# ONLINE PEER TUTORING SYSTEM

## A MINI PROJECT REPORT

**Submitted by**

**MEGHA PV (TKR21CS057)**

**NANDANA CHANDRAN (TKR21CS072)**

**NANDANA TV (TKR21CS073)**

**NAVYA P (TKR21CS075)**

to

the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology in

*Computer Science and Engineering*

**Department of Computer Science and Engineering**

College of Engineering Trikaripur

July 2024

# DECLARATION

We undersigned hereby declare that the project report “Online Peer Tutoring System”, submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a Bonafide work done by we under the supervision of Mr. Sreeraj S, Assistant Professor Department of Computer Science Engineering. This submission represents the ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that it has adhered to ethics of academic honesty and integrity and has not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

CHEEMENI

MEGHA PV

NANDANA CHANDRAN

NANDANA T

NAVYA P

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## COLLEGE OF ENGINEERING TRIKARIPUR

## CERTIFICATE

This is to certify that the report entitled “**ONLINE PEER TUTORING SYSTEM**", submitted by **Megha PV, Nandana Chandran, Nandana TV, Navya P** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a Bonafede record of the seminar carried out by him/her under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

| | | |
|---|---|---|
| Guide | Coordinator | Head of the Department |
| **Mr. Sreeraj S** | **Mr. Sreeraj S** | **Dr.Naveena A K** |
| Assistant Professor | Assistant Professor | |
| Department of Computer Science and Engineering | Department of Computer Science and Engineering | Department of Computer Science and Engineering |

# ACKNOWLEDGEMENT

We give honor and praise to the LORD who gave us wisdom and enabled us to complete this mini project successfully.

We wish to extend our sincere gratitude to our Principal, **Dr. Mahesh V V** for the help extended by him in providing all the college facilities for our project work. We extend our genuine and heartfelt thanks to **Dr.Naveena A K**, Head of Department, Department of Computer Science and Engineering for having permitted us to undergo this project and for her valuable suggestions, help and support during the study, analysis, design and implementation of this project.

We would like to express our deep gratitude to our project guide **Mr. Sreeraj S**, Assistant Professor in Department of Computer Science and Engineering, project coordinator Mr. Sreeraj s, Assistant Professor in Department of Computer Science and Engineering for their timely suggestions and encouragement through the processes involved in the presentation of the seminar for his wholehearted support and encouragement for the betterment of the seminar presentation and report preparation.

Also, we would like to express our sincere voice of gratitude to all the faculty members of CSE department who gave their support throughout the project. We also express our deepest thanks to our family members and friends for the moral support rendered to us to make this project a success.

# ABSTRACT

The Peer Tutoring Platform Project aims to create an online platform connecting students seeking academic assistance with their peers who are proficient in various subjects. This platform addresses the growing demand for personalized and accessible tutoring services while fostering a collaborative learning environment among students.

In the contemporary educational landscape, the integration of technology is pivotal for enhancing learning experiences and outcomes. The Online Peer Tutoring System is an innovative project designed to revolutionize traditional education by leveraging digital platforms to facilitate seamless communication and collaboration among students, teachers, and administrators. This system provides students with easy access to assignments, tutorials, instructional videos, and real-time peer interactions, fostering a supportive and interactive learning environment. Teachers can effortlessly upload educational materials, enhancing the efficiency and effectiveness of content delivery. Administrators manage user accounts, ensuring the system's organization and security. By transcending geographical barriers, the Online Peer Tutoring System creates a dynamic learning community that promotes academic success and continuous learning. This project explores the design, implementation, and benefits of the system, highlighting its potential to transform educational practices and improve student outcomes in a digitally connected world.

# CONTENTS

# LIST OF FIGURES

# ABBREVIATIONS

- HTML - Hypertext Markup Language
- VS CODE - Visual Studio Code
- SQL - Structured Query Language
- CSS - Cascading Style Sheets
- GUI - Graphical User Interface

# CHAPTER 1

# INTRODUCTION

In the rapidly evolving landscape of education, technology plays a crucial role in transforming traditional learning methodologies. The Online Peer Tutoring System is an innovative project designed to enhance the educational experience by leveraging digital platforms to facilitate seamless communication and collaboration among students, teachers, and administrators. This system is envisioned as a comprehensive solution to bridge the gap between conventional classroom settings and modern, technology-driven education.

At its core, the Online Peer Tutoring System aims to empower students by providing them with easy access to a wide range of educational resources. Upon logging in, students can view assignments, tutorials, and instructional videos, all curated to support their learning journey. Additionally, the system offers a platform for real-time chats with peers, enabling students to discuss topics, ask questions, and share insights. This peer-to-peer interaction is a cornerstone of the system, fostering a collaborative and supportive learning environment where students can learn from one another.

Teachers also benefit significantly from this system. The user-friendly interface allows educators to upload assignments, tutorials, and videos with ease, ensuring that high-quality educational content is always available to students. This functionality not only streamlines the process of content delivery but also allows teachers to focus more on engaging with students and less on administrative tasks. By centralizing educational materials in one accessible platform, the system enhances the efficiency and effectiveness of teaching. Administrators play a vital role in maintaining the integrity and functionality of the Online Peer Tutoring System. With the ability to add or remove students and teachers, administrators ensure that the platform remains organized and up-to-date. This management capability is essential for creating a secure and well-regulated environment where both students and teachers can thrive.

This project report delves into the intricate design, implementation, and myriad benefits of the Online Peer Tutoring System. It highlights the system's potential to transform educational practices by providing a more flexible, interactive, and collaborative learning environment. Through detailed analysis and insights, the report underscores the system's capacity to enhance student outcomes, making it a valuable addition to the modern educational toolkit.

## 1.1 BACKGROUND

The concept of peer tutoring has long been recognized as an effective educational strategy, wherein students support each other's learning under the guidance of a teacher. Traditional peer tutoring typically involves face-to-face, which, while beneficial, can be limited by logistical challenges such as scheduling conflicts and geographical constraints. With the rapid advancement of technology and the increasing prevalence of digital learning platforms, there has been a significant shift towards online educational tools that offer greater flexibility and accessibility.

The Online Peer Tutoring System emerges from the need to adapt traditional peer tutoring methods to a digital format, making them more accessible and efficient. This system integrates several key functionalities: it allows teachers to upload assignments and tutorials, enables students to read these materials, and facilitates real-time communication between peers through chat features. Such a platform not only supports asynchronous learning, where students can access materials at their convenience, but also promotes synchronous interaction, where immediate feedback and collaborative problem-solving can occur.

Recent studies have shown that peer-assisted learning can lead to improved academic performance, higher engagement levels, and enhanced social skills among students. By moving peer tutoring to an online environment, we can overcome the limitations of physical presence and provide a scalable solution that caters to a diverse student population. Moreover, the COVID-19 pandemic has underscored the importance of having robust online education systems in place, as schools and universities worldwide were compelled to shift to remote learning almost overnight.

The Online Peer Tutoring System is designed to harness the benefits of peer-assisted learning within a flexible and user-friendly digital platform. It aims to create an inclusive educational environment where knowledge is shared freely, and students can benefit from the collective expertise of their peers. By facilitating efficient communication and interaction, this system not only enhances learning outcomes but also fosters a sense of community and collaboration among students.

## 1.2 SCOPE

The Online Peer Tutoring System is designed to create an interactive and collaborative educational platform for students and teachers. Teachers can upload assignments, tutorials and videos Students have access to these educational materials and can engage in real-time chats with peers to discuss assignments and clarify doubts. Administrative features include user management. The technological framework ensures scalability and integration with existing Learning Management Systems (LMS). This system aims to enhance the quality of education by making it more accessible, engaging, and effective for all users. engaging, and effective for all users.

The scope of an online peer tutoring system is broad and multifaceted, aimed at enhancing the educational experience through various features and functionalities. Such a system provides 24/7 accessibility, allowing students from remote or underserved areas to benefit from flexible learning opportunities. It supports a wide range of subjects and educational levels, with specialized tutors available for advanced topics. Personalization is key, with tailored learning experiences.

## 1.3 OBJECTIVE

The objective of the Online Peer Tutoring System is to revolutionize the educational landscape by providing a versatile and interactive digital platform that fosters effective communication, collaboration, and knowledge sharing among students and teachers. By leveraging technology, the system aims to enhance access to educational resources, facilitate peer-to-peer learning, support teacher-student interaction, promote collaborative learning experiences, ensure flexibility and accessibility, and streamline administrative processes. Ultimately, the system seeks to empower both students and teachers to actively engage in the learning process, leading to improved academic outcomes and the creation of a dynamic and supportive learning community.

## 1.4 PROPOSED SYSTEM

The Online Peer Tutoring System is a comprehensive digital platform designed to enhance the educational experience by facilitating effective communication and collaboration among students and teachers. Key features include:

Teacher Module: Allows teachers to upload assignments, tutorials and videos.

Student Module: Enables students to access learning materials, engage in real-time chats with peers and share documents.

Peer Interaction: Promotes collaborative learning through study groups and peer discussions, fostering a supportive learning environment.

Administration: Offers student and teacher management and robust security measures to ensure smooth operation and data privacy.

Technological Framework: Ensures scalability and seamless integration with existing Learning Management Systems.

This system aims to improve educational outcomes by providing a interactive, and user-friendly platform for both students and teachers.

## **Advantages**:

The Online Peer Tutoring System offers:

Enhanced Accessibility: Learning materials are accessible anytime, anywhere.

Improved Communication: Real-time chats and forums facilitate instant feedback and interaction.

Collaborative Learning: Students can work together and support each other’s academic growth.

Efficient Resource Management: Easy upload and organization of educational materials.

Data-Driven Insights: Analytics provide valuable information for informed decision-making.

This system fosters an engaging, inclusive, and efficient learning environment for students and teachers alike.

# CHAPTER 2

# SOFTWARE REQUIREMENTS SPECIFICATION

## 2.1 PRODUCT OVERVIEW

Our Online Peer Tutoring Platform is designed to create an enriching and collaborative educational environment for students, teachers, and administrators. Students can access a diverse range of educational resources, including videos, tutorials, and assignments, all uploaded by their teachers. This ensures that students have the necessary materials to enhance their learning experience. Additionally, the platform features a chat function that allows students to communicate with their peers, fostering a collaborative and supportive learning community where they can discuss topics, share ideas, and seek help from one another. For teachers, the platform offers a user-friendly portal to upload and manage educational content. Teachers can easily upload videos, tutorials, and assignments, making them readily available to students. They can also track student progress, monitor assignment submissions, and provide timely feedback, ensuring that students stay on track with their studies and receive the support they need. Administrators play a crucial role in maintaining the platform's functionality and organization. They have the authority to manage user accounts, adding or removing students and teachers as needed. Overall, our Online Peer Tutoring Platform is a comprehensive solution designed to enhance learning, support collaboration, and streamline educational management.

## 2.2 PRODUCT FUNCTIONALITY

- Students can sign up and log in to access the platform.
- Students can read assignments and tutorials.
- Students can access learning materials.
- Students can chat with their peers for collaborative learning.
- Students can upload documents.
- Teachers can upload and manage assignments and tutorials.

- Teachers can upload videos.
- Admins can add or remove students and teachers.
- User-friendly GUI.
- Real-time communication support.
- Scalable architecture.
- Compliance with educational data privacy regulations.

## 2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

- Real-Time Communication: Efficient support for real-time chat functionality.
- Data Security: Robust encryption and role-based access control.
- Scalability: Scalable architecture to handle varying loads.
- User Interface: Maintain a user-friendly interface for all user roles.
- Regulatory Compliance: Adherence to educational data privacy regulations.
- Integration: Seamless integration with existing school management systems.

## 2.4 HARDWARE REQUIREMENTS

System - Intel3core or higher

Memory - 4GB RAM and above

Monitor - 14'Color Monitor

Mouse - Optical mouse

## 2.5 SOFTWARE REQUIREMENTS

Operating system - Windows10 and above

IDE - VS CODE

Front End – HTML, CSS,ReactJs

JavaScript for interactive elements.

Nodejs and Express for server-side scripting.

MySQL for the database.

Extra tool –chatengine.io

## 2.6 FUNCTIONALITY REQUIREMENT

### STUDENT

- Login: Students log in to their account.
- Browse Content: Access and read assignments, tutorials and videos.
- Engage with Peers: Use the chat feature to discuss assignments or share knowledge and can upload documents.

### TEACHER

- Login: Teachers log in to their account.
- Upload Content: Upload assignments, tutorials and videos.
- View current assignments and tutorials.

### ADMIN

- Login: Admins log in to their account.
- Manage Users: Add or remove students and teachers as needed.
- Monitor System: Oversee activities and ensure the smooth operation of the platform.

## 2.7 USECASE DIAGRAM

A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user's perspective, aiding in the communication between stakeholders and guiding the development process. It is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circle or ellipses, The actors are often shown as stick figures.

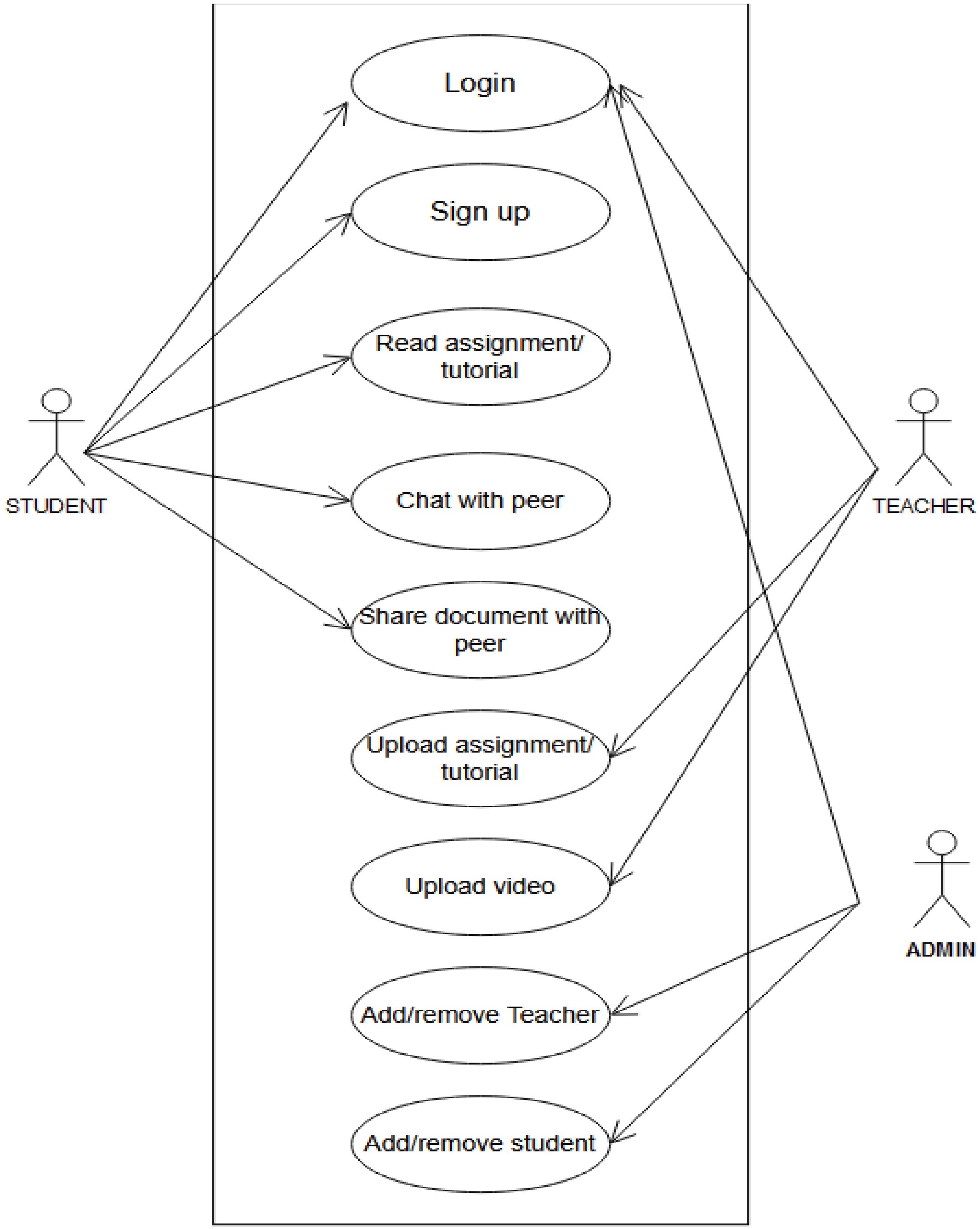


**FIG2.7: USECASE**

# CHAPTER 3

# SYSTEM ARCHITECTURE DESIGN

System architecture design is the conceptual model that defines the structure, behavior, and more views of a system. It involves the design and specification of the hardware, software, and network infrastructure necessary to meet the requirements and objectives of a particular system. Design process is the process through which designers design interface in software or electronic devices with an emphasis on aesthetics or style and is termed user interface design. Here we use different design processes like dataflow diagram, activity diagram, use case diagram to implement our project.

## 3.1 DESIGN PHASE

It is a visual representation of the system architecture. It shows the connections between the various components of the system and indicates what functions each component performs. The design phase is a critical stage in the development of the Online Peer Tutoring System, as it lays the foundation for the entire project. It serves as the blueprint for the project, ensuring that all subsequent phases of development are aligned with the intended functionality, user experience, and overall objectives.

**Importance of the Design Phase in the Online Peer Tutoring System Project**

1. Clear Vision and Direction:

- The design phase helps establish a clear vision and direction for the project. By defining the system architecture, user interfaces, data flows, and interactions, it ensures that everyone involved has a common understanding of what the final product will look like and how it will operate.
- This clarity helps in aligning the efforts of all team members and stakeholders, reducing the risk of misunderstandings and scope creep.

2. Requirement Validation and Feasibility:

- During the design phase, detailed analysis of the requirements is performed to ensure that they are feasible and realistic. This phase helps in identifying any gaps or inconsistencies in the requirements, allowing for adjustments before moving to development.

3. System Architecture and Scalability:

- The design phase involves creating a comprehensive system architecture that outlines how different components will interact and integrate. This includes defining the database schema, application logic, user interfaces, and communication protocols.
- A good design ensures that the system is scalable, maintainable, and adaptable to future changes. It allows for easy addition of new features and modules without major rework.

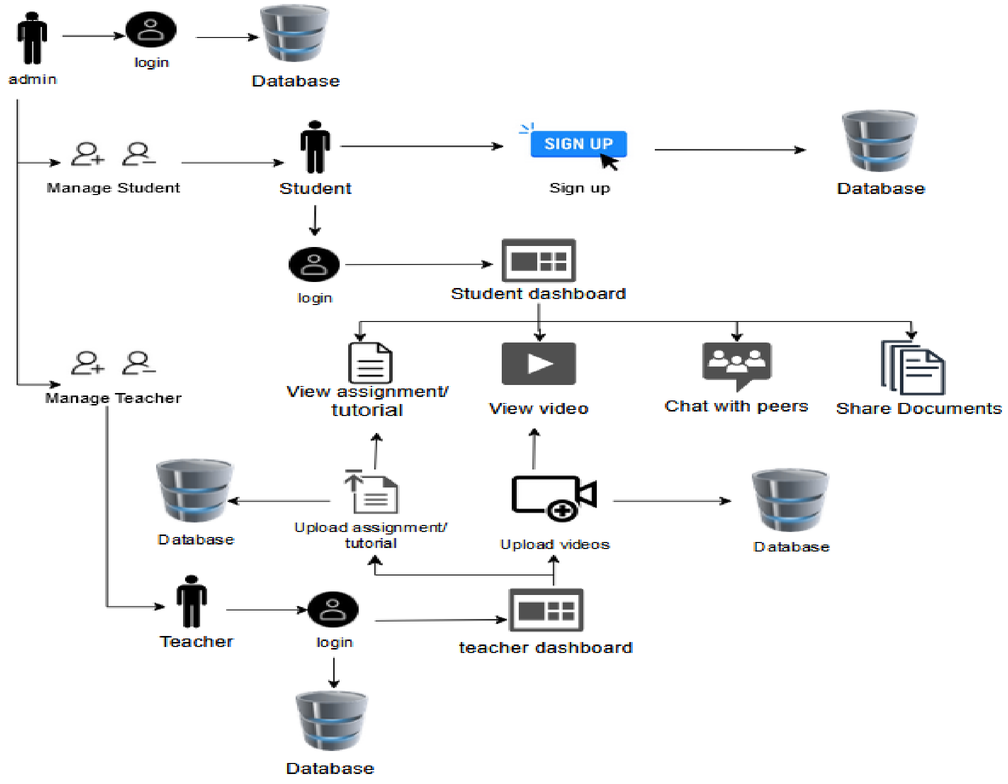
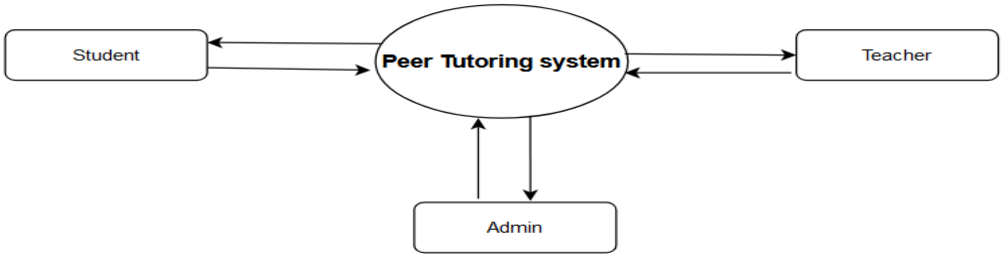


FIG3.1: DESIGN PHASE

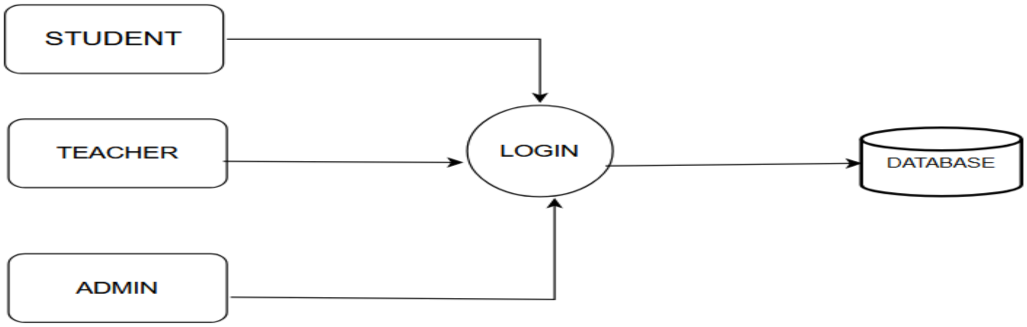
## 3.2 DATAFLOW DIAGRAM

The data flow diagram shown below illustrates the general structure of the system. Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams.

### LEVEL0



### LEVEL 1



## LEVEL 2

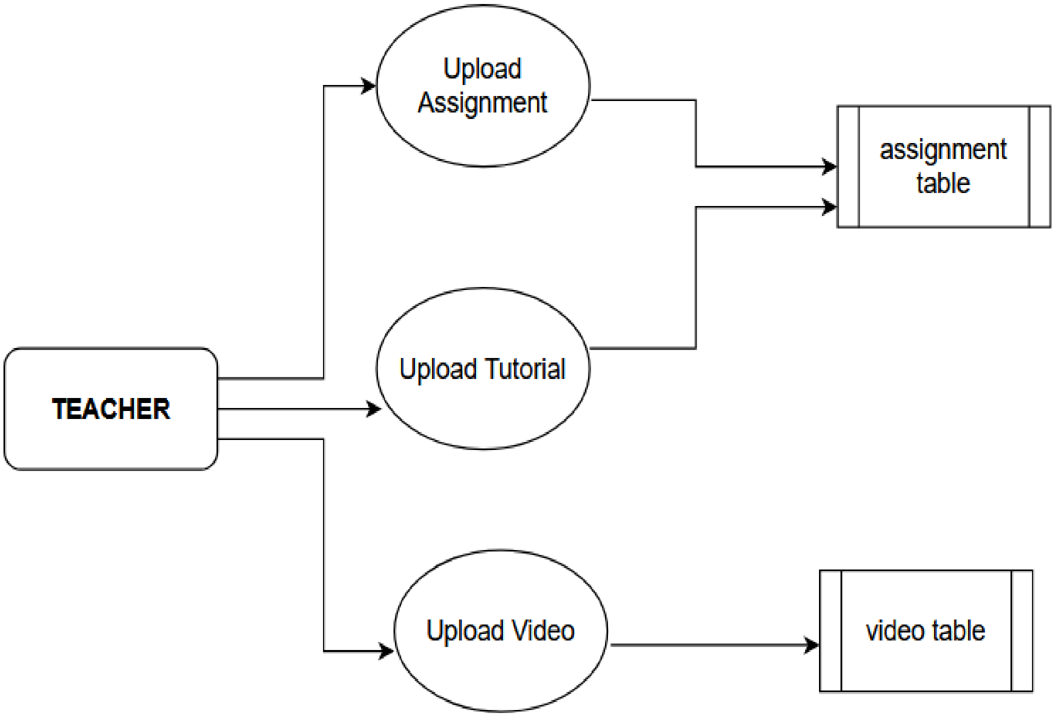


FIG3.2:DATA FLOW DIAGRAM

## 3.3 ACTIVITY DIAGRAM

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modelling how a collection of use cases coordinates to represent business workflows

**Importance of Activity Diagrams**

An Activity Diagram is a vital tool in the analysis and design phases of the Online Peer Tutoring System. It provides a detailed visual representation of the workflows and processes within the system, illustrating the sequence of activities and the flow of control from one activity to another. Here are the key reasons why Activity Diagrams are important for this project:

1. Detailed Workflow Representation:

- Activity diagrams capture the step-by-step workflow of various processes in the system. For the Online Peer Tutoring System, this could include logging in, accessing tutorials, uploading assignments, and managing user accounts.
- By visualizing these workflows, stakeholders can clearly see how different activities interconnect and progress, ensuring a thorough understanding of each process.

2. Enhanced Communication:

- Like Data Flow Diagrams, activity diagrams provide a clear and intuitive way to communicate complex processes. They help bridge the gap between technical and non-technical stakeholders by presenting processes in a visual format that is easy to understand.
- This improves collaboration and ensures that everyone involved has a common understanding of the system's operations.

3. Identifying Bottlenecks and Inefficiencies:

- By mapping out the flow of activities, activity diagrams can help identify potential bottlenecks or inefficiencies in the system. For example, they can reveal unnecessary steps or redundant processes that can be streamlined.
- This optimization can enhance the system's performance and improve the user experience for students, teachers, and administrators.

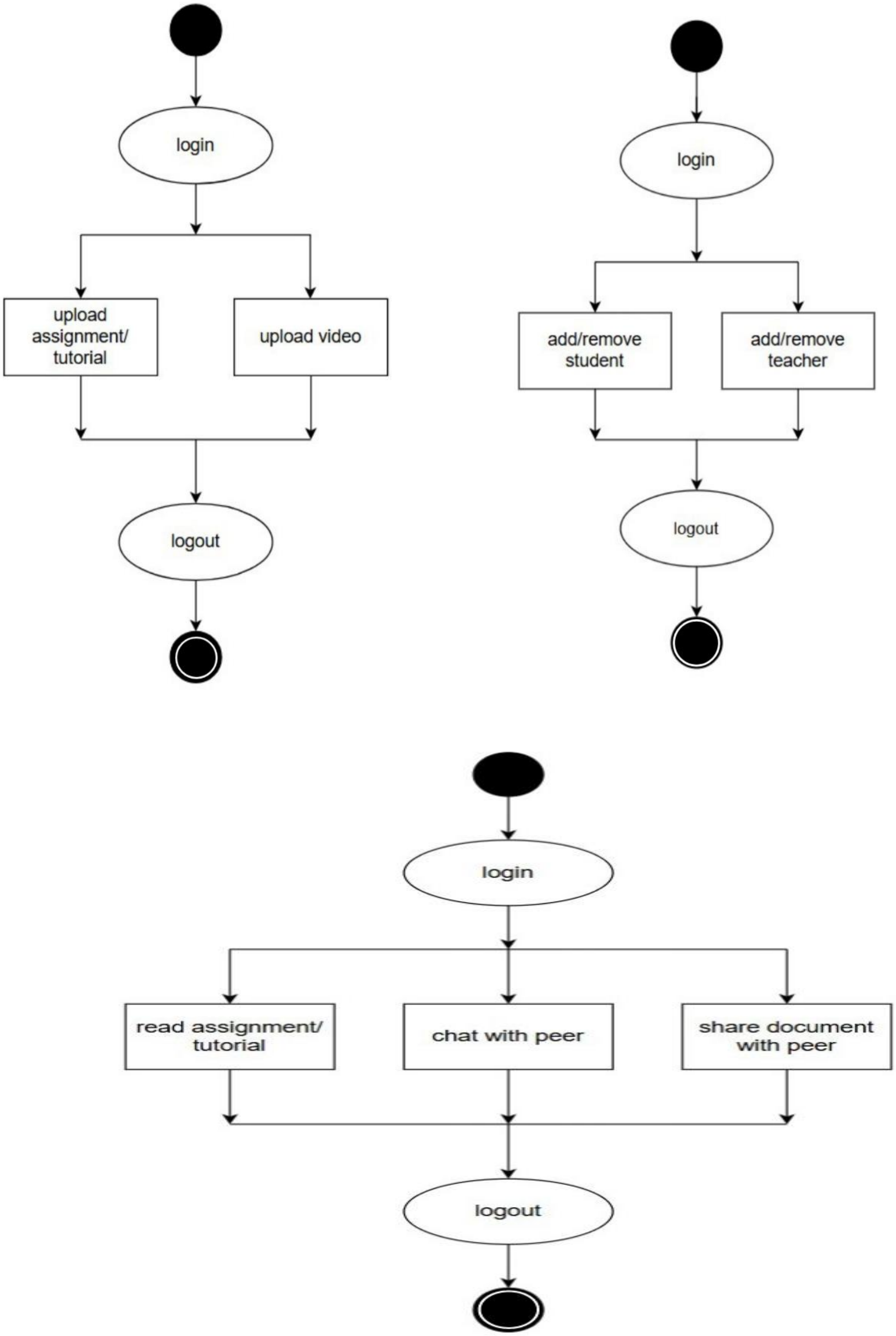


FIG3.3: ACTIVITY DIAGRAMS

## 3.4 DATABASE DESIGN

Database design is the organization of data according to a database model. Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

**Importance of a Database Design Diagram**

A Database Design Diagram is an essential tool in the development of the Online Peer Tutoring System, encompassing entities such as admin, student, teacher, assignment, and videos. . Here are the key reasons why a Database Design Diagram is important for this project:

1. Clear Visualization of Data Structure:

- The diagram provides a clear and organized visualization of the data structure, showing how different entities such as admin, student, teacher, assignment, and videos relate to each other.
- This visualization helps stakeholders understand how data is organized and interconnected, facilitating better decision-making during the design and development phases.

2. Efficient Database Design:

- By detailing the entities and their relationships, the database design diagram ensures an efficient database design. It helps in structuring the database to minimize redundancy, avoid data anomalies, and maintain data integrity.
- For instance, relationships between students and assignments or teachers and videos can be precisely mapped, ensuring that data is stored in a logical and efficient manner.

4. Identification and Resolution of Potential Issues:

- The diagram helps identify potential issues such as missing entities, incorrect relationships, or redundant attributes early in the design phase. Resolving these issues early can prevent costly modifications later in the development process.
- It ensures that all necessary data requirements are captured, reducing the risk of data-related problems during system implementation.

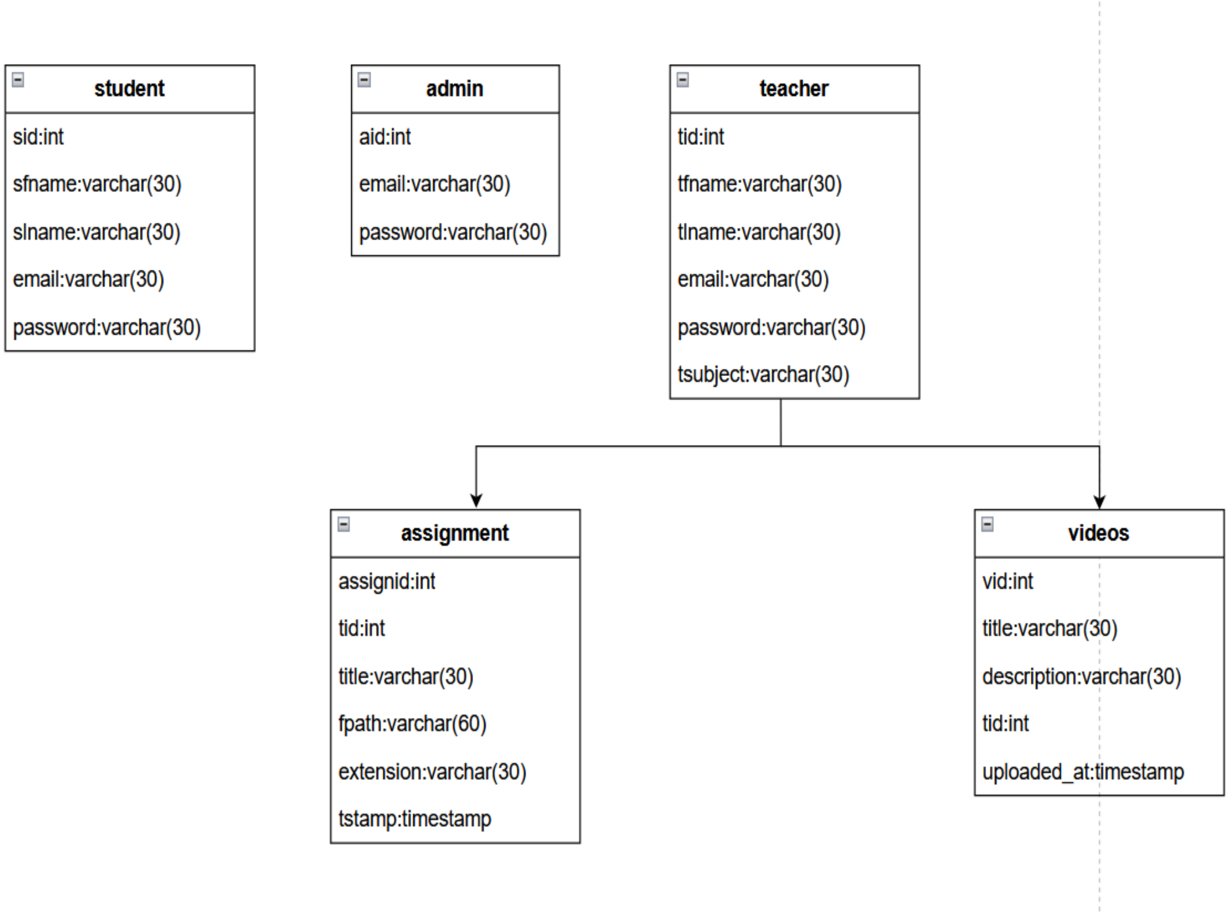


FIG3.4: DATABASE DESIGN

# CHAPTER 4

# IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

Implementing the Peer Tutoring Platform Project involves several key steps to ensure its success. To build an online tutoring system using HTML, CSS, JavaScript, MySQL, Node.js, and Express, we start with the backend by setting up Node.js and Express for server-side logic. We define routes for functionalities such as uploading tutorial videos, managing assignments, and sharing documents. MySQL is used to create and manage the database schema, including tables for users (teachers, students, and admins), tutorial videos, assignments, and shared documents. Teachers upload tutorial videos and assignments through dedicated endpoints, storing the files on the server and data in the MySQL database, making them accessible to students via secure routes.

For the frontend, we use HTML and CSS to design a responsive and user-friendly interface, and JavaScript to handle dynamic interactions. The front end allows teachers to upload content and students to access tutorials, assignments, and shared documents. Additionally, a real-time chat feature is implemented using chatengine.io enabling students to communicate with each other instantly.

Admins can add or remove users and assign roles through a secure interface, ensuring proper access control and management of the platform. Document sharing is facilitated by enabling students to upload and share files, with uploads stored on the server and tracked in the MySQL database. This comprehensive system ensures data security through robust authentication and authorization mechanisms, providing a seamless experience for online tutoring with video tutorials, assignments, real-time chat, and document sharing capabilities.

## 4.1 Login

```
app.get("/login", function (req, res) {
  res.sendFile(path.join(__dirname, "public", "html", "studentlogin.html"));
});
const roleToTableMap = {
  student: 'student',
  teacher: 'teacher',
  admin: 'admin'
};

app.post('/login', async (req, res) => {
  const { email, password, role } = req.body;
  const tableName = roleToTableMap[role];

  if (!tableName) {
      return res.status(400).send('Invalid role specified');
  }

  const query = SELECT * FROM ${mysql.escapeId(tableName)} WHERE email = ?;

  pool.query(query, [email], async (error, results) => {
      if (error) {
          console.error('Database error:', error);
          return res.status(500).send('Database error');
      }

      if (results.length === 0) {
          return res.status(401).send('Login failed: Invalid email');
      }
      function checkAuthentication(req, res, next) {
        if (req.session.userId && req.session.role) {
          next();
        } else {
          res.status(403).send("Not authorized");
        }
      }

      function checkRole(role) {
        return function(req, res, next) {
          if (req.session.role === role) {
            next();
          } else {
            res.status(403).send("Access denied");
          }
        }
      }
      const user = results[0];
```

```
   // Check password
   const isMatch = password === user.password//await bcrypt.compare(password, user.password);
   if (!isMatch) {
     return res.status(401).send('Login failed: Invalid password');
   }

   req.session.userId = user.tid; // assuming id is a field in your user record
   req.session.role = role;

   if (role === 'teacher') {
    req.session.tid = user.tid; // Assuming 'tid' is the identifier for teacher ID in your 'teacher' table
   }

   if (role === 'student') {
    req.session.sid = user.sid; // Assuming 'tid' is the identifier for teacher ID in your 'teacher' table
   }
   // Redirect based on role
   switch (role) {
     case 'student':
      const sqlAssignments = 'SELECT title, fpath FROM assignment';
      pool.query(sqlAssignments, (err, assignments) => {
       if (err) {
        console.error('Database error on assignments:', err);
        return res.status(500).send('Failed to retrieve assignments');
       }

       const sqlTutorials = 'SELECT title, fpath FROM tutorial';
       pool.query(sqlTutorials, (err, tutorials) => {
        if (err) {
         console.error('Database error on tutorials:', err);
         return res.status(500).send('Failed to retrieve tutorials');
        }

        res.render('studentdashboard', { assignments: assignments, tutorials: tutorials });
        const sqlVideos = 'SELECT title  FROM videos';
pool.query(sqlVideos, (err, videos) => {
 if (err) {
  console.error('Database error on videos:', err);
  return res.status(500).send('Failed to retrieve videos');
 }

 res.render('courses', {

  videos: videos
 });   });

       });
      });
      break;

       break;
     case 'teacher':
       res.redirect('/teacherdashboard.html');
       break;
     case 'admin':
            res.redirect('/admindashboard.html');
            break;
        default:
            res.status(401).send('Role does not have a designated dashboard');
            break;
    }

  });
});
```

# Sign up

```
app.get("/signup.html", function (req, res) {
  res.sendFile(path.join(__dirname, "public", "html", "signup.html"));
});

app.post("/signup.html", function (req, res) {
    const { sfname, slname, email, password } = req.body;
    var sql = "INSERT INTO student(sfname, slname, email, password) VALUES ?";
    var values = [[sfname, slname, email, password]];

    pool.query(sql, [values], function (error, results) {
      if (error) {
        console.error("Failed to insert data:", error);
        res.status(500).send("Error registering new student");
        return;
      }
      res.send("Student Register successful with ID: " + results.insertId);
    });
  })
```

# Add Students

```
app.get("/managestudents.html", function (req, res) {
  res.sendFile(
    path.join(__dirname, "public", "html", "admin", "managestudents.html")
  );
});app.post("/managestudents.html", function (req, res) {
    const { sfname,slname, email } = req.body;
    var sql = "INSERT INTO student(sfname,slname, email) VALUES ?";
    var values = [[sfname,slname, email]];

    pool.query(sql, [values], function (error, results) {
      if (error) {
        console.error("Failed to insert data:", error);
        res.status(500).send("Error registering new student");
        return;
      }
      res.send("Student Register successful with ID: " + results.insertId);
    });
  });
```

# Add Teachers

```
app.get("/manageteachers.html", function (req, res) {
   res.sendFile(
     path.join(__dirname, "public", "html", "admin", "manageteachers.html")
   );
 });
 app.post("/manageteachers.html", function (req, res) {
   const { tfname,tlname, email,tsubject } = req.body;
   var sql = "INSERT INTO teacher(tfname, tlname,email,tsubject) VALUES ?";
   var values = [[tfname,tlname, email,tsubject]];

   pool.query(sql, [values], function (error, results) {
     if (error) {
       console.error("Failed to insert data:", error);
       res.status(500).send("Error registering new teacher");
       return;
     }
     res.send("teacher Register successful with ID: " + results.insertId);
   });
 });
```

# Assignment Upload

```
app.get("/assignment.html", function (req, res) {
  res.sendFile(path.join(__dirname, "public", "html", "teacher", "assignment.html"));
});

app.post('/assignment.html', upload.single('file'), function(req, res) {
  if (!req.file) {
    return res.status(400).send('No file uploaded.');
  }
  if (!req.body.title) {
    return res.status(400).send('Assignment title is required.');
  }

  const title = req.body.title;
  const tid = req.session.tid; // Assumes session management is setup
  const fPath = req.file.path; // Correct variable name usage

  const sql = 'INSERT INTO assignment (tid, title, fpath, tstamp) VALUES (?, ?, ?, NOW())';
  const values = [tid, title, fPath];

  pool.query(sql, values, function(error, results) {
    if (error) {
      console.error('Error uploading assignment:', error);
      return res.status(500).send('Error uploading assignment');
    }
    res.send('Assignment uploaded successfully');
```

```
 });
```

# Assignment Retrieval

```
app.set('view engine','ejs');
 app.get('/studentdashboard', (req, res) => {
  const sqlAssignments = 'SELECT title, fpath FROM assignment';
  pool.query(sqlAssignments, (err, assignments) => {
    if (err) {
      console.error('Database error on assignments:', err);
      res.status(500).send('Failed to retrieve assignments');
      return;
    }

    const sqlTutorials = 'SELECT title, fpath FROM tutorial';
    pool.query(sqlTutorials, (err, tutorials) => {
      if (err) {
        console.error('Database error on tutorials:', err);
        res.status(500).send('Failed to retrieve tutorials');
        return;
      }

      res.render('studentdashboard', { assignments: assignments, tutorials: tutorials });
    });
  });
});


app.get('/assignment/:fpath', function(req, res) {
  const fpath = req.params.fpath;
  console.log("Requested file path:", fpath);

  if (!fpath) {
    return res.status(400).send('Bad Request: File path is undefined');
  }

  // Assuming fpath is a full path, you might need to extract just the filename
  const filename = path.basename(fpath); // Extracts the filename from the path

  res.download(fpath, filename, function(err) {
    if (err) {
      console.error('Error downloading file:', err);
      res.status(404).send('File not found');
    } else {
      console.log('File downloaded successfully:', fpath);
    }
  });
});
```

## Chat

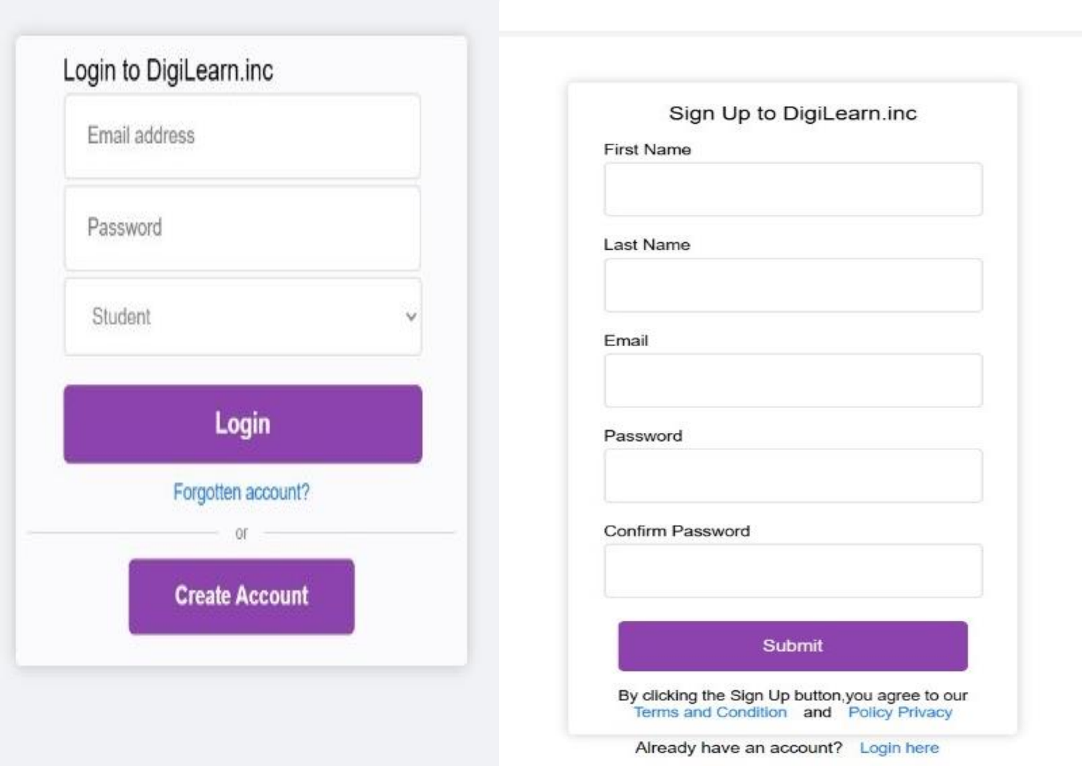
```
const express = require("express");
const cors = require("cors");
 const app = express();
app.use(express.json());
app.use(cors({ origin: true }));
const axios = require("axios");
app.post("/authenticate", async (req, res) => {
  const { username } = req.body;
  try {
    const r = await axios.put(
      "https://api.chatengine.io/users/",
      { username: username, secret: username, first_name: username },
      { headers: { "Private-Key": "XXXXXXXXX" } }
    );
    return res.status(r.status).json(r.data);
  } catch (e) {
    return res.status(e.response.status).json(e.response.data);
  }

});
 app.listen(3001);
```

# CHAPTER 5
# RESULTS AND DISCUSSION

In this section, experiments were analyzed to evaluate the performance of the system. As per the objective of our project we were able to complete. We created the front end of the system using HTML, CSS, JAVA SCRIPT and it is completed.

The implementation of the online peer tutoring system has yielded a robust platform where teachers can effortlessly upload tutorial videos and assignments, enhancing the educational resources available to students. Students can seamlessly access and engage with these materials, including reading assignments and watching tutorials. The system also facilitates peer interaction through a chat feature, fostering collaborative learning and discussion. Additionally, students can share documents, further enriching the collaborative environment. Administrative functionalities are well-supported, allowing admins to manage user roles by adding new teachers and students. Overall, this system has significantly streamlined the educational process, promoting a dynamic and interactive learning experience.



By clicking the Sign Up button,you agree to our Terms and Condition and Policy Privacy

Already have an account? Login here

**FIG 5.1: LOGIN/SIGNUP**

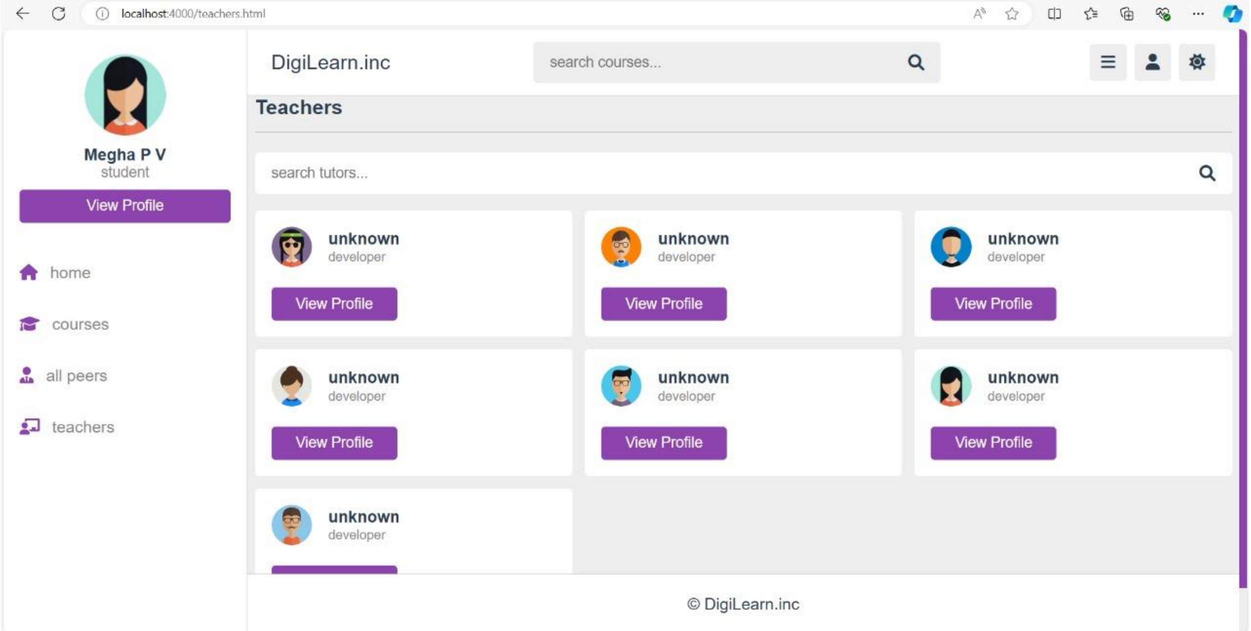


**Fig 5.2 Student Dashboard**

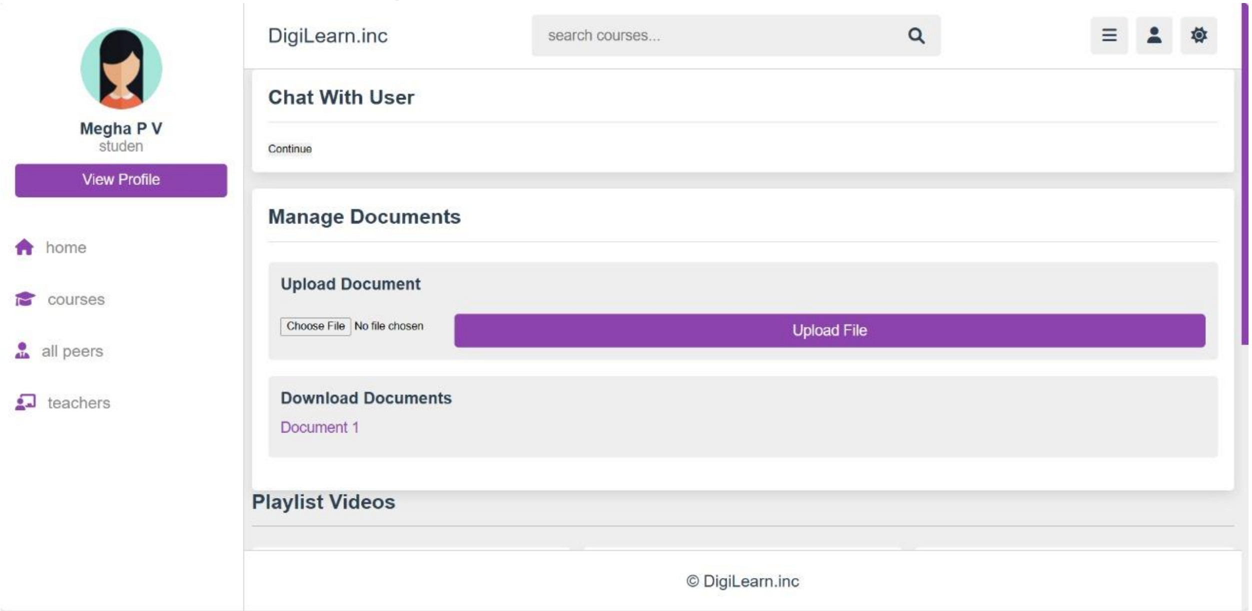


**Fig 5.2.1: Chat With Peer**

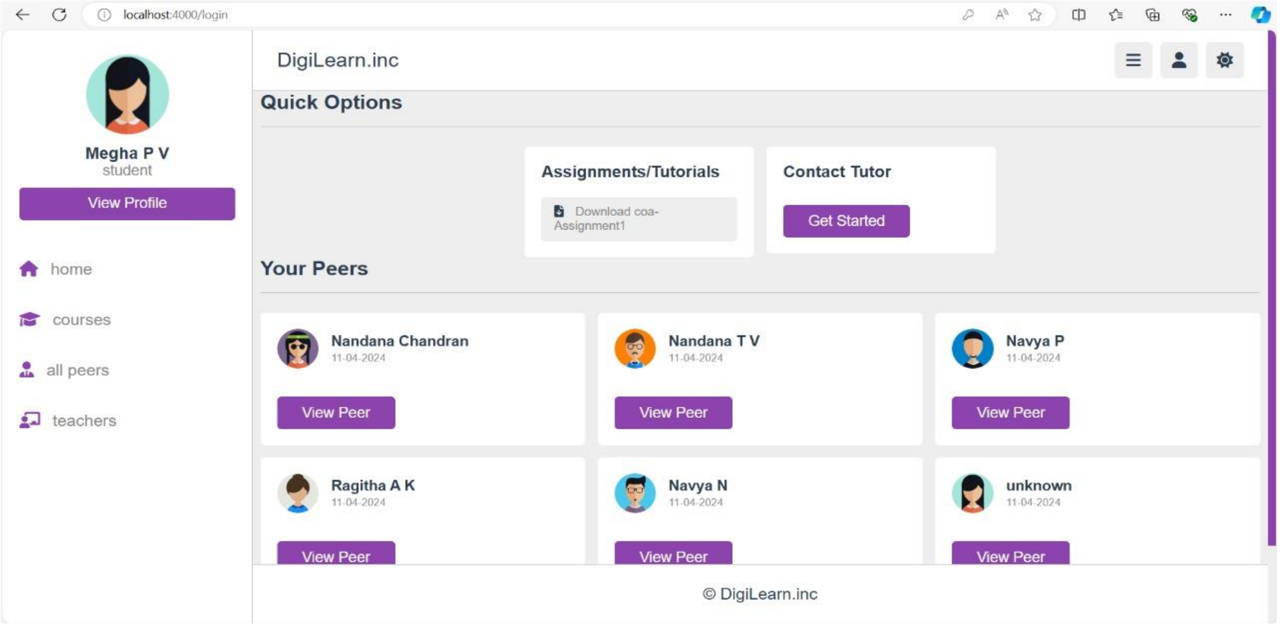


**Fig 5.2.2: View Assignments/tutorials**

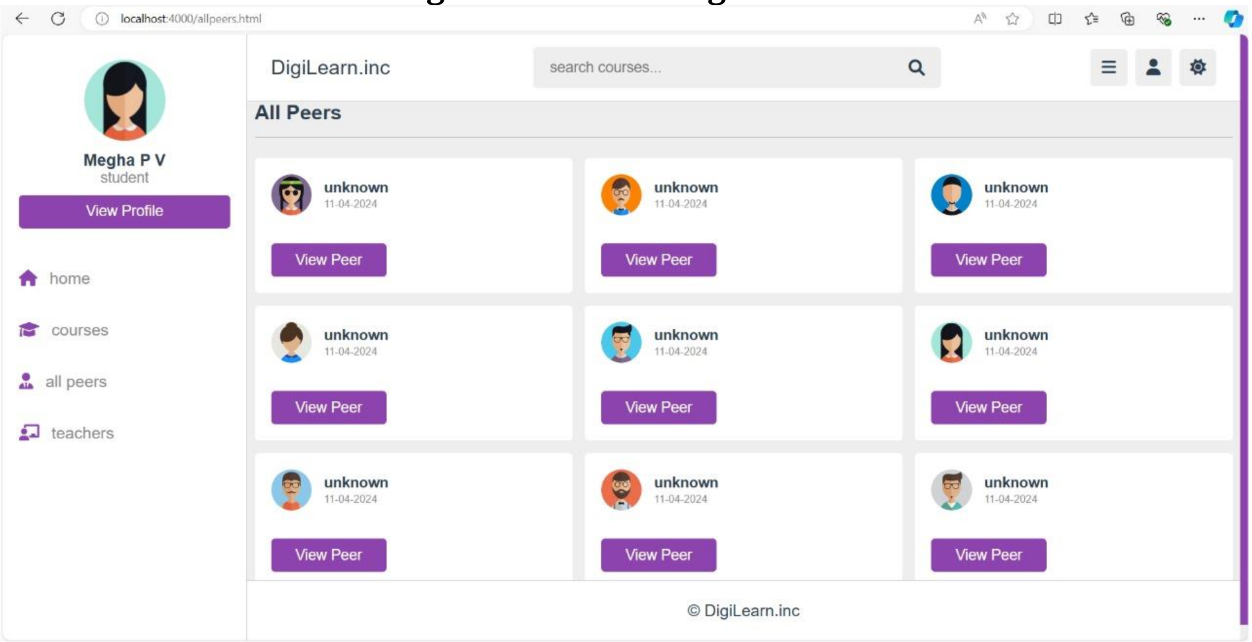


**Fig 5.2.3: All peers**

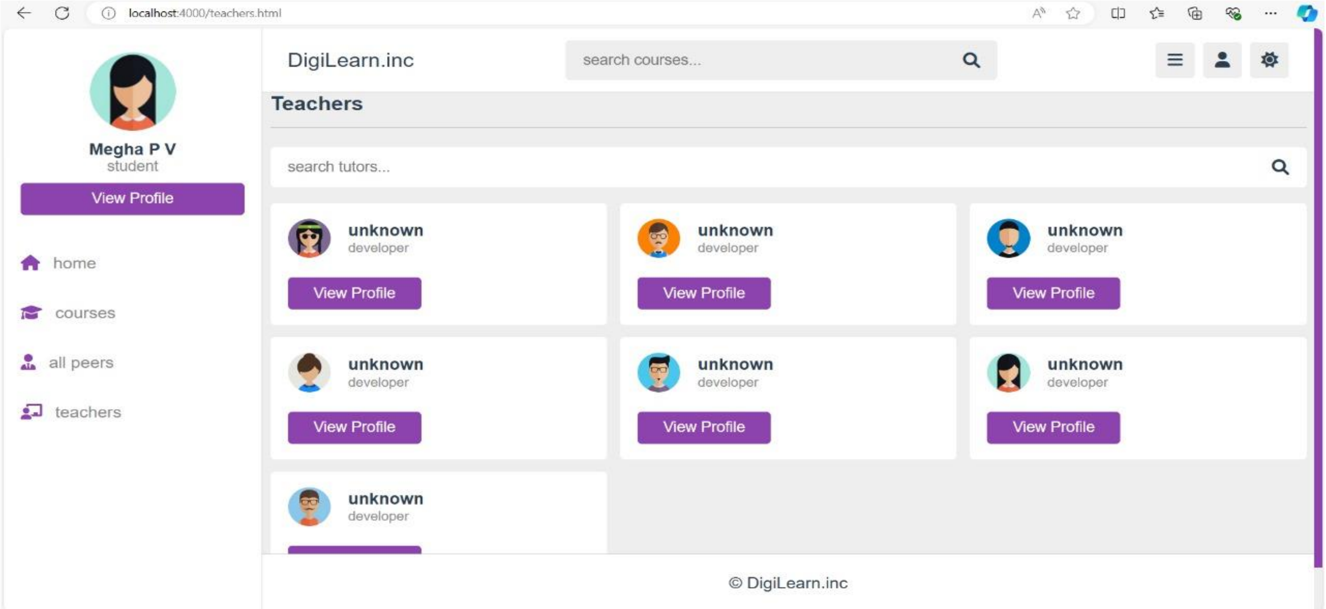


**Fig 5.2.4: Teachers**

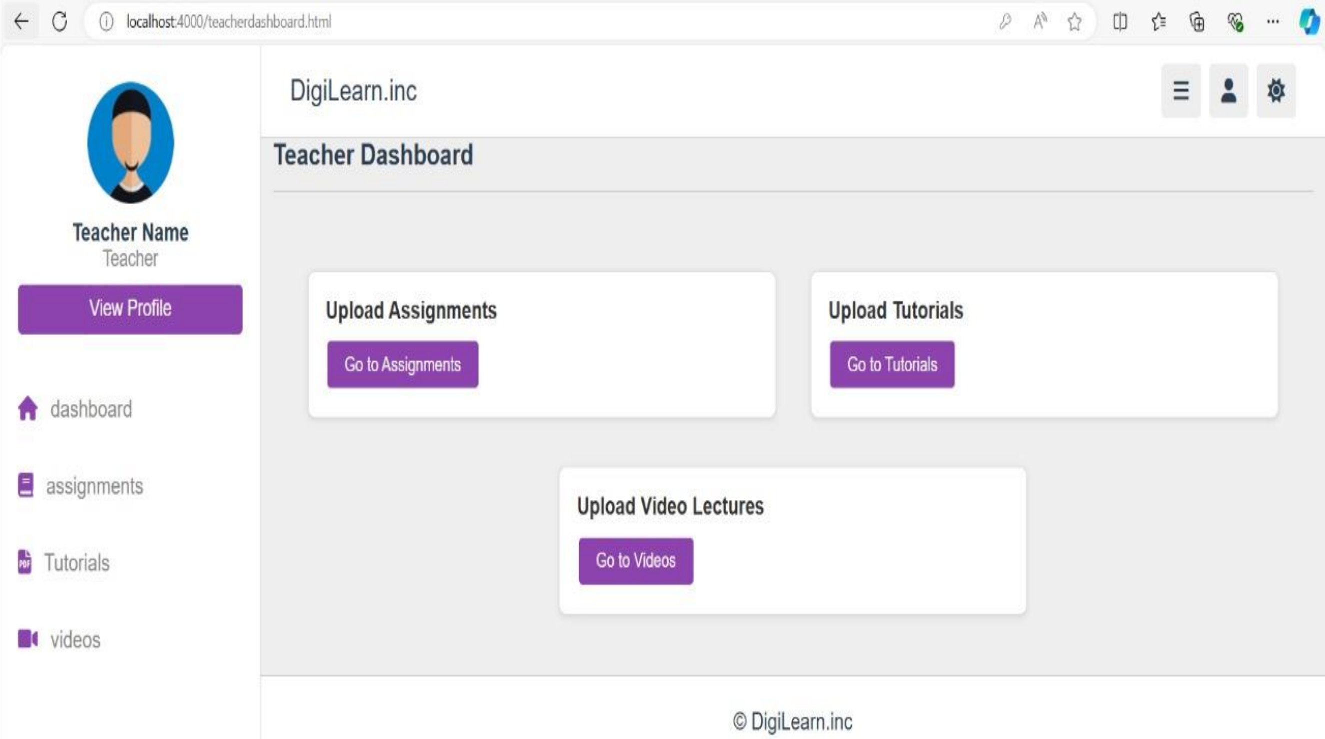
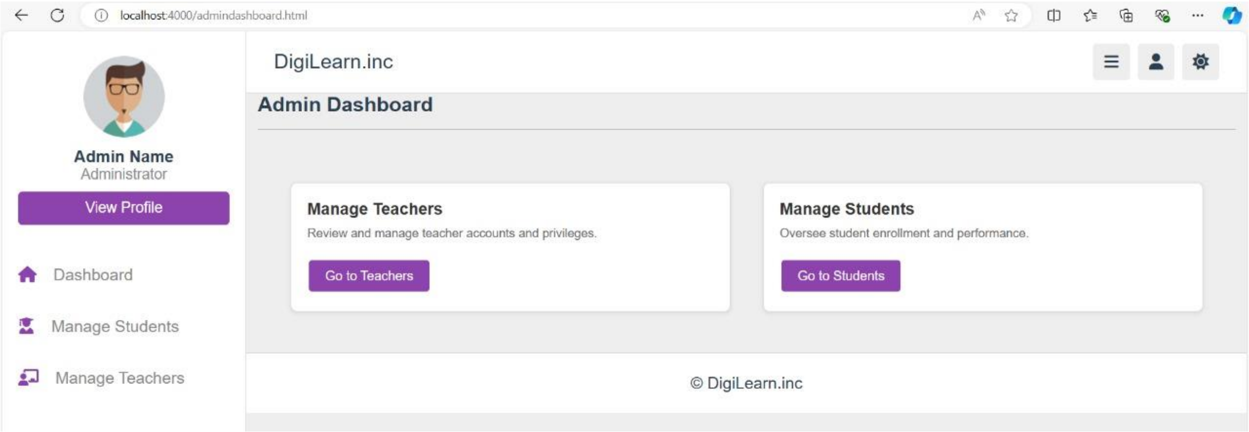


**Fig 5.3: Teacher Dashboard**



Admin Dashboard

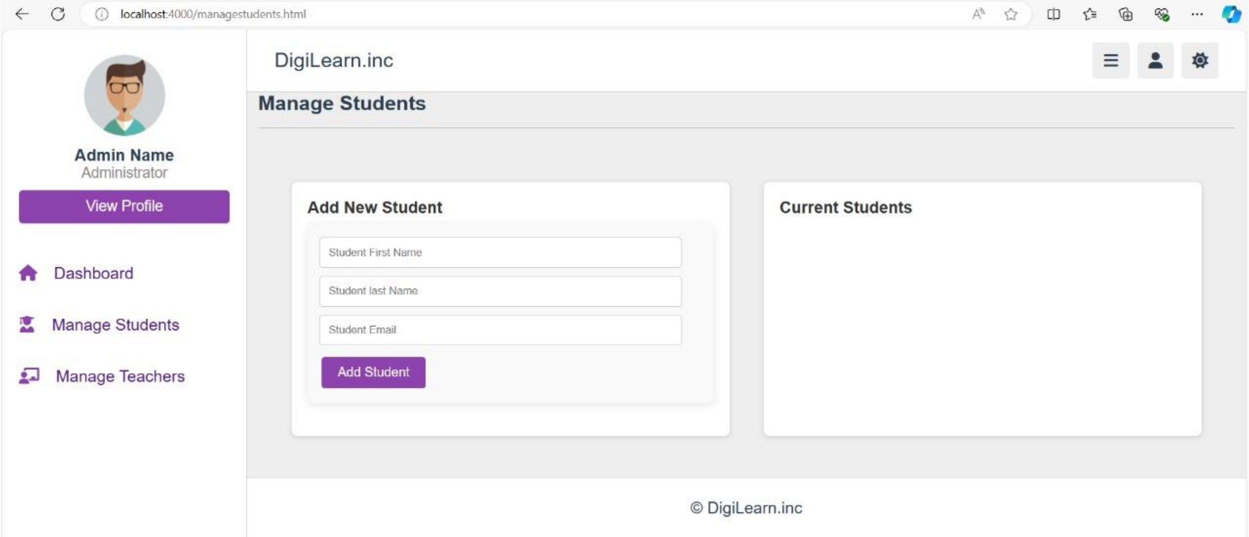
Manage Teachers

Review and manage teacher accounts and privileges.

Manage Students

Oversee student enrollment and performance.

**Fig 5.4:Admin Dashboard**

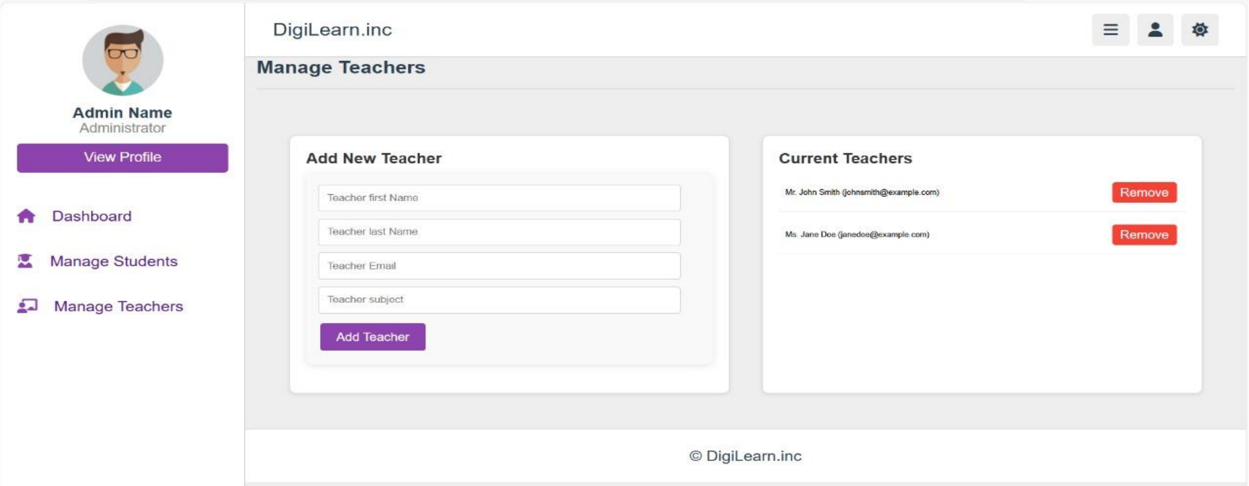


Manage Students

Add New Student

Current Students

**Fig 5.4.1:Manage Students**



Manage Teachers

Add New Teacher

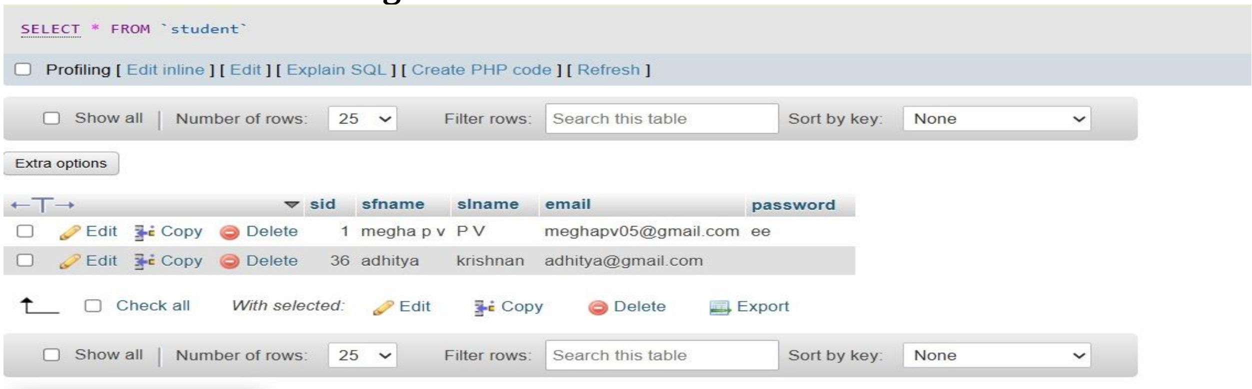
Current Teachers

Mr. John Smith (johnsmith@example.com)

Ms. Jane Doe (janedoe@example.com)

**Fig 5.4.2: Manage Teachers**

## Fig 5.5: DATABASES

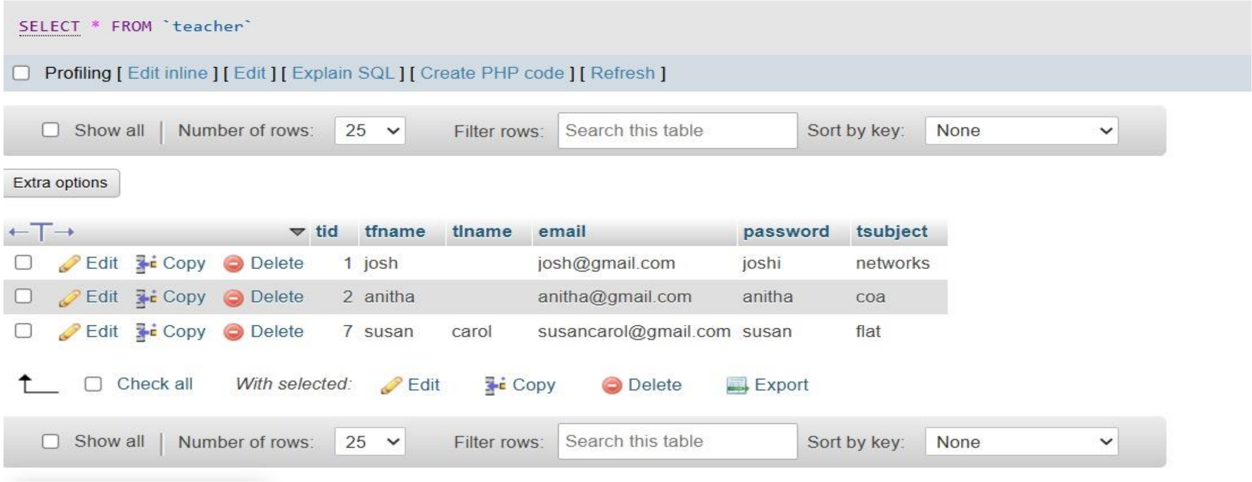


```
SELECT * FROM `student`
```

□ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
□ Check all With selected: Edit Copy Delete Export

| | sid | sfname | slname | email | password |
|---|---|---|---|---|---|
| □ Edit Copy Delete | 1 | megha p v | P V | meghapv05@gmail.com | ee |
| □ Edit Copy Delete | 36 | adhitya | krishnan | adhitya@gmail.com | |

Fig 5.5.1 student

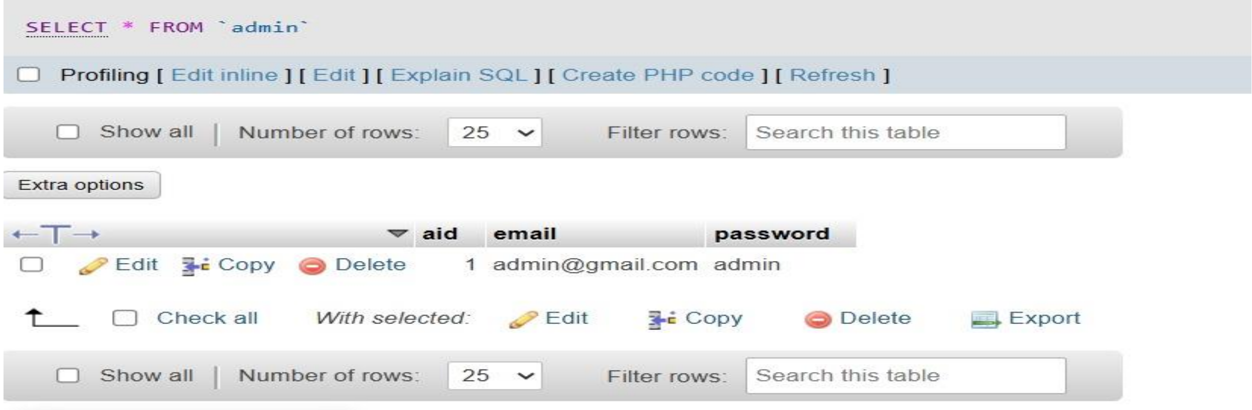


```
SELECT * FROM `teacher`
```

□ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
□ Check all With selected: Edit Copy Delete Export

| | tid | tfname | tlname | email | password | tsubject |
|---|---|---|---|---|---|---|
| □ Edit Copy Delete | 1 | josh | | josh@gmail.com | joshi | networks |
| □ Edit Copy Delete | 2 | anitha | | anitha@gmail.com | anitha | coa |
| □ Edit Copy Delete | 7 | susan | carol | susancarol@gmail.com | susan | flat |

Fig 5.5.2 teacher

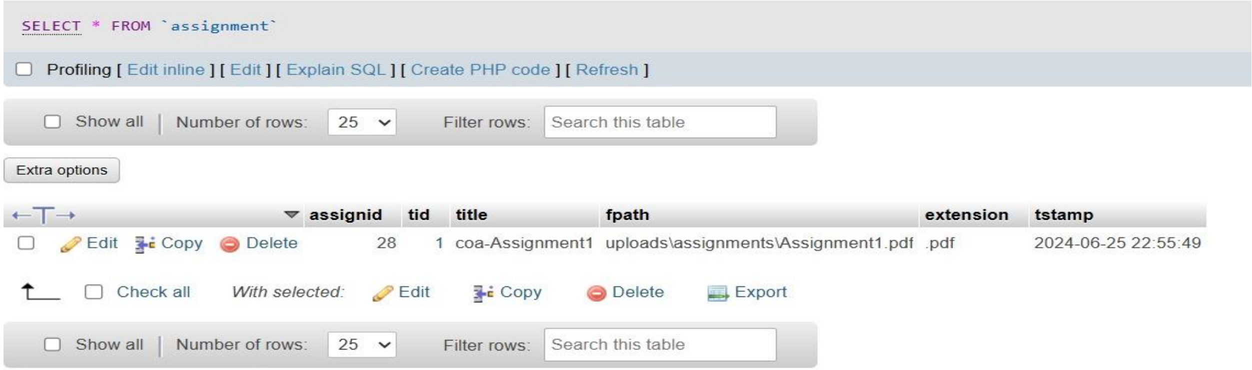


```
SELECT * FROM `admin`
```

□ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options
□ Check all With selected: Edit Copy Delete Export

| | aid | email | password |
|---|---|---|---|
| □ Edit Copy Delete | 1 | admin@gmail.com | admin |

Fig 5.5.3 admin

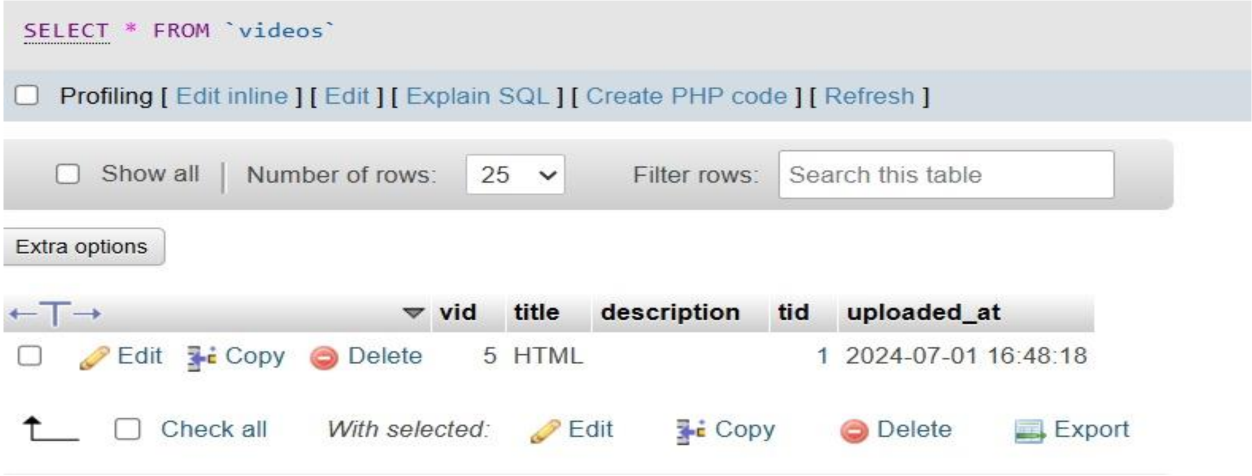


```
SELECT * FROM `assignment`
```

| assignid | tid | title | fpath | extension | tstamp |
|---|---|---|---|---|---|
| 28 | 1 | coa-Assignment1 | uploads\assignments\Assignment1.pdf | .pdf | 2024-06-25 22:55:49 |

□ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export
□ Show all | Number of rows: 25 Filter rows: Search this table

**Fig 5.5.5 assignment**



```
SELECT * FROM `videos`
```

| vid | title | description | tid | uploaded_at |
|---|---|---|---|---|
| 5 | HTML | | 1 | 2024-07-01 16:48:18 |

□ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options
□ Edit Copy Delete
□ Check all With selected: Edit Copy Delete Export

**Fig 5.5.6 videos**

CHAPTER 6

# CONCLUSION

The Online Peer Tutoring System represents a significant advancement in the realm of education, offering a comprehensive digital platform that promotes collaboration, communication, and knowledge sharing among students and teachers. Throughout this project, we have explored the various features and functionalities of the system, highlighting its potential to revolutionize traditional learning methodologies.

As we conclude this project, it is evident that the Online Peer Tutoring System has the potential to significantly enhance the quality of education by making it more accessible, engaging, and effective for all users. The Online Peer Tutoring Platform is not just a tool but a transformative approach to education. By embracing technology and leveraging the power of collaboration, we can create a brighter future for education, one where every student has the opportunity to thrive and succeed.

# BIBILOGRAPHY

## WEBSITES

- www.chatengine.io
- www.google.com
- www.youtube.com
- www.wikipedia.com
- www.geeksforgeeks.com
- www.javatpoint.com
- www.stackoverflow.com