JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

COURSE FEEDBACK SYSTEM

A Phase 1 Report on Developing a Scalable Web Platform for Course Feedback System Using MERN Stack

[1]Ms.Prathiksha, [2]Dr. Shashidhar Kini K
[1]Student, [2]Professor & Head,
[1]Department of Master of Computer Applications,
[1]Srinivas Institute of Technology, Valachil Mangalore, Karnataka , India

Abstract : In today's academic landscape, continuous feedback from students is critical for ensuring high-quality education and fostering academic excellence. This paper presents a Phase 1 report on the development of a Course Feedback System using the MERN stack (MongoDB, Express.js, React.js, Node.js). The system provides a secure and interactive platform for students to submit course feedback, allowing institutions to analyze trends, monitor teaching effectiveness, and support curriculum improvements. Key functionalities include user authentication, feedback submission, role-based dashboards, and real-time feedback analytics. The design emphasizes scalability, usability, and secure data handling. The report outlines the planning, system architecture, and implementation methodology for creating a dynamic course feedback platform aligned with institutional quality assurance goals.

IndexTerms **- Course Feedback System, MERN stack, Web Application, Student Feedback, Educational Quality Improvement, NoSQL Database.**

I. INTRODUCTION

Course feedback collection and analysis have become vital components of academic quality assurance in modern higher education. As institutions seek to continuously improve teaching standards and student experiences, feedback systems are increasingly recognized as critical tools for gathering actionable insights, promoting transparency, and fostering a culture of responsiveness. A well-designed feedback mechanism strengthens institutional reputation, supports faculty development, and helps identify areas for curriculum enhancement. However, many institutions still rely on outdated methods like paper surveys, email forms, or basic spreadsheets—approaches that are often inefficient, prone to low participation, and fail to provide timely, actionable insights in today's digital-first world.

The rise of modern web development frameworks and the need for real-time, scalable, and user-centric systems have led to a shift in how feedback is collected and managed. This has paved the way for digital platforms such as dedicated Course Feedback Systems, which offer centralized online environments where students can submit feedback, faculty can review reports, and administrators can analyze trends to drive academic improvements. These platforms not only streamline the feedback process but also foster a culture of open communication and continuous development.

This paper presents a Phase 1 implementation of a Course Feedback System designed and developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The system is built as a full-stack web application that enables institutions to collect, manage, and analyze feedback efficiently while offering features such as secure user authentication, customizable feedback forms, real-time response tracking, analytics dashboards, and report generation. Emphasizing responsiveness, data security, and scalability, the system aims to meet the needs of students, faculty, and administrators across diverse educational settings.

The architecture of the platform is designed to support future enhancements such as advanced sentiment analysis, AI-based recommendations, mobile app integration, and API-based interoperability with existing learning management systems (LMS). The initial phase includes detailed planning, system architecture design, database schema development, frontend/backend integration, and pilot testing with users. This research focuses on creating a sustainable, user-friendly feedback system that aligns with institutional goals and student expectations. By adopting this digital solution, institutions can modernize their feedback processes, enhance academic quality, and foster a data-driven culture of continuous improvement.

II. Ease of Use

The Course Feedback System is designed with a strong emphasis on accessibility, simplicity, and responsiveness to ensure maximum participation among students, instructors, and administrators.

The system features:

- A user-friendly React.js interface with Material-UI components ensuring intuitive navigation.
- Simplified feedback forms with Likert scale ratings and optional comment fields.
- Mobile-friendly responsive design allowing feedback submission from any device.
- Dynamic role-based dashboards: Students can submit feedback, instructors can view course-wise reports, and administrators can monitor overall feedback trends.
- Real-time feedback acknowledgment and submission tracking.

The use of JWT-based secure login ensures a seamless and secure authentication process. Real-time analytics dashboards provide instructors and administrators with immediate insights into feedback metrics, enabling timely interventions.

1. Prepare Your Paper Before Styling

Before finalizing the development and documentation of the Course Feedback System, special emphasis was placed on completeness, clarity, and system reliability. Initial requirements were gathered through stakeholder interviews with students, faculty members, and administrators to identify the necessary functionalities and user expectations. During the system design phase, frontend wireframes were created to simulate user navigation across feedback forms and dashboards, while the backend architecture and database schema were optimized using MongoDB to ensure scalable and efficient feedback storage. Security measures such as password hashing, JWT-based authentication, and role-based access control were incorporated to protect user data. Modular and maintainable frontend components were developed using React.js. Agile methodologies, with iterative weekly sprints, focused on continuous frontend and backend integration, API testing, and user interface refinements. To ensure the documentation quality, the final report was structured following IEEE-style research paper guidelines, covering key sections such as Introduction, Literature Review, Technology Stack, Methodology, Results, and Bibliography. Multiple rounds of revisions were conducted to enhance clarity, maintain technical accuracy, and ensure adherence to academic publication standards.

2. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even if they have been defined in the abstract. Do not use abbreviations in section titles unless unavoidable.

In this paper, the following abbreviations and acronyms are used:

- **API:** Application Programming Interface
- **MERN:** MongoDB, Express.js, React.js, Node.js
- **JWT:** JSON Web Token
- **UI:** User Interface
- **UX:** User Experience
- **DB:** Database
- **UAT:** User Acceptance Testing
- **CRUD:** Create, Read, Update, Delete
- **HTTPS:** HyperText Transfer Protocol Secure
- **CSS:** Cascading Style Sheets
- **HTML**: HyperText Markup Language
- **IDE**: Integrated Development Environment

III. RESEARCH METHODOLOGY

This section outlines the methodology adopted to develop a full-stack web platform for fostering engagement between students and academic institutions through course feedback collection. It encompasses the user population, data sources, theoretical framework, and the development and testing tools used throughout the initial phase of the Course Feedback System implementation.

3.1 Study Population and Sample

The system is designed to serve students from undergraduate and postgraduate programs across multiple academic departments. User requirements were gathered through surveys and interviews with student representatives and faculty members. Three primary user roles were identified: Students, Instructors, and Administrators. The inclusion criteria required active enrollment and valid institutional credentials, while feedback responses were anonymized to encourage honest participation.

3.2 Data and Sources of Data

Primary data was collected through structured online feedback forms integrated within the web platform. The datasets collected consist of quantitative ratings across multiple course evaluation parameters (such as content quality, teaching effectiveness, and fairness in evaluation) and qualitative open-ended comments offering specific suggestions or concerns. Additionally, each submission is automatically timestamped to track temporal feedback trends. Course metadata, including course identifiers, department codes, and instructor information, was imported from the institution's academic records. Student identification details were linked securely through authenticated login processes while maintaining user anonymity in the feedback data. All datasets were stored in a MongoDB database and managed through Mongoose schemas, ensuring robust data validation, data typing, and the enforcement of schema consistency. Data indexing and optimization techniques were also applied to enhance query performance during feedback retrieval and reporting.

3.3 Theoretical framework

This study adopts a multi-theoretical design perspective integrating frameworks that prioritize both technology development and user adoption. System Design Theory guides the platform's architecture by enforcing modular development and separation of concerns across backend, frontend, and database layers. The Technology Acceptance Model (TAM) is used to evaluate how students and faculty accept digital platforms based on perceived usefulness and ease of use. Additionally, Continuous Improvement Theory supports the iterative refinement of teaching and learning processes based on regular feedback analysis. The theoretical foundation also considers user motivation factors such as course engagement, transparency of evaluation, and academic satisfaction. These perspectives inform both technical decisions (e.g., modular feedback modules, responsive dashboards) and usability enhancements (e.g., simplified navigation, quick access to reports). By incorporating user behavior theories and system architecture principles, the project aims to balance performance efficiency with intuitive, inclusive engagement mechanisms suited to varied academic users

3.4 .Development Tools and Web Technologies

This phase integrates both conventional web technologies and modern development tools to implement and evaluate a scalable and responsive alumni portal. The core objective is to develop a functional full-stack web application enabling alumni engagement through networking, content sharing, and institutional collaboration. Frontend development used React.js with Material-UI components, integrated with Axios for API communication. Backend development utilized Node.js and Express.js for secure RESTful APIs, incorporating middleware for validation and authentication. MongoDB was used as the NoSQL database with collections managed via Mongoose. Authentication was implemented using JWT tokens. Unit and integration testing were conducted using Jest, and the system was deployed on cloud platforms like Render and Netlify with MongoDB Atlas as the database backend.

3.4.1 Descriptive Statistics

Initial analysis of stakeholder data for the Course Feedback System involved frequency counts, response distributions, and statistical summaries of key feedback areas (e.g., most common course concerns, preferred feedback submission methods, and areas for improvement). These insights were instrumental in prioritizing features during the system design, ensuring that the platform meets the needs of students, faculty, and administrators while aligning with institutional goals for enhancing academic quality.

3.4.2 Exploratory Feature Mapping

Wireframing tools like Figma and Miro were used to map feature relationships and navigation flows for the Course Feedback System. Information architecture was visualized through tree diagrams, representing page hierarchies, content organization, and user access roles. This approach ensured that the platform's design was intuitive, user-friendly, and aligned with the needs of students, faculty, and administrators, while also supporting efficient navigation and streamlined feedback processes..

3.4.3 Backend Architecture

Node.js and Express.js were used to build the RESTful API endpoints for the Course Feedback System developed with the MERN stack. These endpoints handled essential functionalities such as user authentication, profile creation, feedback submission, and report generation..

3.4.4 Database Design

MongoDB was chosen for its flexible document structure, allowing easy scalability and data management. Collections for users, events, jobs, and messages were created, with Mongoose schemas ensuring data consistency across the Course Feedback System.

.

3.4.5 Frontend Framework

The frontend of the Course Feedback System was built with React.js, utilizing component-based rendering and hooks for dynamic interactivity. Key UI features included search filters, event registration cards, profile edit modules, and admin dashboards for efficient feedback management

3.4.6 .API Integration

Axios-based API calls ensured reliable communication between frontend and backend. JSON Web Tokens (JWT) were used to secure protected routes and prevent unauthorized access.

3.4.7 Testing Strategies

Unit tests using Jest were written for API endpoints in the Course Feedback System to ensure functionality and reliability. Manual testing with simulated users verified role-based content visibility and navigation, while integration testing ensured data consistency across pages.

3.4.8 Deployment

The Course Feedback System was deployed with the backend on platforms like **Heroku** or **Render**, and the frontend hosted on **Netlify**. **MongoDB Atlas** was used for cloud database hosting, ensuring scalability and reliability.

IV. RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Table 4.1: Descriptive Statics

Metric	Value (Sample)
Number of Feedback Entries	450
Average Content Quality Rating	4.2 / 5
Average Teaching Method Rating	4.0 / 5
Average Evaluation Fairness Rating	4.1 / 5

Table 4.1 The Course Feedback System received a total of 450 feedback entries, providing a substantial dataset for analysis. The average content quality rating was 4.2 out of 5, indicating that students generally felt the course content was well-organized and valuable. The average teaching method rating stood at 4.0 out of 5, suggesting a positive but slightly lower perception of teaching methods compared to content quality. Additionally, the average evaluation fairness rating was 4.1 out of 5, reflecting students' belief that the course evaluations were generally fair. These metrics highlight areas of strength, such as content quality, while also indicating opportunities for improvement in teaching methods and evaluation fairness.

V. ACKNOWLEDGMENT

The author wishes to express sincere gratitude to the Project Guide and Head of the Department of Computer Applications, Dr. Shashidhar Kini K, for his invaluable guidance, insightful suggestions, and continuous encouragement throughout the development of the Course Feedback System. Special thanks are extended to the Principal, Dr. Shrinivasa Mayya D, for providing a supportive academic environment that made this project possible within the institution. The author also acknowledges the management of Srinivas Institute of Technology for their direct and indirect support during the course of this research.

Appreciation is due to all faculty members and non-teaching staff of the MCA department for their timely assistance, technical advice, and encouragement throughout the project phases. Lastly, the author is deeply thankful to parents, friends, and peers whose unwavering motivation and moral support were instrumental in completing this phase of the work successfully.

REFERENCES

[1] Brad Traversy, *"MERN Stack Front To Back: Full Stack React, Redux & Node.js"*, Udemy, 2020.

[2] MongoDB Documentation, *"MongoDB Manual"*, Available at: https://www.mongodb.com/docs/

[3] Express.js Documentation, *"Express - Node.js web application framework"*, Available at: https://expressjs.com/

[4] React Documentation, *"Getting Started – React"*, Meta (Facebook), Available at: https://react.dev/

[5] Node.js Documentation, *"Node.js v18.x Documentation"*, Available at: https://nodejs.org/en/docs/

[6] Netlify Docs, *"Deploying React Applications on Netlify"*, Available at: https://docs.netlify.com/

[7] Heroku Dev Center, *"Deploying Node.js Apps on Heroku"*, Available at: https://devcenter.heroku.com/articles/deploying-nodejs

[8] MongoDB University, *"Mongoose for MongoDB"*, Available at: https://mongoosejs.com/docs/

[9] Figma Help Center, *"Getting Started with Figma"*, Available at: https://help.figma.com/

[10] Postman Docs, *"Postman API Platform Documentation"*, Available at: https://learning.postman.com/