CLASSCONNECT

PROJECT REPORT SUBMITTED TO MAHATMA GANDHI UNIVERSITY, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS

BY

AJIL SAJI

Register Number: 230021078057

DEPT. OF COMPUTER APPLICATIONS

B.V.M HOLY CROSS COLLEGE

CHERPUNKAL, KOTTAYAM 686 584

October 2025

Department of Computer Applications

B.V.M HOLY CROSS COLLEGE

CERTIFICATE

Certified that the report entitled **"CLASSCONNECT"** is the bona fide record of the project work done by **Mr. Ajil Saji (Reg. No. 230021078057)** under our guidance and supervision and is submitted in partial fulfillment of the **Bachelor degree in Computer Applications**, awarded by Mahatma Gandhi University, Kerala.

Ms. Brigit Thomas
Project Guide

Ms. Seena S Nair
Head of the Department

Submitted for Project Evaluation on ------/-----/------

Internal Examiner

External Examiner

DECLARATION

I hereby declare that the project work entitled "**CLASSCONNECT"** submitted in partial fulfillment of the requirements for the award of the Bachelor of degree in Computer Applications from BVM Holy Cross College, Cherpunkal, is record of bona fide work done under the guidance of **Ms. Brigit Thomas** (Assistant Professor, Dept. of Computer Applications)

Date: Name: Ajil Saji

Place: Cherpunkal Reg No. 230021078057

ACKNOWLEDGEMENT

Dedicating this project to the Almighty God, whose abundant grace and mercies enabled its successful completion, we would like to express our profound gratitude to all the people who inspired and motivated us to make this project a success. We express our heartfelt thanks to our Principal, **Rev. Dr. Baby Sebastian**, for his warm support and valuable suggestions in accomplishing the project. We would also like to place our deep sense of gratitude to **Ms. Seena S Nair** (Head of the Department of Computer Applications) for her guidance in carrying out this project work. We are profoundly grateful to **Ms. Brigit Thomas** (Assistant Professor, Department of Computer Applications), our Project Guide, for her valuable guidance, suggestions, and assessment throughout the project. We also extend our sincere thanks to all other faculty members of the Department of Computer Applications for their assistance and encouragement. Last but not least, we would like to thank our friends for their cooperation and support throughout the project.

ABSTRACT

Classconnect is a role-based, web-powered class management and communication hub that centralizes announcements, resources, and interactions for students, teachers, and administrators. The platform supports four distinct user roles — Student, Teacher, Admin, and Super Admin — each with specific permissions and access levels to ensure security, privacy, and efficient management. 4 5 Core features include a ClassFeed system for posting updates and sharing files, a Poll Section with real-time voting results and one-vote-per-user enforcement, a Feedback Section that allows category-based feedback with optional anonymity and star ratings, and a Question Paper Section for streamlined access to study materials. The platform also includes a Notification System, Profile Management, and Role-Based Access Control to restrict sensitive features (e.g., user list, post deletion) to authorized users only. Built using PHP, MySQL, HTML, CSS, and JavaScript, Classconnect adopts a modular architecture for scalability and maintainability. Security measures such as session management and (planned) password hashing are incorporated to safeguard user data. By unifying communication channels, Classconnect reduces the need for fragmented messaging apps and manual announcements, offering a responsive, easy-to-use web platform accessible from any device. Future enhancements will focus on real-time updates (WebSockets), mobile app integration, teacher student private messaging, and analytics dashboards for better academic insights.

CONTENTS

1. INTRODUCTION

1.1.PROJECT OVERVIEW

ClassConnect is a role-based, web-based classroom collaboration and resource-sharing platform designed to centralize communication between students and teachers. The system was developed as a mini-project using the LAMP stack (Linux/Apache/MySQL/PHP) with standard web technologies (HTML, CSS, JavaScript) for the frontend and MySQL as the backend database. ClassConnect enables secure user registration and authentication with support for multiple roles — Super Admin, Admin (Teacher), and Student. Each user sees a personalized dashboard based on their role: teachers and admins can create posts, upload files (study materials, question papers), manage polls, and monitor feedback, while students can view class posts, participate in polls, and submit feedback. 9 A key feature of ClassConnect is the ClassFeed, a dynamic, real-time feed where teachers and admins can share announcements, resources, and updates with their batch or class. Posts are displayed in reverse chronological order, and admins can edit or delete posts, ensuring only relevant information is available. Other major modules include: • Feedback System: Students can submit feedback with optional anonymity, star ratings, and categories. Admins can view, reply, and track feedback status. • Poll System: Supports single-choice/multiple-choice polls, live vote counts, poll expiration, and an archive for past polls. • Question Paper Section: Teachers can upload question papers, and students can download them easily. • Notification System: Displays recent updates, sorted by upload time, and can be triggered from the dashboard. • Profile Section: Shows user details (name, admission number, register number, role, email, class, course, DOB) in a popup view. ClassConnect follows software engineering best practices, including modular code structure, database normalization, and secure SQL queries (prepared statements). The system is designed to be scalable and extensible, allowing easy addition of future features such as private messaging, file version history, and AI-powered recommendations. This project not only provides a functional platform for real classroom communication but also serves as a comprehensive case study in full-stack web application development, covering all phases of the Software Development Life Cycle (SDLC): requirement gathering, system design, implementation, testing, and deployment.

1.2.ORGANIZATION PROFILE

Organization Name: BVM Holy Cross College

Location: Cherpunkal, Kottayam, Kerala, India

Year of Establishment: 1995

Type of Institution: Self-Financing College

Affiliation: Mahatma Gandhi University

Management: Holy Cross Forane Church, Cherpunkal

B.V.M Holy Cross College is a pioneering self-financing college affiliated to Mahatma Gandhi University, Kottayam. The college belongs to Roman Catholic Syrian Christian Community. It was founded in 1995 by the Holy Cross Forane Church, Cherpunkal of Diocese of Palai under the patronage of Palai Diocese. Situated in the renowned Infant Jesus pilgrimage centre alongside an idyllic ambience on the banks of Meenachil river, the college is enveloped in an aura that facilitates overall development of a student.

Vision: To be a leading institution of higher education that empowers students to become responsible citizens and ethical professionals.

Mission: To provide quality education that promotes academic excellence, holistic development, and social commitment.

Core Values:

- Academic Integrity
- Student-Centric Approach
- Social Responsibility
- Inclusiveness

• Excellence in all Endeavors

Contact Information:

BVM Holy Cross College

Cherpunkal, Kottayam,

Kerala-686529

Connect with BVM

Phone: 04822-267520

9446640157

Email: bvmhcc@gmail.com

Website: https://bvmcollege.com/

2.SYSTEM CONFIGURATION

2.1 HARDWARE SPECIFICATION

- CPU: Intel i3 or equivalent
- RAM:4 GB (8 GB recommended)
- Storage: 20 GB free

2.2 SOFTWARE SPECIFICATION

- OS Windows /Linux/macOS
- Web Server : Apache(XAMPP/WAMP/LAMP)or Ngnix + PHP-FPM
- PHP: 7.4/8.x
- Database : MySQL / MariaDB
- Toll Used: Git ,VS code ,phpMyAdmin

3.SYSTEM STUDY

The System Study provides an in-depth examination of the current classroom communication practices, identifies gaps and inefficiencies, and explains how the proposed system 11 (ClassConnect) addresses these shortcomings. It is divided into three parts: Existing System, Proposed System, and Feasibility Study.

3.1. EXISTING SYSTEM

The current system of class communication and resource sharing relies heavily on informal and manual methods, with no dedicated technological integration. Students and teachers typically use WhatsApp groups, emails, or word of mouth to share announcements, study materials, or important updates, which is often time-consuming and prone to miscommunication. Access to shared resources such as notes, question papers, or feedback is limited and scattered, depending on individual contributions and personal storage, leading to reduced collaboration and overall effectiveness.

Interaction between students and teachers outside the classroom is minimal, making it difficult to address academic concerns or provide timely updates on schedules, events, or assignments. File and document sharing is unstructured, with no centralized repository to organize or retrieve past materials efficiently. User roles and permissions are not defined, so sensitive actions like deleting content or managing users are unregulated.

Feedback collection is usually verbal or handled through separate forms, lacking a structured mechanism for anonymity, categorization, or tracking responses. Similarly, class-level decisions such as polls or surveys are managed informally, which can result in confusion and lack of transparency. Event reminders and notifications are also communicated irregularly, making it harder for students to stay updated.

Overall, the existing system faces challenges in terms of efficiency, collaboration, and organization, highlighting the need for a more streamlined, centralized, and technology-driven solution for the class.

Limitations of Existing System:

- Lack of Centralization: No single platform to manage communication, study materials, and class activities in one place.

- Unstructured File Sharing: Notes, question papers, and assignments are scattered across multiple apps and personal devices.
- No Role Management: Teachers/admins and students have the same level of control, leading to confusion and lack of authority.
- Communication Gaps: Important updates often get lost in crowded chat groups or are missed due to irregular notifications.
- No Proper Feedback Mechanism: Feedback is usually verbal or handled through basic forms, with no anonymity or tracking system.
- Limited Collaboration: Group discussions, polls, and event planning are unorganized, reducing overall class participation.
- Manual Processes: Many tasks like collecting feedback, tracking attendance, or sharing updates rely on manual efforts, increasing errors and delays.
- Poor Accessibility: Past resources (e.g., notes, question papers) are difficult to retrieve since they aren't stored in a structured repository.
- Scalability Issues: As the class size grows, existing informal methods struggle to handle the increased communication and resource-sharing needs.
- Data Security Concerns: No secure system for managing student data, credentials, or sensitive information.

3.2.PROPOSED SYSTEM(CLASSCONNECT)

The proposed ClassConnect platform provides a centralized, technology-driven solution to the limitations of the existing manual and fragmented class communication methods. By leveraging modern web technologies, ClassConnect streamlines communication, enhances collaboration, and improves the overall classroom experience. Through a clean and user-friendly interface, students and teachers gain access to a single platform where they can share study materials, participate in polls, provide feedback, and stay updated on events and notifications.

One of the major strengths of ClassConnect is its ability to centralize all academic resources and communication into a single, accessible hub. This ensures that students can easily retrieve notes, assignments, question papers, and announcements without relying on scattered apps or

personal contacts. With defined user roles, admins (teachers) have greater control over content management, while students can focus on collaboration and learning.

The platform also enables seamless communication between students and teachers by offering structured features such as real-time notifications, a dedicated feedback system with optional anonymity, and interactive polls for decision-making. This transparency and responsiveness foster an inclusive environment where every student's voice can be heard.

Efficiency is further improved by introducing features such as categorized file uploads, automated organization of resources, and an integrated class feed for posts and discussions. Unlike traditional methods, ClassConnect ensures that information is never lost and can be retrieved whenever needed.

A built-in feedback mechanism allows students to rate, suggest, and submit their concerns in an organized way, while admins can track and respond effectively. This creates a continuous improvement loop that enhances the quality of teaching and learning.

With a scalable architecture, ClassConnect can easily adapt to larger groups, additional courses, or extended functionalities such as event calendars, advanced role management, and performance tracking. By providing a secure, centralized, and structured environment, ClassConnect not only addresses the inefficiencies of the current system but also lays the foundation for a more collaborative and technology-enabled academic experience.

3.3. FEASIBILITY STUDY

1.Technical Feasibility:

ClassConnect is built using open-source, widely supported technologies such as PHP**,** MySQL**,** HTML**,** CSS, and JavaScript, deployed on a LAMP/XAMPP stack. It supports role-based access control, CRUD operations for posts, polls, and feedback, and is designed to be fully responsive for mobile and desktop users. Since no specialized hardware is required, the system can run on any standard server or personal computer. The modular database design ensures that the system can be easily expanded with new features in the future.

2. Operational Feasibility:

The system is user-friendly and intuitive, requiring only a web browser to access. Dashboards are customized based on user roles:

- Admins/Teachers can create and manage posts, polls, and feedback.
- Students can view posts, submit feedback (with an option for anonymity), and participate in polls.

  This role-based separation minimizes confusion, reduces errors, and eliminates the need for extensive training, making adoption smooth and efficient.

3. Economic Feasibility:

ClassConnect is highly cost-effective, as it relies on free and open-source tools such as XAMPP**,** MySQL**,** VS Code**,** and GitHub. Since there are no licensing fees or specialized infrastructure costs, the overall development and deployment expenses are negligible. This makes the system an affordable solution for academic institutions.

4. Schedule Feasibility:

The system was developed in a phased approach over a 4–6 week mini-project timeline. Core modules such as ClassFeed**,** Feedback**,** Polls**,** and Notifications were implemented incrementally, ensuring smooth integration and testing at each stage. This structured development process allowed the project to meet deadlines without delays.

3.4. REQUIREMENT SPECIFICATION

Software Requirement Specification (SRS) is the requirements document that provides the technical specification for the design and development of the software. This document improves the quality of the system by formalizing communication between the developer and the user, and provides accurate information for proper documentation. A description of each function required to solve the problem is presented in the functional description. The behavioral description section of the specification examines the operation of the software as a result of external events and internally generated control characteristics. The validation criteria act as an implicit review of all requirements to ensure they are complete and testable. The proposed system has the following requirements:

3.4.1 FUNCTIONAL REQUIREMENTS

1. User Registration and Authentication
    - Users must be able to create an account and log in securely.
    - Session management should validate active users.
    - Role-based authentication: Admins/Teachers vs Students.
2. ClassFeed
    - Users can post text updates, announcements, and discussions.
    - Posts may include attachments (images, PDFs, notes).
    - Students can view, like, or comment on posts.
    - Admins can delete or moderate posts.
3. Files Module
    - Users can upload and share files (notes, assignments, presentations, etc.).
    - Files should be categorized for easy retrieval.
    - Search functionality should allow users to find files quickly.
4. Feedback System
    - Students can submit feedback with categories (academic, non-academic, facility-related, etc.).
    - Feedback can be anonymous or named.
    - Feedback may include text and optional star ratings.
    - Admins can track feedback status (pending, resolved) and reply to it.
5. Timetable
    - Admins/Teachers can upload and update the class timetable.
    - Students can view daily/weekly timetables.
    - Changes in timetable should trigger notifications.

6. Academics Module

- Teachers/Admins can upload student marks for exams, assignments, and assessments.
- Students can view their own marks in a secure, role-based manner.
- Marks should be organized by subject, exam type, and date.
- The system should prevent students from seeing marks of other students.
- Teachers/Admins can update or modify marks if needed.

7. Polls

- Admins/Teachers can create polls with single-choice or multiple-choice options.
- Students can cast one vote per poll.
- Polls should display live results after voting.
- Polls should have an expiration date/time and move to archive once completed.

8. Question Papers Repository

- Users can upload previous year question papers.
- Papers should be organized by subject and exam type.
- Students can view/download stored question papers.

9. Teacher's Corner

- A dedicated section for teachers to share academic resources, notes, and announcements.
- Students can access teacher-uploaded materials directly.
- Teachers can update or remove their own content.

10. Notification System

- Displays updates for posts, new files, polls, timetable changes, and feedback responses.
- Notifications should be sorted by upload time.

- Notifications should be visible via a popup when the user clicks the notification button.

11. Profile Management

- Each user has a profile showing: name, admission no, register no, email, role, DOB, class, and course.
- Profile details should be accessible in a popup box.

3.4.2 NON-FUNCTIONAL REQUIREMENT

1. Usability:

- The system must have a clean, user-friendly interface accessible through any web browser.
- Responsive design should support both desktop and mobile devices.

2. Reliability:

- The system should ensure consistent performance without data loss during normal operation.

3. Scalability:

- The modular design should allow easy addition of new features (e.g., attendance tracking, grading).

4. Security:

- User authentication must be secure (future upgrade: password hashing & encryption).
- Access control should prevent unauthorized use of admin features.

5. Performance:

- The system should handle multiple users simultaneously without significant delays.

4.SYSTEM DESIGN

4.1 INTRODUCTION

System design is the process of defining the architecture, components, modules, interfaces, and data of a system in order to meet the specified requirements. For ClassConnect, the system design phase transforms the requirements identified in the system study into a detailed technical blueprint for implementation. The focus of the design phase is to create a clear, maintainable, and scalable solution that supports all ClassConnect functionalities such as ClassFeed, Polls, Feedback, Notifications, Question Paper Management, and Role-Based Access Control.

The system is designed with a client-server architecture where:

- The frotend (HTML,CSS,JavaScript) provides an intutice and responsive interface for student ,teacher and admin.
- The backend (PHP + MySQL) handles authenticatin ,database operation ,and business logic securely.

Special care has been taken to ensure:

- Role-based access control ,so students, teachers,and admins have different permissions.
- Data consistency and security ,with prepared statements ,hashed passwords and validated user inputs.
- Scalability , allowing new feactures like event calanders,chat modules or mobile app integration to be added easily in the future.
- User experience,with a dashboard centric design that gives students quick access to posts ,polls,notification,and uploaded materials.

This design ensure that ClassConnect not only meets its current requirnments but is also flexible enough for future enhancements .It serves as a blueprint that bridges the gap between requirnment analysis and the actual coding process.

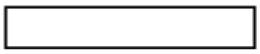
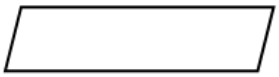
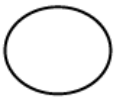
4.2 SYSTEM FLOWCHART

The classical system flowchart approach to describing and documenting a system will be presented.These system flowchart are also used in the structured approach that is ,from the general to detailed ,of the system developments life cycle. Because they have been used to described systems for many years ,they are still common in many buisnesses.

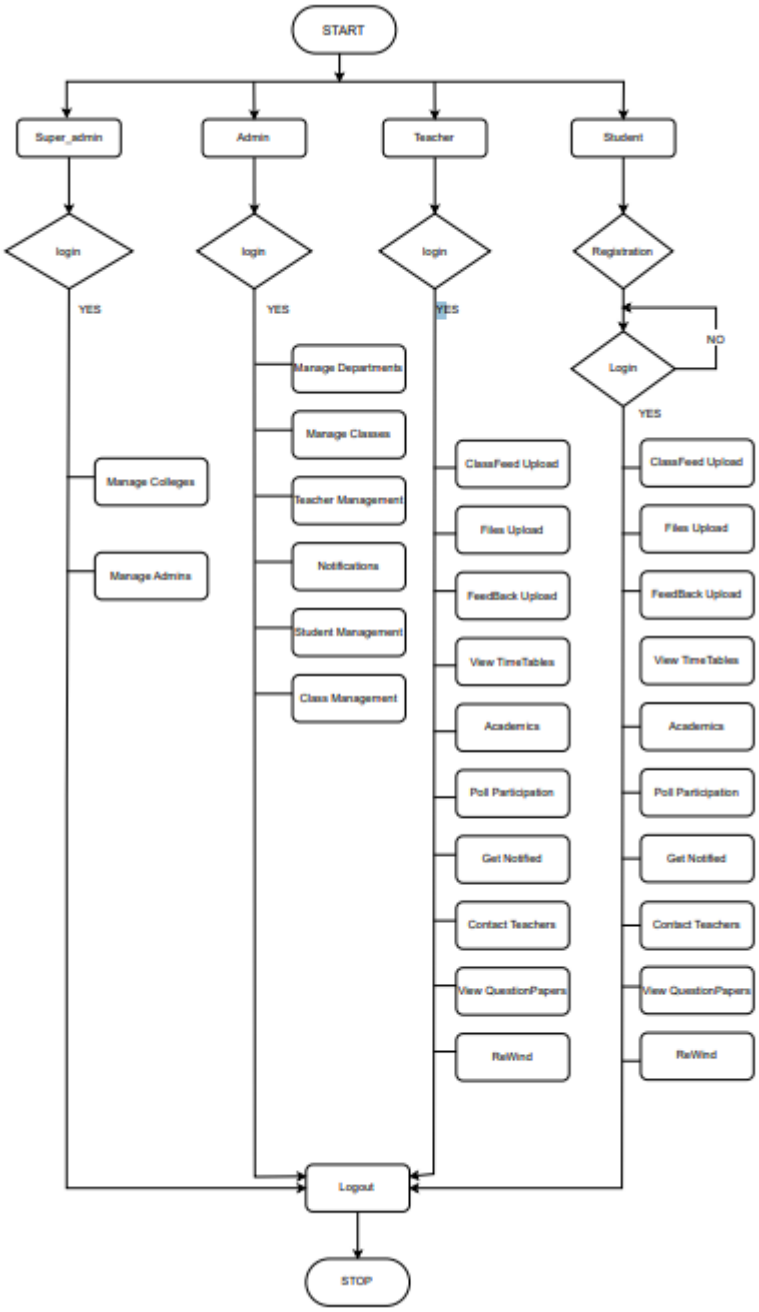
They help analysts, designers, and developers communicate system logic clearly, identify potential bottlenecks, and improve efficiency in the design process. Even with the rise of modern modeling techniques like data flow diagrams (DFDs) and Unified Modeling Language (UML) diagrams, system flowcharts continue to be used widely due to their simplicity, clarity, and ability to present a complete picture of system operations.

Moreover, system flowcharts serve as a bridge between system analysis and program design, allowing teams to move smoothly from conceptual understanding to technical implementation.

Basic Flow Chart Symbols:

Process	
Input-Output	
Connector	
Off page connector	
Data Flow	

FLOWCHART


START
Super_admin
Admin
Teacher
Student
login
login
login
Registration
YES
YES
YES
NO
Login
YES
Manage Departments
Manage Classes
Teacher Management
Notifications
Student Management
Class Management
Manage Colleges
Manage Admins
ClassFeed Upload
Files Upload
FeedBack Upload
View TimeTables
Academics
Poll Participation
Get Notified
Contact Teachers
View QuestionPapers
ReWind
ClassFeed Upload
Files Upload
FeedBack Upload
View TimeTables
Academics
Poll Participation
Get Notified
Contact Teachers
View QuestionPapers
ReWind
Logout
STOP

4.3DATABASE DESIGN

The most important aspect of building an application is the design of tables or the database schema. The data stored in the tables must be organized in some manner,which is meaningful. The overall objective in the process of table design has been to treat data as an organizational, resource and as an integrated whole. The organization of data in a database aim to achieve three major objectives, which are given below:

- Data integration
- Data abstraction
- Data independence

Several degrees of normalization have to be applied during the process of table design. The major aim of the process of normalization is to reduce data redundancy and prevent losing data integrity. Data integrity has to be converted at all levels. Pure normalization can access problem related to storage and retrieval of data. During the process of normalization, dependence's can be identified which cause serious problems during deletion and updating. Normalizing also hope in simplifying the structure of the table. The theme behind a database is to handle information as an integrated whole thus making access to information easy, quick, inexpensive and flexible for users.The entire package depends on how the data are maintained in the system.Each table has been designed with a perfect vision. Minor tables have been treated which through takes much space facilitates the process of querying fast and accurate.

PRIMARY KEY

The key is to identify records. Also uniquely notify the not null constraints. 17 FOREIGN KEY The key which references the primary key, is the data inserted in the primary key column of the table.

NORMALIZATION

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example,storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

First normal form (1NF)

sets the very basic rules for an organized database: Eliminate duplicative columns from the same table. Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

Second normal form (2NF)

further addresses the concept of removing duplicative data: Meet all the requirements of the first normal form. Remove subsets of data that apply to multiple rows of a table and place them in the separate tables.

Table 1: Users

Primary key: id

Foreign key: College_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary Key	Unique id
Admission_no	Varchar	50	UNIQUE,NOT NULL	Unique id for student
Email	Varchar	100	UNIQUE	Unique email address
Register_no	Varchar	50	UNIQUE	Unique register number
Dob	Date			Date of birth
course	Varchar	100		Studying course
password	Varchar	225	NOT NULL	User password
role	Enum(‘super_admin’, ‘admin’,’teacher’, ‘student’)		DEFAULT student	Defines role of the users

subject	Varchar	100		Subject of teacher studying
Created_at	Timestamp		DEFAULT	Timestamp of when the user was created
First_name	Varchar	50	NOT NULL	First name of the user
Last_name	Varchar	50	NOT NULL	Second name of the user
College_code	Varchar	50	UNIQUE	College code of the college
College_id	Int	11	Foreign key	College id of the college
Class_code	Varchar	50	UNIQUE	Classcode of the class
Deparment_id	Int	11	UNIQUE	Department of the class
Batch_id	Int	11	UNIQUE	Batch under department

Table 2 : Uploads

Primary key: Id

Foreign key: User_id , File_type_id

Field name	Datatype	Size	Constraints	Description
Id	int	11	Primary key	Unique id
User_id	Int	11	Foreign key	Users id
File_name	Varchar	225	NOT NULL	Name of the file
File_path	Varchar	225	NOT NULL	Path of the file

File_type_id	Int	11	UNIQUE	Type of the fie
Uploaded_at	timestamp		Default	Upload time

Table 3 : Timetable

Primary key: Id

Foreign key: College_id,Batch_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
Filename	Varchar	225	NOT_NULL	File name
File_path	Varchar	225	NOT_NULL	Path to the file
Uploaded_by	Int	11	UNIQUE	User id who upload
Uploaded_by_name	Varchar	225	NOT NULL	User name of the uploader
Uploaded_at	Timestamp		NOT NULL	Uploaded time
College_id	Int	11	Foreign key	Unique college id
Batch_id	Int	11	Foreign key	Unique batch id
Class_code	Varchar	50	NOT NULL	Unique class code

Table 4 : teacher_allocations

Primary key: Id

Foreign key: Teacher_id , Department_id,Batch_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id

Teacher_id	Int	11	Foreign key	User id of teacher
Department_id	Int	11	Foreign key	Department id from department
Batch_id	Int	11	Foreign key	Batch id from batch
Subject	Varchar	100	NOT NULL	Subject taught
Class_code	Varchar	50	NOT NULL	Class code

Table 5: question_papers

Primary key: Id

Foreign key : allocation_id, college_id, uploaded_by

Field name	**Datatype**	**Size**	**Constraints**	**Description**
Id	Int	11	Primary key	Unique id
Title	Varchar	100	DEFAULT	Title of question paper
Subject	Varchar	100	NOT NULL	subject
File_path	Varchar	225	DEFAULT	Path to the file
Uploaded_at	Timestamp		NOT NULL	Uploaded by
Allocation_id	Int	11	Foreign key	Teacher id who upload
College_id	Int	11	Foreign key	College id
Class_code	Varchar	50	NOT NULL	Unique code
Uploaded_by	Int	11	Foreign key	Teacher who upload
Batch_id	Int	11		Batch id

6.poll_votes

Primary key: Id

Foreign key: Poll_id,Option_id,User_id

Field name	**Datatype**	**Size**	**Constraints**	**Description**
Id	Int	11	Primary key	Unique id
Poll_id	Int	11	Foreign key	Poll unique id
Option_id	Int	11	Foreign key	Option unique id
User_id	Int	11	Foreign key	User unique id
Voted_at	Timestamp		NOT NULL	Voted time

Table 7 : poll_options

Primary key: Id

Foreign key: Poll_id

Field name	**Datatype**	**Size**	**Constraints**	**Description**
Id	Int	11	Primary key	Unique id
Poll_id	Int	11	Foreign key	Poll unique id
Option_text	Varchar	225	NOT NULL	Poll options

Table 8 : polls

Primary key: Id

Foreign key: College_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique poll id
Question	Text		NOT NULL	Poll question
Options	Text		NOT NULL	options

Is_anonymous	Tinyint	1	DEFAULT	Anonymous vote
Is_multiple_choice	Tinyint	1	DEFAULT	Multiple choise
Created_at	Datetime		DEFAULT	Created time
Expires_at	Datetime	11	NOT NULL	Expiring time
College_id	Int	11	Foreign key	Unique id
status	Tinyint	1	DEFAULT	Status of poll

Table 9: notifications

Primary key: Id

Foreign key : College_id , Creator_id , Target_user_id , Batch_id ,Classfeed_id

Field name	**Datatype**	**Size**	**Constraints**	**Description**
Id	Int	11	Primary key	Unique id
Post_id	Int	11	DEFAULT	Unique post id
Message	Text		NOT NULL	message
File_path	Varchar	225	DEFAULT	Path to the file
Created_at	Datetime		DEFAULT	Created at
Uploaded_by	Varchar	50	DEFAULT	Uploaded by
Creator_id	Int	11	Foreign key	User id of sender
Target_role	Enum('student','admin')		DEFAULT	Role
Target_user_id	Int	11	Foreign key	Receiver id
Is_read	Tinyint	1	DEFAULT	Is read
Classfeed_id	Int	11	Foreign key	Classfeed unique id
College_id	Int	11	Foreign key	College unique id

Batch_id	Int	11	Foreign key	Batch unique id
Class_code	Varchar	50	NOT NULL	Class Code

Table 10 : notes

Primary key : Id

Foreign key: Uploaded_by

Field name	Datatype	Size	Constraints	Description
Id	In	11	Primary key	Unique note id
Title	Varchar	100	DEFAULT	Title
Description	Text		DEFAULT	Description
File_path	Varchar	225	DEFAULT	Path to the file
Uploaded_by	Int	11	Foreign key	Uploaded user
Uploaded_at	timestamp		NOT NULL	User name who uploaded

Table 11:file_type numbers

Primary key: Id

Foreign key: File_type_id

Field name	**Datatype**	**Size**	**Constraints**	**Description**
Id	Int	11	Primary key	Unique id

File_type_id	Int	11	Foreign key	Unique file type id
Batch_id	Int	11	NOT NULL	Unique id for batch
Type_label	varchar	225	NOT NULL	Type of the file
Created_at	Timestamp		NOT NULL	Created time

Table 12:file_types

Primary key: Id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
Type_name	Varchar	50	UNIQUE	File type names

Table 13: file_categories

Primary key: Id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
name	Varchar	100	NOT NULL	Categories name

Table 14: files

Primary key: Id

Foreign key: Subject_id,User_id,College_id,Batch_id,File_type_number,Fil_type_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
File_name	Varchar	225	DEFAULT	File name
Title	Varchar	100	NOT NULL	title

File_path	Varchar	225	NOT NULL	Path to the file
Uploaded_at	Timestamp		DEFAULT	Uploaded time
File_type	Enum('material' ,'assignement','others')		DEFAULT	Type of the file
Subject	Varchar	100	NOT NULL	Subject
Subject_id	Int	11	Foreign key	Unique subject id
Register_no	Varchar	50	DEFAULT	Register number
Admission_no	Varchar	50	DEFAULT	Admission number
Comment	Text		DEFAULT	Comment
User_id	Int	11	Foreign key	Unique id of user
Deadline	Datetime		DEFAULT	deadline
Status	Enum('onTime','late')		DEFAULT	Status of upload
College_id	Int	11	Foreign key	Unique college id
Batch_id	Int	11	Foreign key	Unique batch id
Class_code	Varchar	50	NOT NULL	classcode
File_type_number	Int	11	Foreign key	Unique number
File_type_id	int	11	Foreign key	Unique file type id

Table 15:departments

Primary key: Id

Foreign key: College_id

Field name	Datatype(size)	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
College_id	int	11	Foreign key	Unique college id
Department_name	Varchar	100	DEFAULT	Department name
Created_at	timestamp		DEFAULT	Created time

Table 16:feedback

Primary key: Id

Foreign key: User_id , Batch_id , Teacher_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
User_id	Int	11	Foreign key	Unique user id
Category	Varchar	50	NOT NULL	Feedback category
Comment	Text		NOT NULL	Comment
anonymous	Tinyint	1	DEFAULT	Identity of voter
Rating	Int	11	DEFAULT	Rating
Reply	Text		DEFAULT	Reply
Created_at	Timestamp		NOT NULL	Created time
Feedback_text	Text		NOT NULL	Feedback text
Is_anonymous	Tinyint	1	DEFAULT	Anonymous vote
Batch_id	Int	11	Foreign key	Unique batch id
Subject	Varchar	100	NOT NULL	subject
Teacher_id	Int	11	Foreign key	Unique teacher id

Table 17: colleges

Primary key: College_id

Field name	Datatype	Size	Constraints	Description
College_id	Int	11	Primary key	Unique college id
College_code	Varchar	50	NOT NULL	College code
College_name	Varchar	225	NOT NULL	College name
Address	Text		DEFAULT	Address of college
Contact_email	Varchar	225	DEFAULT	Contact email
Contact_phone	Varchar	20	DEFAULT	Contact number
Created_at	Timestamp		NOT NULL	College added time

Table 18 : classfeed

Primary key: Id

Foreign key : User_id , College_id ,Batch_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
User_id	Int	11	Foreign key	Unique user id
Post_type	Varchar	50	NOT NULL	Post type
Subject	Varchar	100	DEFAULT	Subject of classfeed
Message	Text		NOT NULL	message
File_path	Varchar	225	DEFAULT	Path to the file
Created_at	Datetime		DEFAULT	Created time
College_id	Int	11	Foreign key	Unique college id
Batch_id	Int	11	Foreign key	Unique batch id
Class_code	Varchar	50	NOT NULL	Class code

Table 19:classes

Primary key: Id

Foreign key: College_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
Class_name	Varchar	225	NOT NULL	Name of class
Class_code	Varchar	50	NOT NULL	Unique code
College_id	Int	11	Foreign key	Unique college id
Created_at	Timestamp		NOT NULL	Created time

Table 20:batches

Primary key: Id

Foreign key: Department_id , College_id

Field name	Datatype	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
Department_id	Int	11	Foreign key	Unique department id
College_id	Int	11	Foreign key	Unique college id
Batch_name	Varchar	225	NOT NULL	Batch name
Class_code	Varchar	20	DEFAULT	Unique code

Table 21 : marks

Primary key: Id

Foreign key: Student_id ,Teacher_id,Batch_id

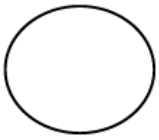
Field name	Data type	Size	Constraints	Description
Id	Int	11	Primary key	Unique id
Student_id	Int	11	Foreign key	Unique student id
Teacher_id	Int	11	Foreign key	Unique teacher id
Batch_id	Int	11	Foreign key	Unique batch id
subject	Varchar	100	NOT NULL	Subject
Exam_type	Varchar	100	NOT NULL	Type of the exam
Marks_obtained	Decimal	5,2	NOT NULL	Marks obtained
Max_maks	Decimal	5,2	NOT NULL	Maximum marks
Uploaded_at	Timestamp		NOT NULL	Uploaded time

4.4 DATAFLOW DIAGRAM

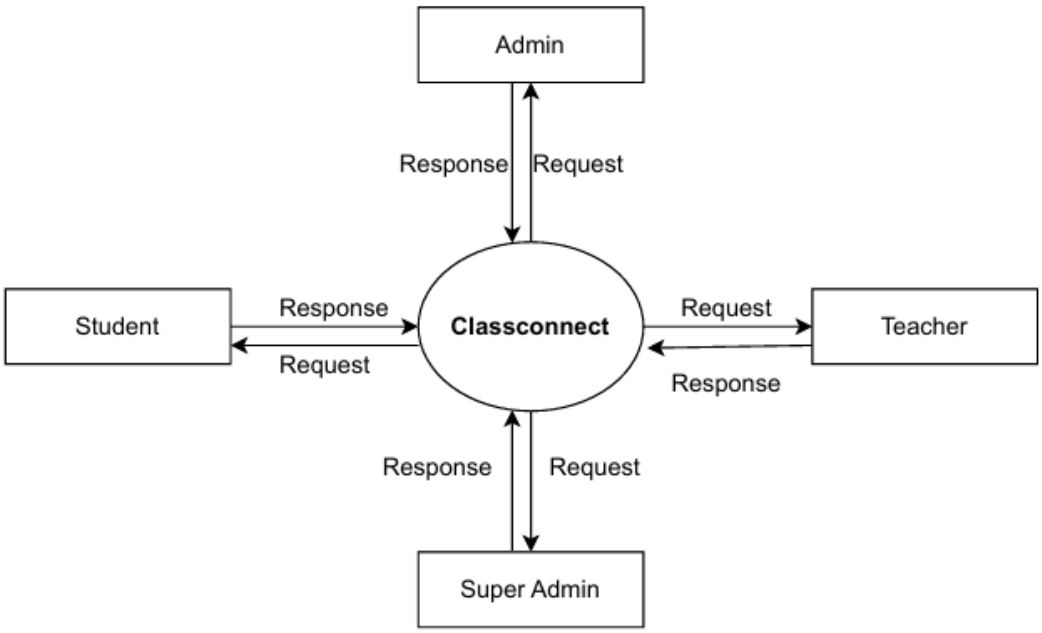
A DataFlow Diagram (DFD) is a graphical representation of the “flow” of data through an information system .A data flow diagram canals used for the visualization of data processing. DataFlow Diagram is a common practice for a designer to draw a context-level DataFlow Diagram first which shows the interaction between the system and out side entities . A DataFlow Diagram is a network that describes the flow of data and processes that change ,or transform , data throughout the system. This network is constructed by using a set of symbols that do not imply a physical implementation .It is a graphical tool for structure analysis of the system requirements .DFD models a system by using external entities from which data flows to a process ,which transforms the data and creates ,output data flows which go to other processes or external entities or files .Data in files may also flow to processes as inputs .There are various symbols used in a DFD .Bubbles represent the CLASSCONNECT processes. Named arrows indicate the data flow. External entities are represented by rectangles and are outside the system such as venders or customers with whom the system interacts. They either supply or consume data are called sinks. Data are stored in a data store process in the system. Each component in a DFD is labelled with a descriptive name, Process names are further identified with a number. The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the inputs(source),

all in a format that meet the user's requirements. The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow

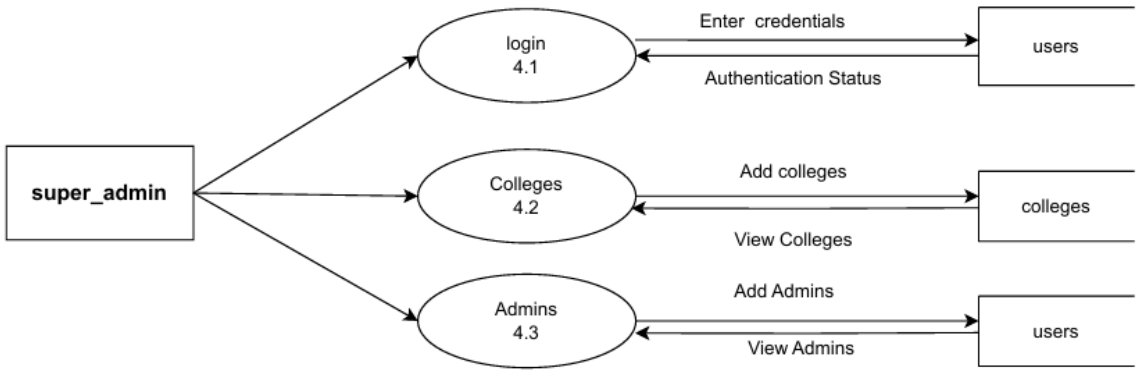
In the normal convention a DFD has four major symbols:

	It represents data source or destination
	It represents the data store
	It represents the data flow
	It represents a process

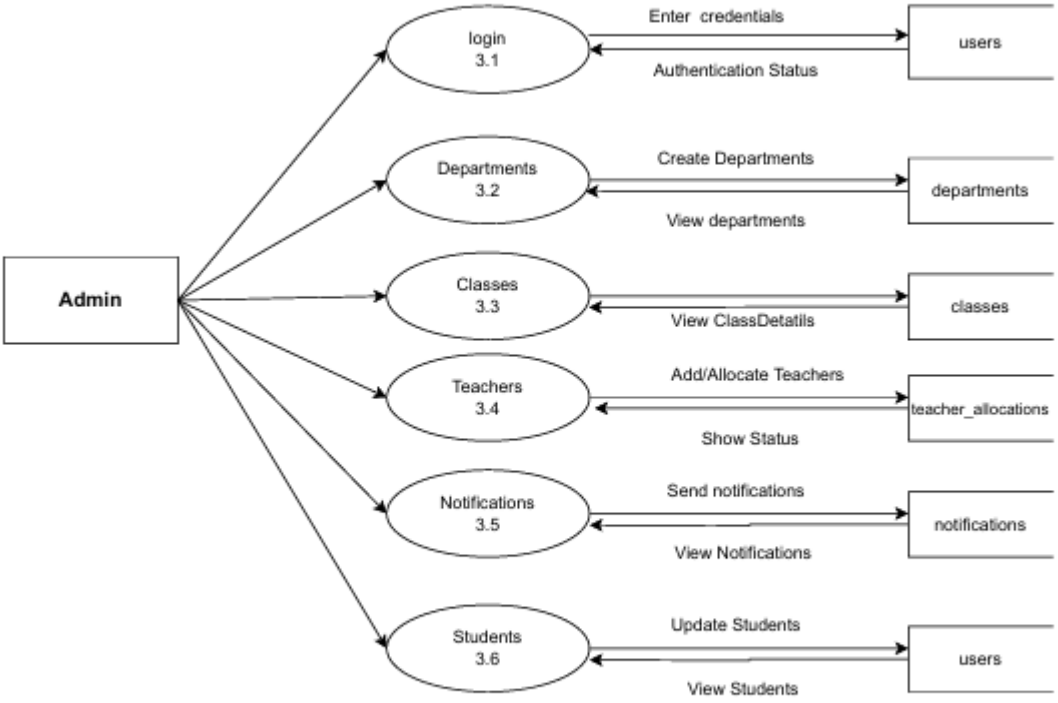
LEVEL 0 DFD (Classconnect)



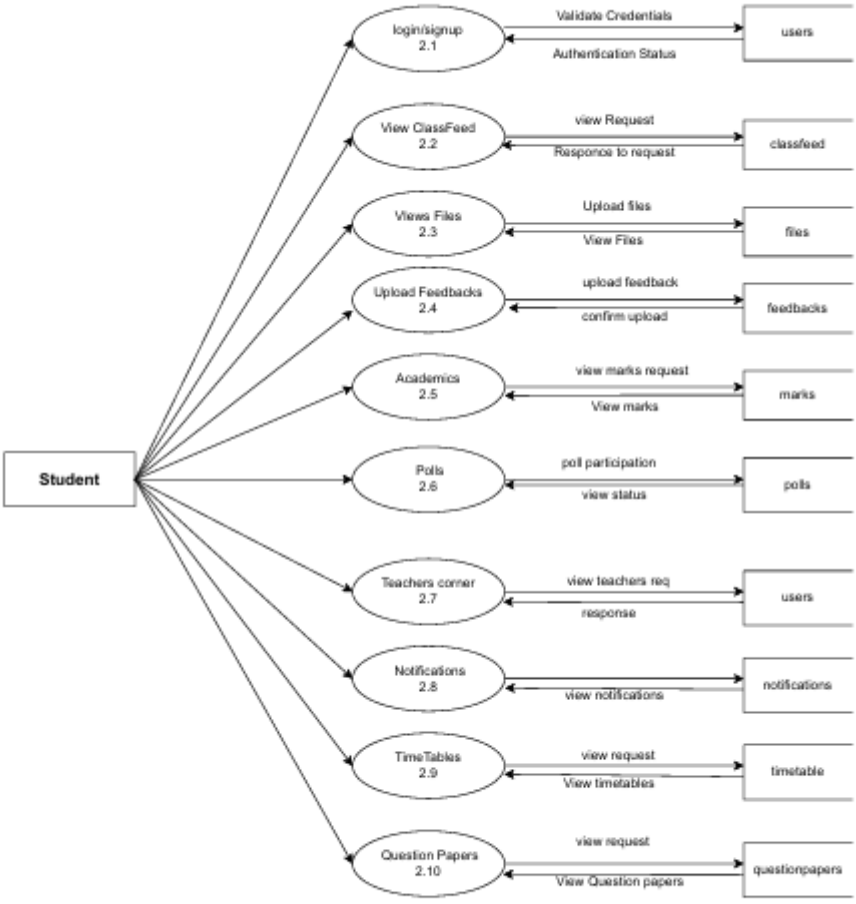
LEVEL 1 DFD (Super_admin)



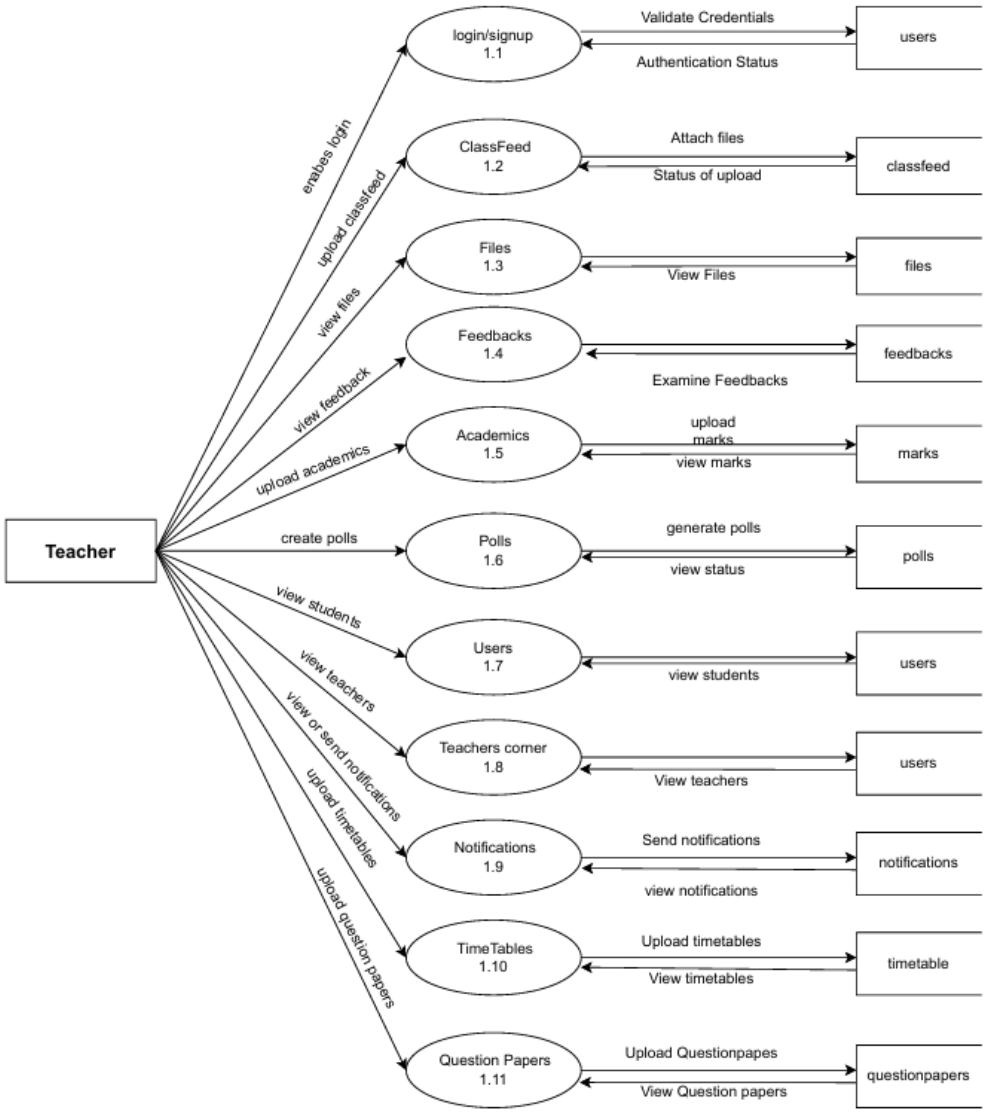
LEVEL 1 DFD(Admin)



LEVEL 1 DFD(Student)

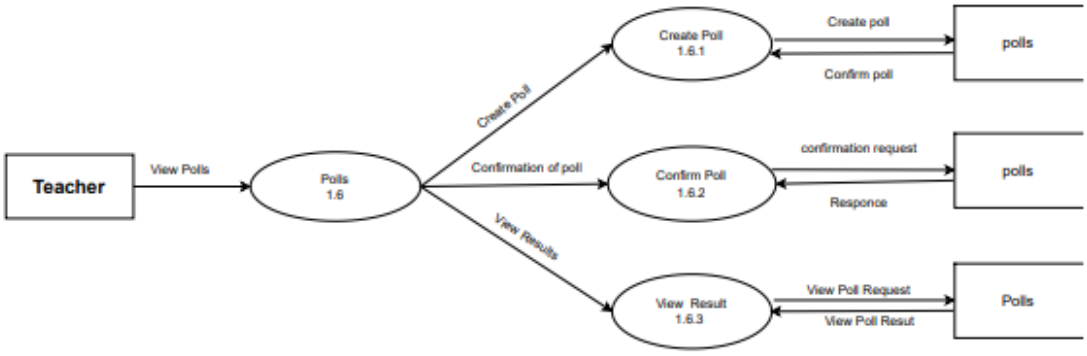


LEVEL 1 DFD (Teacher)



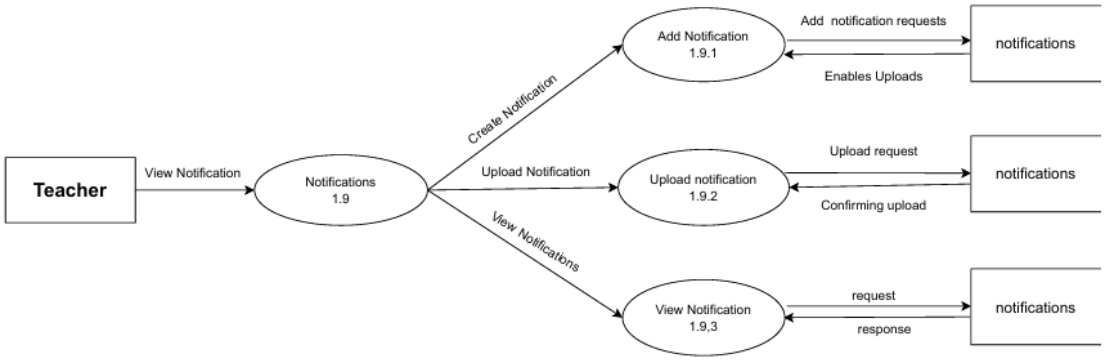
LEVEL 2 DFD(Teacher)

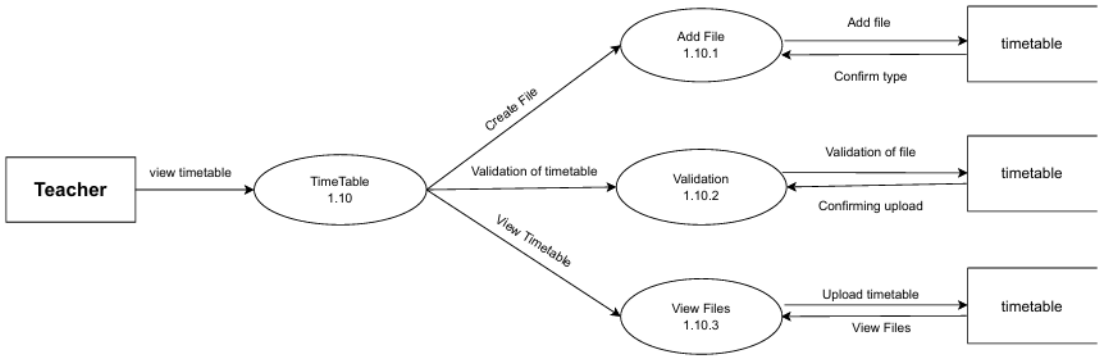

Teacher
view results
Results
1.6
View Results
1.5.1

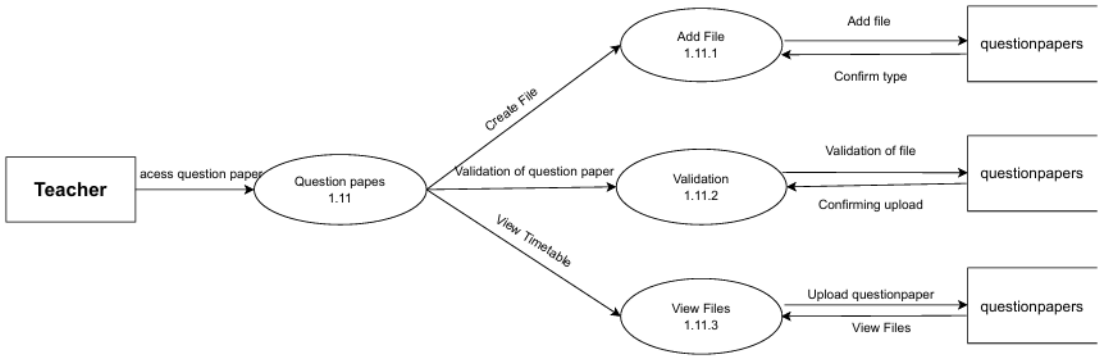

Teacher
View Polls
Polls
1.6
Create Poll
Confirmation of poll
View Results
Create Poll
1.6.1
Confirm Poll
1.6.2
View Result
1.6.3
Create poll
Confirm poll
confirmation request
Responce
View Poll Request
View Poll Resut
polls
polls
Polls


Teacher
Students
Users
1.7
View Student details
View Students
1.7.1
Request
Response
users

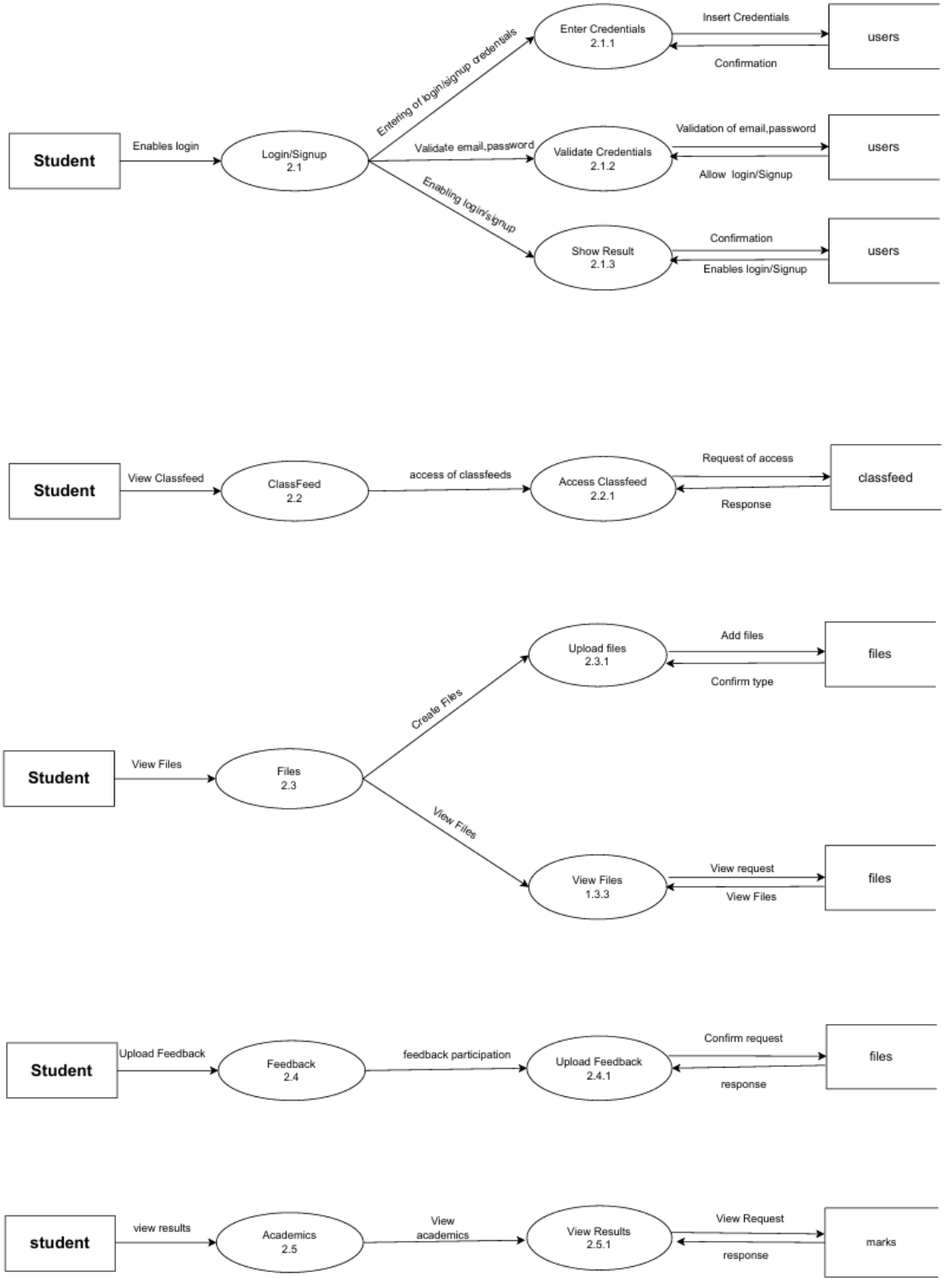

Teacher
View teachers
Teacher's Corner
1.8
View Teachers
View Teachers list
1.8.1
View teachers request
View teachers
teacher_allocations

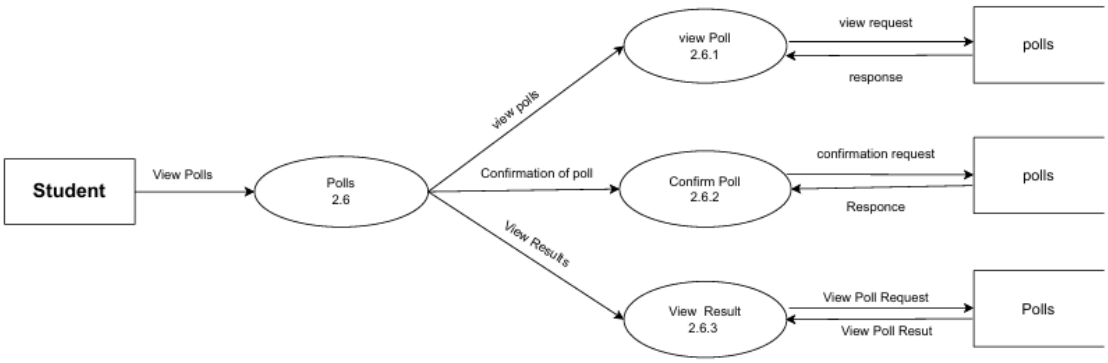

Teacher
View Notification
Notifications
1.9
Create Notification
Upload Notification
View Notifications
Add Notification
1.9.1
Upload notification
1.9.2
View Notification
1.9.3
Add notification requests
Enables Uploads
Upload request
Confirming upload
request
response
notifications
notifications
notifications

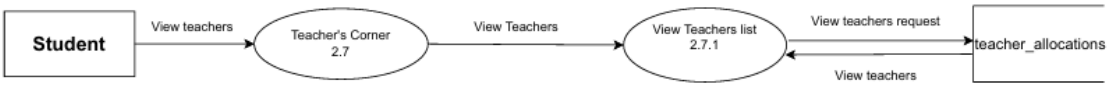

Teacher
view timetable
TimeTable
1.10
Create File
Validation of timetable
View Timetable
Add File
1.10.1
Validation
1.10.2
View Files
1.10.3
Add file
Confirm type
Validation of file
Confirming upload
Upload timetable
View Files
timetable
timetable
timetable


Teacher
acess question paper
Question papes
1.11
Create File
Validation of question paper
View Timetable
Add File
1.11.1
Validation
1.11.2
View Files
1.11.3
Add file
Confirm type
Validation of file
Confirming upload
Upload questionpaper
View Files
questionpapers
questionpapers
questionpapers

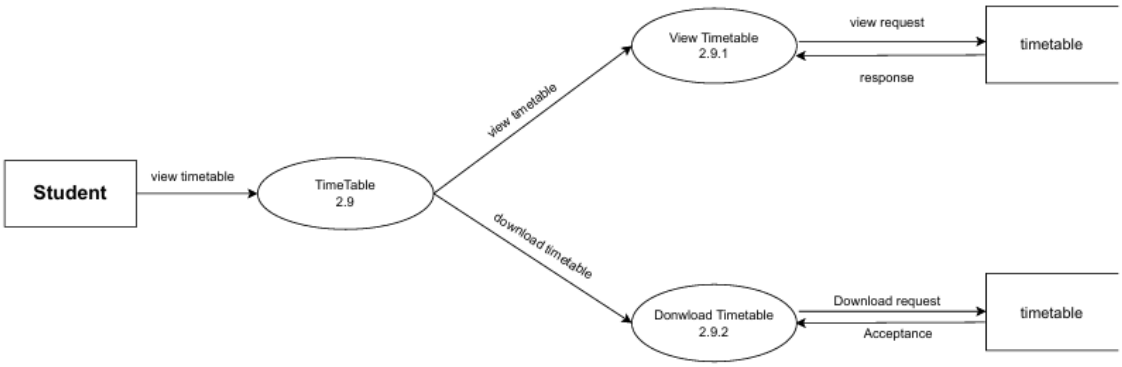
LEVEL 2 DFD(Student)

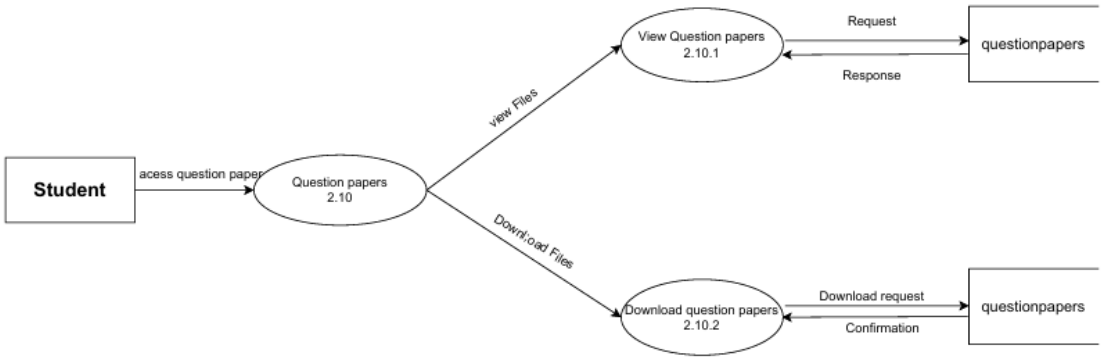



Student
View Polls
Polls
2.6
view polls
Confirmation of poll
View Results
view Poll
2.6.1
Confirm Poll
2.6.2
View Result
2.6.3
view request
response
confirmation request
Responce
View Poll Request
View Poll Resut
polls
polls
Polls

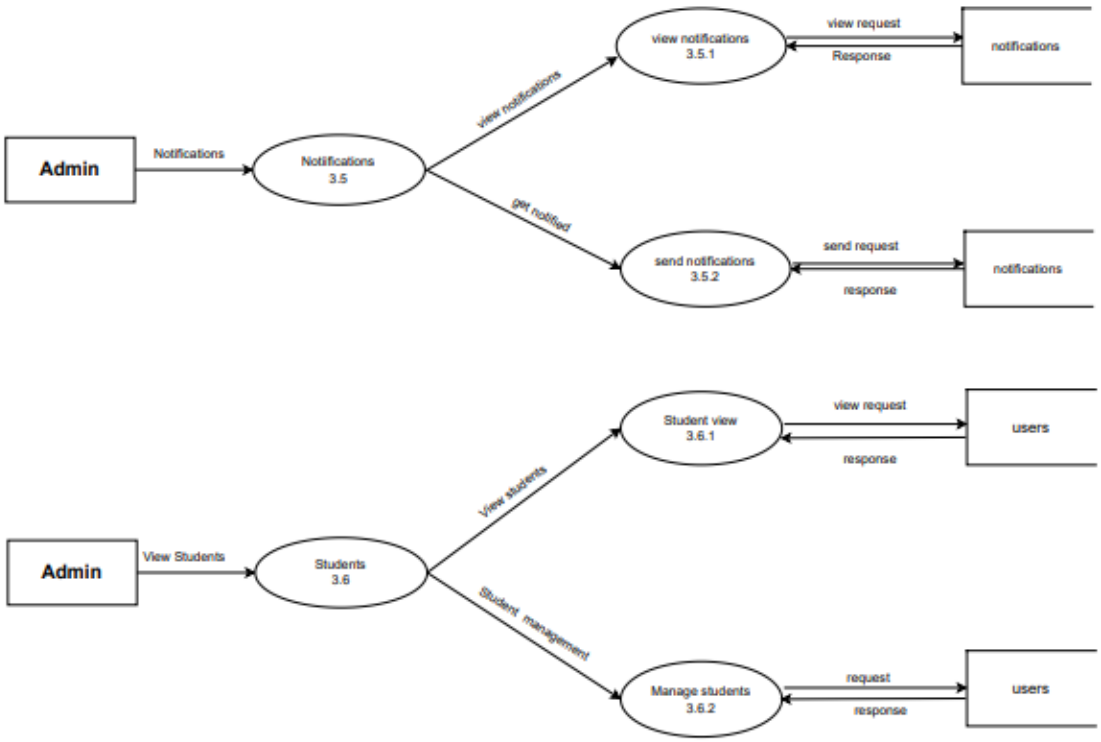

Student
View teachers
Teacher's Corner
2.7
View Teachers
View Teachers list
2.7.1
View teachers request
View teachers
teacher_allocations


Student
View Notification
Notifications
2.8
view notifications
view notifications
'2.8.1
view request
confirmation
notifications


Student
view timetable
TimeTable
2.9
view timetable
download timetable
View Timetable
2.9.1
Donwload Timetable
2.9.2
view request
response
Download request
Acceptance
timetable
timetable


Student
acess question paper
Question papers
2.10
view Files
Downl oad Files
View Question papers
2.10.1
Request
Response
questionpapers
Download question papers
2.10.2
Download request
Confirmation
questionpapers

LEVEL 2 DFD(Admin)



4.5 INPUT DESIGN – CLASSCONNECT

Admin Module:

- Login Page**:**
  Admin enters credentials to access the admin dashboard.
- Dashboard Navigation**:**
  Access sections like User Management, ClassFeed management, Polls, Feedback, and Notifications.
- User Management**:**
  Add, update, or remove students and teachers; assign roles.
- ClassFeed Management**:**
  Moderate posts and attached files (delete/update).

- Poll Management**:**
  Create, edit, or delete polls; monitor results.
- Feedback Handling**:**
  Track feedback submissions, respond, and update status.
- Notification Management**:**
  Send system-wide notifications to students and teachers.

Teacher Module:

- Login Page**:**
  Teacher enters credentials to access their dashboard.
- Dashboard Navigation**:**
  Access sections like ClassFeed, Polls, Feedback, Academics, and Files.
- ClassFeed Interaction**:**
  Create posts, upload files, and respond to comments.
- Poll Management**:**
  Create polls for students, monitor votes, and view results.
- Feedback Handling:
  View and respond to student feedback.
- Academics Module**:**
  Upload and update student marks.
- File Upload:
  Upload academic resources, notes, or study materials.

Student Module:

- Registration Page:
  Create an account to access Classconnect.

- Login Page:
  Enter credentials to access student dashboard.
- ClassFeed Interaction:
  View posts, comment, and like posts.
- Poll Participation:
  Vote in polls and view results.
- Feedback Submission:
  Submit categorized feedback with optional anonymity.
- View Academics:
  Access personal marks uploaded by teachers.
- File Access:
  Download or view notes, assignments, and question papers.
- Notifications:
  Receive alerts on new posts, polls, feedback responses, and timetable changes.

4.6 OUTPUT DESIGN –CLASSCONNECT

Admin Module:

- Dashboard Overview**:**
  Summary of all posts, polls, feedback status, and notifications.
- User List:
  Display all students and teachers with details and roles.
- ClassFeed Reports:
  List of all posts and files, with options to moderate or delete.
- Poll Results:
  Overview of all polls with total votes.
- Feedback Reports:
  View all feedback, status, and responses.

- Notifications:
  List of all system notifications sent.

Teacher Module:

- Dashboard Overview:
  Summary of posts, polls, student feedback, and uploaded files.
- ClassFeed Display:
  View own posts and attached files; see student interactions.
- Poll Results:
  Poll outcomes and participation statistics.
- Feedback Reports:
  Student feedback directed to the teacher with response options.
- Academics Reports:
  Student marks uploaded by the teacher, organized by subject and exam.
- File Repository:
  Uploaded academic resources organized by subject.

Student Module:

- Dashboard Overview:
  Summary of unread posts, active polls, feedback status, and notifications.
- ClassFeed Display:
  List of posts with files and comments.
- Poll Results:
  View results after voting.
- Feedback Status:
  Confirmation of feedback submission and response tracking.
- Academics Display:
  Personalized marks view.

- File Repository:

  Access to notes, assignments, and question papers.

- Notifications Popup:

  Alerts for posts, polls, feedback replies, and timetable changes.

5.SYSTEM DEVELOPMENT

5.1 INTRODUCTION

This section provides a high-level overview of the system development process for ClassConnect:

Requirements Gathering:

- Define functional and non-functional requirements based on student and teacher needs.
- Determine input/output designs, algorithms, and technologies to be used (PHP, MySQL, HTML, CSS, JavaScript).
- Collect input from students, teachers, and admin stakeholders to understand collaboration, communication, and academic management requirements.

Data Collection and Preparation:

- Prepare data required for ClassConnect, including student and teacher details, academic records, posts, polls, feedback, and files.
- Ensure data consistency, proper categorization, and relevance to support features like ClassFeed, Polls, Feedback, Academics, and Notifications.

System Development:

- Develop ClassConnect based on the defined requirements and feedback from stakeholders.
- Implement modules for user registration, ClassFeed, polls, feedback, academic marks, notifications, and file sharing.
- Design an intuitive user interface for Admins**,** Teachers**,** and Students, ensuring ease of navigation and task completion.
- The development process is iterative, allowing incorporation of new data, feedback, and features to improve system performance and user satisfaction.

5.2 MENU LEVEL DESCRIPTION

Admin Menu:

- Home Page**:** Landing page for admin to access all sections like User Management, ClassFeed, Polls, Feedback, Academics, and Notifications.
- User Management: View, add, update, or remove student and teacher accounts.
- ClassFeed Management: Monitor and moderate posts and files shared by users.
- Poll Management: Create, update, and archive polls.
- Feedback Management: View and respond to student feedback.
- Academics Management: Oversee academic records and marks uploaded by teachers.
- Notifications: Send alerts and updates to students and teachers.

Teacher Menu:

- Home Page**:** Landing page showing dashboard summary (posts, polls, feedback, academics).
- ClassFeed**:** Create posts, upload files, and interact with students' comments.
- Polls: Create polls, monitor votes, and view results.
- Feedback: View and respond to feedback from students.
- Academics: Upload and manage student marks.
- Files: Upload study materials and resources for students.
- Notifications: Receive alerts about new posts, feedback, and polls.

Student Menu:

- Home Page: Dashboard showing unread posts, active polls, feedback responses, and notifications.

- ClassFeed: View posts, comment, and like content.
- Polls: Participate in polls and view results after voting.
- Feedback: Submit feedback with optional anonymity.
- Academics: View marks uploaded by teachers.
- Files: Access study materials, notes, and question papers.
- Notifications: Receive alerts for posts, poll updates, feedback replies, and timetable changes.

5.3 PROCESS SPECIFICATION

1. Administrative Functions:

1. Manage Users**:** Admin can add, update, or remove student and teacher accounts.
2. Moderate ClassFeed: Admin can monitor and delete inappropriate posts or files.
3. Poll Oversight: Admin can create polls, update settings, and archive old polls.
4. Feedback Handling**:** Admin can view student feedback and reply as needed.
5. Notifications: Admin can send system-wide notifications to teachers and students.

2. Teacher Functions:

1. Create Posts: Teachers can add posts with optional file attachments in ClassFeed.
2. Poll Management**:** Teachers can create polls, track votes, and view results.
3. Feedback Response: Teachers can view and reply to student feedback.
4. Upload Marks: Teachers can enter and update student academic marks.

5. Upload Files**:** Teachers can upload notes, assignments, or other academic resources.
6. Notifications**:** Teachers receive alerts for new posts, feedback, and poll activities.

3. Student Functions:

1. View ClassFeed**:** Students can read posts, comment, and like posts.
2. Participate in Polls**:** Students can vote and see poll results.
3. Submit Feedback**:** Students can provide feedback with categories and optional anonymity.
4. View Academics**:** Students can check marks uploaded by teachers.
5. Access Files: Students can download notes, question papers, and study materials.
6. Receive Notifications**:** Students are notified about new posts, poll updates, feedback responses, and timetable changes.

6. SYSTEM TESTING

6.1 TESTING METHODS

The purpose of system testing is to identify and correct errors in the Classconnect system. Testing is a vital part of software quality assurance and represents the final review of design, logic, and coding. Since Classconnect is a web-based platform intended for class communication and management, thorough testing ensures smooth performance, reliability, and a bug-free experience for students and admins.

Unit Testing

Unit testing focuses on testing individual modules of ClassConnect such as the Login System, Class Feed, Feedback Section, Poll System, and Question Paper Upload Module. Each module was tested separately to verify that it delivers the expected output for given inputs. Validation and logical conditions were tested at the module level. Example: The login module was tested to ensure only valid credentials grant access, while invalid data triggers proper error messages. Similarly, the feedback form validators were tested for required fields, text length limits, and valid rating inputs.

Integration Testing

After all modules were verified individually, they were integrated to test the interaction between them. The main focus of this testing was to ensure that components communicate correctly — for example, when a student logs in and posts in the Class Feed, the notification system and database should update accordingly. Integration testing ensured that connected modules such as authentication, posting, file uploads, and user roles worked seamlessly together without data mismatches or conflicts.

Validation Testing

Validation testing ensures that classconnect fulfils all functional requirements as expected by the users (students,teachers,admin).It verifies that the system perfoms in accordance with the original specifications. All forms in classconnect (eg registration

from,feedback form ,poll creation form)includes validations such as required field,validator,regular expression validator and compare validator to maintain data integrity.For example email fields only accepts proper email formats,and numeric fields reject alphabetic inputs

System Testing

System testing is the final stage before deployment, where the complete ClassConnect system was tested as a whole. The goal was to verify that the entire system works according to the defined requirements and performs smoothly under realistic conditions.

This includes testing all functionalities such as login/logout, feed posting, feedback submission, poll participation, and file uploads. Any detected issues were fixed before the acceptance phase to ensure a stable release version.

Verification Testing

Verification testing ensured that each stage of Classconnect's development — from documentation to coding — followed the specified requirements. This testing verified whether the system was built correctly according to design. It involved reviewing code, flowcharts, and module structures to confirm consistency with the intended functionality before moving on to validation.

Usability Testing

Usability testing focused on evaluating how easily students and admins can use Classconnect. The system was tested by actual users to check navigation flow, clarity of interface, and ease of performing tasks like posting, viewing polls, and submitting feedback. The result showed that Classconnect is intuitive and user-friendly, requiring minimal training for new users.

Regression Testing

Whenever changes or updates were made — such as UI improvements or backend modifications — regression testing was performed to ensure that new code did not break existing features. For instance, after implementing the role-based access system, previously working modules like the feed and notifications were re-tested to confirm stable performance.

Testing Environment & Tools

Classconnect was tested on a local development environment using:

- Backend**:** PHP 8.x, MySQL Database
- Frontend**:** HTML5, CSS3, JavaScript, Bootstrap
- Testing Tools**:** Browser Developer Tools, XAMPP Server Logs, Manual Testing via Test Data

Debugging (Black Box Testing & White Box Testing)

BlackBox Testing:
This method was used to test the external behavior of ClassConnect without looking into the internal code. For example, the signup form was tested with invalid emails, blank fields, and incorrect passwords to verify validation messages and response handling.

WhiteBox Testing:
This method focused on the internal logic and flow of PHP scripts and SQL queries.
Developers checked conditions, loops, and data handling functions to ensure the logic worked correctly behind the scenes.

6.2 TEST PLAN ACTIVITIES

Testing started alongside the development of Classconnect and continued through to final validation. The test plan defined the scope, approach, and schedule for each testing phase, along with assigning responsibilities to team members.

Test Plan Included**:**

- Test Unit Specification**:** Modules like login, feed, feedback, polls, and profile were identified as test units.
- Features to be Tested**:** Authentication, role-based access, CRUD operations, file uploads, and real-time interactions.
- Approach for Testing**:** Combination of manual testing and black box testing.
- Test Deliverables: Test cases, test data, bug reports, and system test results.
- Personnel Allocation: Developers performed unit and integration testing; system validation and usability testing were done by selected end users (students and admins).

7.SYSTEM IMPLEMENTATION

System implementation is the process of transforming the ClassConnect system from the design and development phase into a fully functional, operational platform accessible to students and administrators. The implementation phase ensures that the developed system works effectively in the real environment and meets all user requirements.

The following key activities were carried out during the implementation of ClassConnect:

1. Installation

   The ClassConnect application was installed on a local server environment using XAMPP, which provides support for PHP and MySQL. The database was created, configured, and connected to the backend scripts to ensure smooth data flow between the application and the database.

2. Configuration

   After installation, system configurations were customized to match the specific requirements of the college environment. This included setting up user roles (Admin and Student), database connections, and defining parameters such as file upload limits and session timeouts. Each configuration ensured secure and efficient use of the system.

3. Data Migration

   Initial data, such as user credentials, class information, and sample posts, were entered into the database to simulate real-time usage. Since ClassConnect was a newly developed system, no old system migration was required, but the module supports future import of existing class data if needed.

4. Testing

   Before full deployment, comprehensive testing was performed to ensure that all components functioned as expected. This included unit testing**,** integration testing, and system testing. Each module—such as the login system, class feed, polls, and feedback—was tested using various test cases to identify and correct bugs. Testing confirmed that the system met performance and functional requirements.

   End-users, including students and administrators, were provided with basic training on how to use the ClassConnect interface.

6. Documentation

Comprehensive documentation was created for the ClassConnect system. It includes:

- User Manual**:** Provides step-by-step instructions for operating the system.
- Technical Documentation**:** Contains details on system architecture, database schema, and source code structure.
- Maintenance Guide**:** Describes procedures for updating and troubleshooting the system.

7. Deployment

After successful testing and configuration, ClassConnect was deployed in a live environment. The application was hosted locally for initial evaluation and can be migrated to an online server for wider accessibility.

8. Maintenance

Regular maintenance activities were planned to ensure that the system remains stable and up to date. This includes checking for bugs, optimizing database performance, and ensuring compatibility with newer browsers and PHP versions.

9. Monitoring

The system performance and user activity are continuously monitored through logs and database records. This helps in identifying potential issues such as failed login attempts, large file uploads, or inactive users, allowing timely corrective actions.

10. Upgrades

Classconnect was designed with scalability in mind. Future upgrades, such as adding teacher-specific dashboards, real-time chat, or mobile app integration, can be implemented without altering the existing system architecture.

11. Support

Ongoing technical support is provided to handle any issues faced by users. The admin team can monitor reports, fix bugs, and manage user accounts to ensure smooth operation and user satisfaction.

8.CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT

CONCLUSION

The Classconnect project successfully demonstrates the design and development of a dynamic, user-friendly class management and collaboration system that enhances communication between students and administrators. Built using PHP, MySQL, HTML, CSS, and JavaScript, the system provides a secure and organized platform for sharing announcements, conducting polls, collecting feedback, and managing class-related updates.

Throughout the development process, the Prototype Model was followed, which allowed for iterative improvements and early validation of system requirements. The system's modular architecture ensures scalability, while the user interface emphasizes simplicity and accessibility for all users.

Core objectives achieved include:

- Secure user authentication with session handling and role-based access (Admin and Student).
- A fully functional Class Feed module for posting, commenting, and file sharing.
- An interactive Poll System for collecting student opinions in real time.
- A Feedback Section supporting anonymous and detailed feedback with rating features.
- Question Paper Management, enabling easy upload and organization of academic materials.
- Proper database management with relationships between colleges, departments, batches, and users.
- A responsive and structured user interface that adapts well to different devices.

ClassConnect demonstrates how a web-based application can simplify classroom communication, improve collaboration, and bring all academic interactions under one digital roof. The system meets its goals of being reliable, easy to use, and secure for both admins and students.

SCOPE FOR FUTURE ENHANCEMENT

While Classconnect successfully fullfills its core objectives, there are several opportunities to expand its functionality and improve user experience in future updates:

1. Enhanced User Experience

- Implement a dark/light mode theme for better accessibility.
- Enable profile customization (profile pictures, bios, and status).
- Add real-time notification**s** using AJAX or Web Sockets for instant updates.
- Introduce a mobile app version using Flutter for Android/iOS access.

2. Advanced Communication Features

- Add private chat or group chat between students and admins.
- Implement voice/video meeting integration using APIs like Zoom or Google Meet.
- Support emoji reactions and tagging in class posts for better engagement.

3. Academic Enhancements

- Include a subject management module for organizing materials by course.
- Integrate attendance tracking and assignment submission systems.
- Enable grade posting and academic performance reports.

4. Admin and Security Improvements

- Add multi-admin support with separate permissions (teacher, coordinator, HOD, etc.).
- Implement two-factor authentication (2FA) for increased login security.
- Introduce activity logs to monitor user interactions.
- Provide automated backup of databases and uploaded files.

5. Performance and Scalability

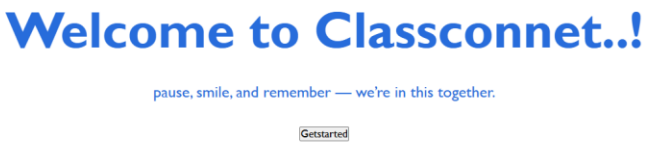
- Optimize backend performance using caching and pagination for large datasets.
- Host the project on cloud servers (AWS, Azure, or Digital Ocean) for better scalability.
- Use Docker containers for easier deployment and maintenance.

6. Smart Analytics and Insights

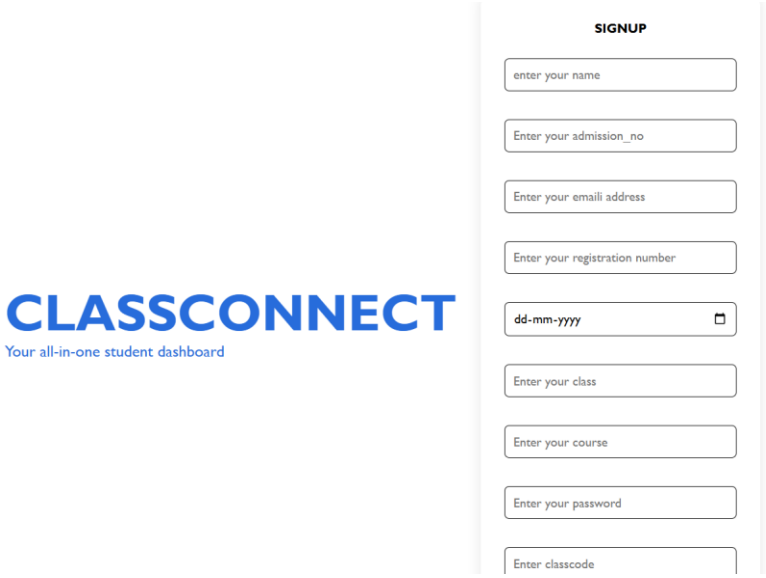
- Implement a dashboard that displays activity analytics, feedback summaries, and poll insights.
- Use AI-based suggestions for improving student engagement or identifying inactive users.

9.SCREEEN LAYOUT

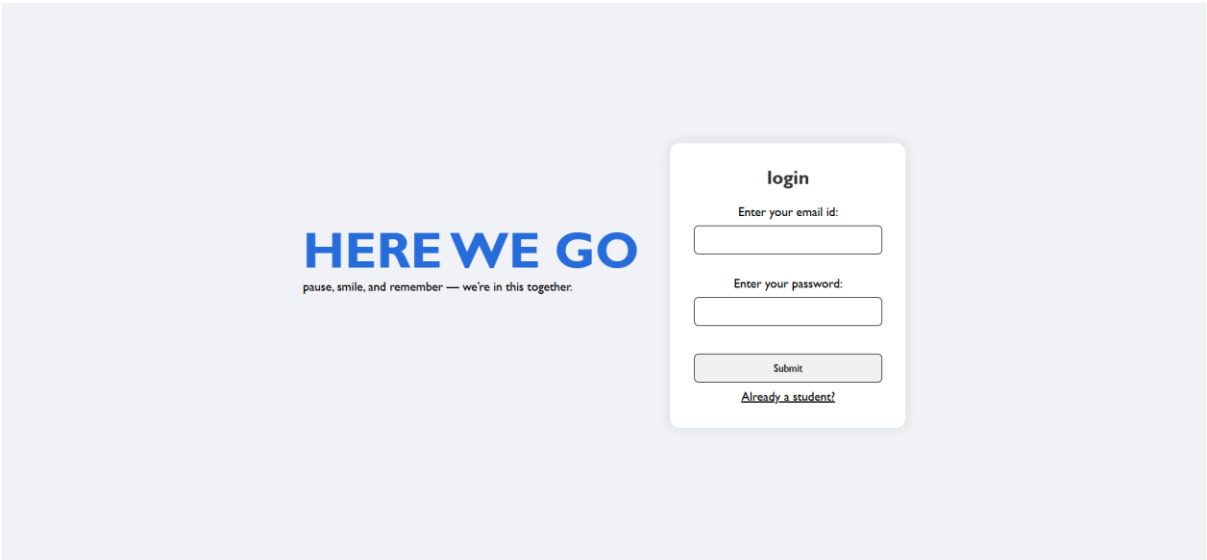
Landing Screen (super_admin/admin/teacher/student)



signup page (student)

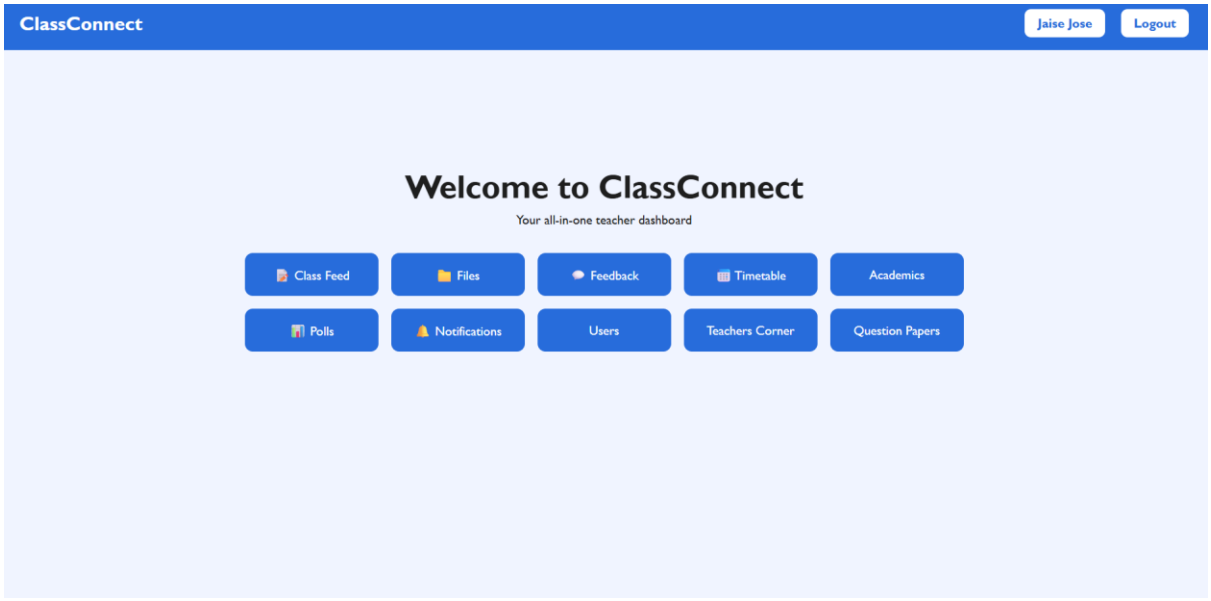


LoginPage (super_admin/admin/teacher/student)

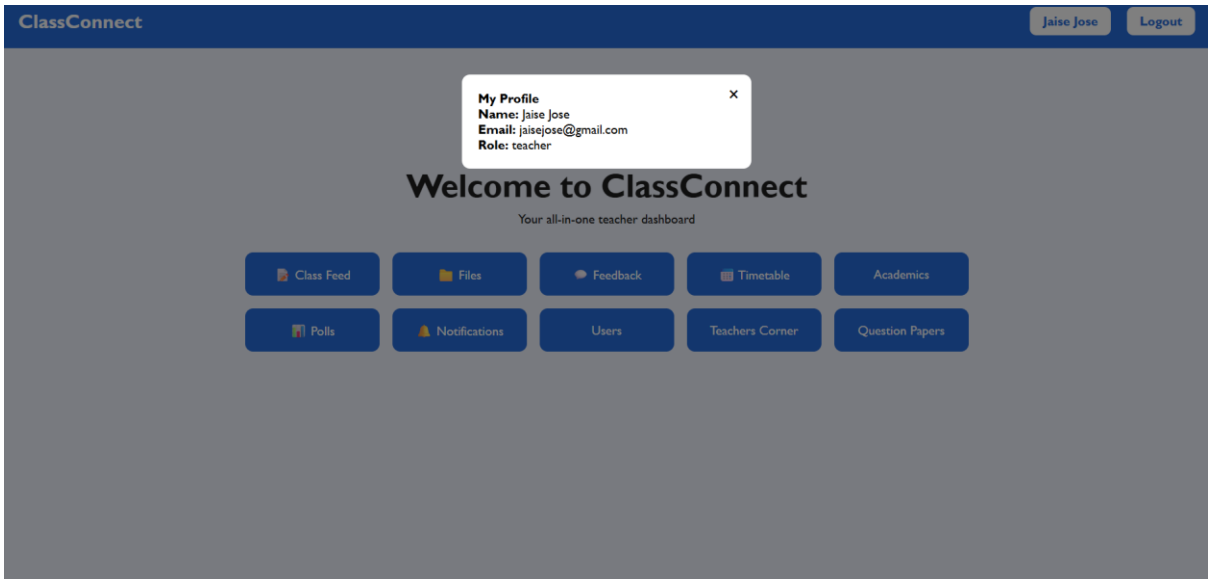


TEACHER SIDE

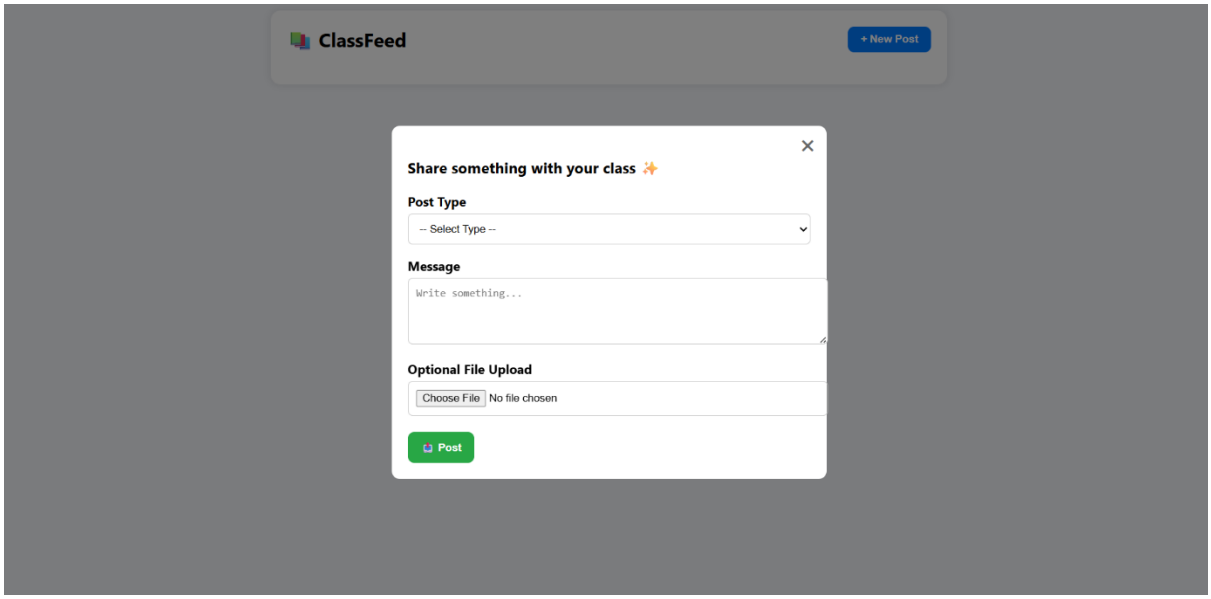
Teacher Dashboard

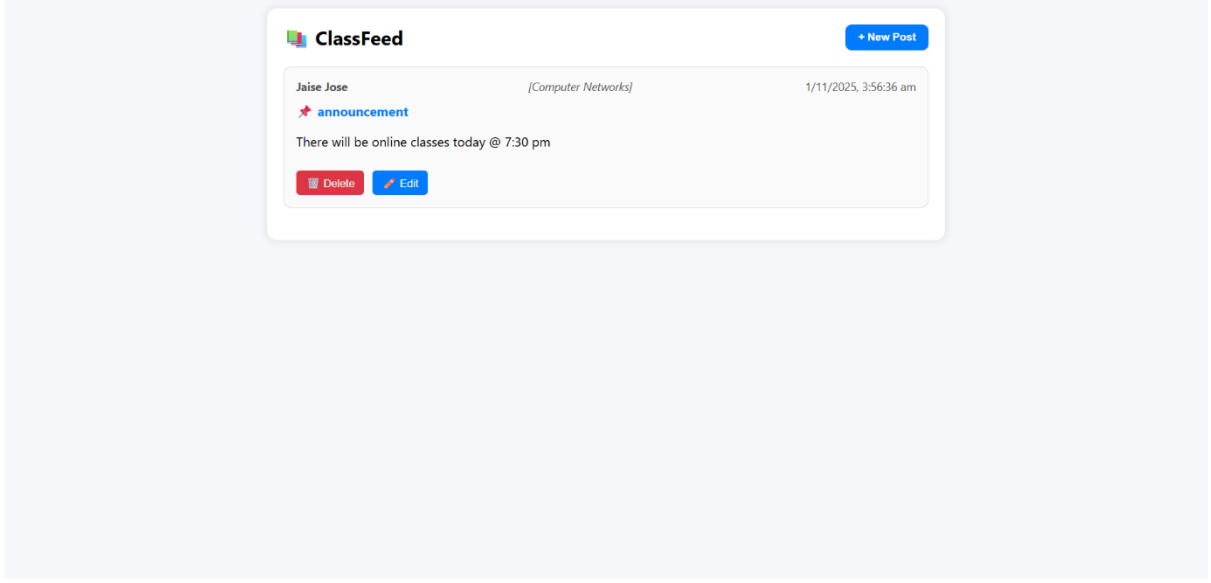


Profile View

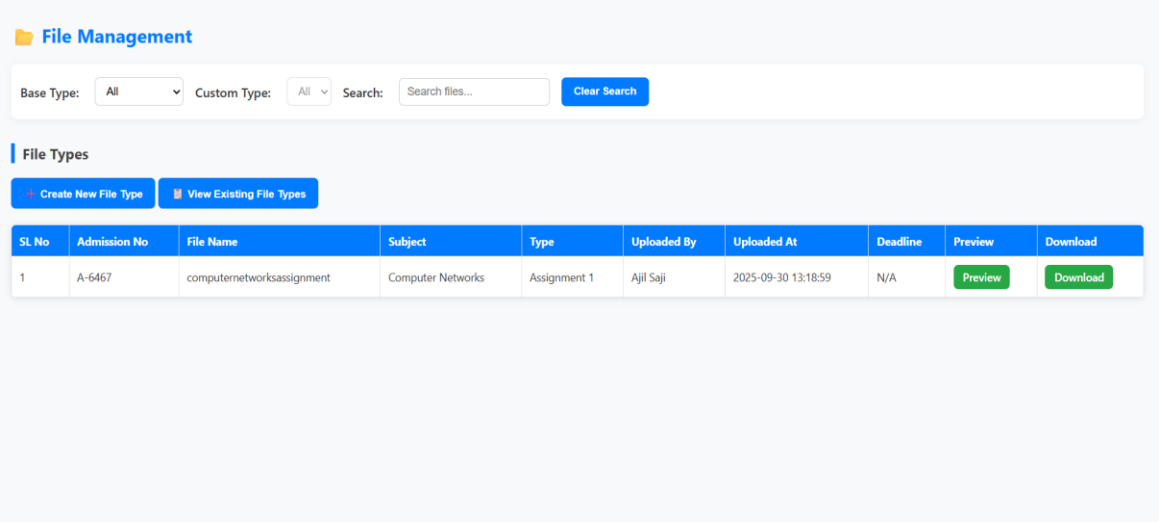


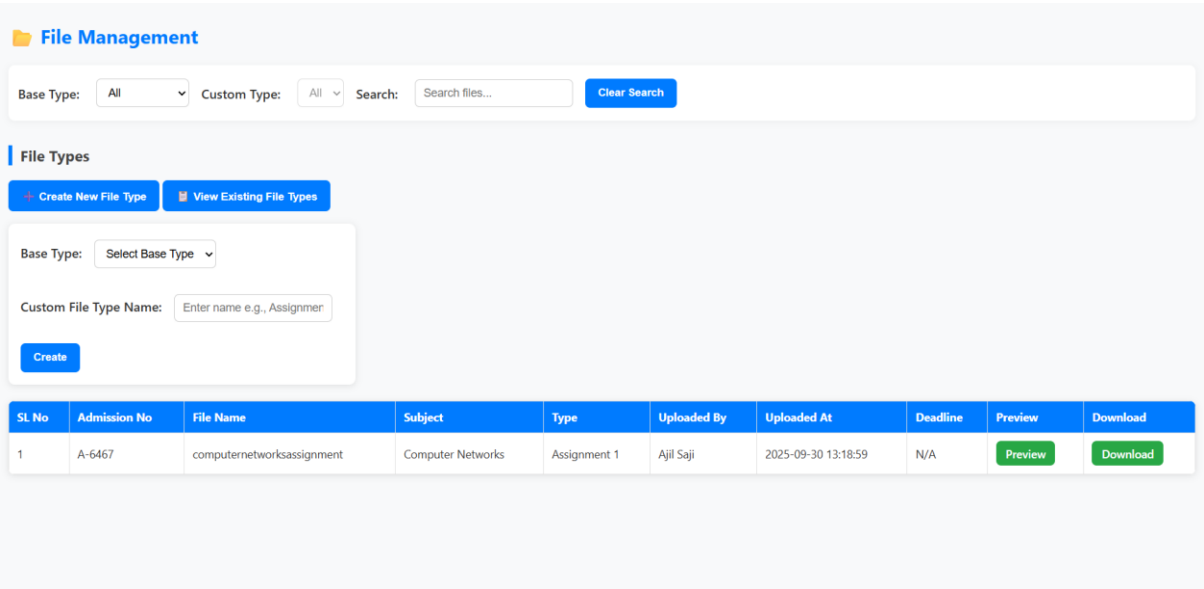
ClassFeed Upload



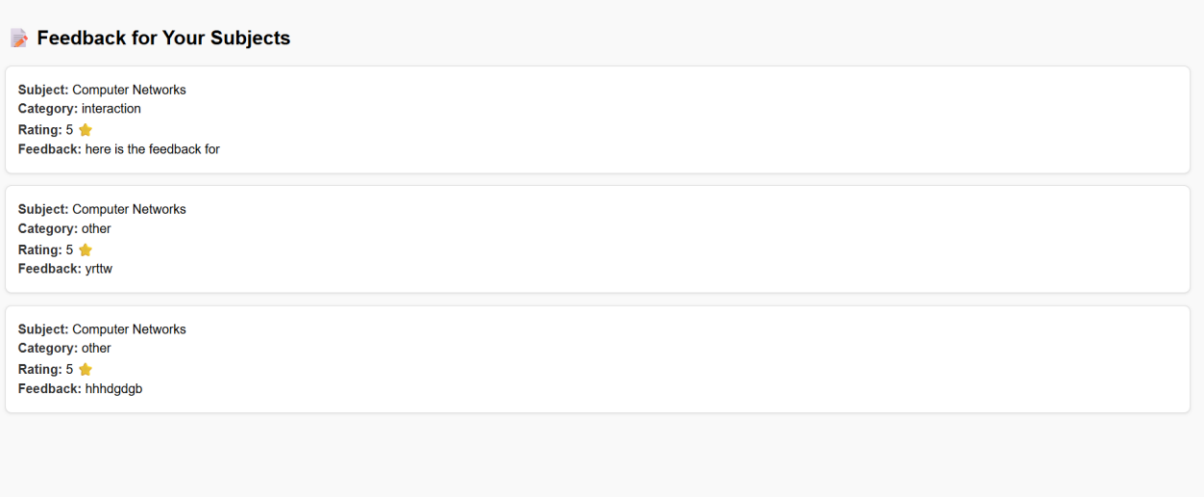


File View

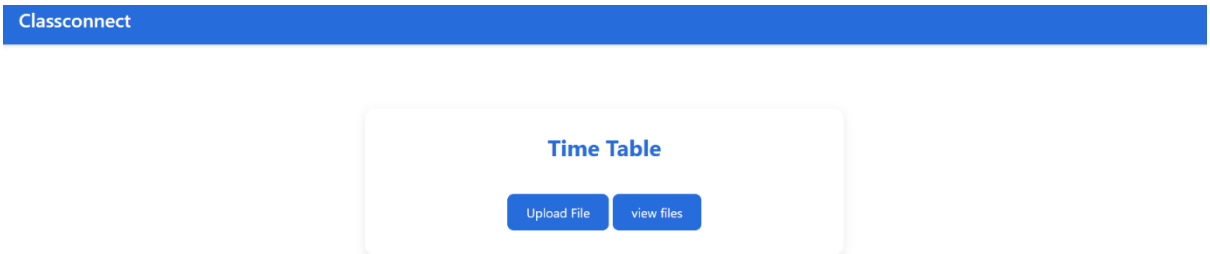

File Management
Base Type: All
Custom Type: All
Search: Search files...
Clear Search
File Types
Create New File Type
View Existing File Types
SL No
Admission No
File Name
Subject
Type
Uploaded By
Uploaded At
Deadline
Preview
Download
1
A-6467
computernetworksassignment
Computer Networks
Assignment 1
Ajil Saji
2025-09-30 13:18:59
N/A
Preview
Download

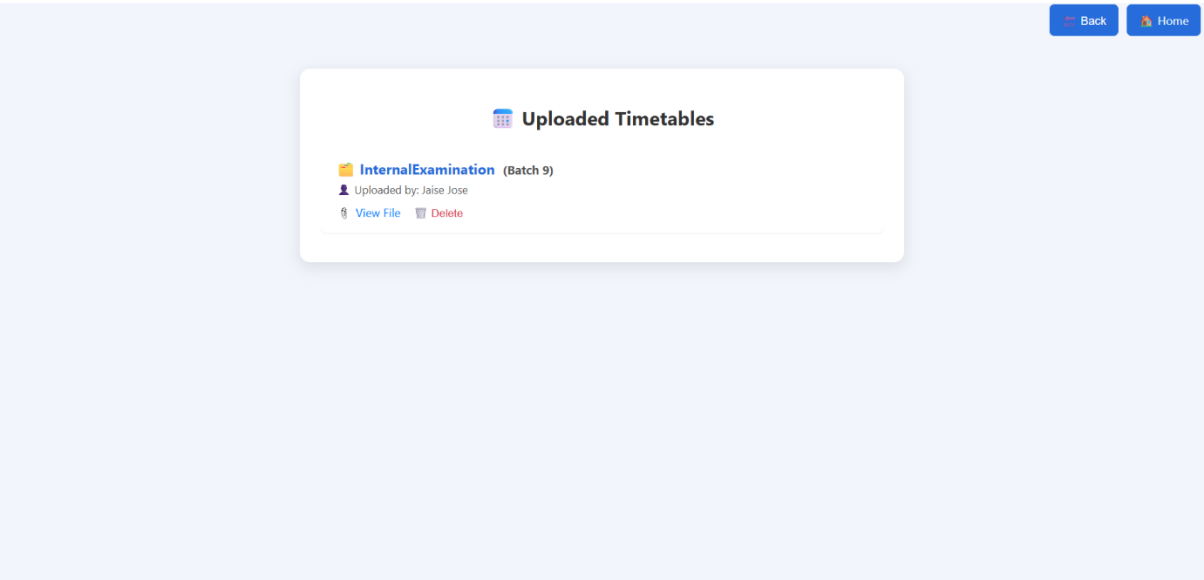

File Management
Base Type: All
Custom Type: All
Search: Search files...
Clear Search
File Types
Create New File Type
View Existing File Types
Base Type: Select Base Type
Custom File Type Name: Enter name e.g., Assignmer
Create
SL No
Admission No
File Name
Subject
Type
Uploaded By
Uploaded At
Deadline
Preview
Download
1
A-6467
computernetworksassignment
Computer Networks
Assignment 1
Ajil Saji
2025-09-30 13:18:59
N/A
Preview
Download

Feedback

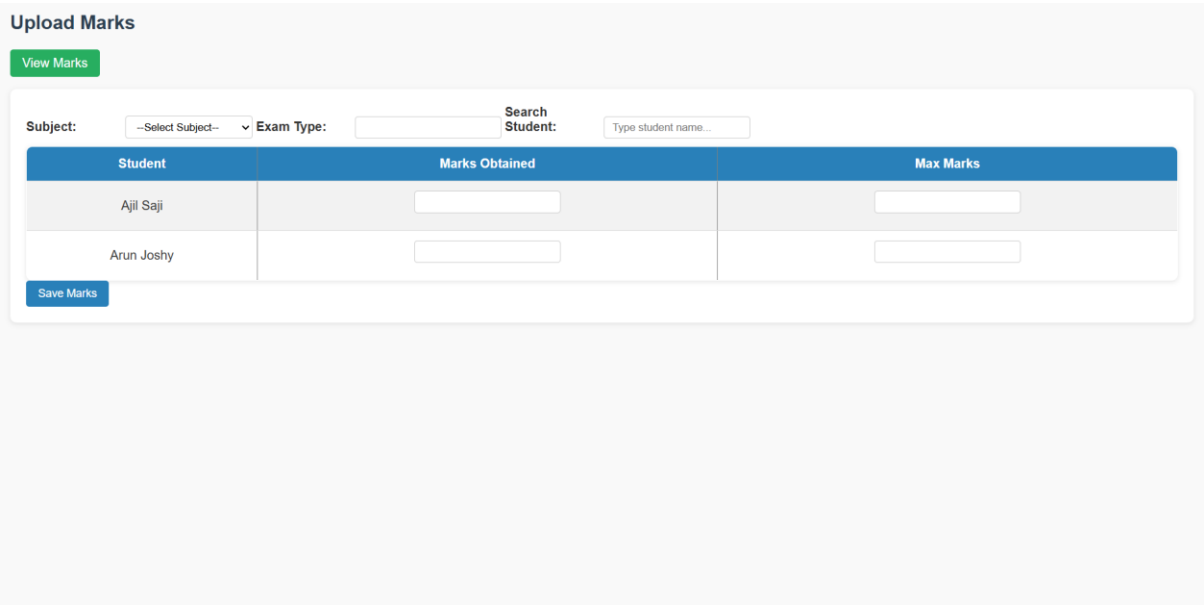


Timetable

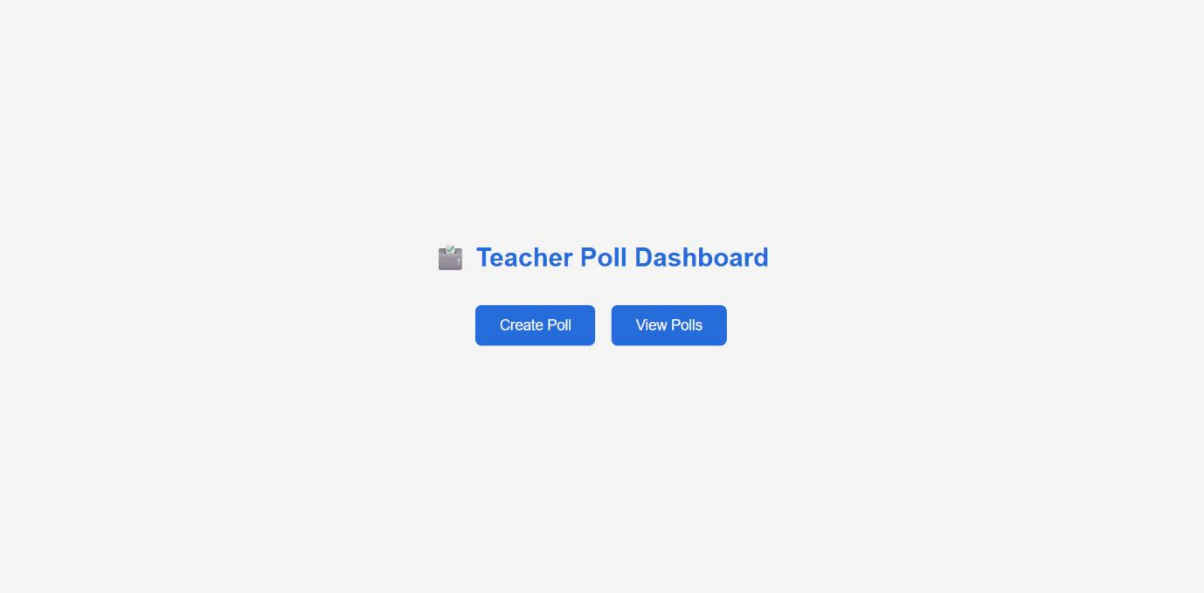


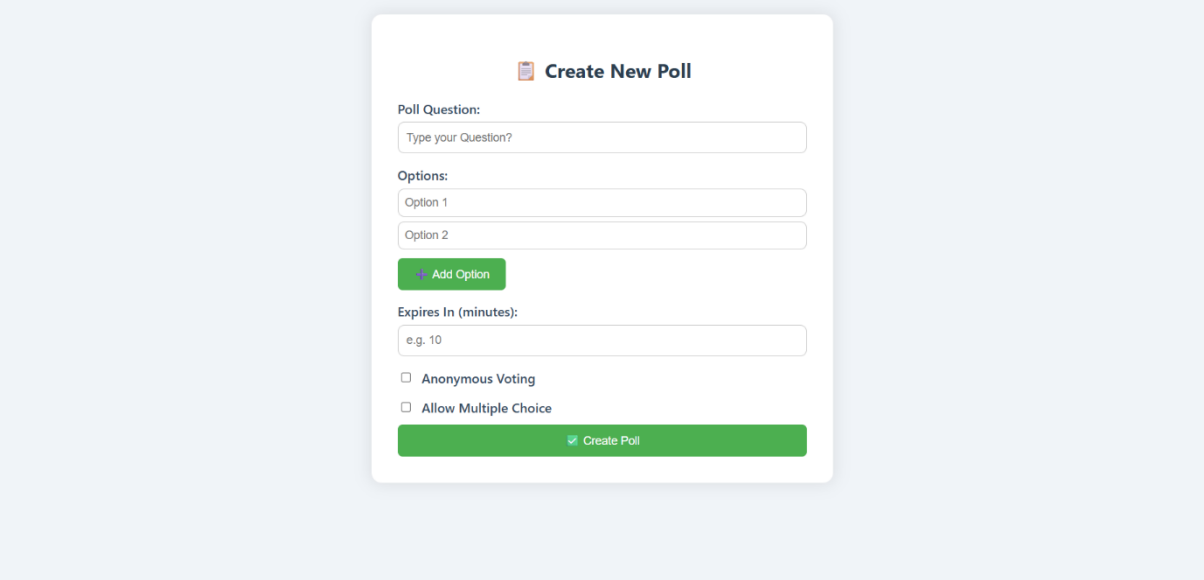

Back
Home
Uploaded Timetables
InternalExamination (Batch 9)
Uploaded by: Jaise Jose
View File
Delete

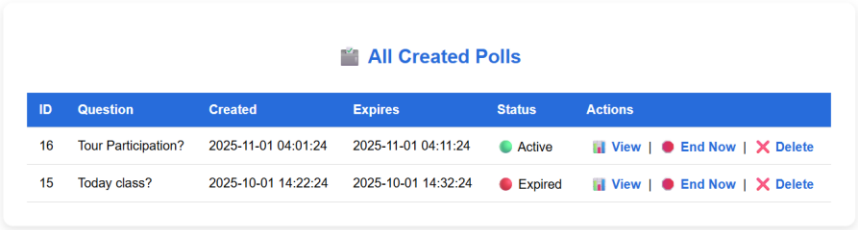
Academics

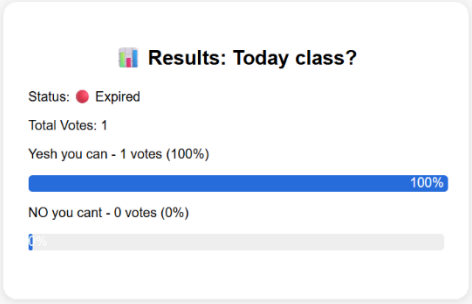

Upload Marks
View Marks
Subject:
--Select Subject--
Exam Type:
Search Student:
Type student name...
Student
Marks Obtained
Max Marks
Ajil Saji
Arun Joshy
Save Marks

Polls

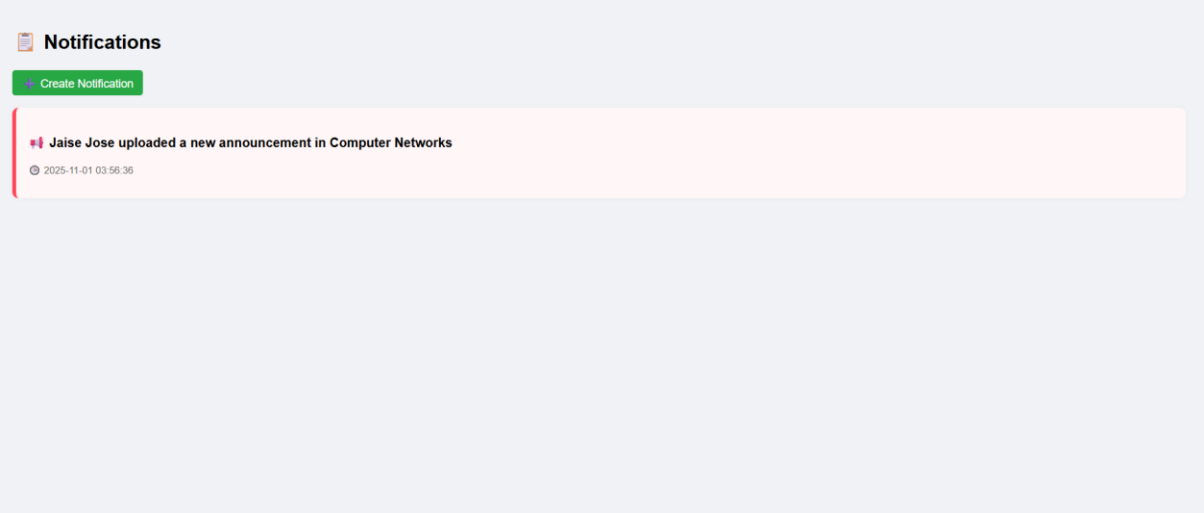

Teacher Poll Dashboard
Create Poll
View Polls


Create New Poll
Poll Question:
Type your Question?
Options:
Option 1
Option 2
Add Option
Expires In (minutes):
e.g. 10
Anonymous Voting
Allow Multiple Choice
Create Poll


All Created Polls
ID
Question
Created
Expires
Status
Actions
16
Tour Participation?
2025-11-01 04:01:24
2025-11-01 04:11:24
Active
View | End Now | Delete
15
Today class?
2025-10-01 14:22:24
2025-10-01 14:32:24
Expired
View | End Now | Delete


Results: Today class?
Status: Expired
Total Votes: 1
Yesh you can - 1 votes (100%)
100%
NO you cant - 0 votes (0%)
0%

Notification

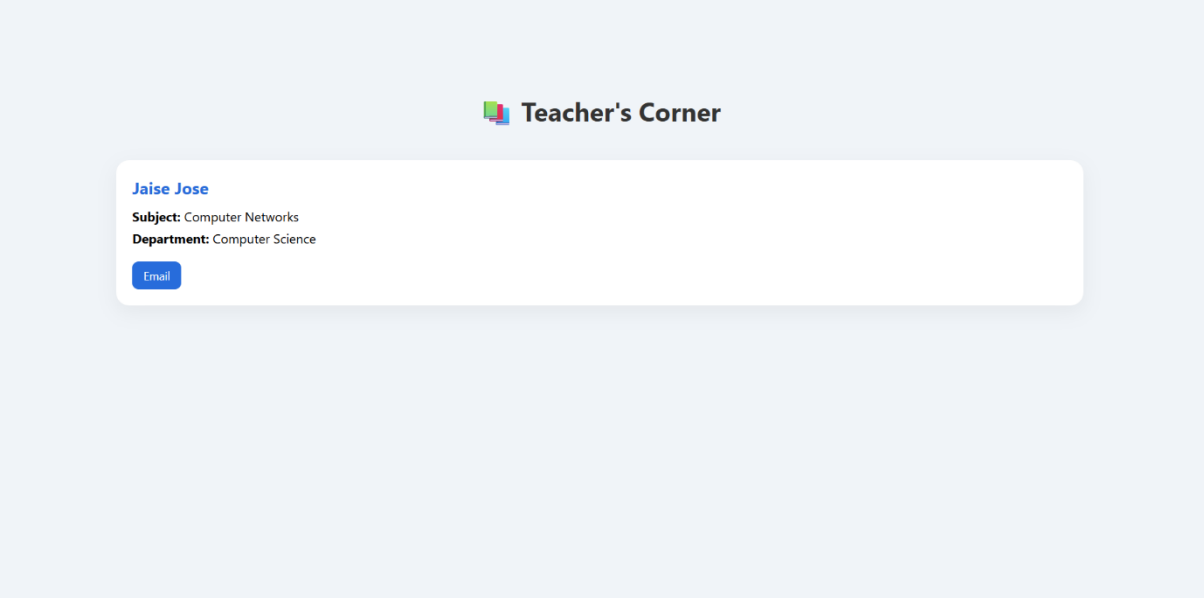


Student lists

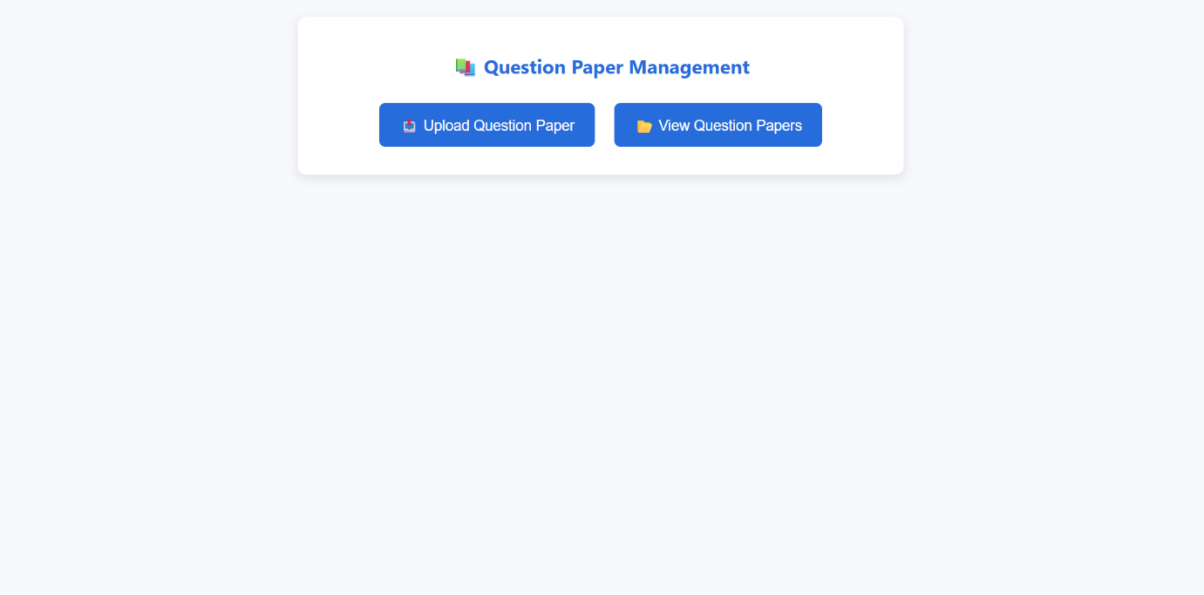
Students in Your Batch

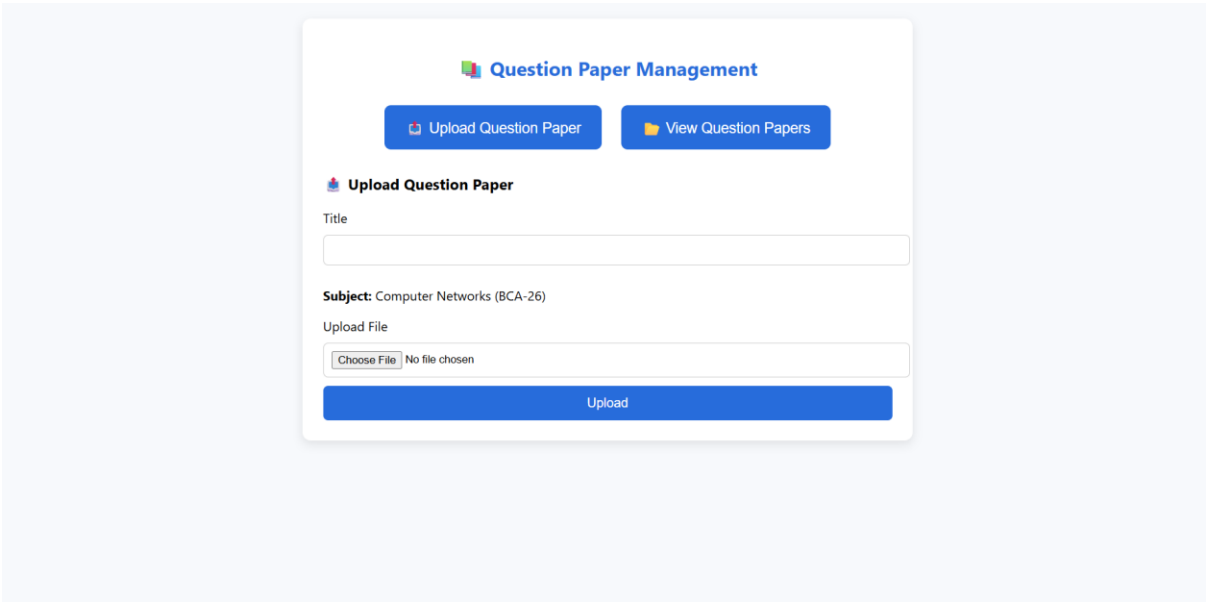
Sl. No.	Name	Admission No	Email	Register No	DOB	Course	Created At
1	Ajil Saji	A-6467	ajilsaji@gmail.com	5452252525	2025-09-16	BCA	2025-09-29 17:33:23
2	Arun Joshy	A-6564	arunjoshy@gmail.com	2323454	2025-10-05	bca	2025-10-01 22:27:56

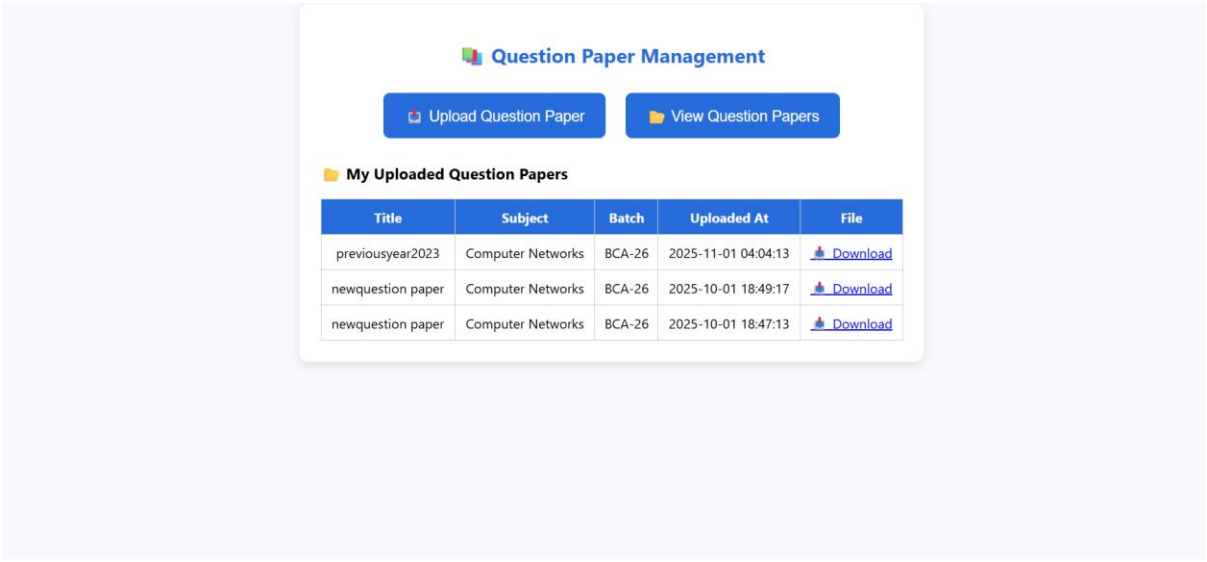
Teachers corner



Question paper

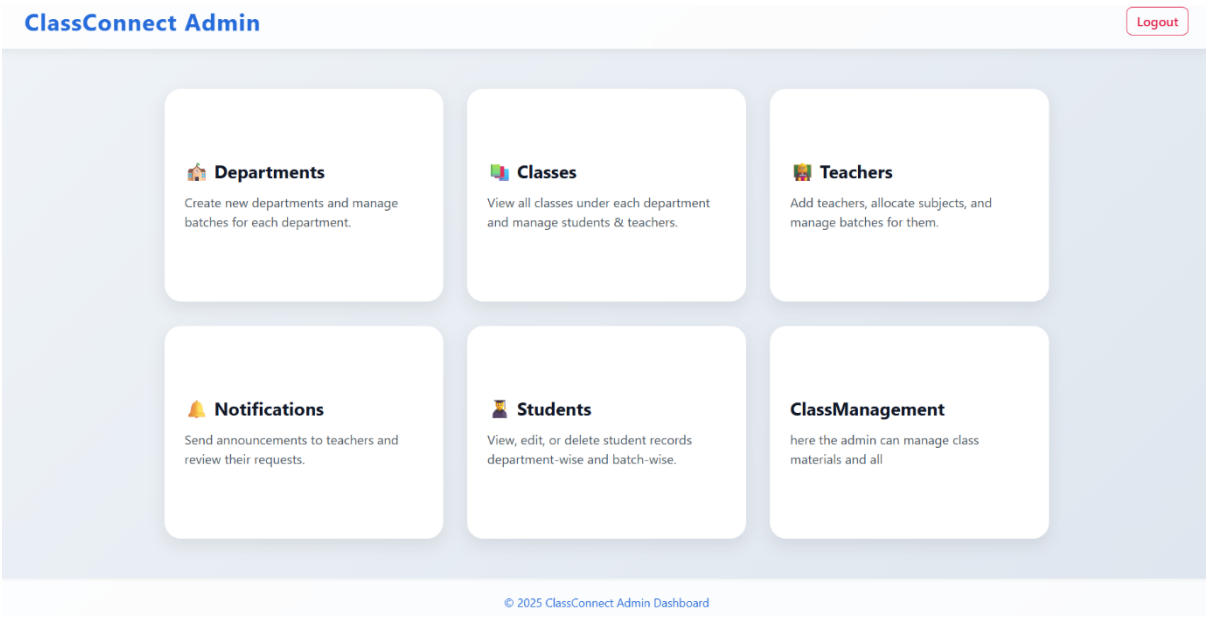



Question Paper Management
Upload Question Paper
View Question Papers
Upload Question Paper
Title
Subject: Computer Networks (BCA-26)
Upload File
Choose File
No file chosen
Upload

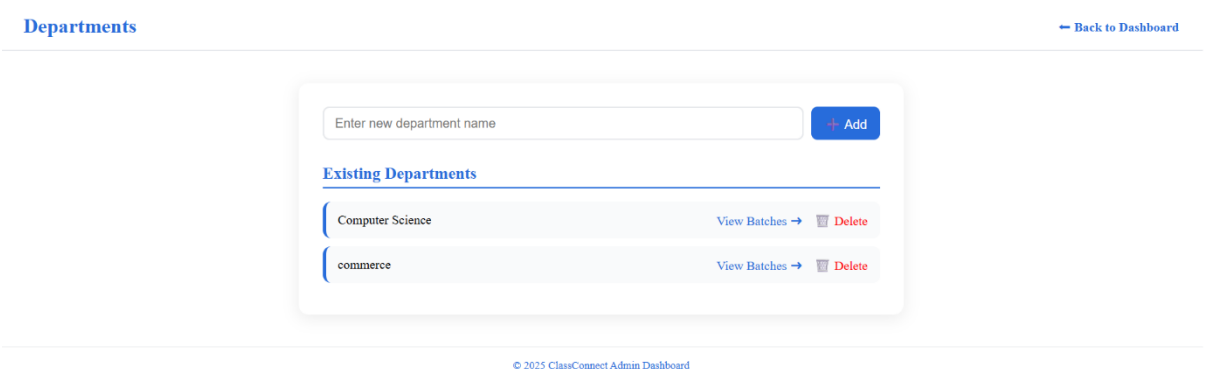

Question Paper Management
Upload Question Paper
View Question Papers
My Uploaded Question Papers
Title
Subject
Batch
Uploaded At
File
previousyear2023
Computer Networks
BCA-26
2025-11-01 04:04:13
Download
newquestion paper
Computer Networks
BCA-26
2025-10-01 18:49:17
Download
newquestion paper
Computer Networks
BCA-26
2025-10-01 18:47:13
Download

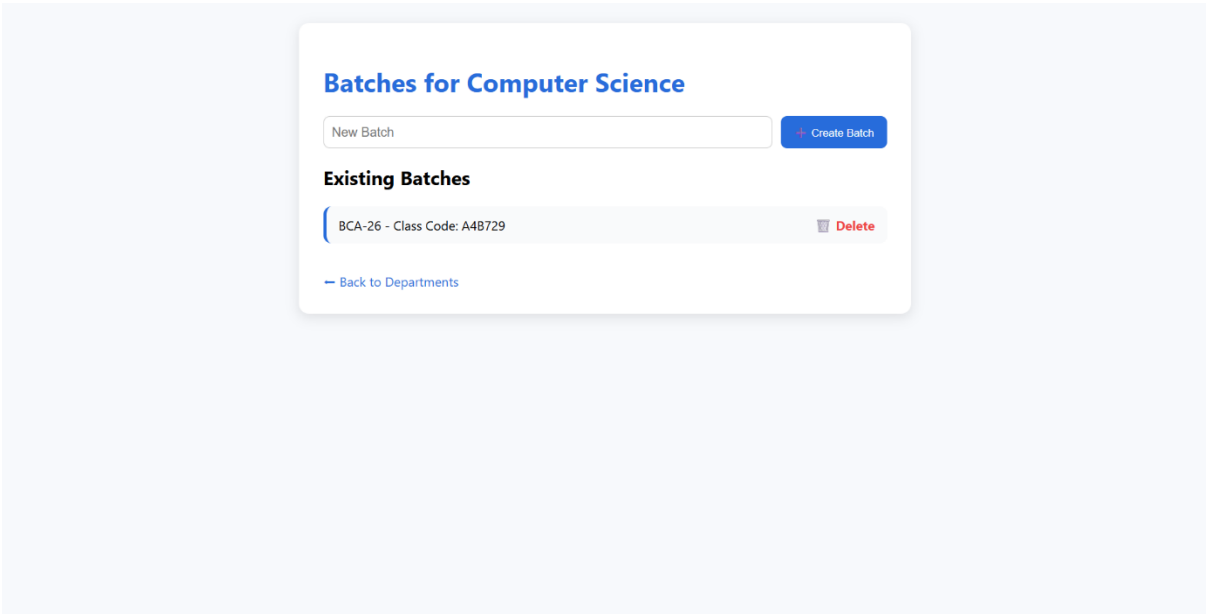
ADMIN SIDE

Admin Dashboard

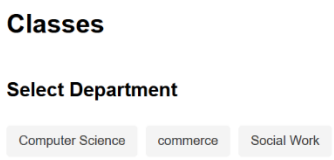


Departments




Batches for Computer Science
New Batch
Create Batch
Existing Batches
BCA-26 - Class Code: A4B729
Delete
Back to Departments

Classes


Classes
Select Department
Computer Science
commerce
Social Work

Classes

← Back to Departments

Available Batches

BCA-26

Classes

← Back to Batches

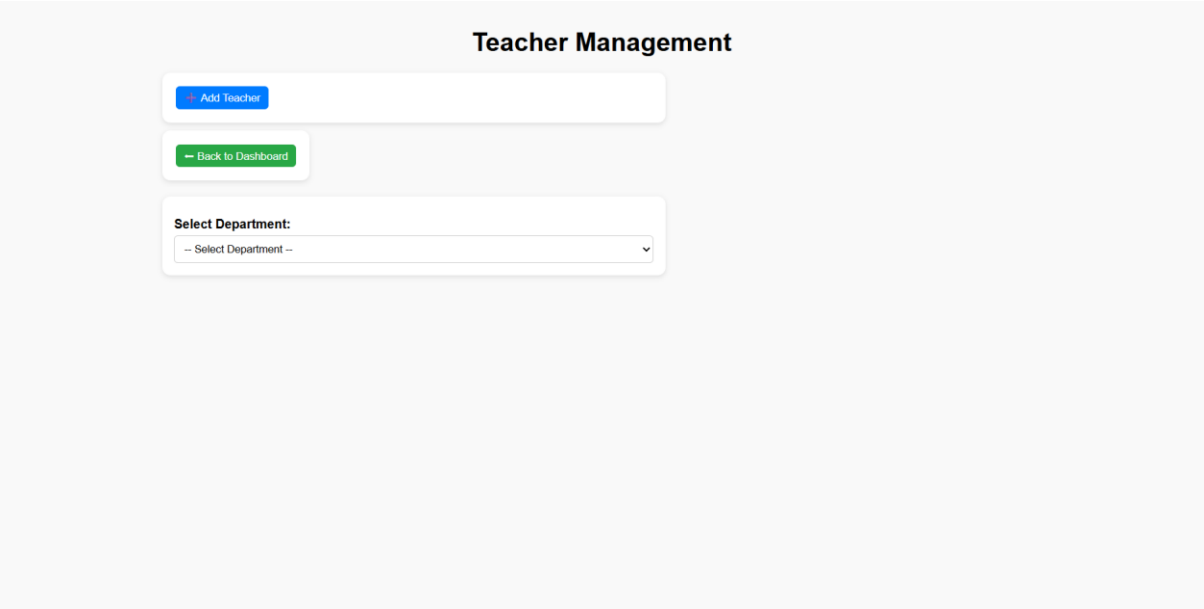
Students in Batch

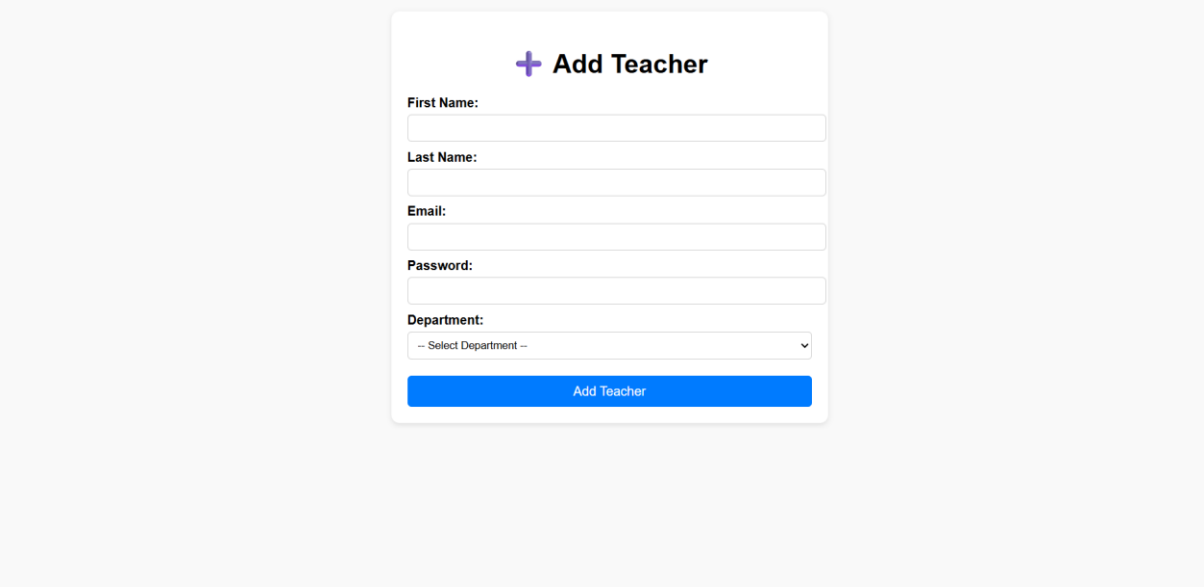
Admission No	Name	Email	Register No	Actions
A-6467	Ajil Saji	ajilsaji@gmail.com	5452252525	Edit Remove
A-6564	Arun Joshy	arunjoshy@gmail.com	2323454	Edit Remove

Teachers in Batch

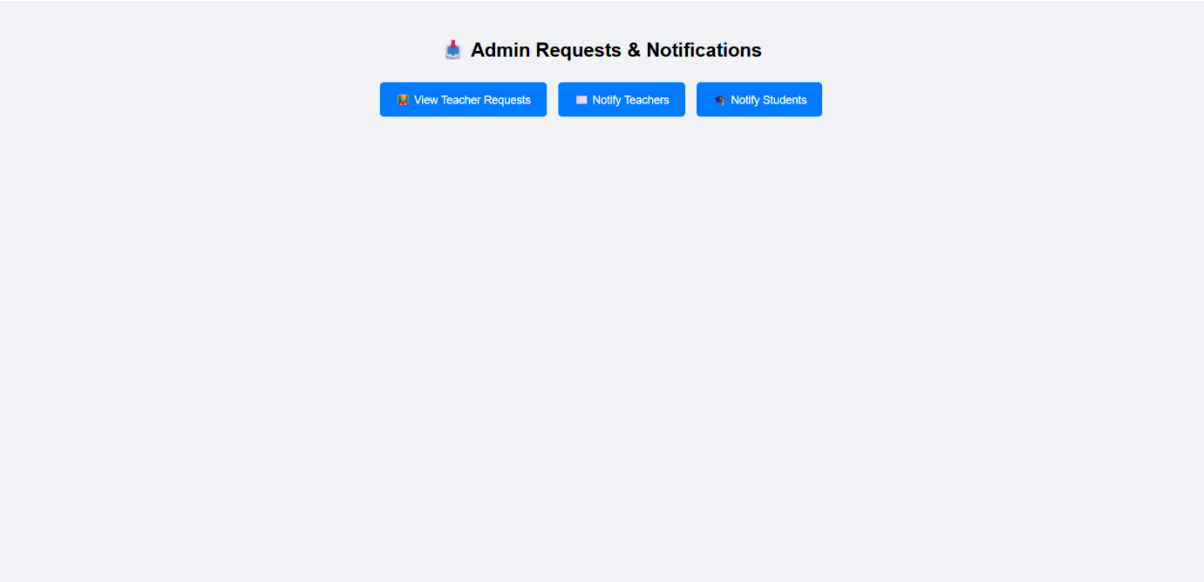
Name	Email	Subject	Actions
Jaise Jose	jaisejose@gmail.com	Computer Networks	Edit Remove

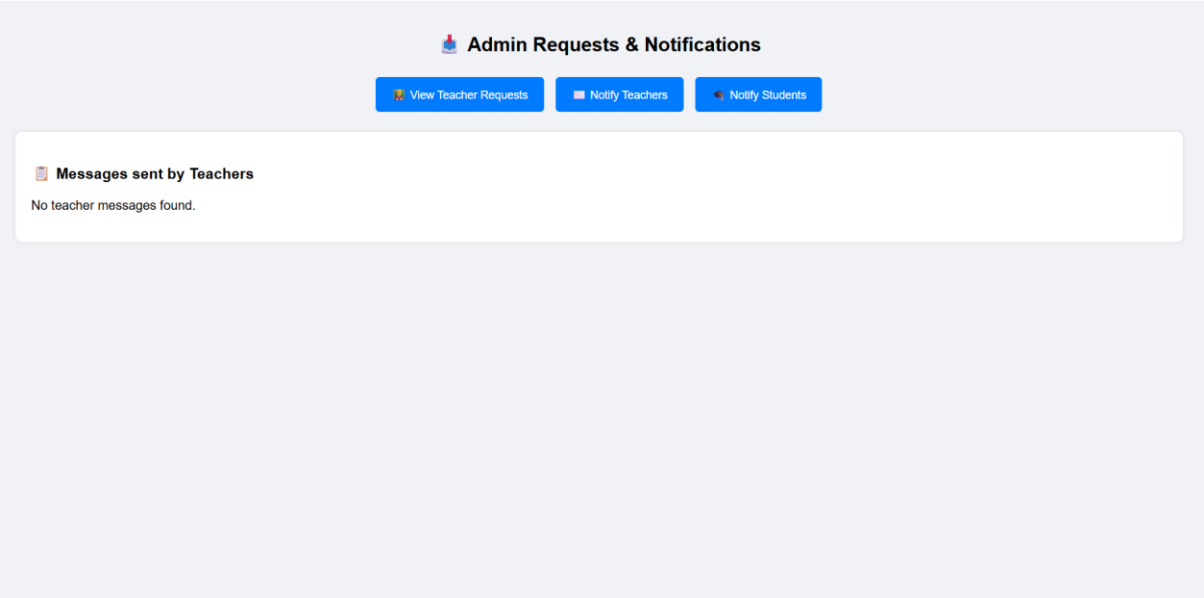
Teachers

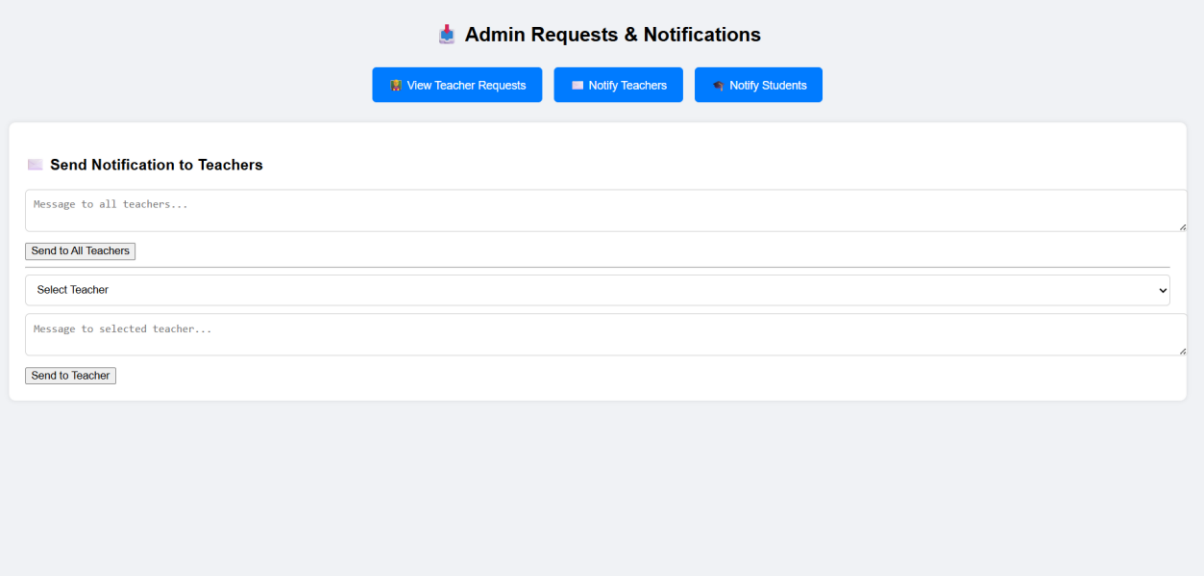

Teacher Management
Add Teacher
Back to Dashboard
Select Department:
-- Select Department --

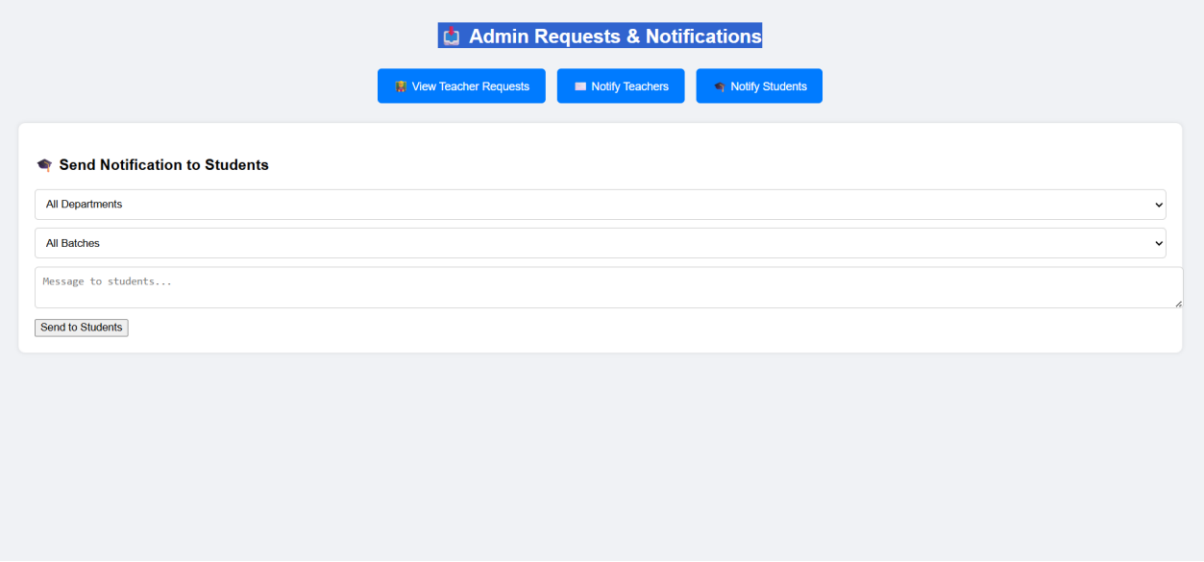

Add Teacher
First Name:
Last Name:
Email:
Password:
Department:
-- Select Department --
Add Teacher

Notification

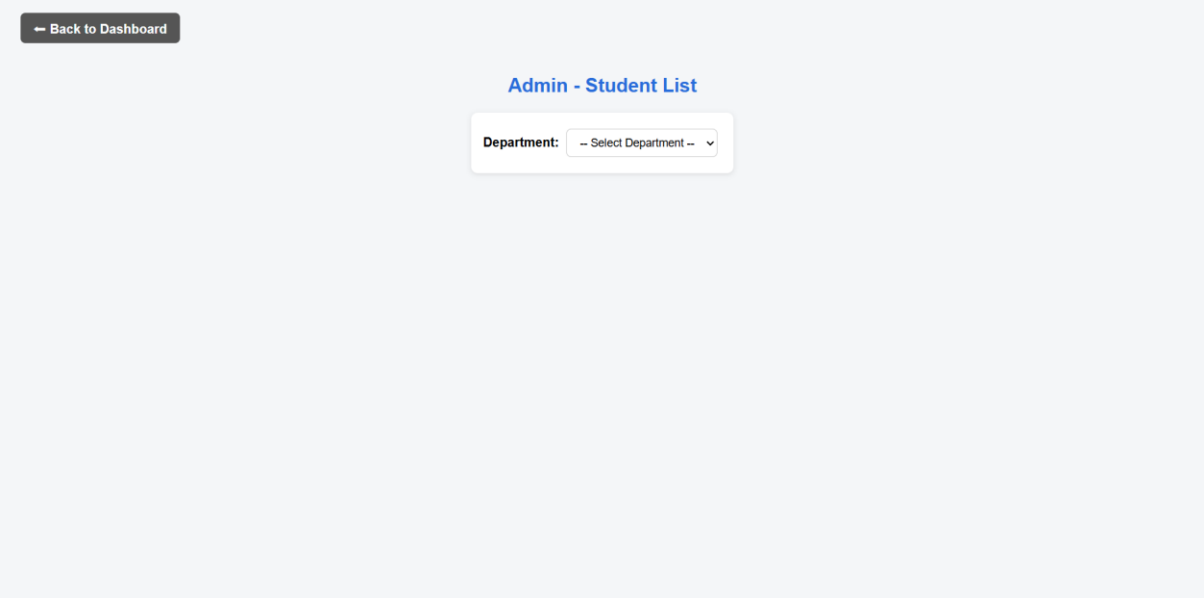


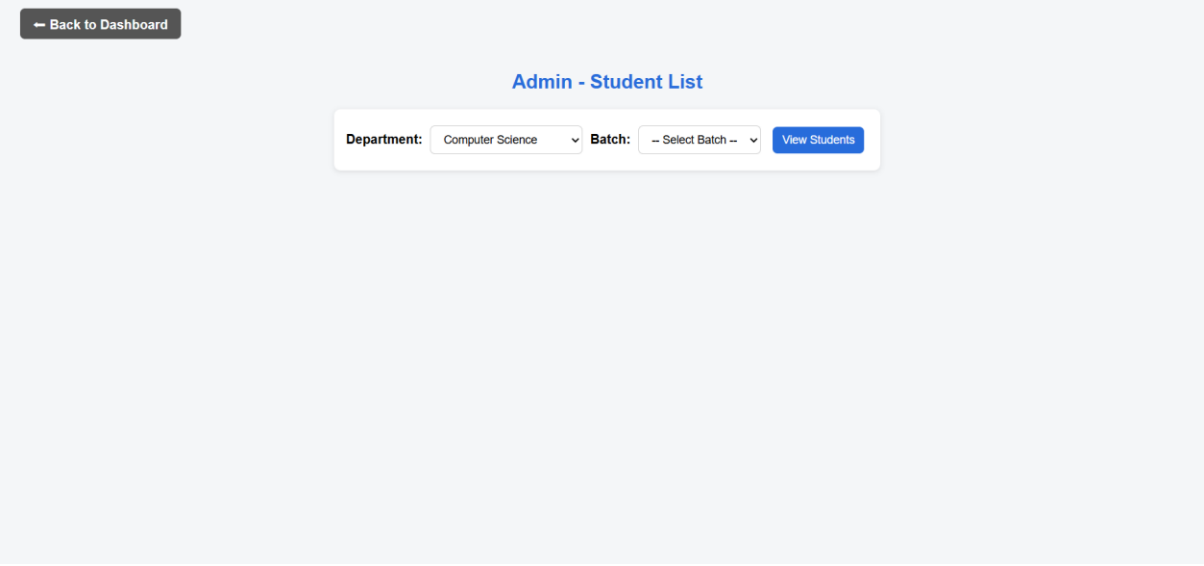


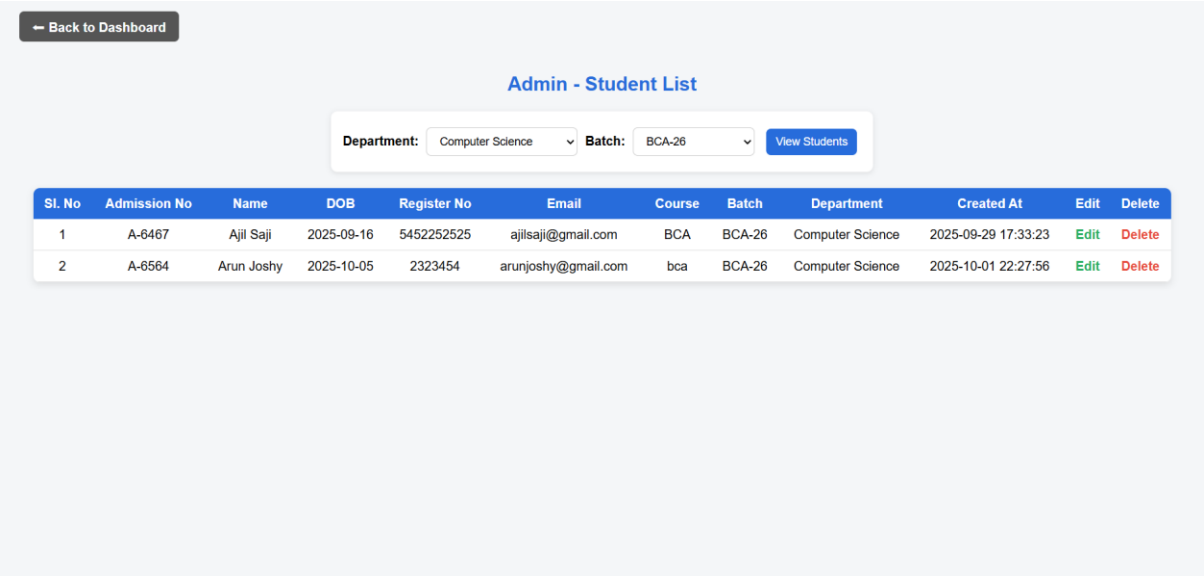

Admin Requests & Notifications
View Teacher Requests
Notify Teachers
Notify Students
Send Notification to Teachers
Message to all teachers...
Send to All Teachers
Select Teacher
Message to selected teacher...
Send to Teacher

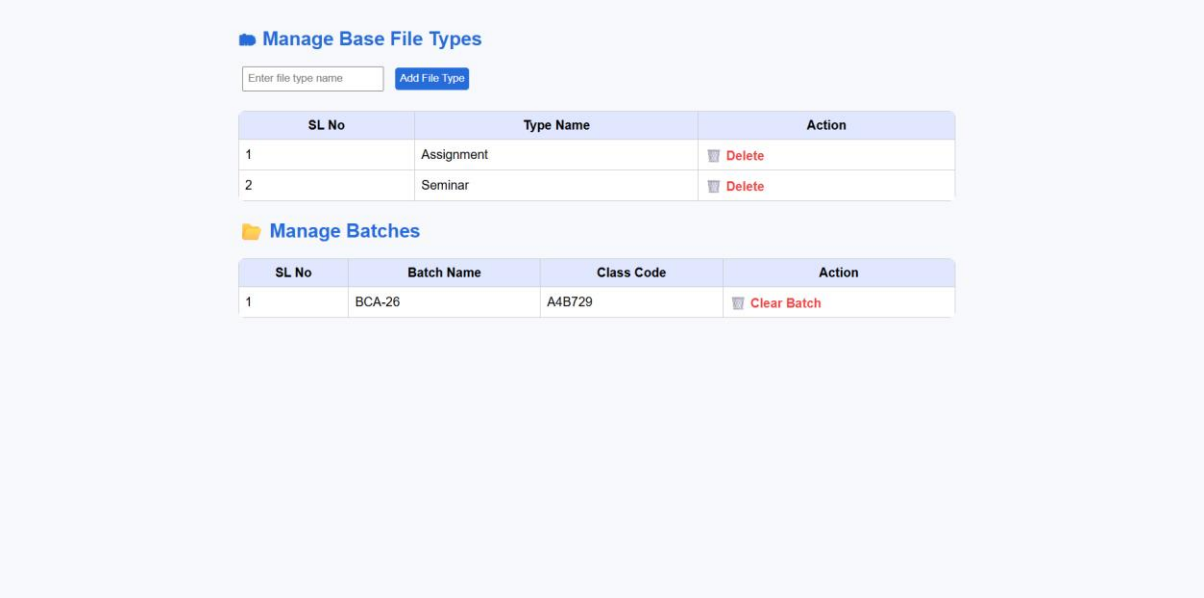

Admin Requests & Notifications
View Teacher Requests
Notify Teachers
Notify Students
Send Notification to Students
All Departments
All Batches
Message to students...
Send to Students

Student management



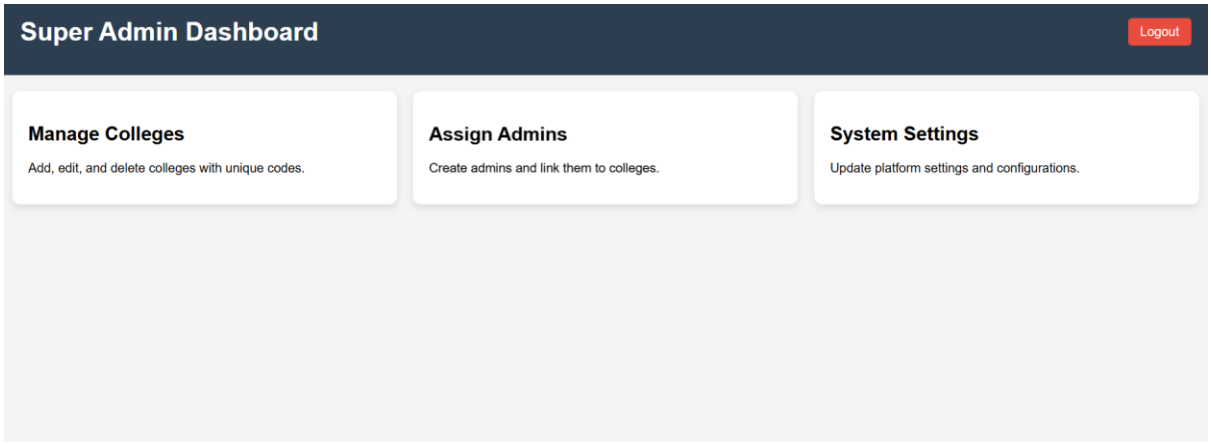



← Back to Dashboard
Admin - Student List
Department: Computer Science
Batch: BCA-26
View Students
Sl. No | Admission No | Name | DOB | Register No | Email | Course | Batch | Department | Created At | Edit | Delete
1 | A-6467 | Ajil Saji | 2025-09-16 | 5452252525 | ajilsaji@gmail.com | BCA | BCA-26 | Computer Science | 2025-09-29 17:33:23 | Edit | Delete
2 | A-6564 | Arun Joshy | 2025-10-05 | 2323454 | arunjoshy@gmail.com | bca | BCA-26 | Computer Science | 2025-10-01 22:27:56 | Edit | Delete

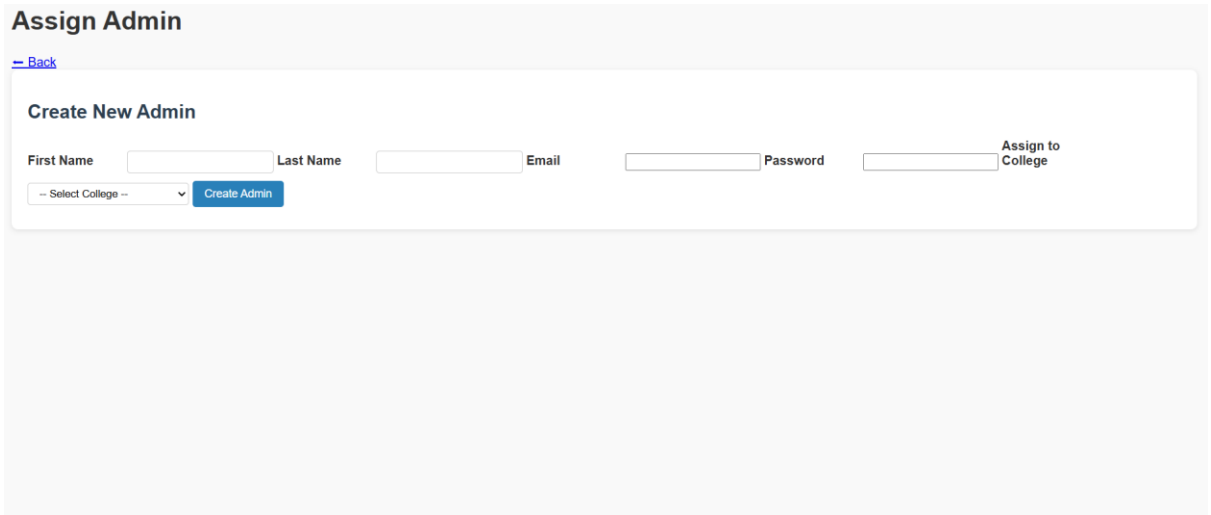

Manage Base File Types
Enter file type name
Add File Type
SL No | Type Name | Action
1 | Assignment | Delete
2 | Seminar | Delete
Manage Batches
SL No | Batch Name | Class Code | Action
1 | BCA-26 | A4B729 | Clear Batch

SUPER-ADMIN

Superadmin Dashboard

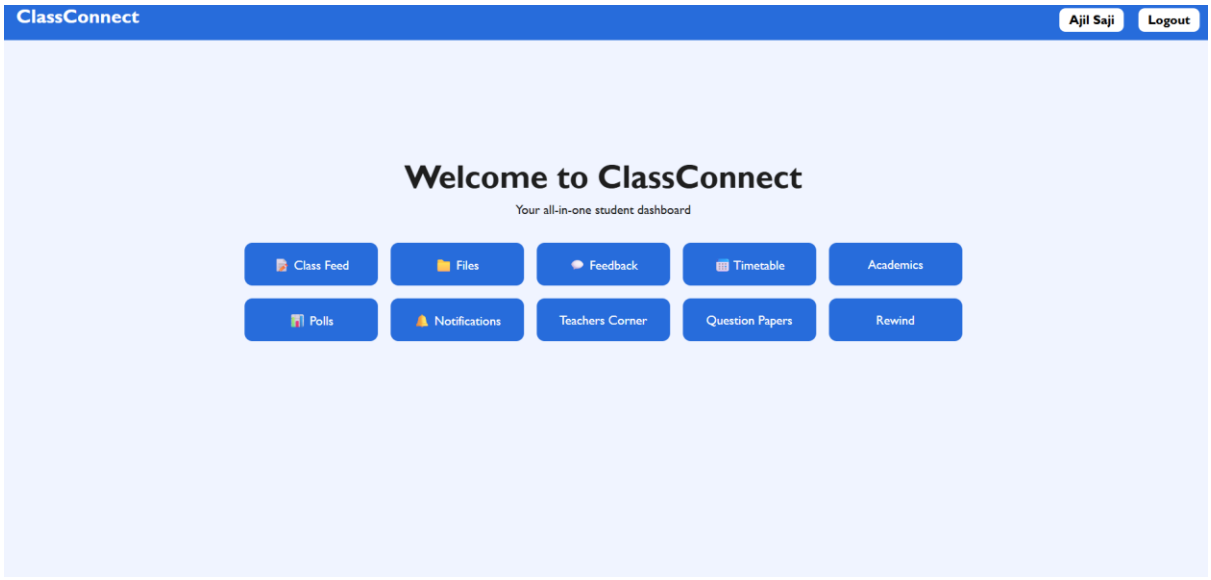


Assign admins

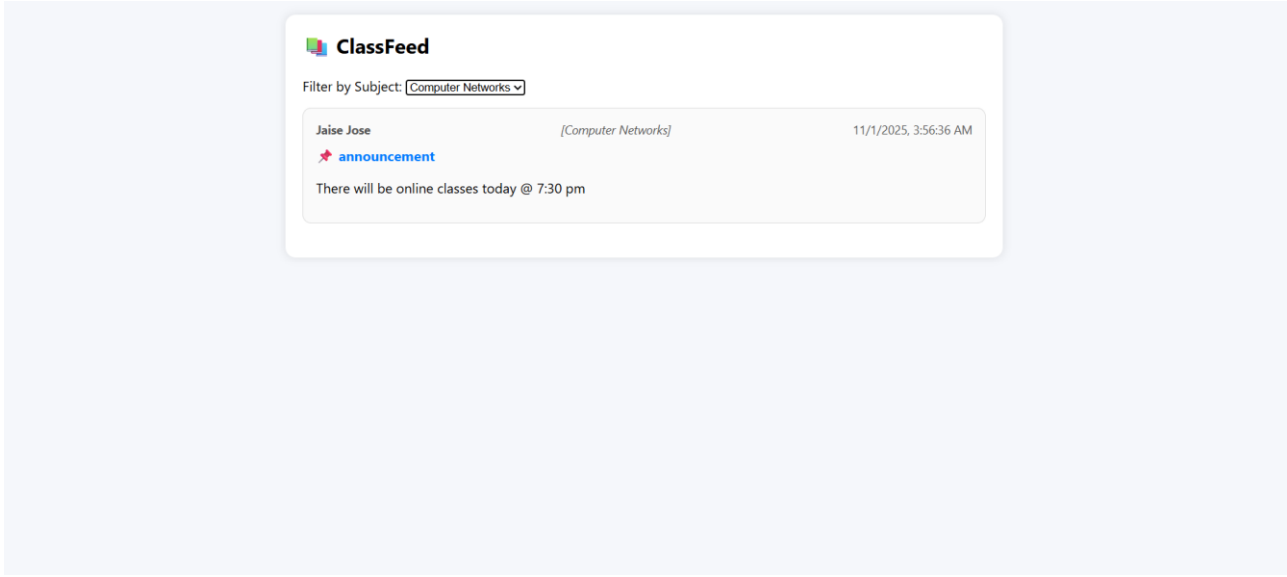


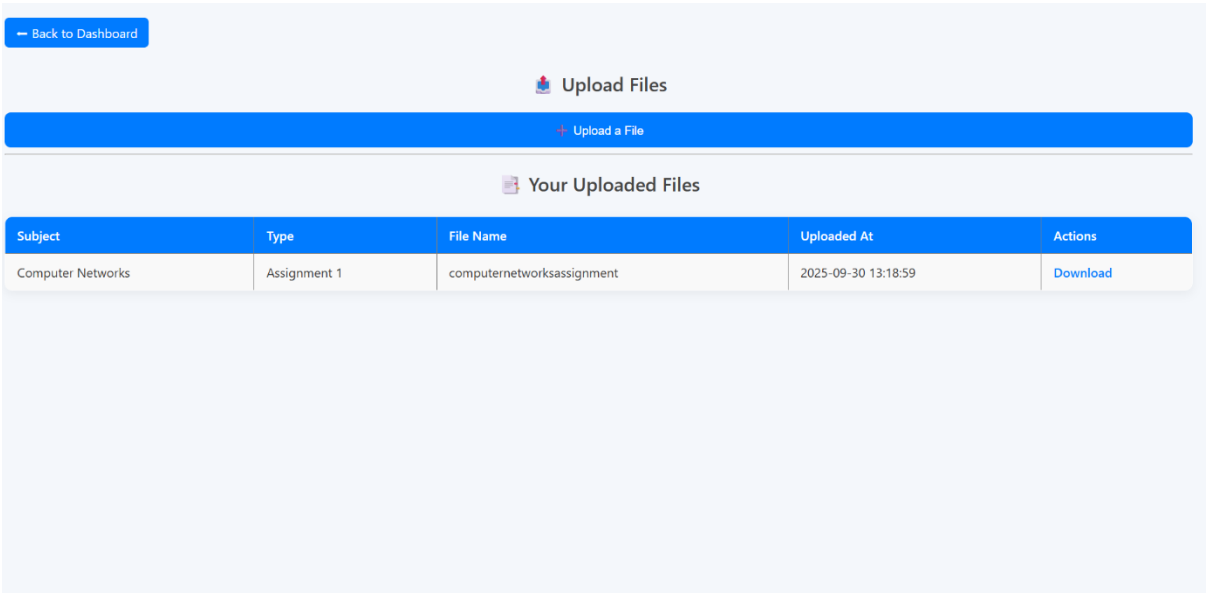
STUDENT SIDE

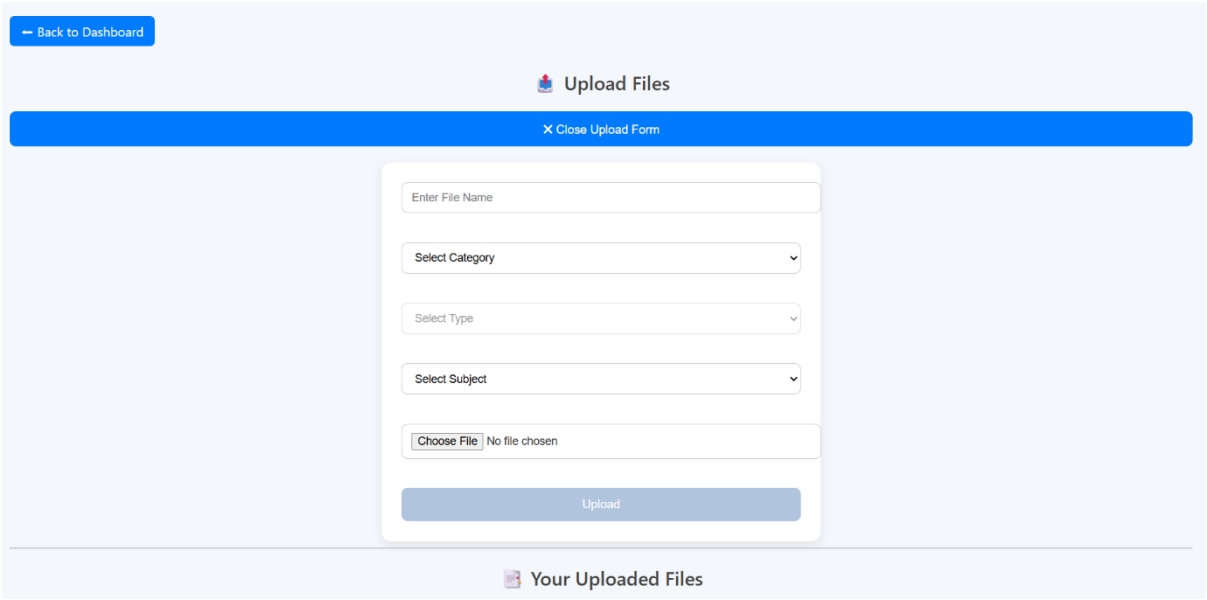
Student Dashboard



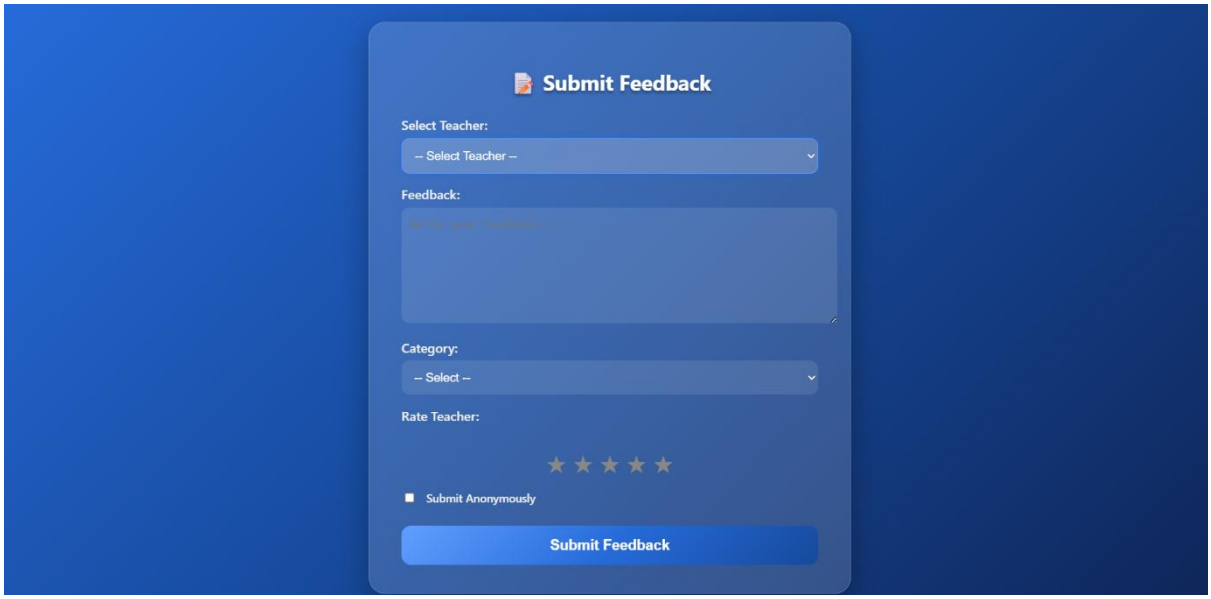
Classfeed



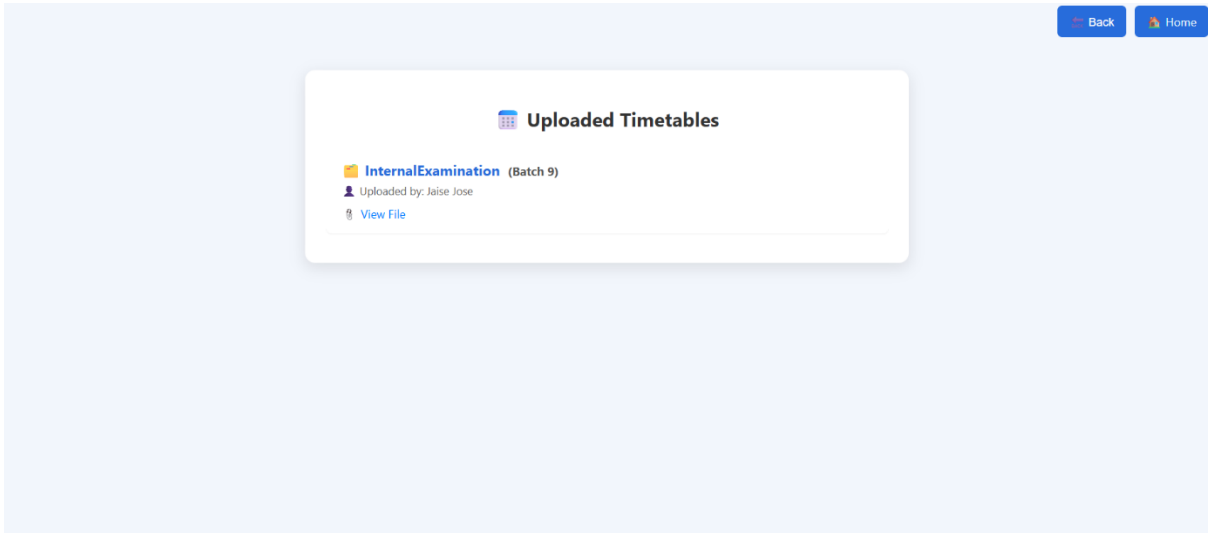

Back to Dashboard
Upload Files
Upload a File
Your Uploaded Files
Subject
Type
File Name
Uploaded At
Actions
Computer Networks
Assignment 1
computernetworksassignment
2025-09-30 13:18:59
Download


Back to Dashboard
Upload Files
Close Upload Form
Enter File Name
Select Category
Select Type
Select Subject
Choose File
No file chosen
Upload
Your Uploaded Files

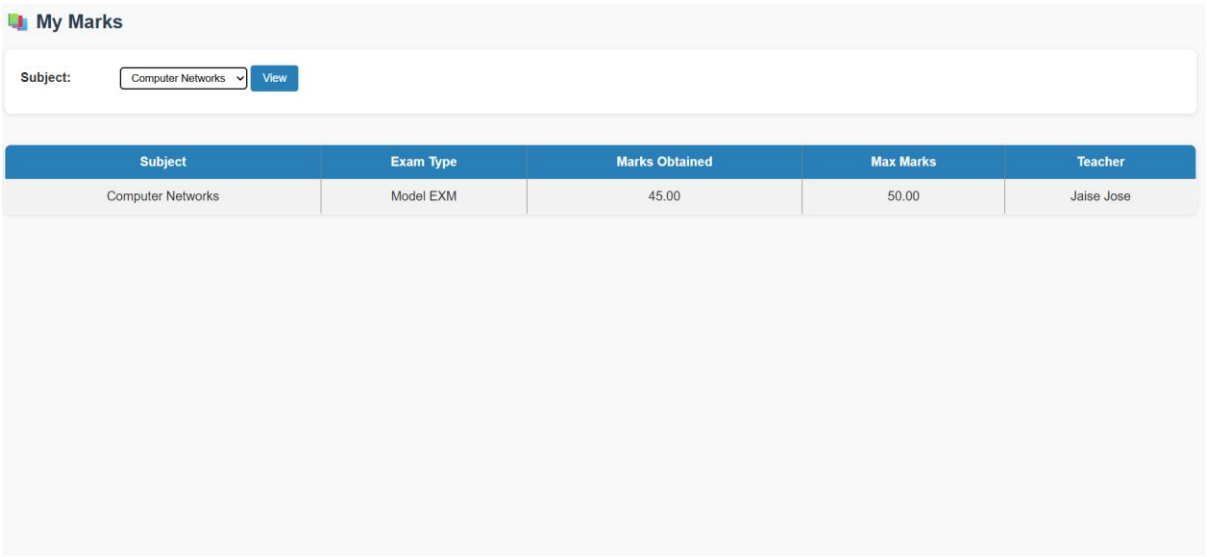
Feedback



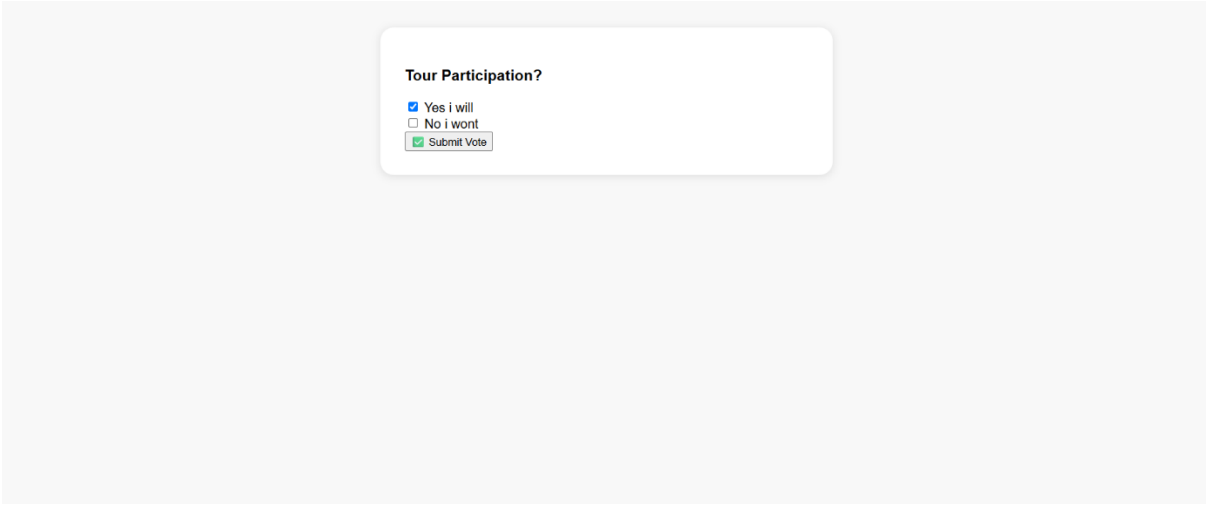
Timetable



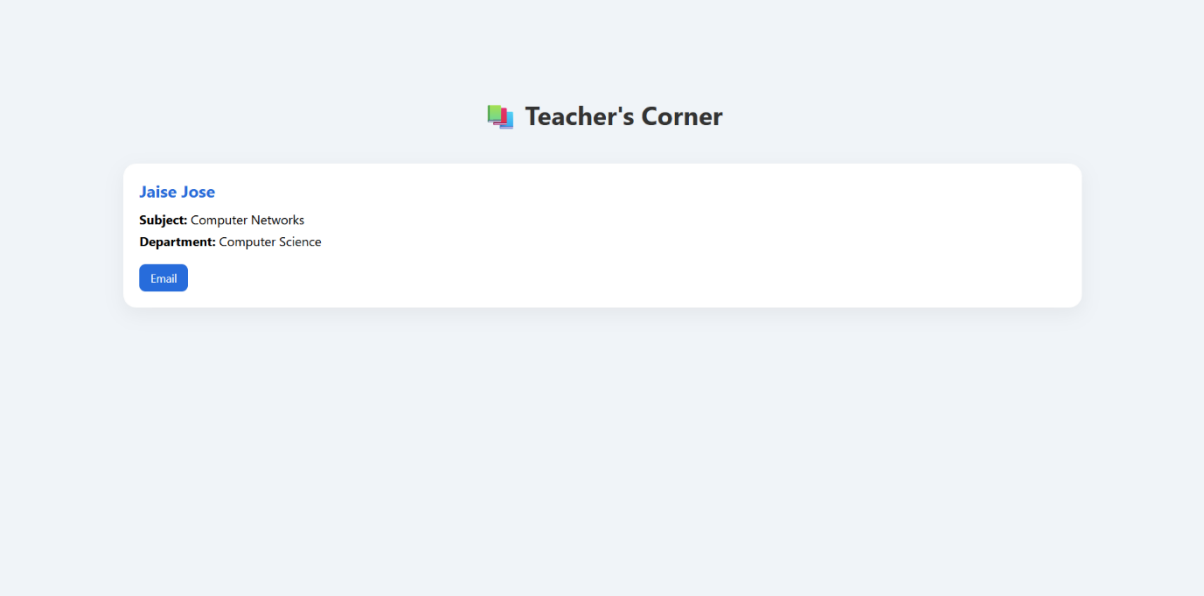
Academics



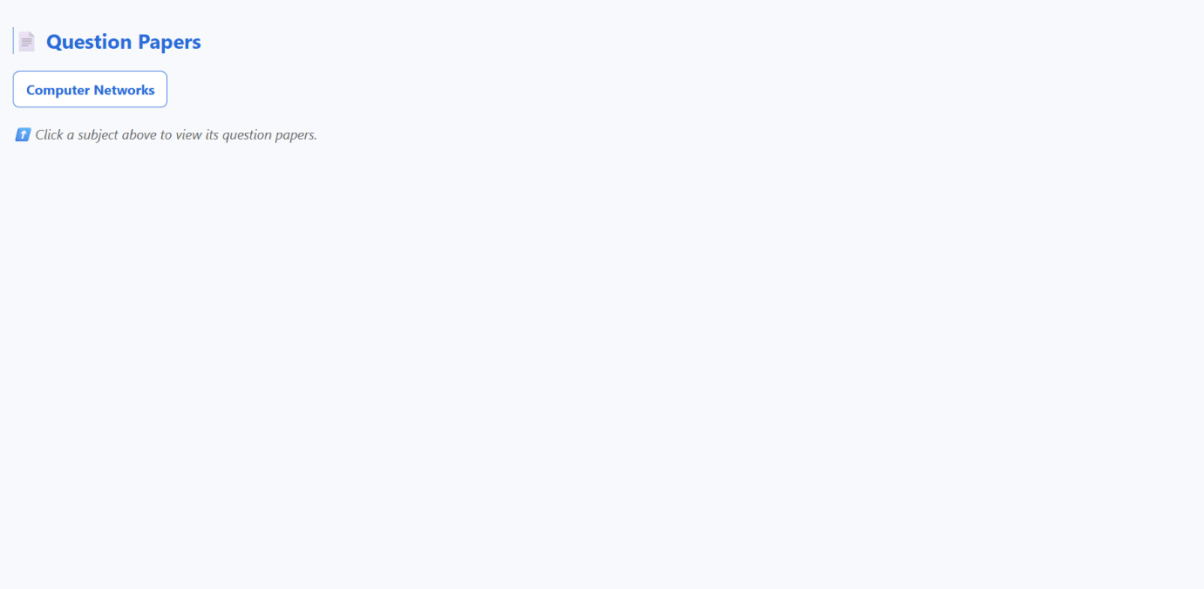
Polls

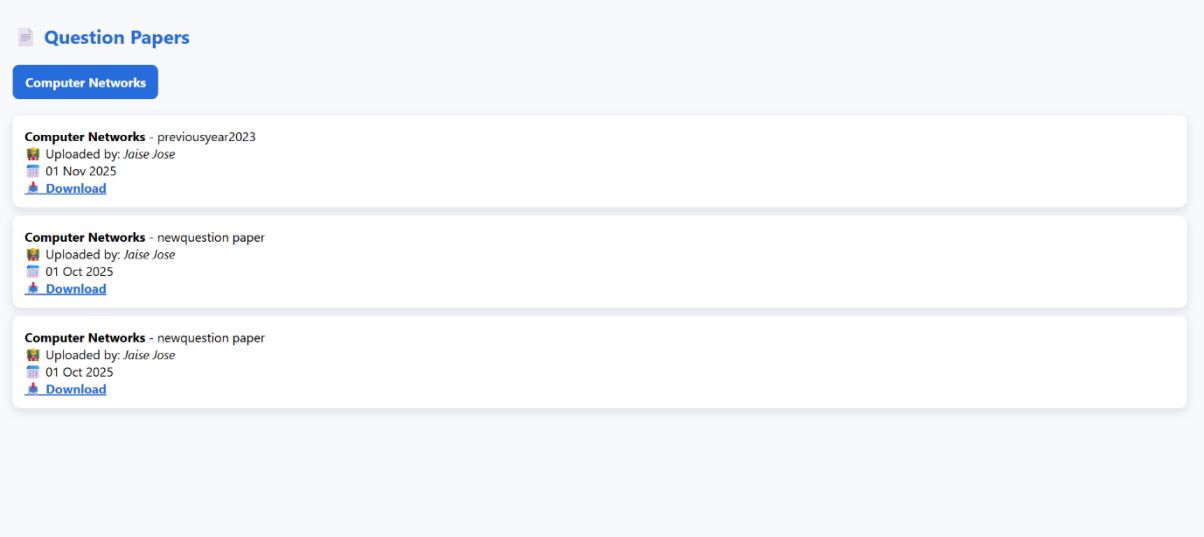


Teachers corner

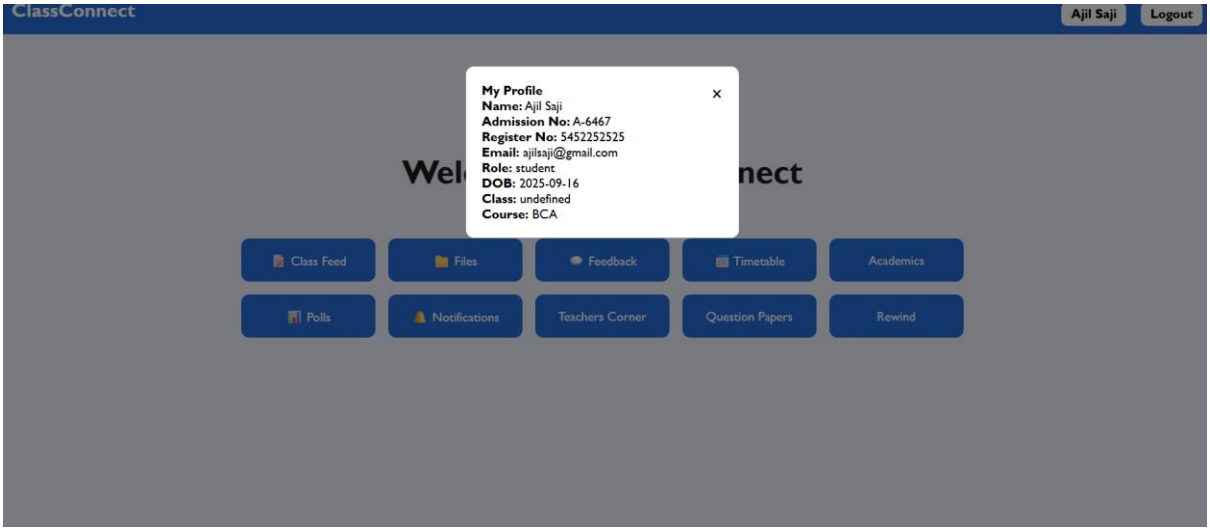


Question papers





Student's profile



10.BIBLOGRAPHY

TEXT REFERENCE

- Software Engineering: A practitioners Approach by Roger.S.Pressman7
- Programming PHP: Rasmus Lerdorf and Kevin Tatore, Shroff Publishers & Distributors Pvt. Ltd
- Binu M.B and Amaladevi V.C Web Programming using PHP, Prakash Publications, Changanacherry,2019.
- Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5 by RobinNixon.

WEB REFERENCE

- https://github.com/ajilaries/Classconnect.git
- Course: The Complete Full-Stack Web Development Bootcamp | Udemy
- MySQL Tutorial
- Evergreen_16x9_Python_UGC_20s_la.EN