

---

# ONLINE STUDENT FEEDBACK SYSTEM

Dr P.Suresh

Associate Professor, Department of Computer Science & Engineering, Anurag Engineering  
College, kodada, suryapeta, Telangana.

---

Submitted: 19-04-2025

Accepted: 16-05-2025

Published: 23-05-2025

---

## Abstract

In traditional educational institutions, collecting feedback from students has largely relied on outdated manual processes, such as paper forms or unstructured digital tools. These methods are inefficient, prone to manipulation or errors, and lack the ability to analyze data at scale. They also compromise the privacy of student opinions, leading to biased or dishonest feedback. The absence of role-based control, secure data storage, and automated reporting makes the feedback process cumbersome and disconnected from the academic improvement cycle.

This project proposes a secure and scalable Online Student Feedback System (OSFS) that allows students to provide structured, anonymous feedback through a web-based platform. The system is developed using PHP, HTML, CSS, and MySQL, and is deployed using XAMPP on a localhost or LAN. It includes role-based modules for students, faculty, and administrators, enabling one-time feedback submission, automatic report generation, and feedback visualization. The project aims not only to improve institutional feedback workflows but also to give student developers practical experience in software design, database modeling, and secure web application development.

*This is an open access article under the creative commons license*  
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



---

## 1. INTRODUCTION:

Student feedback serves as an essential component in evaluating academic performance, teaching methodologies, and institutional policies. Without a reliable mechanism to capture this feedback, colleges miss out on critical insights that could lead to better classroom experiences and improved educational outcomes. Educational accreditation bodies and quality control departments also rely on this data to make informed decisions.

In most Indian universities and colleges, feedback collection is still done using printed forms or basic Google Forms. While these may appear convenient, they lack essential features such as user authentication, response validation, data privacy, and automatic analytics. This often results in incomplete, biased, or manipulated responses. In addition, these methods require extensive manual work in compiling and analyzing feedback, making the process slow and ineffective.

The Online Student Feedback System (OSFS) was conceptualized and developed to address these limitations. It offers a centralized platform where students log in securely, submit structured feedback for each subject/faculty, and ensures only one submission per course. The feedback data is stored in a secure MySQL database and analyzed by the admin module to generate reports. The system ensures student anonymity while giving institutions better control over the feedback process and decision-making.

---

## 2. LITERATURE SURVEY:

Before designing Online Student Feedback System, several existing systems and research papers were analyzed. Each had its strengths and weaknesses, which helped shape the requirements of our project.

### Comparative Table:

S.No	Author / System	Method / Tool Used	Key Findings / Limitations
1	Sharma et al. (2022)	Paper-based manual forms	Low engagement, poor privacy, manual compilation required
2	Google Forms in Colleges	Cloud forms, no login	No identity control, duplicate entries possible, lacks visualization
3	Ravi et al. (2023)	Mobile feedback app	Good UX, but needs internet and Android device
4	Existing Internal Tools	Excel sheets & forms	Data scattered, no real-time reports, limited access
5	Proposed OSFS (2025)	Web app with PHP/MySQL	Role-based access, anonymous, local/LAN compatible

### Analysis:

The most common feedback tools used in colleges today either rely on printed forms or basic cloud-based forms (like Google Forms or Microsoft Forms). Although these tools are free and easy to use, they lack features like login-based access, feedback frequency control, and structured reporting. Mobile-based apps provide better usability but often require high maintenance, backend APIs, and constant internet access.

From these limitations, it became clear that the ideal system must:

- Work **offline or on college LAN**
- Enforce **one-time feedback per course**
- Allow **anonymous** yet **authenticated** access
- Store feedback in a **relational database** for easy reporting
- Have **modular code** for future improvements

These points directly shaped the architecture and goals of OSFS.

## 3. EXISTING SYSTEM:

In the majority of colleges today, feedback collection still follows either a fully manual or semi-digital approach. The manual system involves printed paper forms, which are distributed in classrooms and later collected for analysis. In some cases, institutions use online tools like Google Forms, but these are mostly used without any login validation or data protection mechanisms.

### **Key Limitations of Existing Systems:**

- **No Authentication:** Anyone with the form link can submit feedback. There's no way to verify if it's a real student.
- **Duplicate Entries:** Students can submit multiple responses, distorting the data.
- **Lack of Anonymity:** Manual forms may accidentally reveal the identity of students, leading to dishonest feedback.
- **Manual Data Handling:** Feedback needs to be manually entered into Excel or typed, which is error-prone and slow.
- **Limited Access Control:** Faculty members can sometimes access raw feedback before it's reviewed.
- **Scalability Problems:** Difficult to use at department or institution level.

### **Real-World Example:**

A common situation in most colleges is the use of end-semester paper-based feedback forms. These forms are later handed to the admin office, where data entry clerks manually type each line into Excel — a process that may take days and introduces human error.

These outdated systems fail to capture the full value of student feedback and discourage students from participating honestly. Hence, a need for a secure, digital, structured, and easy-to-manage system arises.

## **4. PROPOSED SYSTEM:**

The **Online Student Feedback System (OSFS)** is a web-based platform developed to overcome the inefficiencies of current feedback methods. It focuses on providing a secure, role-based environment where students, faculty, and administrators each interact with the system differently.

### **Key Objectives:**

- Make feedback **anonymous but accountable** (one-time per student)
- Enable **real-time digital access** for reports
- Allow admins to **analyze trends** and improve faculty performance
- Reduce paper usage and manual effort

### **System Roles & Functionalities:**

#### **1. Student Module:**

- Login using student credentials
- Submit feedback only once per subject/faculty
- Rate on 5–10 structured questions (e.g., subject clarity, teaching style)
- Add optional comments
- Feedback is stored anonymously in DB

#### **2. Faculty Module:**

- Login securely
- View summary of feedback submitted for their subjects
- No access to individual comments or student identity

#### **3. Admin Module:**

- Add faculty, subjects, and students
- Assign faculty to subjects
- Generate feedback reports (charts, tables)
- Export/download reports as PDF or Excel (future upgrade)

### Tech Stack:

- **Frontend:** HTML5, CSS3, JS (optional enhancements later)
- **Backend:** PHP 7+ with secure session handling
- **Database:** MySQL with relational tables
- **Hosting:** XAMPP (localhost), or LAN/IP-based access

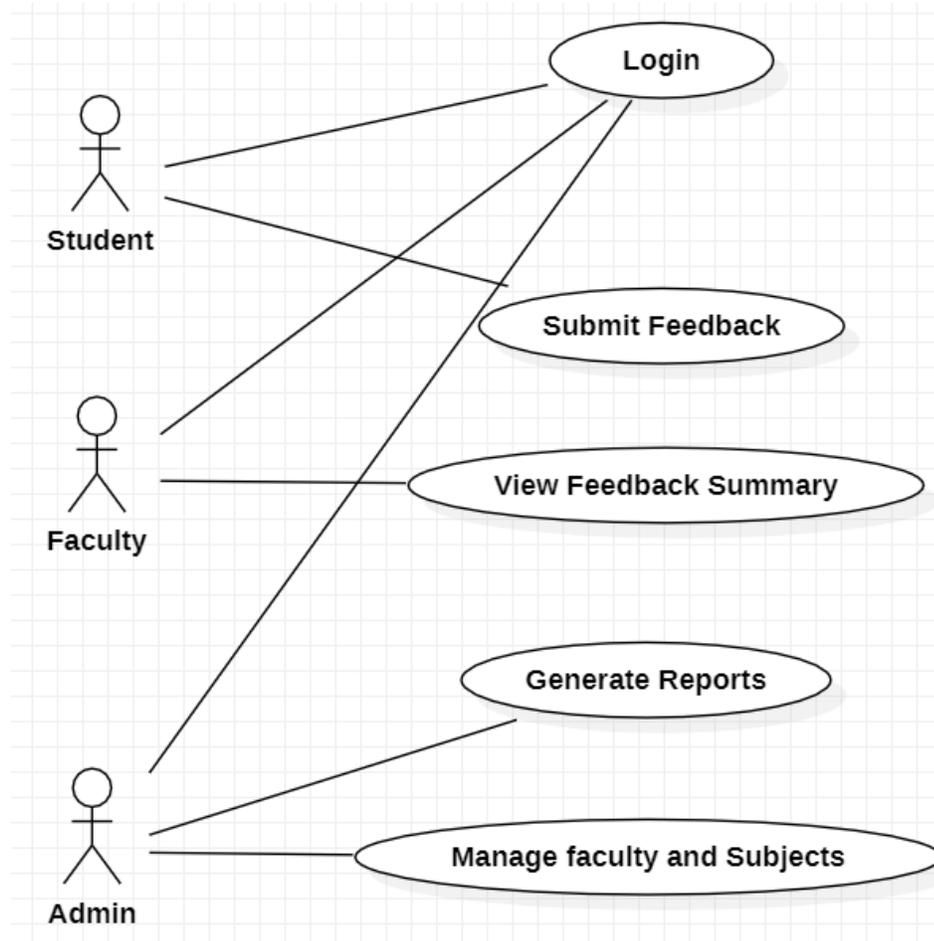


Figure 1: Use Case Diagram

## 5. SYSTEM DESIGN

The design of OSFS focuses on **modularity, security, and role-based functionality**. It follows a standard web application architecture with separation of concerns (frontend, backend, and database layers).

### a) Logical Architecture

#### Layers:

1. **User Interface (UI):** HTML forms + CSS, role-based pages (Student, Faculty, Admin)
2. **Controller Layer (PHP):** Receives form inputs, validates them, executes DB queries
3. **Database Layer (MySQL):** Normalized tables:
  - students
  - faculty

- feedback
- subjects
- users (login credentials)

**Login Flow:**

- User enters credentials
- Session is created
- User is routed to their dashboard (role-based)

**Feedback Flow:**

- Student → Fill form → Submit → DB entry → Disable further submission

**b) Physical Architecture**

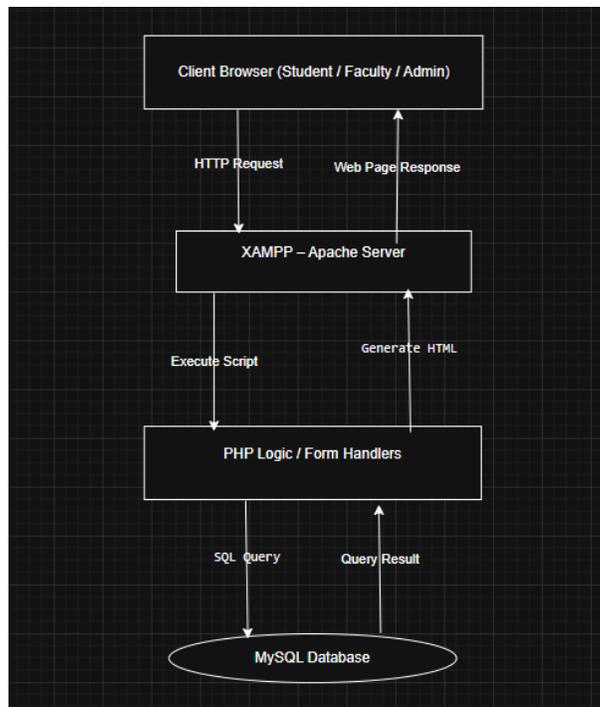
Browser (Client) → Apache Server (PHP) → MySQL Database → Reports / UI Rendered

- Hosted using XAMPP
- Can be deployed on **localhost** (personal PC) or **LAN IP** (college lab use)
- Accessible via `http://localhost/student_feedback/` or `http://192.168.X.X/student_feedback/`

**c) Security & Access Control**

- Password-protected login (can be hashed for higher security)
- Session-based user access
- Students can't submit feedback more than once
- SQL inputs are sanitized using PHP
- Faculty cannot edit or see individual response

*The architecture diagram below was created using draw.io, a free and open-source diagram tool.*



**Figure 3: System Architecture Diagram**

## 6. IMPLEMENTATION:

### Frontend (UI):

- Separate login forms for **Student**, **Faculty**, and **Admin**
- Feedback form: radio buttons (1–5 rating), optional comment box
- Admin dashboard: table of feedback entries, links to graphs

### Backend Logic (PHP):

- Login validation using sessions
- Check if student has already submitted feedback (flag in DB)
- Insert feedback into DB only once
- Redirect users based on role

### Database Tables:

- students(student\_id, name, course\_id, email, password)
- faculty(faculty\_id, name, department)
- subjects(subject\_id, name, faculty\_id)
- feedback(feedback\_id, student\_id, subject\_id, rating1, rating2, ..., remarks)
- admin(admin\_id, username, password)

### Functionality Summary:

Module	Feature
Student	Login, Submit Feedback, Logout
Faculty	Login, View Summary Report
Admin	Manage data, Generate Reports

*The diagram below was created using StarUML, a modern UML modeling tool.*

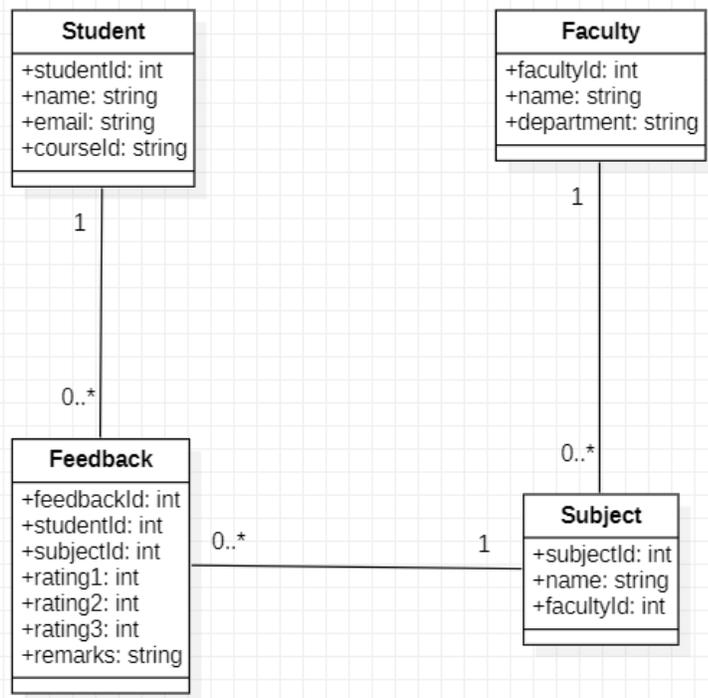


Figure 2: ER Diagram (Class Based)

## 7. RESULTS & DISCUSSION

The Online Student Feedback System was tested locally using XAMPP on a Windows-based environment and also on a local LAN in a lab setting. A sample database was populated with mock students, faculty, and subjects to test core functionalities.

### Functional Results:

- **Login Validation:** Each user (student/faculty/admin) could only access their specific dashboard based on session variables.
- **One-Time Feedback Enforcement:** Students were blocked from resubmitting feedback for the same subject/faculty after the first submission.
- **Data Entry:** All ratings and comments were accurately stored in the feedback table with timestamp and unique IDs.
- **Faculty Reports:** Faculty could see summarized feedback (average rating per question, count of entries) for their assigned subjects.
- **Admin Reports:** Admins could filter by faculty or subject and download raw tables for further analysis.
- **Feedback Integrity:** SQL injection attempts were blocked using input sanitization, and no duplicate data entries were possible due to constraints.

### Discussion:

The system proves to be scalable and effective for small to medium-sized institutions. It ensures that:

- Students feel safe giving honest feedback
- Admins avoid long hours of manual data compilation
- Faculty receives timely, data-backed insights

Future testing with larger datasets (100+ students across 10 subjects) will validate performance and latency. The architecture supports LAN-wide access but can be extended to web-hosting for college-wide deployment.

## 8. CONCLUSION AND FUTURE SCOPE

### Conclusion:

The Online Student Feedback System offers a complete digital solution to the outdated and error-prone process of manual feedback collection. It automates and secures the feedback cycle through:

- Role-based access
- Anonymous structured responses
- One-time feedback submission enforcement
- Organized data storage
- Report generation

The project not only solved a real academic need but also gave the development team practical experience in:

- Database normalization
- Session management in PHP
- CRUD operations
- System architecture planning

The use of XAMPP ensures easy deployment in any college lab without internet dependency, making it ideal for academic environments.

### Future Scope:

To enhance its usability and power, the system can be improved in the following ways:

1. **Sentiment Analysis:** Use Python NLP tools to analyze comments for positive/negative trends.
2. **Email Notifications:** Send summary feedback reports to faculty via automated emails.
3. **PDF Report Export:** Allow admin to download individual faculty reports in PDF format.
4. **Mobile App Integration:** Build React Native app to submit feedback via phones.
5. **Graphical Dashboards:** Use Chart.js to visualize trends over time.
6. **Multilingual Interface:** Support regional languages to improve accessibility.

## 9. REFERENCES

1. Sharma, P. et al. (2022). "Modernizing Student Feedback with Digital Platforms." International Journal of Advanced Research in Computer Science (IJARCS).
2. Mozilla Developer Network. (2023). "HTML, CSS, and JavaScript Documentation." Available at: <https://developer.mozilla.org>
3. MySQL. (2023). "MySQL 8.0 Reference Manual." Available at: <https://dev.mysql.com/doc>
4. PHP Group. (2023). "PHP Manual." Available at: <https://www.php.net/manual>
5. Diagrams.net. (2023). "draw.io – Online Diagram Tool." Available at: <https://draw.io>
6. StarUML. (2023). "StarUML – Modern UML Tool." Available at: <https://staruml.io>