CLASSCONNECT

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

BY,

Ajil Saji
Register Number:230021078057

DEPT. OF COMPUTER SCIENCE
B.V.M HOLY CROSS COLLEGE CHERPUNKAL,
KOTTAYAM 686 584 August
2025

Department of Computer Science
BVM HOLY CROSS COLLEGE CHERPUNKAL

CERTIFICATE

Certified that the report entitled **Classconnect** is a bonafide record of the mini project work done by 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.

HOD
Seena S Nair

Project Guide
Brigit

Submitted for Project Evaluation on:

External Examiner

DECLARATION

I hereby declare that the mini 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 bonafide work done under the guidance of Mrs. Brigit Jose.

Name:Ajil Saji

Reg.No:230021078057

Place: Cherpunkal

Date:

ACKNOWLEDGEMENT

Dedicating this mini project to the Almighty God whose abundant grace and mercies enabled successful completion,we would like

to express our profound gratitude to all the people who had inspired and motivated us to make this mini project a success. We would like to place our deep sense of gratitude to Mrs.Seena S Nair (Head of the department of Computer Applications) for her guidance in carrying out this project work. We are profoundly grateful to Mrs.Brigit Jose as the 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 throughout the project. Last, but not least we would like to thank our friends for their co-operation and encouragement.

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.

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.

Table Of Contents

1.Introduction

ClassConnect is a collaborative web platform designed to simplify communication, resource sharing, and classroom engagement for both students and teachers. In modern academic life, information is often scattered across multiple channels — WhatsApp groups, emails, and handwritten notes — leading to confusion and missed updates. ClassConnect solves this by providing a centralized hub where students and teachers can connect, share posts, view announcements, upload and download study materials, give feedback, participate in polls, and stay updated on class activities.

The platform is built with a role-based system that ensures security and personalization. Teachers and admins can create posts, upload question papers, manage polls, and monitor feedback, while students can view posts, interact with polls, and submit feedback. This ensures that each user sees a dashboard tailored to their role, improving usability and reducing clutter.

ClassConnect is intentionally designed to be modular and extensible. The backend manages authentication, user roles, ClassFeed operations (create, read, update, delete posts), notifications, polls, and feedback tracking. The frontend is structured with a clean, responsive UI that makes it easy to navigate across devices. This separation of concerns allows for easy updates and future feature additions, such as private messaging, advanced analytics, or AI-powered study suggestions.

By integrating all key classroom interactions into a single platform, ClassConnect enhances collaboration, reduces friction, and keeps everyone on the same page — literally.

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.

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.

2.System Specification

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 Nginx + PHP-FPM
- PHP: 7.4 / 8.x
- Database: MySQL / MariaDB
- Frontend: HTML5, CSS3, JavaScript (ES6)
- Tools 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

(*ClassConnect*) addresses these shortcomings. It is divided into three parts: **Existing System**, **Proposed System**, and **Feasibility Study**.

3.1 Existing System

In most classrooms, communication and resource sharing are handled through:

- **WhatsApp/Telegram groups** for announcements and updates.
- **Emails or Google Drive links** for sharing study materials and question papers.
- **Manual feedback forms** or informal chats for student feedback.
- **Physical notice boards** for event announcements.

Limitations of Existing System:

- **Scattered Information:** Updates and resources are spread across multiple platforms, making them hard to track.
- **No Centralized Dashboard:** Students must check multiple places for announcements, notes, and feedback.
- **Lack of Role-Based Access:** No proper control over who can post or delete information.
- **No Structured Feedback Mechanism:** Student feedback is often lost or ignored without proper tracking or replies.
- **No Polling/Engagement Tools:** Teachers cannot easily conduct quick surveys or gather opinions.
- **Limited Notification System:** Important updates can be missed if students don’t check WhatsApp or notice boards on time.

3.2 Proposed System (ClassConnect)

ClassConnect overcomes these limitations by providing a centralized, role-based, web platform for communication and collaboration.

Key Features of the Proposed System:

- **User Authentication:** Secure login system with role-based dashboards (Super Admin, Admin/Teacher, Student).
- **ClassFeed:** Centralized post feed for announcements, file sharing, and updates — displayed in reverse chronological order.
- **Feedback System:** Students can submit categorized feedback with optional anonymity, star ratings, and comments. Admins can view, reply, and mark feedback as resolved.
- **Poll System:** Teachers can create polls with single/multiple-choice answers, live vote counts, expiration dates, and view archived results.
- **Question Paper Section:** Teachers can upload question papers and materials; students can easily access and download them.
- **Notification System:** Displays updates (sorted by upload time) in a popup notification panel for better visibility.
- **Profile Section:** Displays user details (name, admission no, register no, class, course, email, role, DOB).
- **Secure Data Handling:** Uses prepared statements for SQL queries to prevent SQL injection and role checks to prevent unauthorized access.

This proposed solution brings all communication, feedback, and resources into one place, saving time, improving transparency, and ensuring that everyone stays updated.

3.3 Feasibility Study

- **Technical Feasibility:**

ClassConnect is developed using open-source, widely supported technologies (PHP, MySQL, HTML, CSS, JavaScript) on a LAMP/XAMPP stack. It supports role-based access control, CRUD operations for posts, polls, and feedback, and is fully responsive. No specialized hardware is required, and the modular database design makes future expansion easy.

- **Operational Feasibility:**

The system is simple and intuitive, requiring only a web browser. Students and teachers get a dashboard customized to their roles. Teachers can manage posts, polls, and feedback, while students can view posts, submit feedback, and vote in polls. This reduces confusion and eliminates the need for training.

- **Economic Feasibility:**

ClassConnect uses free tools and frameworks (XAMPP, MySQL, VS Code, GitHub), so the cost of development and deployment is negligible. It is highly cost-effective for academic institutions.

- **Schedule Feasibility:**

The system was developed in a phased manner over a 4–6 week mini-project schedule. Features like ClassFeed, Feedback, Polls, and Notifications were implemented incrementally, making integration smooth and meeting the project deadline comfortably.

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 Secure Authentication**
   - Users must be able to register with details such as name, email, admission number, register number, class, course, date of birth, and role.
   - Secure login system to redirect users to their role-specific dashboard (Admin / Student).
2. **Role-Based Access Control**
   - **Admins/Teachers:** Can post in ClassFeed, upload question papers, create polls, view all users, reply to feedback, and manage posts/files.
   - **Students:** Can view posts, vote in polls, submit feedback, and download question papers.
3. **ClassFeed (Central Feed)**
   - Admins/Teachers can create posts with optional file attachments.
   - Posts are displayed in reverse chronological order.

- Admins can delete posts and uploaded files.

4. **Feedback Section**
    - Students can submit feedback with optional anonymity.
    - Feedback can be categorized (Academics, Events, Facilities, etc.).
    - Admins can reply to feedback, mark them as resolved, and track status.
5. **Poll Section**
    - Admins can create single-choice or multiple-choice polls.
    - Students can vote once per poll.
    - Poll results are shown after voting.
    - Poll expiration and archived results are supported.
    - Admins can end polls early or clear poll data.
6. **Question Paper Section**
    - Admins can upload question papers categorized by subject and exam type.
    - Students can search, filter, and download question papers.
7. **Notification System**
    - Notifications must show new posts, polls, feedback replies, and question paper uploads.
    - Notifications are sorted by upload time and displayed in a popup.
8. **Profile Section**
    - Users can view their profile details including name, admission number, register number, email, role, class, course, and date of birth.
9. **Search and Filter**
    - Students can search posts, filter question papers, and view feedback history.
10. **Responsive User Interface**

- The system must work smoothly across desktop, tablet, and mobile devices.

3.4.2 Non-Functional Requirements

1. **Security**
    - Passwords must be stored using hashing.
    - Use prepared statements to prevent SQL injection.
    - Sessions should validate user roles before granting access.
2. **Performance**
    - ClassFeed and notifications should load quickly, even with multiple posts and users.
    - Database queries must be optimized for concurrent access.
3. **Usability & Accessibility**
    - The UI must be clean, responsive, and easy to use.
    - Error/success messages must be displayed clearly.
4. **Maintainability**
    - Code should be modular with proper separation between frontend (HTML, CSS, JS) and backend (PHP, MySQL).
    - System should allow easy addition of new features like chat, event calendar, and mobile app integration.
5. **Reliability**

- The system must prevent unauthorized access and ensure data consistency.
- Proper validation of file uploads and poll deadlines is mandatory.

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 **frontend** (HTML, CSS, JavaScript) provides an intuitive and responsive interface for students, teachers, and admins.

- The **backend** (PHP + MySQL) handles authentication, database operations, 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 features like event calendars, 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, notifications, and uploaded materials.

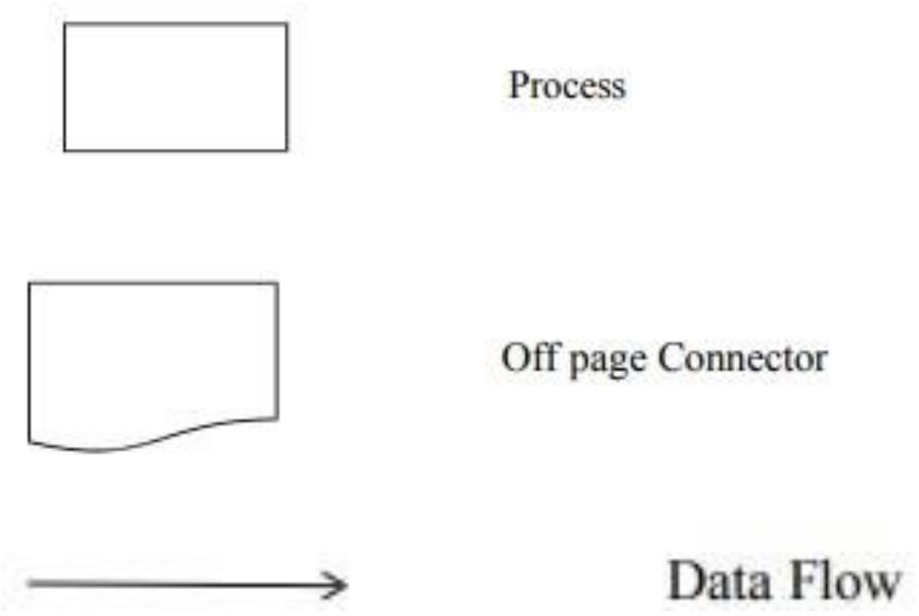
This design ensures that ClassConnect not only meets its current requirements but is also flexible enough for future enhancements. It serves as a blueprint that bridges the gap between requirement analysis and the actual coding process.

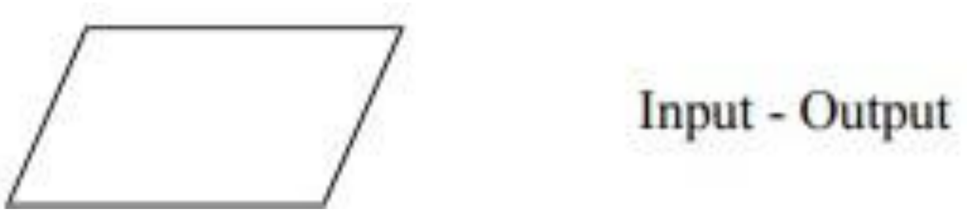
a programmer or database personal.it is creative in both art and technology.

4.2. System Flowchart

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

Basic Flow chart Symbols:





4.3.Database 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.

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)

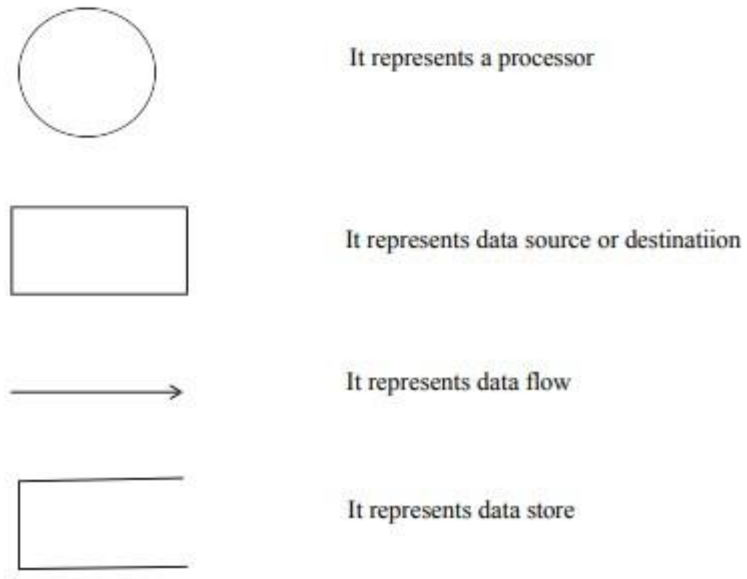
sets the very basic rules for an organized database: Eliminate duplicative columns from the same table. Create separate tables for each group of related data and identify each row with a unique column or set of columns (the primary key).

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 **4.5.Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system. A data flow diagram can also be used for the visualization of data processing. Data Flow Diagram is a common practice for a designer to draw a context-level Data Flow Diagram first which shows the interaction between the system and outside entities. A Data Flow 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 structured 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 processes. Named arrows indicate the data flow. External entities are represented by rectangles and are outside the system such as vendors or customers with whom the system interacts. They either supply or consume data are called sinks. Data is stored in a data store by a 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 meets 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 a processor

It represents data source or destinatiion

It represents data flow

It represents data store

Level-0

9.Biblography

1.Open Ai gpt 4o,4.1

2.Gemini 2.5 flash and Gemini 2.5 pro

3.youtube