

Student Feedback Management System



**BAKLIWAL  
FOUNDATION**

**College of Arts, Commerce & Science**

A

*Project Report*

*On*

Submitted in partial fulfillment of the requirement for the award of degree of

**Bachelor of Computer Application (BCA)**

of

**Kavikulaguru Kalidas Sanskrit University's**

*Submitted by*

**Onkar Rajaram Raising**

Aayush Anil Gurav

Dhammadeep Harshboh Shirsath

*Under the guidance of*

**Prof. Divya Patil.**



Kavikulaguru Kalidas Sanskrit Vishwavidyalaya's

**Bakliwal Foundation College of Arts, Commerce & Science**

Vashi.

**BATCH: 2022-2025**



Kavikulaguru Kalidas Sanskrit Vishwavidyalaya's

**Bakliwal Foundation College of Arts, Commerce & Science**

Vashi.

**CERTIFICATE**

This is to certify that the project entitled **Student Feedback System** undertaken at the PCP Center: Bakliwal Foundation College of Arts, Commerce & Science, Vashi, New Mumbai by **Mr.Onkar Rajaram Raising** holding **PRN no: 2022018100094802**, studying **Bachelor of Computer Application** Semester – VI has been satisfactorily completed as prescribed by the Kavikulaguru Kalidas Sanskrit Vishwavidyalaya, during the year 2024-2025.

**Project In-charge**

**Co-Ordinator**

**External Examiner**

**Internal Examiner**

**Principal**

## Declaration

I hereby declare that the project **Student Feedback System** is the result of my own efforts and has been developed as a part of my academic curriculum. This project has been carried out with the objective of creating a user-friendly and efficient online shopping platform that addresses the evolving needs of customers and seller.

The contents of this report reflect the conceptual framework and core functionalities of the web application. While every effort has been made to ensure the accuracy and functionality of the system, this project may still be subject to further refinement and enhancement based on future feedback and technological advancements.

This project has been completed under the guidance of **Prof. Divya Patil**, whose insights and support have been invaluable throughout the development process. I take full responsibility for the content of this report and the originality of the work presented.

**Onkar Rajaram Raising**

(Signature & Date)

## Acknowledgment

I would like to express my sincere gratitude to all those who have supported and guided me throughout the development of the project "**Student Feedback System.**"

Firstly, I extend my heartfelt thanks to my faculty guide, **Prof. Divya Patil** for their invaluable guidance, continuous support, and constructive feedback, which helped me shape this project from start to finish. I am also thankful to my institution, **Bakliwal Foundation College of Arts, Commerce & Science**, for providing the resources and environment necessary for learning and development.

I am deeply grateful to my friends and family for their constant encouragement and motivation during every phase of the project. Their support played a crucial role in helping me overcome challenges and stay focused.

I also thank respected faculty members – **Prof. Mohammad Umar Shaikh, H.O.D Prof. Sneha Shashikant Lokhande Prof. Kalyani Akshay Kulkarni Principle Dr. Sharad Kumar Shah** for their kind support and help throughout the project. I appreciate all the online communities and resources that contributed valuable insights and knowledge, aiding me in the successful completion of this academic endeavor.

Finally I express my deep regards to all of those who stretch their helping in the execution of my project.

## Table of Contents

| <b>Sr No.</b> | <b>Title</b>                                       | <b>Page No.</b> |
|---------------|--|-----------------|
| <b>1</b>      | <b>Abstract</b>                                    | <b>1</b>        |
| 1.1           | Introduction Of Project                            |                 |
| 1.2           | About Client                                       |                 |
|               |  |                 |
| <b>2</b>      | <b>Literature Survey</b>                           | <b>2</b>        |
|               |  |                 |
| <b>3</b>      | <b>Acquisition of Knowledge</b>                    | <b>3 - 4</b>    |
|               |  |                 |
| <b>4</b>      | <b>Domain Knowledge</b>                            | <b>5</b>        |
| 4.1           | Introduction                                       |                 |
| 4.2           | Advantages of Domain Knowledge                     |                 |
|               |  |                 |
| <b>5</b>      | <b>System Study</b>                                | <b>6</b>        |
| 5.2           | Benefits of proposed system                        |                 |
|               |  |                 |
| <b>6</b>      | <b>Problem Definition &amp; Scope of Project</b>   | <b>7</b>        |
| 6.1           | Objective  |                 |
|               |  |                 |
| <b>7</b>      | <b>Requirement Analysis</b>                        | <b>8 – 9</b>    |
| 7.1           | Analysis   |                 |
| 7.2           | Feasibility Study                                  |                 |
| 7.3           | Hardware & Software Requirements                   |                 |
|               |  |                 |
| <b>8</b>      | <b>Estimation &amp; Planning</b>                   | <b>10</b>       |
| 8.1           | Gantt Chart  |                 |
|               |  |                 |
| <b>9</b>      | <b>Methodology</b>                                 | <b>11 - 13</b>  |
| 9.1           | Spiral Model                                       |                 |
| 9.2           | Objectives of Spiral Model and its advantages      |                 |
|               |  |                 |
| <b>10</b>     | <b>Operating Tools and Development Environment</b> | <b>14</b>       |
|               |  |                 |
| <b>11</b>     | <b>Designing</b>                                   | <b>15 – 19</b>  |
| 11.1          | Use-case Diagram                                   |                 |
| 11.2          | Class Diagram                                      |                 |
| 11.3          | Activity Diagram                                   |                 |
| 11.4          | Sequence diagram                                   |                 |
|               |  |                 |
| <b>12</b>     | <b>Implementation</b>                              | <b>20 – 25</b>  |
| 12.1          | Screenshots of projects                            |                 |

|           |                                      |           |
|-----------|--------------------------------------|-----------|
|           |                                      |           |
| <b>13</b> | <b>Testing</b>                       | <b>26</b> |
| 13.1      | Unit Testing                         |           |
| 13.2      | System Testing                       |           |
| 13.3      | Operational Testing                  |           |
| 13.4      | Maintenance                          |           |
|           |                                      |           |
| <b>14</b> | <b>Limitations &amp; Enhancement</b> | <b>27</b> |
|           |                                      |           |
| <b>15</b> | <b>Bibliography</b>                  | <b>28</b> |
|           |                                      |           |

## 1. ABSTRACT

### 1.1 Introduction

The Student Feedback Management System (SFMS) is a web-based solution developed to simplify and digitize the feedback collection process in educational institutions. The traditional method of collecting feedback through pen-and-paper forms is not only time-consuming but also inefficient in terms of data analysis and storage.

This project introduces an advanced and interactive platform where:

- ❖ Students can anonymously evaluate teachers based on various academic and behavioral parameters through a structured questionnaire.
- ❖ Teachers can view analyzed feedback in real-time, helping them reflect and improve their teaching methodologies.
- ❖ Admins can manage the feedback cycle, assign teachers to subjects, analyze overall performance, and generate academic quality reports.

This system ensures:

- ❖ Confidential feedback submission to encourage honest responses.
- ❖ Role-based access control for Admins, Teachers, and Students.
- ❖ Dynamic charts and graphs to visualize performance.
- ❖ Scalable architecture allowing use across various departments.

It bridges the communication gap between students and faculty and supports continual improvement in academic standards using data-driven decision-making.

### 1.2 About the Client

This project was developed as part of the BCA Final Year curriculum under the Mentorship of Prof. Mohammed Umar Shaikh and the academic supervision of Bakliwal Foundation College of Arts, Commerce & Science, Vashi, affiliated with Kavikulaguru Kalidas Sanskrit Vishwavidyalaya.

## 2. LITERATURE SURVEY

Student feedback systems are crucial in improving the quality of education, but many institutions still rely on outdated methods. Research reveals:

- ❖ Paper-based feedback is:
  - Prone to manipulation or data loss.
  - Time-consuming to analyze.
  - Not confidential, reducing student participation.
- ❖ Digital solutions such as Google Forms or basic survey tools:
  - Lack advanced analytics or real-time data visualization.
  - Do not support role-based control or secure authentication.
  - Cannot integrate directly with academic systems for data mapping (teachers-subjects).

Existing Academic Platforms (like Moodle or ERP):

- ❖ Are too complex and over-engineered for simple feedback needs.
  - ❖ Do not allow detailed visual analysis specific to individual faculty.
- The Proposed SFMS overcomes these issues:
- ❖ Built from the ground up using PHP, MySQL, Tailwind CSS, and Chart.js.
  - ❖ Provides a dedicated Admin panel for monitoring feedback statistics.
  - ❖ Ensures easy scalability and user-friendliness.
  - ❖ Offers real-time results, filters by semester, teacher, subject, and CSV export.



### 3. ACQUISITION OF KNOWLEDGE

This project helped the team gain hands-on experience with real-world software engineering practices and tools across all stages of the Software Development Life Cycle (SDLC).

Key Areas of Learning:

#### Frontend Development

- ❖ **HTML5** for semantic layout and structure.
- ❖ **Tailwind CSS** for utility-first styling and responsiveness.
- ❖ **JavaScript** for dynamic elements like charts, filters, modals, and form validation.
- ❖ **Chart.js** for rendering real-time, animated pie charts and bar graphs for feedback analysis.

#### Backend Development

- ❖ **PHP** (with PDO):
  - Secure database interaction using prepared statements.
  - Implementing login sessions and role-based routing.
  - Server-side form handling for feedback submissions, teacher additions, profile updates.
- ❖ Authentication System:
  - Role-specific login (Admin, Teacher, Student).
  - Session-based control for security.

#### Database Management (**MySQL**)

- ❖ Designing normalized relational tables:
  - users, teachers, subjects, teacher\_subjects, feedback, admins.
- ❖ Writing complex SQL joins for feedback analysis.
- ❖ Maintaining data integrity and relationships between courses, users, and feedback records.

#### Practical DevOps Skills

- ❖ File organization in a modular structure (includes, admin, teacher, assets, etc.).
- ❖ Deployment on local **XAMPP** server.
- ❖ Handling image uploads securely using **PHP** and database references (profile\_image column).

#### Real-World Exposure

- ❖ Emulating actual academic workflows.
- ❖ Addressing practical user experience challenges (mobile-friendliness, animations, filters).

- ❖ Developing admin tools for efficient management and troubleshooting.

## 4. DOMAIN KNOWLEDGE

### 4.1 Introduction

The Student Feedback Management System (SFMS) belongs to the domain of Academic Management Systems. It specifically addresses the quality assurance practices in educational institutions by enabling systematic feedback collection, analysis, and reporting.

### 4.2 Key Domain Concepts

#### 4.2.1. Academic Evaluation Systems

- ❖ Evaluation of teaching quality based on:
  - Communication skills
  - Knowledge delivery
  - Class engagement
  - Subject clarity
- ❖ Typically collected via semester-end feedback forms.

#### 4.2.2. Educational Roles and Access

- ❖ Admin:
  - Controls subject-teacher assignments.
  - Views analytics and manages feedback data.
- ❖ Teachers:
  - Access feedback summaries.
  - Modify personal profile and photo.
- ❖ Students:
  - Submit anonymous, one-time feedback per subject-teacher pair.

#### 4.2.3. Feedback Analysis

- ❖ Scores calculated per question (Q1–Q10).
- ❖ Averages are plotted on interactive pie charts using Chart.js.
- ❖ Admins can filter data by semester, subject, and teacher.

#### 4.2.4. Digital Transformation in Academia

- ❖ Replaces paper forms with digital dashboards.
- ❖ Ensures faster analysis, reduced manual errors, and real-time access.
- ❖ Offers cross-platform accessibility via browsers.

## 5. SYSTEM STUDY

### 5.1 Overview of Existing System

- ❖ Traditional feedback mechanisms are paper-based, leading to:
  - Manual compilation of scores.
  - Loss of confidentiality.
  - Administrative delays in analysis.

### 5.2 Benefits of Proposed SFMS

- ❖ Role-Based Dashboards
  - Custom views for Admin, Teacher, and Student.
  - Reduces complexity and increases usability.
- ❖ Visual Feedback Analytics
  - Pie charts for each question (Q1–Q10) to visualize performance.
  - Clickable analytics on Admin Dashboard.
- ❖ Database-Driven Architecture
  - Real-time storage and retrieval of feedback.
  - Admins can filter or export data as needed.
- ❖ Security and Efficiency
  - Secure login via PHP sessions.
  - PDO prevents SQL injection.
- ❖ Teacher Profile Management
  - Teachers can update name, contact, and profile image.
- ❖ Responsive UI
  - Tailwind CSS ensures smooth mobile and desktop experience.
- ❖ Time-saving
  - Immediate access to analytical reports without manual tabulation.

## 6. PROBLEM DEFINITION & SCOPE OF PROJECT

### 6.1 Problem Definition

In most institutions, student feedback is still collected using offline forms, which:

- ❖ Discourages candid feedback due to lack of anonymity.
- ❖ Involves tedious manual work for result compilation.
- ❖ Delays important academic decisions based on feedback.

There is no centralized digital solution that allows:

- ❖ Role-specific access and permissions.
- ❖ Dynamic feedback questions with analytical visualization.
- ❖ Real-time filtering and export.

### 6.2 Scope of Project

The project aims to design and develop a responsive, scalable, and secure feedback platform where:

- ❖ Admins can create teacher-subject mappings and monitor analytics.
- ❖ Teachers can view feedback and manage profiles.
- ❖ Students can anonymously submit feedback for each assigned teacher.

### 6.3 Key Modules

- ❖ Authentication Module (Multi-role login)
- ❖ Admin Panel (Dashboard, teacher/subject management, feedback analytics)
- ❖ Teacher Panel (Profile view, feedback analytics)
- ❖ Student Feedback Page (Questions Q1–Q10, teacher selection, feedback submission)
- ❖ Database Module (6+ tables, fully normalized)

## 7. REQUIREMENT ANALYSIS

### 7.1 Analysis

#### ➤ User Requirements

- ❖ Students must be able to:
  - Log in securely.
  - View their assigned subjects and respective teachers.
  - Submit one feedback per subject-teacher.
- ❖ Teachers should be able to:
  - Log in and view feedback summary.
  - Update personal info and profile image.
- ❖ Admins must:
  - Manage users, subjects, and assignments.
  - Analyze feedback visually and in tabular form.
  - Export feedback to CSV.
  - Delete or reset feedback if needed.

#### ➤ Functional Requirements

- ❖ Login system with role-based redirection.
- ❖ Form to collect 10-question ratings per teacher.
- ❖ Analytics dashboard using Chart.js.
- ❖ Export and delete functions for feedback.

#### ➤ Non-Functional Requirements

- ❖ Responsive UI for all screen sizes.
- ❖ Modular code structure (PHP includes).
- ❖ Secure session handling using PHP.

### 7.2 Feasibility Study

#### Technical Feasibility

- ❖ PHP 8, MySQL, Chart.js—all lightweight and open-source.
- ❖ Compatible with any browser and OS.
- ❖ Runs locally on XAMPP/WAMP or deploys to shared

#### hosting. Operational Feasibility

- ❖ Simple UI ensures low learning curve.
- ❖ Teachers/Admins trained in <10 minutes.
- ❖ Data is centralized, secure, and easy to back up.

#### Economic Feasibility

- ❖ Free software stack (PHP, MySQL, Chart.js, Tailwind).
- ❖ No recurring licensing costs.

- ❖ Maintenance possible by in-house tech teams.

### **7.3 Hardware and Software Requirements**

#### Hardware

- ❖ Minimum: i3 Processor, 2 GB RAM, 500 MB free storage.
- ❖ Recommended: i5/i7, 4 GB

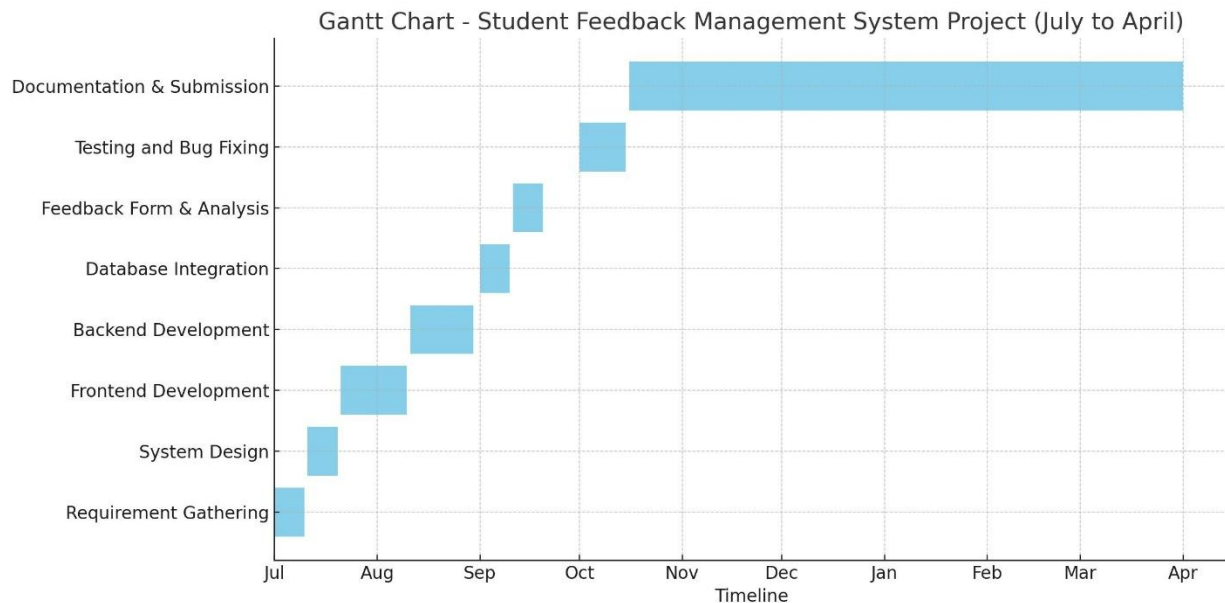
#### RAM. Software

- ❖ Backend: PHP 8.x, MySQL 5.7+
- ❖ Frontend: HTML5, Tailwind CSS, JS
- ❖ Server: XAMPP/WAMP/LAMP
- ❖ Browser: Chrome/Firefox (latest)

## 8. ESTIMATION AND PLANNING

To ensure the successful execution of the project within a defined timeframe, a structured plan was prepared using the Gantt chart model.

| <u>Phase</u>                          | <u>Description</u>  |
|---------------------------------------|---|
| <b>Requirement Gathering</b>          | <b>Collected data on feedback mechanisms from institutions and mentors.</b>       |
| <b>System Design</b>                  | <b>Created UML diagrams and ER schema.</b>  |
| <b>Frontend Development</b>           | <b>Developed HTML and Tailwind-based UI.</b>                                      |
| <b>Backend Development</b>            | <b>Implemented PHP backend, user roles, form processing, and CRUD operations.</b> |
| <b>Database Integration</b>           | <b>Connected PHP to MySQL using PDO.</b>  |
| <b>Feedback Form &amp; Analysis</b>   | <b>Created 10-question form, chart visualization using Chart.js.</b>              |
| <b>Testing and Bug Fixing</b>         | <b>Unit &amp; system testing, performance checks.</b>                             |
| <b>Documentation &amp; Submission</b> | <b>Black book preparation and code repository management.</b>                     |





## 9.METHODOLOGY

### 9.1 Spiral Model

The Spiral Life Cycle Model is a type of iterative software development model which is generally implemented in high risk projects. It was first proposed by Boehm. In this system development method, we combine the features of both, waterfall model and prototype model. In Spiral model we can arrange all the activities in the form of a spiral

### 9.2 Objective of Spiral Model

Each loop in a spiral represents a development phase and we can have any number of loops according to the project.

#### **To determine the objectives, alternatives and constraints**

We try to understand the product objectives, alternatives in design and constraints imposed because of cost, technology, schedule, etc.

#### **Risk analysis and evaluation of alternatives.**

Here we try to find which other approaches can be implemented in order to fulfill the identified constraints. Operational issues are addressed here. Risk mitigation is in focus in this phase. And evaluation of all these factors determines future action.

#### **Execution of that phase of development.**

In this phase we develop the planned product. Testing is also done. In order to do development, waterfall or incremental approach can be implemented.

#### **Planning the next phase.**

Here we review the progress and judge it considering all parameters. Issues which need to be resolved are identified in this phase and necessary steps are taken. Subsequent loops of spiral model involve similar phases. Analysis and engineering efforts are applied in this model. Large, expensive or complicated projects use this type of life cycle. If at any point of time one feels the risk involved in the project is a lot more than anticipated, one can abort it. Reviews at different phases can be done by an in-house person or by an external client.

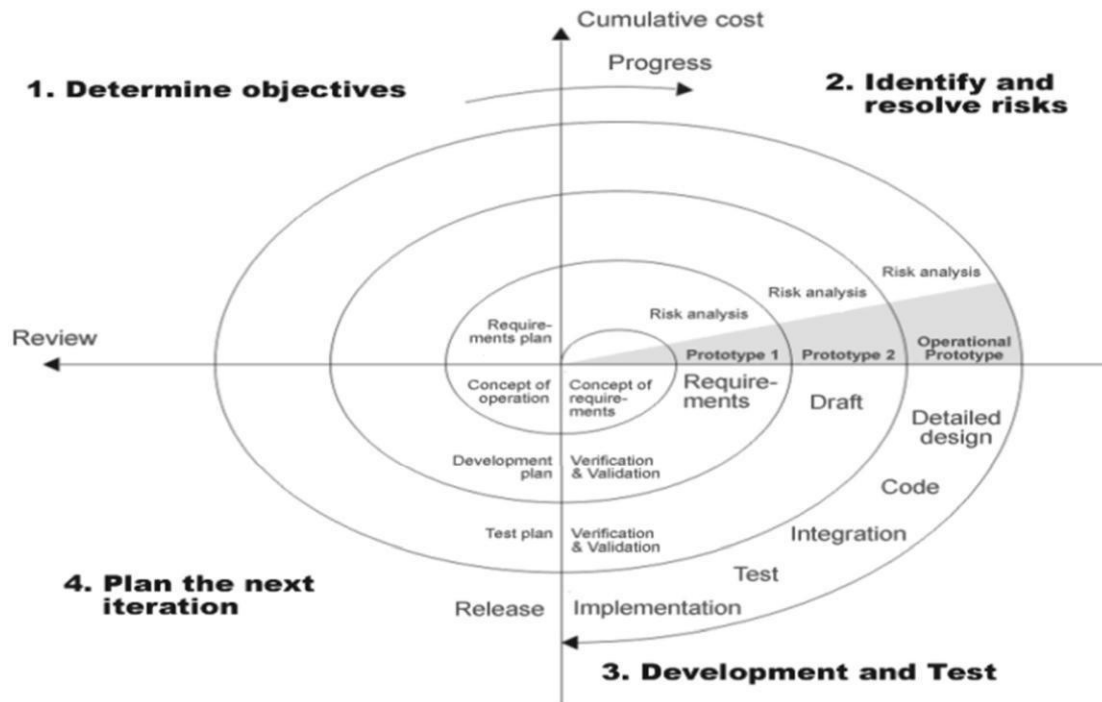
### **Advantages of Spiral Model.**

- ❖ Spiral Life Cycle Model is one of the most flexible SDLC models in place. Development phases can be determined by the project manager, according to the complexity of the project.

- ❖ Project Monitoring is very easy and effective. Each phase, as well as each loop, requires a review from concerned people. This makes the model more transparent.
- ❖ Risk management is one of the in-built features of the model, which makes it extra attractive compared to other models.
- ❖ Changes can be introduced later in the life cycle as well. And coping with these changes isn't a very big headache for the project manager.
- ❖ Project estimates in terms of schedule, cost etc become more and more realistic as the project moves forward and loops in spiral get completed.
- ❖ It is suitable for high risk projects, where business needs may be unstable. A highly customized product can be developed using this.

### Phases of Spiral Model

- ❖ **Plan:** In this phase, the objectives, alternatives and constraints of the project are determined and are documented. The objectives and other specifications are fixed in order to decide which strategies/approaches to follow during the project life cycle.
- ❖ **Risk Analysis:** This phase is the most important part of "Spiral Model". In this phase all possible (and available) alternatives, which can help in developing a cost effective project are analyzed and strategies are decided to use them. This phase has been added specially in order to identify and resolve all the possible risks in the project development. If risks indicate any kind of uncertainty in requirements, prototyping may be used to proceed with the available data and find out possible solution in order to deal with the potential changes in the requirements.
- ❖ **Engineering:** In this phase, the actual development of the project is carried out. The output of this phase is passed through all the phases iteratively in order to obtain improvements in the same.
- ❖ **Customer Evaluation:** In this phase, developed product is passed on to the customer in order to receive customer's comments and suggestions which can help in identifying and resolving potential problems/errors in the software developed. This phase is very much similar to TESTING phase.



## 10. OPERATING TOOLS AND DEVELOPMENT ENVIRONMENT

This section lists the tools used for planning, designing, development, deployment, and testing:

### Hardware Requirements :

- ❖ Processor: Intel i3 and above
- ❖ RAM: 2 GB minimum (4 GB recommended)
- ❖ Storage: Minimum 500 MB

### Software Environment :

#### Frontend Development:

The user interface was built using HTML5 for structure, Tailwind CSS for responsive and clean design, and JavaScript for client-side interactions, including chart rendering and form validation.

#### Backend Development:

The backend was developed using PHP 8.x with PDO for secure database communication. It handled role-based authentication, session management, form processing, and CRUD operations.

#### Database:

MySQL 5.7+ was used to store and manage users, teachers, subjects, and feedback records. The schema was normalized to maintain data integrity and improve performance.

#### Authentication:

A role-based login system was implemented using PHP sessions, ensuring secure and appropriate access for Admin, Teacher, and Student users.

#### Data Visualization:

Chart.js was integrated to render interactive pie charts and bar graphs for feedback analysis, making it easier for admins and teachers to visualize performance.

#### File Uploads:

Teachers can upload profile images using PHP's `move_uploaded_file()` function, with the image paths stored in the database for display.

#### Development Tools:

Visual Studio Code served as the primary IDE. XAMPP was used as the local server (Apache, MySQL, PHP stack), and Git + GitHub were used for version control and collaboration.

#### Browser Testing:

The system was tested in Google Chrome and Mozilla Firefox to ensure cross-browser compatibility and responsiveness.

#### Documentation:

Project documentation, including this black book, was created using Microsoft Word. Canva was used for designing diagrams such as the UML and flowcharts.

## 11. DESIGNING

### Overview

Design document define the overall flow and working concept of the system which will be implemented. There are different approaches of creating design document, one among widely used is UML (Uniform Modeling Language). In order to design the this application, firstly identify all the required interfaces/screens. This results in to documented concept that describes the flow of Application. The concept is then converted into design document with UML diagrams. UML is modeling language used to specify, visualize, modify, construct, and document the artifacts of an object oriented system under development. UML combines technique from data modelling (entity relationship diagrams), business modelling (work flows), object modelling, and component modelling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML diagrams represent two different views of a system model: Static (or structural) view: Emphasizes the static structure of the system using objects, attributes, operations and relationships.

#### The structural view includes:

- ❖ Class diagrams
- ❖ Composite structure diagrams
- ❖ Sequence diagram

### 11.1 Use Case Diagram

#### Actors:

- ❖ Student
- ❖ Teacher
- ❖ Admin

#### Use Cases:

- ❖ Login
- ❖ Give Feedback (Student)
- ❖ View Feedback Summary (Teacher)
- ❖ Manage Teachers/Subjects/Feedback (Admin)
- ❖ Export/Filter Feedback (Admin)

#### *Diagram Description:*

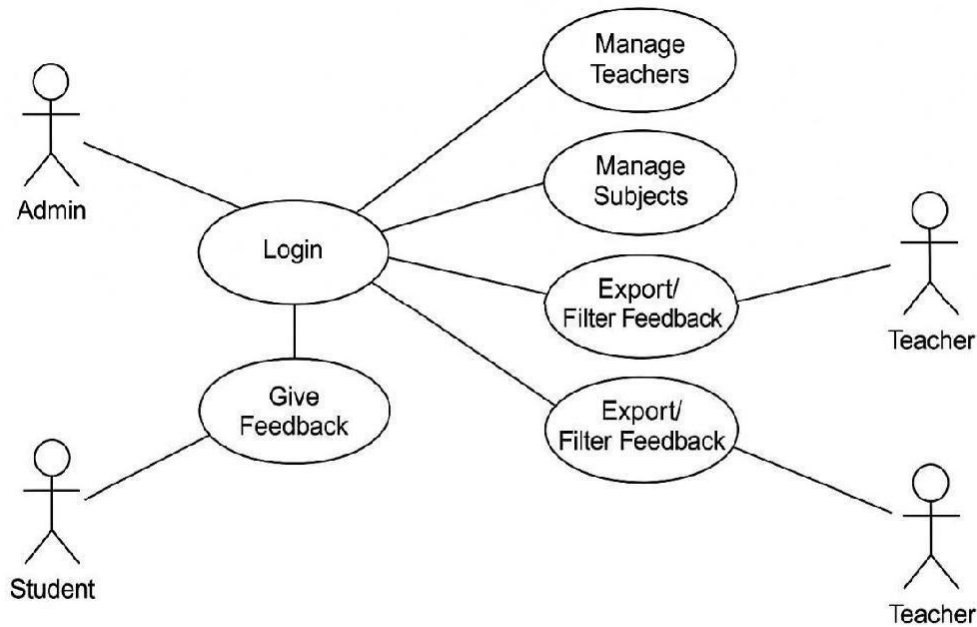
The use case diagram shows the interaction between system actors and modules.

Admin has full control over subject-teacher assignments and analytics.

Teachers only access feedback reports. Students can view subjects and submit feedback once.

**System Use-Case:**

## Student Feedback Management System



### 11.2 Class Diagram

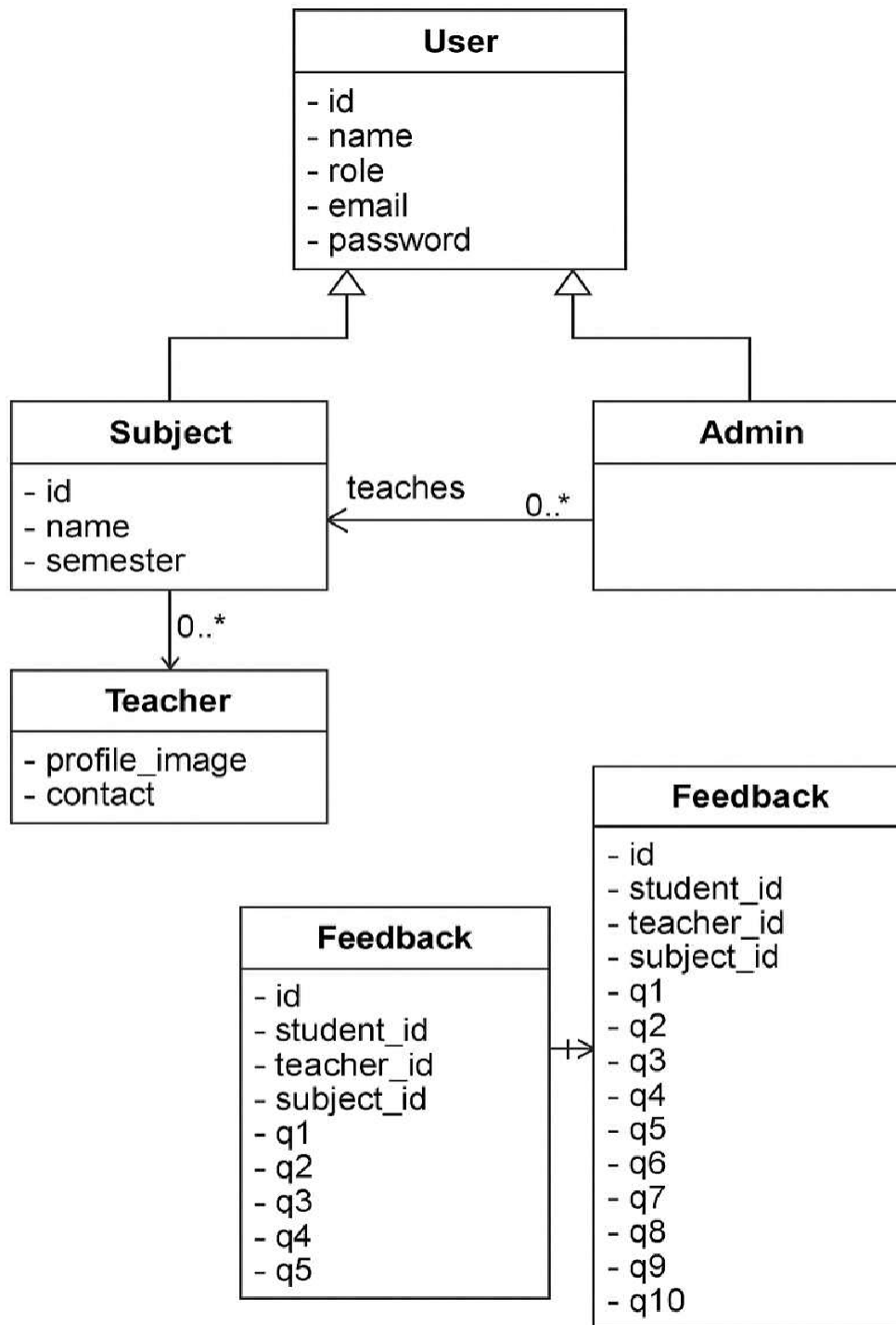
**Main Classes:**

- ❖ **User: Attributes** → id, name, role, email, password
- ❖ **Teacher: Extends User** → profile\_image, contact
- ❖ **Subject: Attributes** → id, name, semester
- ❖ **Feedback: Attributes** → id, student\_id, teacher\_id, subject\_id, q1...q10
- ❖ **Admin: Extends User**

**Relationships:**

- ❖ **One-to-many:** One teacher teaches many subjects.
- ❖ **Many-to-many:** Students give feedback to multiple teachers (via feedback table).

**Class Diagram :**



### 11.3 Activity Diagram

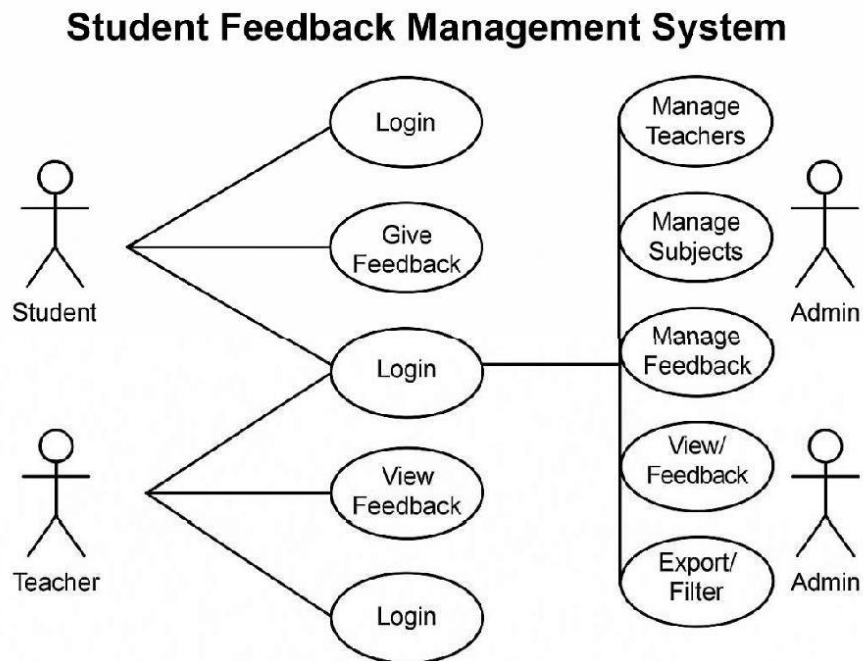
#### Key Flows:

1. Student logs in → Selects subject → Fills Q1–Q10 → Submits feedback → Gets success alert.
2. Admin logs in → Views dashboard → Clicks on teacher → Sees feedback pie chart.
3. Teacher logs in → Clicks profile → Sees average feedback.

#### Flow Elements:

- ❖ Decision boxes for login role.
- ❖ Feedback submission transition ends with DB insert and thank-you screen.
- ❖ Admin dashboard loop for filtering teachers and analyzing data.

#### Activity diagram:





### 11.4 Sequence Diagram

#### Scenario: Feedback Submission

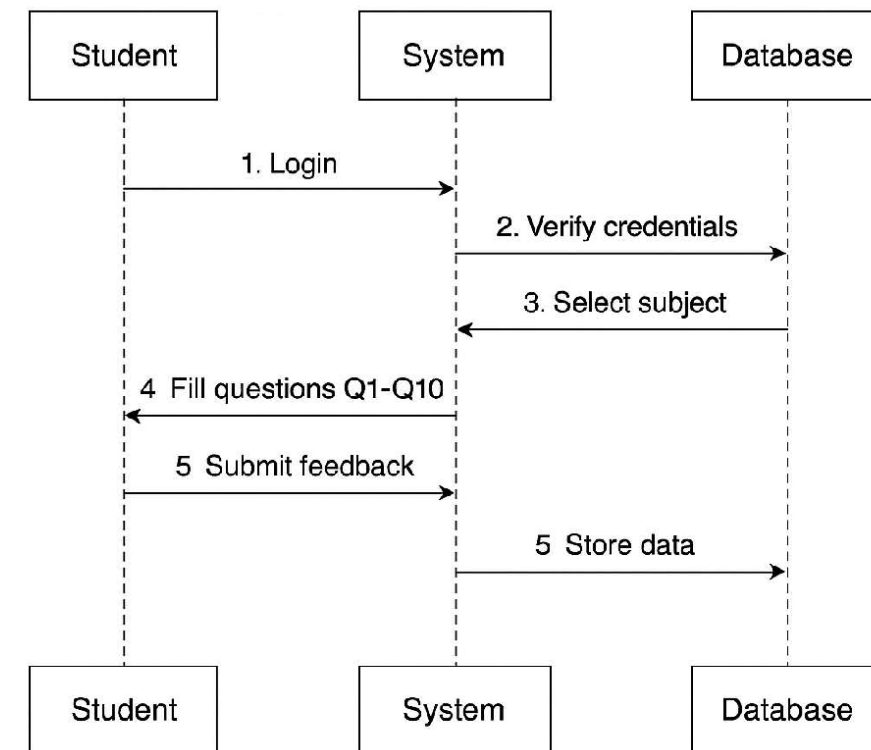
| Order | Action   |
|-------|--|
| 1     | Student logs in → System verifies credentials        |
| 2     | System loads subject list                            |
| 3     | Student selects subject, fills Q1–Q10 form           |
| 4     | Clicks submit → System stores data in feedback table |
| 5     | System shows confirmation message                    |

#### Notes:

- ❖ Async behavior in form submission is simulated using PHP and JS.
- ❖ Role-based flow management is achieved through \$\_SESSION['role'].

#### Activity diagram:

### Feedback Submission



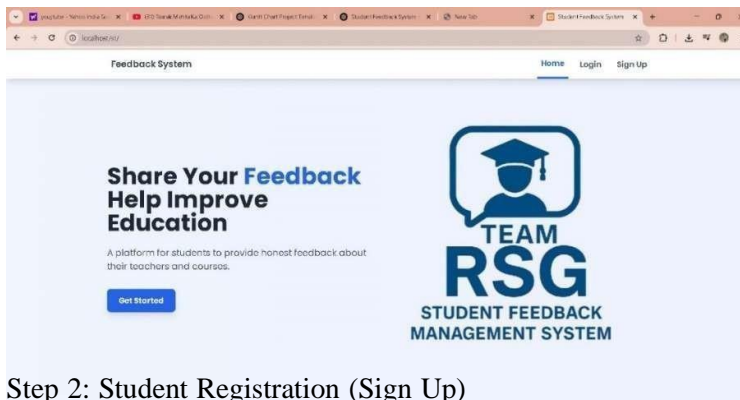
## 12. Implementation

### 12.1 Screenshots

#### Step 1: Home Page – Welcome Screen

Description:

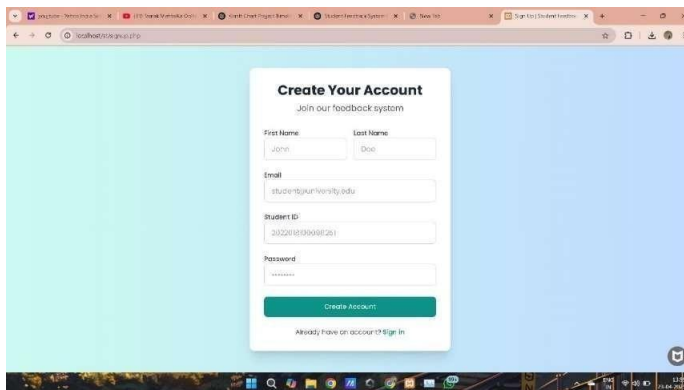
- ❖ This is the landing page of the system.
- ❖ It shows a welcoming message: “Share Your Feedback – Help Improve Education.”
- ❖ There's a “Get Started” button that redirects users to login or signup.
- ❖ TEAM RSG logo and branding are displayed prominently.



#### Step 2: Student Registration (Sign Up)

Description:

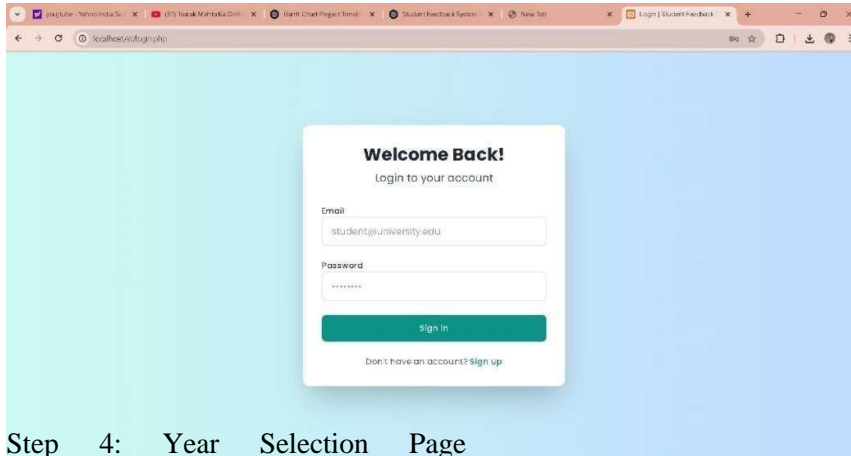
- New users can create an account by entering:
  - ❖ First Name and Last Name
  - ❖ Email ID
  - ❖ Student ID
  - ❖ Password
- After entering details, clicking “Create Account” registers the student in the system.



### Step 3: Student Login

#### Description:

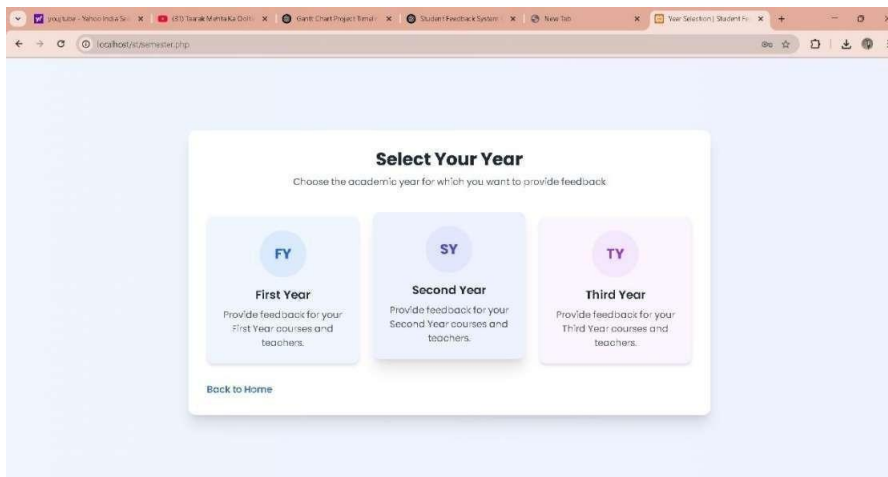
- ❖ Existing users can log in with their registered email and password.
- ❖ A secure session is created for role-based access.



### Step 4: Year Selection Page

#### Description:

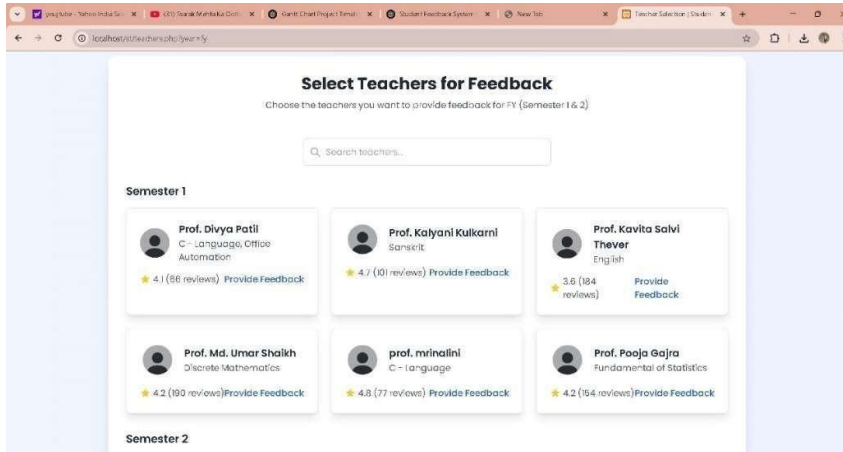
- After login, students are asked to choose their academic year:
  - ❖ First Year (FY)
  - ❖ Second Year (SY)
  - ❖ Third Year (TY)
- This helps the system filter the subjects and teachers assigned to that batch.



## Step 5: Teacher Selection Page

### Description:

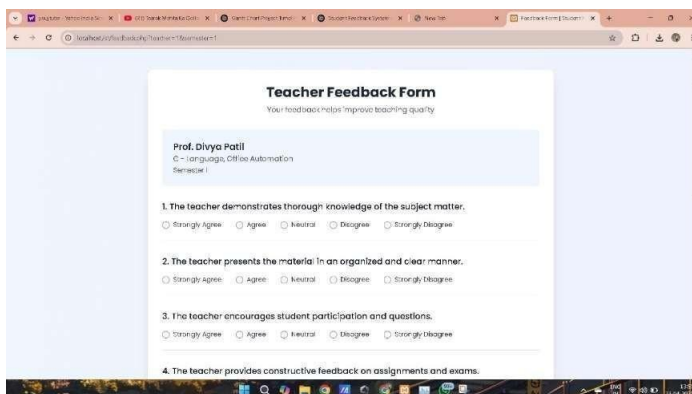
- A list of teachers appears for the selected year.
- Students can see:
  - ❖ Teacher Name & Subject
  - ❖ Semester classification (1 & 2)
  - ❖ Ratings from past students
  - ❖ “Provide Feedback” link next to each teacher



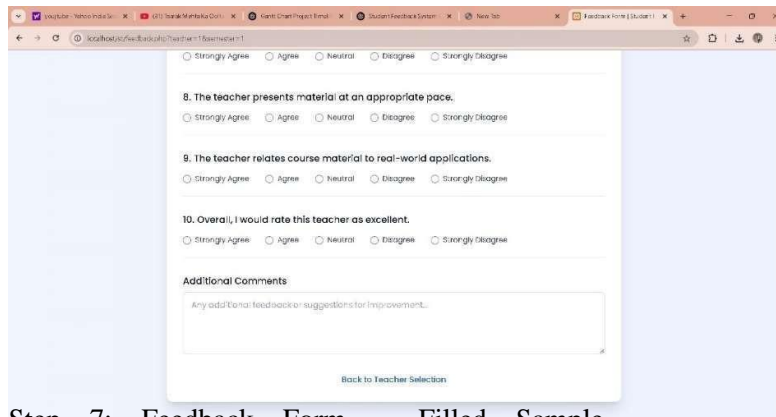
## Step 6: Feedback Form – Teacher Evaluation

### Description:

- ❖ A set of 10 questions (Q1–Q10) is shown.
- ❖ Questions cover various teaching aspects such as clarity, engagement, pace, and overall performance.
- ❖ Each question has 5 radio options: Strongly Agree to Strongly Disagree.
- ❖ A text box allows students to enter additional comments.



## Student Feedback System



Strongly Agree Agree Neutral Disagree Strongly Disagree

8. The teacher presents material at an appropriate pace.

Strongly Agree Agree Neutral Disagree Strongly Disagree

9. The teacher relates course material to real-world applications.

Strongly Agree Agree Neutral Disagree Strongly Disagree

10. Overall, I would rate this teacher as excellent.

Strongly Agree Agree Neutral Disagree Strongly Disagree

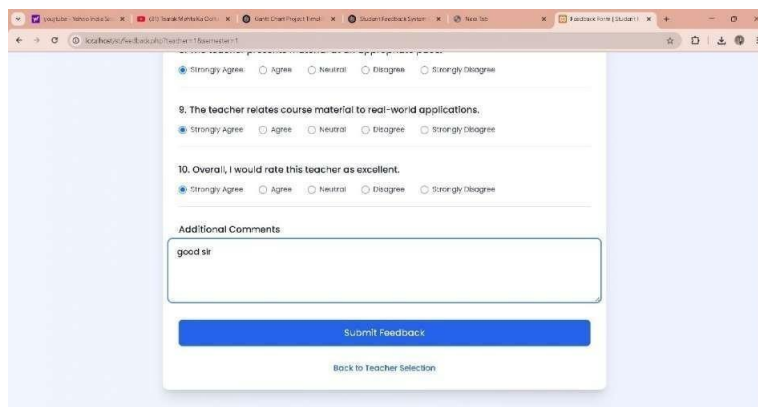
Additional Comments

Any additional feedback or suggestions for improvement...

Back to Teacher Selection

### Step 7: Feedback Form – Filled Sample Description:

- ❖ The student selects answers for all questions.
- ❖ An additional comment “good sir” is written in the textbox.
- ❖ After reviewing, the student clicks “Submit Feedback.”



Strongly Agree Agree Neutral Disagree Strongly Disagree

9. The teacher relates course material to real-world applications.

Strongly Agree Agree Neutral Disagree Strongly Disagree

10. Overall, I would rate this teacher as excellent.

Strongly Agree Agree Neutral Disagree Strongly Disagree

Additional Comments

good sir

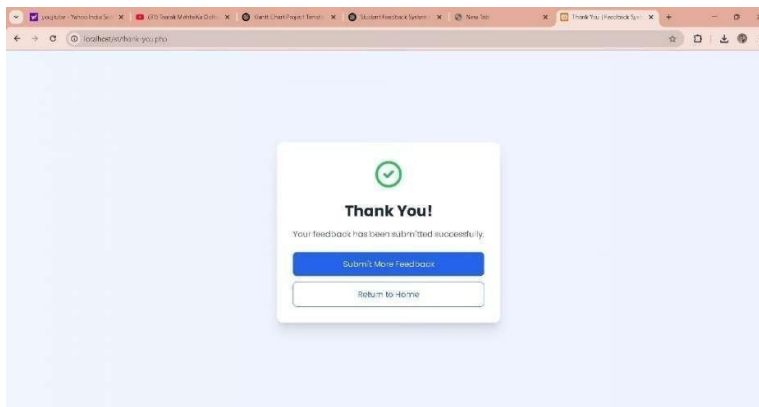
Submit Feedback

Back to Teacher Selection

### Step 8: Feedback Submission Confirmation

#### Description:

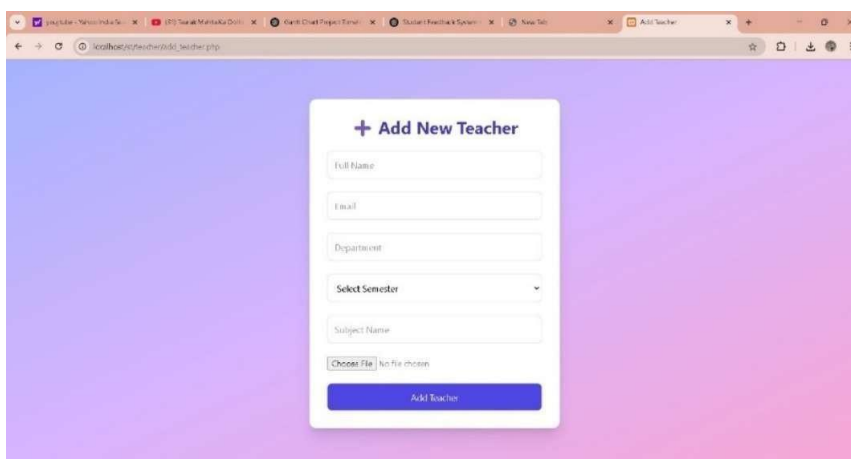
- A success message is displayed: “Thank You! Your feedback has been submitted successfully.”
- Options:
  - ❖ Submit More Feedback
  - ❖ Return to Home



### Step 9: Admin – Add New Teacher (Admin Panel)

#### Description:

- Admin can manually add a new teacher by entering:
  - ❖ Full Name, Email, Department, Semester, Subject
  - ❖ Uploading a profile image
- After clicking “Add Teacher,” the data is inserted into the database and mapped for feedback.



## step 10: Teacher Profile Update Screen

Description:

- Step: Teacher logs in to the system and accesses the profile section.
- Screen: Shows a form with fields for Name, Email, Department, Semester, Subject, and an option to upload a Profile Image.
- Purpose: Allows teachers to update their personal information and profile image

**My Profile**

Full Name  
prof. divya karotra

Email  
xibya99@gmail.com

Department  
Computer Science

Semester  
Semester 6

Subject  
python

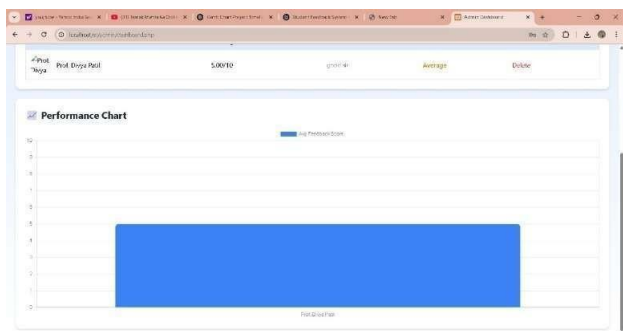
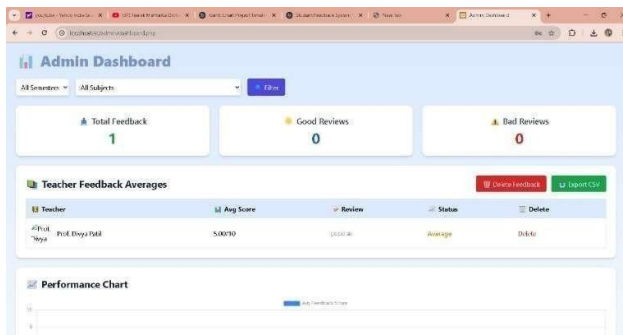
Update Profile Image  
Choose File No file chosen

Update Profile

## Step 11: Performance Chart





Description:

- Step: View teacher performance visually through a bar chart.
- Screen: Displays average feedback score in a bar chart format.
- Purpose: Offers real-time insight into teacher performance using feedback data.







## 13. TESTING

### 13.1 Unit Testing

| <u>Module</u>   | <u>Test Case</u>                       | <u>Result</u>  |
|-----------------|--|--|
| Login           | Valid login with correct credentials   |  Passed |
| Feedback Form   | Feedback submits only once per teacher |  Passed |
| Image Upload    | Valid image gets stored in /uploads/   |  Passed |
| Feedback Export | CSV downloaded correctly               |  Passed |

### 13.2 System Testing

| Component        | Test Scenario                              | Result   |
|------------------|--|--|
| Admin Dashboard  | Loads with correct teacher statistics      |  Passed   |
| Teacher Feedback | Displays chart with correct average values |  Passed   |
| Form Validation  | Prevents empty submissions                 |  Passed |
| Role Routing     | Redirects users based on login role        |  Passed |

### 13.3 Operational Testing

- ❖ System deployed and tested on XAMPP locally.
- ❖ Full user journey tested: login → submit feedback → see analysis.
- ❖ Edge cases tested: duplicate feedback, profile without image, blank form.



## 14. LIMITATIONS & FUTURE ENHANCEMENTS

### ● Current Limitations

- ❖ No email verification: Anyone with credentials can log in.
- ❖ Static feedback questions: Questions Q1–Q10 are fixed.
- ❖ No real-time notifications: Teachers aren't notified when feedback is submitted.
- ❖ Single language UI: English-only interface.
- ❖ Limited analytics: Pie charts only, no heatmaps or comparative graphs.

### ● Proposed Enhancements

| Feature                           | Benefit  |
|-----------------------------------|--|
| Add SMS/Email Notifications       | Alerts teachers/admins on feedback submission        |
| Question Pool System              | Dynamic questions per department/subject             |
| Comparative Analytics             | Compare teachers by subject and semester             |
| Real-time Graph Refresh           | Update charts without reloading dashboard            |
| Mobile App Version                | Create Android/iOS app using Flutter or React Native |
| Multi-language Support            | Include Marathi/Hindi for wider accessibility        |
| Attendance + Feedback Integration | Link attendance records with feedback participation  |

## 15. REFERENCES

- ❖ <https://www.w3schools.com/>
- ❖ <https://php.net/manual/en/index.php>
- ❖ <https://tailwindcss.com/>
- ❖ <https://www.chartjs.org/>
- ❖ <https://stackoverflow.com/>
- ❖ <https://github.com/>
- ❖ <https://www.mysql.com/>