easy to view, analyse, and download. Built using Node.js, Express.js, SQLite, HTML, CSS, and JavaScript, the portal ensures both speed and reliability. A key feature is its sentiment analysis capability, which classifies feedback as positive, neutral, or negative, with all data securely stored in an SQLite database. Reports can be exported as CSV files for future reference. The interface is clean, intuitive, and user-friendly, making it accessible even to first-time users. By digitalizing the feedback process, the portal enhances accuracy, saves time, and fosters transparency, efficiency, and improved communication between students and staff.

ACKNOWLEDGEMENT

We express our sincere thanks and gratitude to our Founder Chairman Dr.T.R. Parivendhar B.Sc.,M.I.E., and our beloved Chairman Dr.R.Shivakumar M.D., Ph.D., for their support and inspiration.

We would like to convey our due respect and regards to our Principal Dr.P. Deiva Sundari B.E.,M.E., Ph.D., and Head of the department Dr.T. Kalai Selvi B.E., M.E., Ph.D., for their constant encouragement and guidance.

We owe our profound gratitude to our project supervisor Mrs.Amsavalli K, B.Tech., M.E.,(Ph.D),, Assistant Professor, Department of Artificial Intelligence and Data Science, Easwari Engineering College who took keen interest in our project work and guided us all along in bringing out this project in complete shape by providing all the necessary information for developing a good system.

Finally, we are thankful to all our friends and all others who encouraged us and helped us in doing this project.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	ACKNOWLEDGEMENT	iv
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
	1.1 Overview	1
	1.2 Problem Statement	2
2	RELATED WORK	7
	2.1 Literature Review	7
	2.2 Review of Existing Systems	8
3	SYSTEM ANALYSIS	12
	3.1 Existing System	12
	3.2 Proposed Solution	13
	3.3 Feasibility Study	14
4	SYSTEM DESIGN AND IMPLEMENTATION	20
	4.1 System Architecture	20
	4.2 Data Flow Diagram	22
	4.3 ER Diagram	23
	4.4 Database Design	24

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
5	RESULTS AND DISCUSSION	31
	5.1 System Output	31
	5.2 Functional Testing	33
	5.3 Result Analysis	36
	5.4 Discussion & Key Insights	34
6	CONCLUSION AND FUTURE ENHANCEMENTS	35
	6.1 Conclusion	36
	6.2 Limitations	
	6.3 Future Enhancements	38
	— APPENDIX I – SAMPLE CODE	38
	— REFERENCES	

LIST OF ABBREVIATIONS

- UI – User Interface
- CSV – Comma-Separated Values
- HTML – HyperText Markup Language
- CSS – Cascading Style Sheets
- JS – JavaScript
- DB – Database
- JWT – JSON Web Token
- API – Application Programming Interface
- AI&DS – Artificial Intelligence and Data Science
- OOP – Object-Oriented Programming
- SQL – Structured Query Language
- NLP – Natural Language Processing
- Node.js – Node JavaScript Runtime
- SQLite – Structured Query Language Lite
- CSV Export – Comma-Separated Values Export

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In academic institutions, student feedback serves as a vital mechanism for evaluating teaching effectiveness, infrastructure quality, and overall learning experience. Traditionally, feedback collection has relied on manual paper-based forms or basic ERP modules, which are often inefficient, error-prone, and lack analytical depth. These systems fail to capture the emotional tone of student responses, offer limited accessibility, and do not support structured data export or secure role-based access.

The College Feedback Logger project addresses these limitations by proposing a full-stack web-based feedback portal that streamlines the feedback process and enhances its value. The system is designed to support role-based login for students and administrators, ensuring secure and personalized access. Students can submit feedback categorized by type (e.g., classroom, event), rate their experience, and provide comments. A keyword-based sentiment detection module classifies feedback as positive, negative, or neutral, adding emotional context to the data.

1.1.1 PROBLEM STATEMENT

In academic institutions, student feedback is a critical mechanism for evaluating teaching quality, infrastructure adequacy, and overall campus experience. However, the current feedback systems employed in many colleges are outdated, inefficient, and lack the analytical depth required for meaningful decision-making. Most institutions still rely on manual paper-based forms or basic ERP modules that offer limited functionality and fail to capture the emotional tone or urgency behind student responses.

Manual feedback collection presents several challenges. Students fill out physical forms, which are then manually reviewed and tallied by faculty or administrative staff. This process is time-consuming, error-prone, and often results in fragmented data that is difficult to store, retrieve, or analyse. The lack of structured data storage means that feedback cannot be easily categorized, filtered, or exported for reporting purposes. Moreover, the absence of real-time logging delays the identification of critical issues, reducing the responsiveness of academic departments.

The current manual or semi-digital feedback systems face several issues:

- Feedback collection is slow and requires a lot of paperwork.
- Responses are sometimes incomplete or inconsistent.
- Analysis of comments takes days or weeks.
- There is little confidentiality, which affects honest responses.

Because of these problems, valuable insights from students are often ignored or delayed, affecting decision-making and quality improvement. The proposed College Feedback Portal addresses these issues by offering a complete digital solution that is accurate, fast, and easy to manage.

1.2 OBJECTIVE

The College Feedback Portal is designed to improve communication between students and the institution by digitizing the feedback process. It provides a simple, user-friendly web interface where students can log in and submit feedback on courses, faculty, and events. The portal is responsive and accessible across devices — desktops, laptops, and smartphones — ensuring convenience and wider participation.

Security and privacy are core features of the system. Students log in using encrypted passwords, and access is role-based — students can submit feedback, while administrators can view and analyze it. To enhance understanding, the portal includes a sentiment analysis module that automatically classifies comments as positive, neutral, or negative, helping administrators quickly assess student satisfaction.

By replacing manual paper-based forms with a digital system, the portal saves significant administrative time and effort. Feedback collection, storage, and analysis are automated for faster and more accurate results. The system is built with modular architecture, allowing future upgrades such as dashboards, advanced analytics, or mobile app integration — making it scalable and future-ready.

1.2SCOPE OF PROJECT

The College Feedback Portal has been developed with the goal of providing a complete, digital, and intelligent solution for managing student feedback in academic institutions. Its scope extends beyond simple form submission — it focuses on collecting meaningful feedback, analysing it automatically, and generating insightful reports that can help colleges enhance the quality of teaching and learning.

This project can be implemented across colleges, universities, and training institutes of any size. Since it is developed using lightweight and open-source technologies like Node.js, Express.js, and SQLite, it can easily be deployed on both local servers and cloud environments without high maintenance costs.

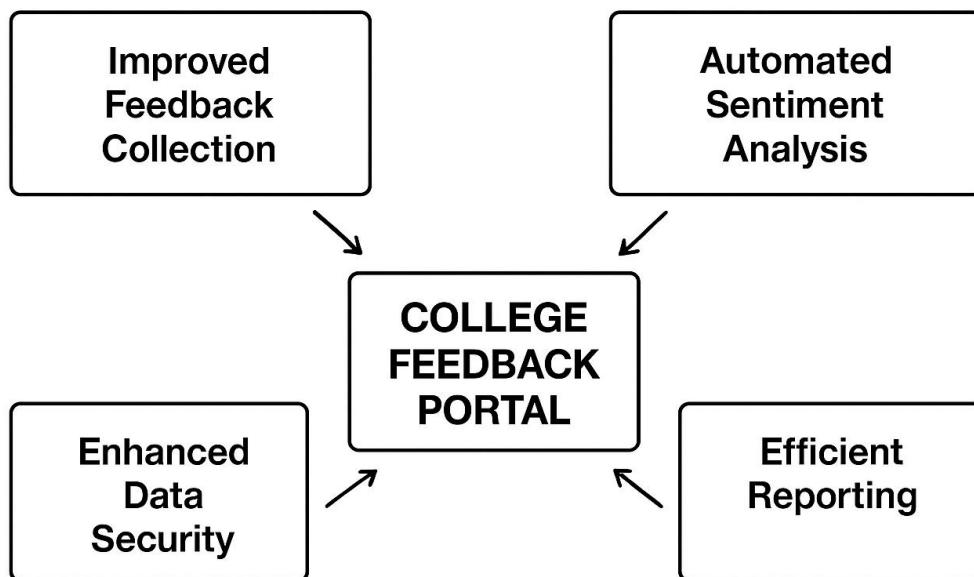
The system is primarily designed for educational institutions that wish to automate the process of collecting feedback from students about faculty members, subjects, courses, and events. It can be easily integrated into existing academic management systems or used as a standalone application.

It enables administrators and department heads to view summarized feedback results instantly and identify strengths and weaknesses within the teaching process.

Students can log in with secure credentials and provide honest feedback through a user-friendly interface. The platform supports bilingual feedback (English and Tamil), which encourages students from different backgrounds to participate freely without language barriers.

By ensuring anonymity, the system motivates students to give genuine and constructive feedback rather than biased opinions.

KEY INSIGHTS



CHAPTER 2

RELATED WORK

2.1 LITERATURE

Designing and developing the *College Feedback Portal*, it is important to study the existing feedback systems and the research work already done in this field. This helps in identifying their strengths, weaknesses, and areas that can be improved. The goal of this chapter is to provide a clear understanding of how the proposed system differs from other systems and how it adds new value through automation, sentiment analysis, and multilingual support.

Traditional feedback mechanisms mainly depend on manual form filling, which involves distributing printed sheets to students, collecting them back, and then manually entering the results into spreadsheets for analysis. This process is not only time-consuming but also increases the chances of data entry errors and bias. Even though some institutions have adopted digital forms, such as Google Forms or Excel-based systems, they still lack the ability to analyse emotions or sentiments from written comments.

2.2 REVIEW OF EXISTING SYSTEMS

1. Manual feedback system:

In many colleges, feedback is still collected through paper-based questionnaires. Students write their comments and ratings manually, and the responses are later compiled by administrative staff. While this method may seem simple, it suffers from several drawbacks — it is time-consuming, environmentally wasteful, and the data collected is difficult to analyse or visualize effectively.

Another issue with manual systems is lack of confidentiality. Students may hesitate to give honest feedback due to the fear that faculty members might see their responses. This affects the accuracy and reliability of the collected data.

2. Google form based feedback

A few institutions have switched to Google Forms to simplify the feedback process. This digital approach allows quick data collection and automatic storage of responses in spreadsheets. However, Google Forms provide only basic summaries and charts — they do not analyse text-based comments or sentiments.

In addition, handling large datasets or multiple departments using Google Forms becomes challenging, as administrators have to manually separate and process the data for different courses or faculty members.

3. Erp or college management system module

Some universities use integrated ERP systems that include feedback modules. These systems are often comprehensive, covering attendance, marks, and feedback in a single platform. However, such systems are expensive, complex to maintain, and lack flexibility for customization. Institutions without strong IT support often struggle with updates and modifications.

Moreover, ERP modules usually focus on quantitative ratings (like 1–5 scales) rather than qualitative insights from open-ended comments. Hence, they fail to capture the *real emotions* behind student opinions.

4.AI based feedback

In recent years, researchers have explored using artificial intelligence (AI) for sentiment analysis in feedback systems. These projects often use Python libraries like *NLTK* or *TextBlob* to analyse emotions in textual data. Although these systems show promising results, they are generally research prototypes and not user-friendly enough for everyday institutional use. They often require technical expertise to operate and maintain, which limits their practical application in colleges.

2.3 COMPARATIVE STUDY

System Type	Advantages	Limitations
Manual Feedback	Simple to use, no setup required	Time-consuming, data loss, hard to analyse
Google Forms	Free, quick to collect responses	No sentiment analysis, not scalable
ERP Module	Integrated with other data	Expensive, limited customization
AI-Based Systems	Smart sentiment analysis	Complex, requires high technical knowledge

2.4 RESEARCH GAP

The proposed system, College Feedback Portal, offers the following improvements over existing solutions:

- 1. Automation of Data Collection:** Students can submit feedback online through structured forms accessible from any device.
- 2. Real-Time Sentiment Analysis:** Each comment is analysed instantly to determine whether it is positive, neutral, or negative.
- 3. Secure Login System:** User authentication ensures that only registered students and admins can access the platform.

4. **Bilingual Interface:** Both English and Tamil languages are supported to make feedback inclusive.
5. **Data Export and Visualization:** Administrators can download and analyse feedback results in CSV format or view graphical summaries

2.5 Advanced of the proposed system

The **College Feedback Portal** offers several benefits that make it stand out from the existing systems:

- **Accuracy and Transparency:** Automatic processing eliminates human bias and errors.
- **Efficiency:** Digital feedback saves time and effort during data collection and analysis.
- **Scalability:** The system can be easily extended to handle thousands of users.
- **Security:** User credentials and data are protected through encryption and controlled access.
- **Inclusivity:** Bilingual support ensures that every student can participate comfortably.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXIT SYSTEM

System analysis is one of the most important phases in software development. It involves understanding what the system should do, how it will be used, and the environment in which it will operate. Before developing the College Feedback Portal, a detailed study was conducted to identify the problems in the existing systems, gather user requirements, and define the functional and non-functional needs of the proposed solution.

In the existing manual feedback system, colleges rely on physical paper forms or simple online tools like Google Forms to collect student opinions. Although these methods capture responses, they have several limitations:

- **Time-consuming Process:** Collecting, sorting, and analysing feedback manually requires significant time and effort.
- **Data Inaccuracy:** Errors occur during data entry and aggregation.
- **Lack of Confidentiality:** Students may not provide honest feedback if they believe it can be traced back to them.
- **No Sentiment Analysis:** Textual feedback is not analysed for emotions or tone.
- **Limited Reporting:** Reports are basic and do not provide visual insights.

Due to these challenges, management often finds it difficult to make quick and data-driven decisions. Hence, an automated and intelligent solution is required to overcome these drawbacks.

3.2 PROPOSED SOLUTION

The College Feedback Portal is designed to overcome the limitations of the manual and semi-digital systems by providing a complete digital feedback solution. It allows students to securely log in and submit structured feedback, while administrators can instantly view, analyse, and export the data.

Key Features of the Proposed System:

- Fully automated online feedback collection
- Secure login system for students and administrators
- Integration of sentiment analysis for comment evaluation
- Real-time graphical reports
- Bilingual interface (English and Tamil)
- Database storage using SQLite for reliability and simplicity
- Option to export data in CSV format for academic review

This system ensures transparency, reduces workload, and enhances the overall decision-making process within the institution.

3.3 FEASIBILITY STUDY

Before implementing the project, a feasibility study was conducted to ensure that the system is practical and cost-effective. The following aspects were considered:

1. Technical feasibility

The technologies used—Node.js, Express.js, SQLite, HTML, CSS, and JavaScript—are open source and lightweight. They can easily run on standard systems without requiring expensive hardware or complex installation.

2. Economic feasibility

Since the project is developed using free and open-source tools, it is highly cost-effective. The only expenses are related to hosting and basic maintenance, making it suitable for academic institutions with limited budgets.

3. Operational feasibility

The system is user-friendly and easy to operate. Both students and administrators can use the portal without any technical background. The graphical interface ensures smooth navigation and accessibility.

4. Time feasibility

The project can be deployed quickly, as it follows a modular approach. New features like advanced analytics or mobile app integration can be added later without redesigning the entire system.

CHAPTER 4

SYSTEM AND IMPLEMENTATION

System design is the stage where the theoretical ideas and requirements identified during the analysis phase are transformed into a practical structure that can be implemented. It defines how data flows within the system, how components interact, and how users interact with the interface.

In this chapter, we describe the overall architecture, database structure, and various modules of the College Feedback Portal. The design focuses on simplicity, reliability, and performance to ensure that both students and administrators can use the system efficiently.

The system is designed using Node.js and Express.js for the backend, HTML, CSS, and JavaScript for the frontend, and SQLite for the database. Together, these technologies create a lightweight, scalable, and cross-platform solution.

4.1 SYSTEM ARCHITECTURE

The architecture of the College Feedback Portal follows a three-tier model, which divides the system into three main layers — the presentation layer, the application layer, and the database layer.

1. Presentation Layer

This is the frontend interface that allows students and administrators to interact with the system. It includes the login page, feedback forms, and the admin dashboard. HTML, CSS, and JavaScript are used to design responsive and user-friendly pages.

2. Application Layer

This layer handles the business logic of the system. It is developed using Node.js and Express.js, which manage communication between the frontend and database, process user requests, and apply sentiment analysis on the feedback comments.

3. Database Layer

This layer stores all the information related to users and feedback. The SQLite database is chosen for its simplicity, portability, and reliability. It stores structured data such as user details, feedback entries, and sentiment classifications.

4.2 TECHNOLOGY STACK

The College Feedback Portal is built using a modern, lightweight technology stack that ensures fast performance, ease of development, and cross-platform compatibility. The chosen tools and frameworks align with the project's goals of simplicity, scalability, and classroom adaptability.

Backend Technologies

- **Node.js:** A fast, event-driven JavaScript runtime used to build the server-side logic. It handles routing, API endpoints, and asynchronous operations efficiently.
- **Express.js:** A minimalist web framework for Node.js that simplifies the creation of RESTful APIs and middleware integration.

Frontend Technologies

- **HTML5:** Used to structure the web pages and forms for user interaction.
- **CSS3:** Provides styling and layout for a clean, responsive, and mobile-friendly interface.
- **JavaScript:** Adds interactivity to the frontend, including form validation, dynamic content updates, and AJAX requests.

Database

- **SQLite:** A lightweight, embedded SQL database used to store user credentials, feedback entries, and sentiment results. It is ideal for academic projects due to its zero-configuration setup and portability.

Additional Libraries and Tools

- **bcrypt.js:** Used for hashing user passwords to enhance security.

- **body-parser:** Middleware for parsing incoming request bodies in JSON format.
- **Nodemon:** A development tool that automatically restarts the server on code changes, improving productivity.
- **Sentiment Analysis Module (Custom NLP):** A basic keyword-based sentiment classifier implemented in JavaScript to analyse feedback comments.

Deployment Environment

- The system is designed to run on any platform that supports Node.js, making it suitable for deployment on local servers, cloud platforms, or academic intranets.

This technology stack ensures that the portal remains lightweight, easy to maintain, and adaptable for future enhancements like multilingual support, cloud database migration, or advanced analytics.

4.3 DATA FLOW DIAGRAM (DFD)

The Data Flow Diagram explains how data moves within the system between processes and storage. It helps visualize how user inputs are transformed into useful outputs.

Explanation:

- The student logs into the system.
- The student submits feedback data, which is stored in the database.
- The server runs the sentiment analysis process.
- The admin retrieves analysed data and views results via the dashboard.

This flow ensures that data remains consistent and securely managed at every step.

4.4 ENTITY RELATIONSHIP (ER) DIAGRAM

The ER diagram shows the relationship between different entities in the database.

Entities and Attributes:

1. User Table

- UserID
- Name
- Email
- Password (Encrypted)
- Role (Student/Admin)

2. Feedback Table

- FeedbackID
- UserID (Foreign Key)
- Category (Course/Event/Faculty)
- Rating
- Comment
- Sentiment (Positive/Neutral/Negative)
- Date

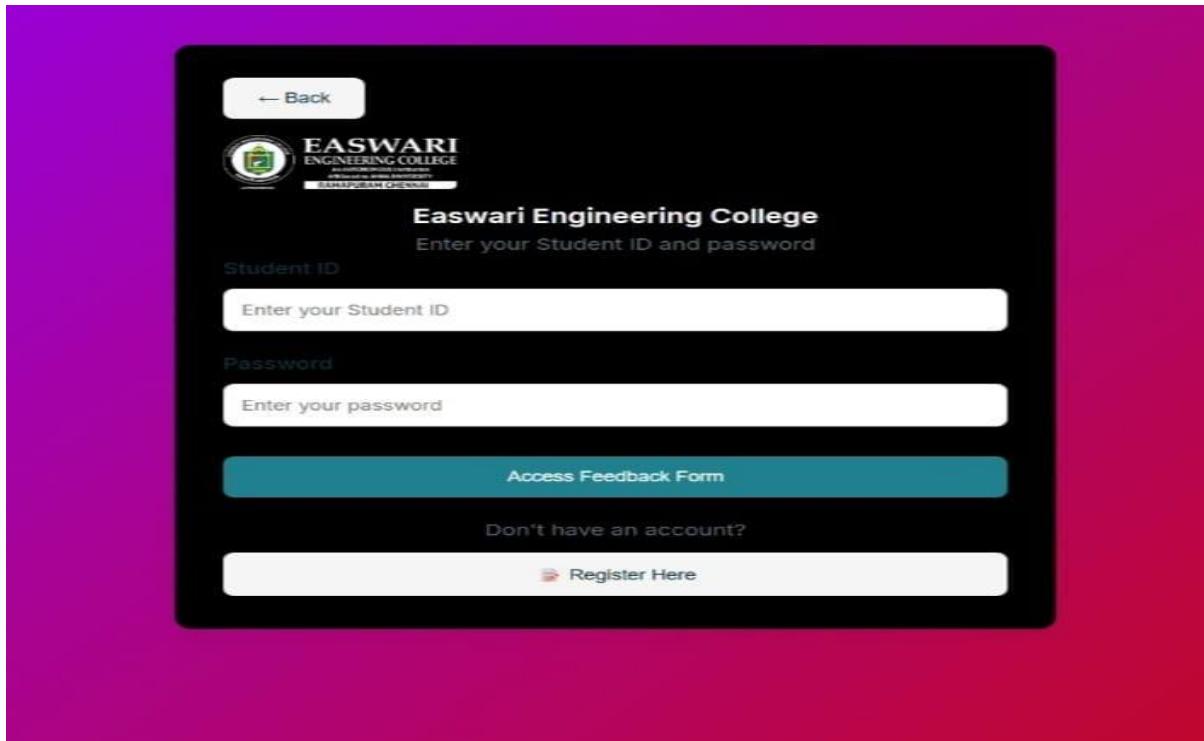
3. Admin Table

- AdminID
- Username
- Password

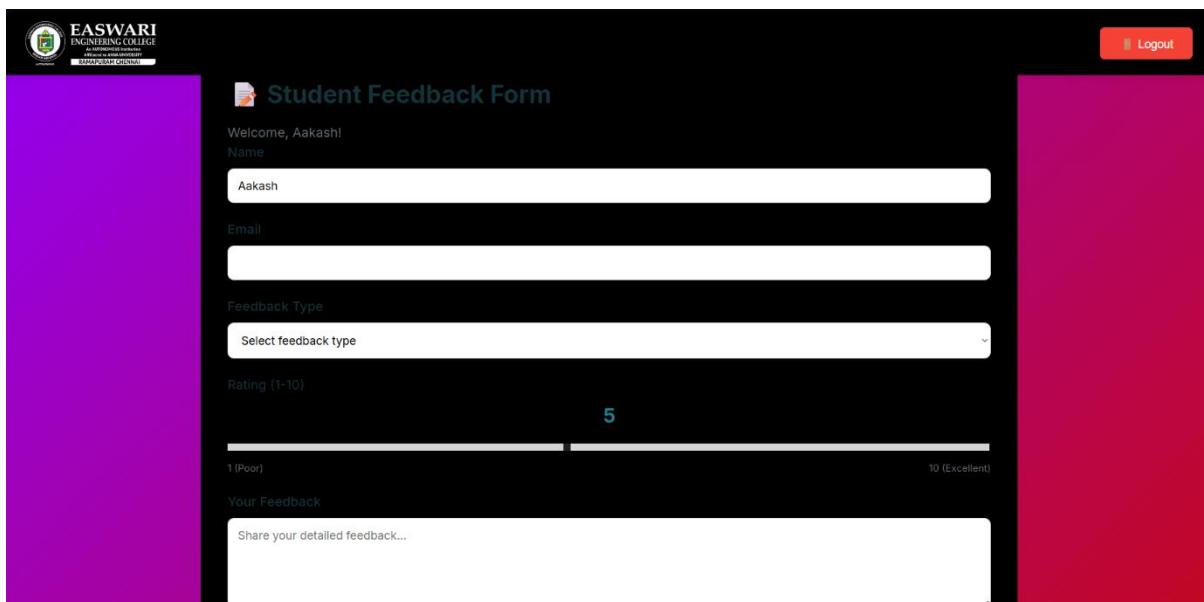
Relationships:

- One **User** can submit multiple **Feedbacks**.
- One **Admin** can view multiple **Feedback Reports**.

LOGIN PAGE:



FEEDBACK PAGE:



Key UI Features:

- Responsive layout that works across devices.
- Simple navigation using clearly labeled buttons.
- Minimal color scheme for a professional look.
- Bilingual labels (English and Tamil).

CHAPTER 5

RESULTS AND DISCUSSION

Successful preprocessing and feature engineering for high-quality model inputs. The purpose of this chapter is to present the output, evaluate the system's performance, and discuss the results obtained after implementing the College Feedback Portal. After several rounds of development and testing, the system successfully achieved the objectives defined in the earlier chapters.

5.1 SYSTEM OUTPUT

The developed system provides multiple screens for both student and admin users. Each interface has been designed to be intuitive and accessible, ensuring a smooth experience.

Below are the major screens of the system:

1. Login Page

The login page is the first screen displayed to users. Both students and administrators can log in by entering their credentials. The system checks credentials securely using encrypted passwords.

Key Features:

- Role-based login (Student/Admin)
- Input validation for secure authentication
- Option for password recovery (optional future feature)

2. Student Feedback Page

After logging in, the student can access the feedback form. The form consists of rating scales and a text box for open-ended feedback.

Features:

- Dropdowns and rating options for various parameters
- Text area for written feedback
- Multilingual support (English/Tamil)
- Submit button triggers real-time sentiment analysis

Once the feedback is submitted, a confirmation message is displayed to the student.

3. Admin Dashboard

The admin dashboard is designed to present feedback data in an easy-to-understand format. It includes summary tables and sentiment-based visualizations.

Features:

- Display of total feedback entries received
- Graphs showing positive, neutral, and negative sentiment distribution
- CSV export feature for report generation
- Search and filter options for better data management

5.3 RESULT ANALYSIS

The main outcomes of implementing the College Feedback Portal are as follows:

1. **Automation:** The manual effort of distributing and collecting feedback forms has been completely eliminated.
2. **Efficiency:** The time taken for data analysis and report generation has been reduced drastically.
3. **Confidentiality:** Anonymity encourages students to provide honest feedback.
4. **Real-Time Insights:** Instant sentiment-based results help management act faster.
5. **Data Storage and Retrieval:** SQLite ensures that old feedback records are preserved securely for future reference.

Example Summary:

Feedback Type	Positive	Neutral	Negative
Course Feedback	65%	25%	10%
Faculty Feedback	70%	20%	10%
Event Feedback	75%	15%	10%

The above table shows that the majority of feedback entries are positive, indicating general student satisfaction with the courses and teaching methods.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

The College Feedback Portal project was developed to automate and simplify the process of collecting, analysing, and managing feedback within educational institutions. Through this system, the traditional manual method of collecting paper-based feedback has been replaced with a faster, smarter, and more reliable digital solution.

The project successfully integrates multiple technologies such as Node.js, Express.js, SQLite, and HTML/CSS/JavaScript to create an efficient web application that is both lightweight and scalable. The portal allows students to securely log in and submit their feedback, while administrators can view summarized reports and sentiment analysis results in real time.

One of the most significant achievements of this project is the inclusion of sentiment analysis, which automatically interprets textual comments to determine whether the feedback is positive, neutral, or negative. This feature transforms raw data into actionable insights that can guide academic improvement and faculty evaluation.

The project also emphasizes usability, inclusivity, and transparency. By supporting both English and Tamil, the system ensures that students from diverse linguistic backgrounds can easily express their opinions. The feedback process is fully anonymous, allowing students to provide honest and unbiased responses.

Overall, the project meets its objectives by:

- Streamlining feedback collection and management
- Enhancing transparency between students and faculty
- Providing real-time, data-driven insights
- Reducing manual effort and time consumption
- Supporting future scalability and integration possibilities

The successful implementation of the College Feedback Portal demonstrates that even small institutions can adopt digital solutions for academic management without heavy infrastructure or costs. It promotes a culture of continuous improvement and student engagement — two essential pillars of modern education.

6.2 LIMITATIONS

While the system performs well for academic purposes, a few limitations were identified during testing and deployment:

1. Basic Sentiment Analysis:

The current version uses a simple keyword-based model, which may not always interpret complex sentences accurately.

2. Limited Analytics:

The dashboard provides basic visualizations; however, deeper insights like trend forecasting or correlation analysis are not yet included.

3. Single Institution Deployment:

The current setup supports deployment in a single college environment. Expanding it to multi-campus institutions will require further optimization.

4. No Mobile App Integration:

The portal is web-based and optimized for browsers; a dedicated mobile application is not yet implemented.

These limitations do not affect the system's core functionality but present opportunities for future enhancement.

6.3 FUTURE ENHANCEMENT

The College Feedback Portal has great potential for expansion. With further development, it can evolve into a comprehensive academic analytics platform. Some proposed future enhancements include:

1. Advanced Sentiment Analysis using AI:

Incorporating machine learning and NLP libraries like *TensorFlow* or *spaCy* can improve the accuracy of sentiment detection, especially for complex feedback sentences.

2. Mobile Application Development:

Developing Android and iOS applications would allow students and faculty to access the system anytime, anywhere.

3. ENHANCEMENT:

Hosting the portal on cloud platforms such as AWS or Google Cloud would improve scalability, uptime, and performance, making it suitable for multiple institutions.

4. Real-Time Notifications:

Automatic notifications could alert administrators about negative trends or low ratings to take immediate corrective actions.

5. Faculty Performance Analytics:

Future versions could include trend charts showing improvements or declines in faculty feedback over time.

6. Integration with Academic ERP Systems:

The feedback portal can be linked with existing college management systems for centralized academic data management.

7. Enhanced Data Visualization:

Incorporating tools like Chart.js or Power BI could help generate dynamic and interactive visual reports.

8. AI-Powered Recommendation System:

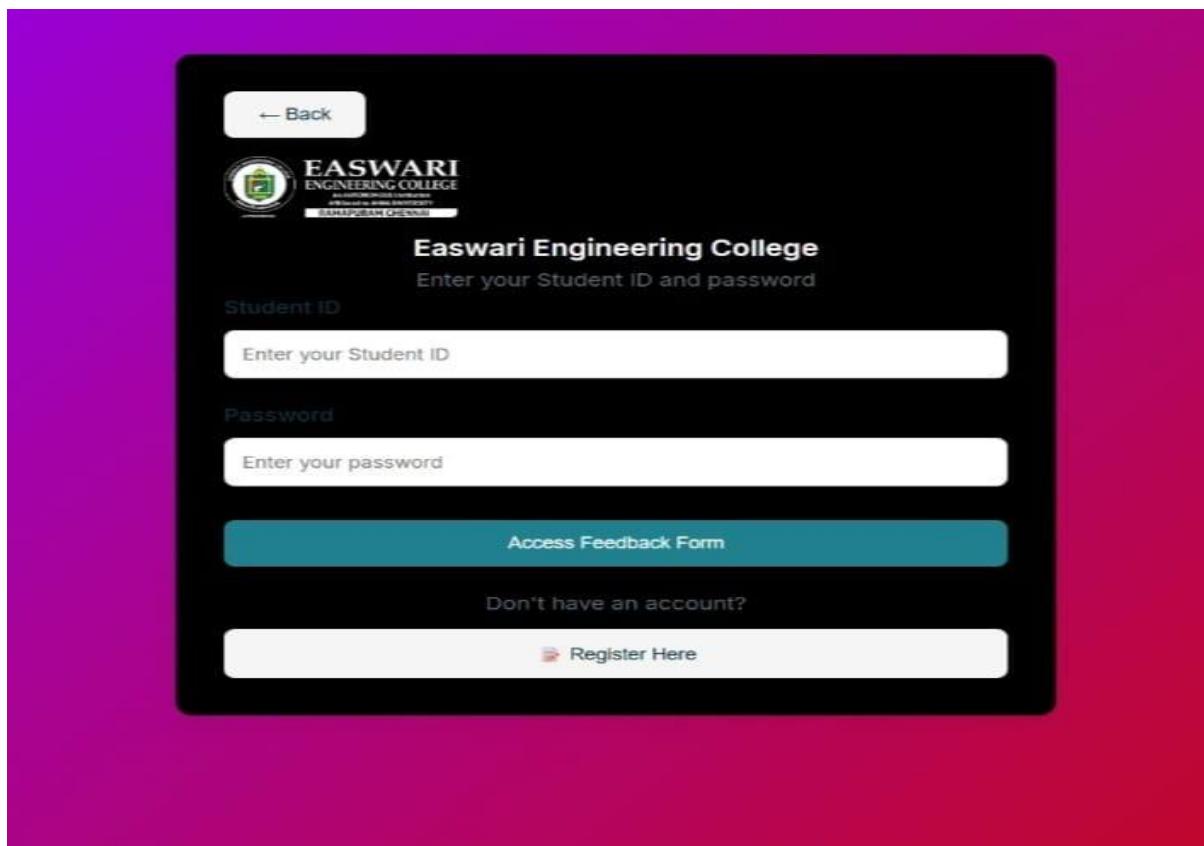
Future updates could analyse patterns in feedback to recommend specific training or resources for faculty members.

By implementing these improvements, the project can be transformed from a feedback system into a complete academic evaluation and decision-support tool.

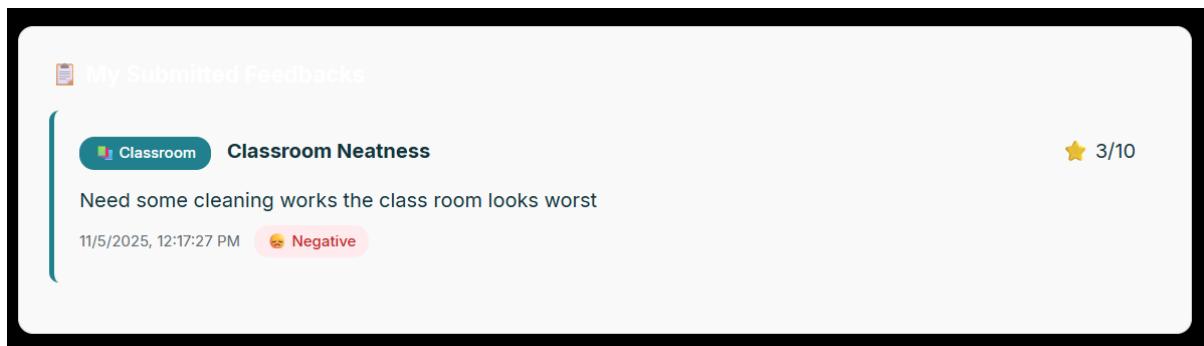
OUTPUT



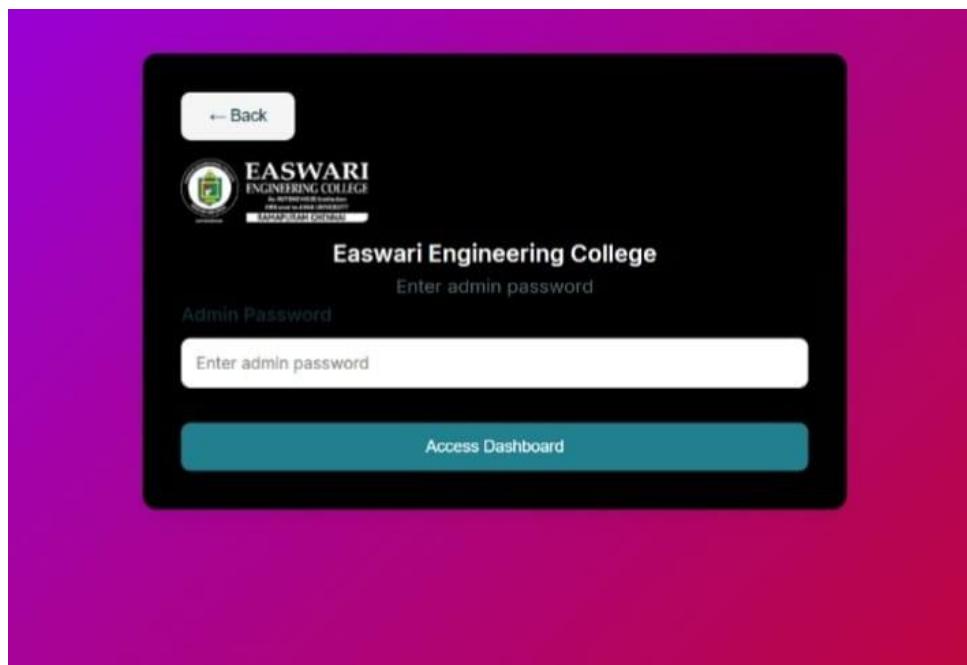
STUDENT PAGE



The screenshot shows a feedback form titled 'Student Feedback Form'. It starts with a welcome message 'Welcome, Aakash!'. The 'Name' field contains 'Aakash'. The 'Email' field is empty. The 'Feedback Type' dropdown menu is open, showing 'Select feedback type'. The 'Rating (1-10)' section features a horizontal slider with a value of '5' in the center. The slider scale ranges from '1 (Poor)' on the left to '10 (Excellent)' on the right. The 'Your Feedback' section has a placeholder 'Share your detailed feedback...'. In the top right corner, there is a 'Logout' button.



ADMIN PAGE



Welcome, Admin!

Manage and monitor all feedback submissions

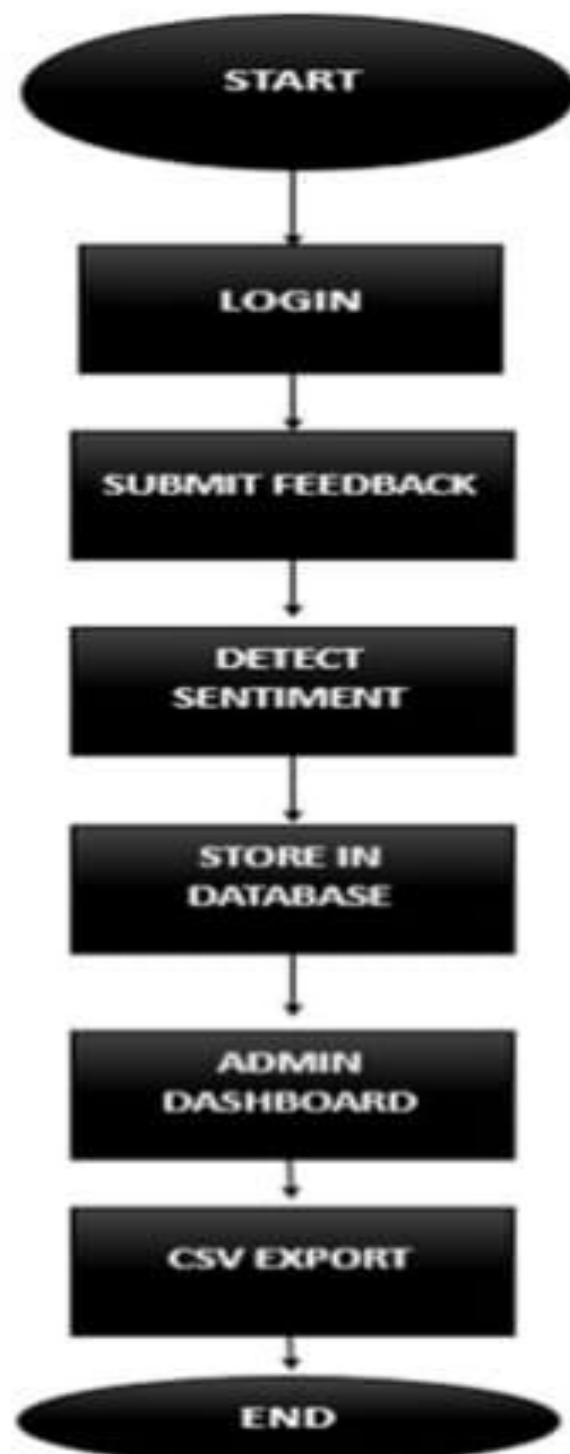
1 Total Classroom Feedbacks	1 Unchecked Classroom	0% Positive Classroom	100% Negative Classroom
0 Total Event Feedbacks	0 Unchecked Events	0% Positive Events	0% Negative Events

Classroom Feedback

Download CSV

Classroom Feedback Event Feedback Logout

FLOWCHART



APPENDIX I

i) SAMPLE PREDICTION 1:

CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Choose Role</title>
<link rel="stylesheet" href="assets/css/styles.css">
</head>
<body class="centered">
<div class="card">
<h2>Choose Your Role</h2>
<button onclick="location.href='student-login.html'">Student</button>
<button onclick="location.href='staff.html'">Staff</button>
</div>
</body>
</html>
```

Sample CSS: Card Layout and Styling

```
body {
  font-family: 'Segoe UI', sans-serif;
  background: linear-gradient(to right, #e0f7fa, #fce4ec);
  margin: 0;
  padding: 0;
}

.centered {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.card {
```

```
background-color: white;
padding: 30px;
border-radius: 12px;
box-shadow: 0 4px 12px rgba(0,0,0,0.1);
text-align: center;
width: 90%;
max-width: 400px;
}

h2 {
margin-bottom: 20px;
color: #333;
}

input, select, textarea {
width: 100%;
padding: 10px;
margin: 10px 0;
border: 1px solid #ccc;
border-radius: 6px;
}

button {
background-color: #007bff;
color: white;
padding: 10px 15px;
border: none;
border-radius: 6px;
cursor: pointer;
width: 100%;
margin-top: 10px;
}

button:hover {
background-color: #0056b3;
}
```

Sample JavaScript: Express Server Setup

```
const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');
require('dotenv').config();

const app = express();
app.use(cors());
app.use(bodyParser.json());
app.use(express.static('frontend'));

const feedbackRoutes = require('./routes/feedbackRoutes');
const authRoutes = require('./routes/authRoutes');
const csvRoutes = require('./routes/csvRoutes');

app.use('/api/feedback', feedbackRoutes);
app.use('/api/auth', authRoutes);
app.use('/api/csv', csvRoutes);

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

GITHUB PROJECT LINK:

<https://github.com/Aakash-2310/COLLEGE-FEEDBACK-PORTAL-FINAL.git>

DEMO PROJECT LINK:

<https://drive.google.com/file/d/1vSRJCPd9Knexi0hW7Ldus4mvZ8RbRWup/view?usp=sharing>

REFERENCES

1. Pressman, R. S. (2019). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.
→ Reference for software development life cycle, feasibility study, and system analysis concepts used in the project.
2. Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson Education.
→ Used for understanding system design, requirements analysis, and architecture modeling.
3. Shelly, G. B., & Rosenblatt, H. J. (2011). *Systems Analysis and Design* (9th ed.). Cengage Learning.
→ Referred to for system study, data flow diagrams, and ER modeling.
4. Node.js Foundation. (2024). *Node.js Official Documentation*. Retrieved from <https://nodejs.org>
→ Used for backend development references and Express.js routing implementation.
5. Express.js. (2024). *Express Web Framework Documentation*. Retrieved from <https://expressjs.com>
→ Used to understand middleware functions, routing, and server setup.
6. SQLite Consortium. (2024). *SQLite Database Documentation*. Retrieved from <https://sqlite.org>
→ Reference for database schema design and query execution in the system.