AI POWERED PLACEMENT PREPERATION PLATFORM

A MINI PROJECT REPORT

Submitted by

NILANI S [211422104317]

NIVETHA D [211422104328]

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

OCT 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“AI POWERED PLACEMENT PREPARATION PLATFORM”** is the bonafide work of **NILANI S (211422104317) & NIVETHA D (211422104328)** who carried out the project work under my supervision.

SIGNATURE
Dr.L.JABASHEELA ,M.E.,Ph.D .,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE ,
COLLEGE,NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE
Mrs.S.LINCY JEMINA,M.E.,Ph.D .
ASSISTANT PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kindwords and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI , Dr. C . SAKTHIKUMAR , M.E. , Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department , **Dr.L.JABASHEELA ,M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs.S.LINCI JEMINA, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project

NILANI S (211422104317)

NIVETHA D (211422104328)

ABSTRACT

This project presents an AI-powered placement preparation platform designed to address key challenges in traditional quiz-based systems, such as lack of personalization, vulnerability to cheating, and administrative inefficiencies. The platform dynamically generates unique quizzes for each student from a large question pool, ensuring a tailored experience and significantly reducing the likelihood of cheating. An AI chatbot is integrated to provide real-time guidance, offering instant feedback and answering queries during quizzes, enhancing the learning experience. To further secure the quiz process, advanced anti-cheating mechanisms, such as tab-switching prevention, are implemented to maintain academic integrity. Faculty members benefit from streamlined quiz creation, assignment, and management features, allowing them to easily administer quizzes to students, while administrators can monitor performance and ensure compliance with examination protocols. Developed using the MERN stack and incorporating artificial intelligence, this platform provides a robust, secure, and adaptive environment for placement preparation, empowering students with personalized learning opportunities and enabling faculty to deliver efficient and effective assessments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The project is centered around an AI-powered platform designed to revolutionize the placement preparation process by providing dynamic, personalized quizzes. In the competitive academic landscape, students require tools that go beyond traditional assessments, offering tailored and effective preparation methods. This platform addresses these needs by generating unique quizzes for each user from a vast question pool, ensuring a fair and customized learning experience. It not only evaluates the students' knowledge but also engages them with content that adapts to their individual progress, creating a more focused preparation environment.

To maintain quiz integrity and prevent cheating, the system incorporates advanced anti-cheating mechanisms, including restrictions on tab switching and real-time monitoring. Additionally, the platform integrates an AI-powered chatbot that provides real-time support to students during quizzes. This chatbot assists with any queries, delivers immediate feedback, and enhances the overall quiz-taking experience. The system’s dynamic nature ensures that every quiz is unique, reducing the chances of sharing answers among users and providing a secure environment for effective learning.

Faculty members benefit from streamlined quiz management features, allowing them to create, assign, and monitor quizzes easily. The administrator plays a crucial role in overseeing system activities and student performance. Built using the MERN stack, the platform is scalable and secure, offering a seamless interface for users. Ultimately, the platform's goal is to provide an adaptive, secure, and interactive solution that prepares students for placement exams in an engaging and efficient manner.

1.1 PROBLEM DEFINITION

- In the competitive world of placement preparation, students face numerous challenges, including access to personalized and secure learning platforms. Many traditional quiz systems lack dynamic content generation, making it easier for students to share answers or cheat. Furthermore, these systems often fail to provide real-time assistance or feedback during quizzes, leaving students without the support they need in high-pressure situations. As students increasingly rely on digital tools for their exam preparation, the absence of a secure, adaptive, and interactive platform limits their ability to prepare effectively for placements.

- Given that most students have access to smartphones, there is a strong need for a placement preparation platform that not only offers personalized quizzes but also ensures security and integrity throughout the learning process. The platform must be easy to use, with intuitive features that adapt to individual student needs and learning progress.

- The objective of this project is to provide a fast, simple, and effective platform for quiz preparation that integrates AI-driven support, dynamic quiz generation, and advanced anti-cheating mechanisms, ensuring a secure and interactive learning environment for students preparing for placements.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Current placement preparation platforms lack dynamic content generation, leading to repetitive question sets that allow answer sharing. The absence of real-time support during quizzes leaves students without guidance, reducing preparation effectiveness. Additionally, weak security measures fail to prevent cheating, such as tab-switching, and the manual setup process for faculty is time-consuming, making it difficult to manage quizzes and track student progress efficiently.

DISADVANTAGES

- Quizzes are not dynamically generated, leading to answer sharing, and there is no real-time assistance or feedback during quizzes.
- Weak anti-cheating mechanisms and a cumbersome manual setup process for faculty hinder the overall system effectiveness.

2.2 PROPOSED SYSTEM

The purpose of this project is to develop a comprehensive AI-powered placement preparation platform that performs the following functions:

- Dynamically generate personalized quizzes from a large pool of questions, ensuring each student receives a unique quiz.
- Provide real-time feedback and assistance during quizzes through an integrated AI chatbot.
- Restrict tab-switching and external resource access to prevent cheating during quizzes.
- Allow faculty to efficiently manage quizzes and track student performance .

ADVANTAGES

- Easy to use with minimal setup, allowing seamless quiz generation and management.
- Ensures secure, personalized quiz-taking with real-time AI support and anti-cheating mechanisms in place.

2.3 DEVELOPMENT ENVIROMENTSOFTWARE REQUIREMENT

- **Operating System**: Windows, MacOS, or Linux (any modern version)
- **Node.js**: Version 14.0 or Higher
- **MongoDB**: Version 4.4 or Higher
- **React.js**: Version 17.0 or Higher
- **Express.js**: Version 4.0 or Higher
- **JavaScript (**ES6+**)**
- **Visual Studio Code (**any modern IDE**)**

HARDWARE REQUIREMENT

- **Computer**: Minimum 8GB RAM and 500GB internal storage for development and testing.
- **Server**: Dedicated server or cloud service (like AWS or Heroku) for hosting.
- **Android Device**: Minimum 2GB RAM and 16GB internal storage for mobile testing.
- **Device with Camera and GPS**: Smartphone with camera and GPS module for location tracking.

CHAPTER 3

SYSTEM DESIGN

3.1 UML DIAGRAMS

3.1.1 Use case diagram:

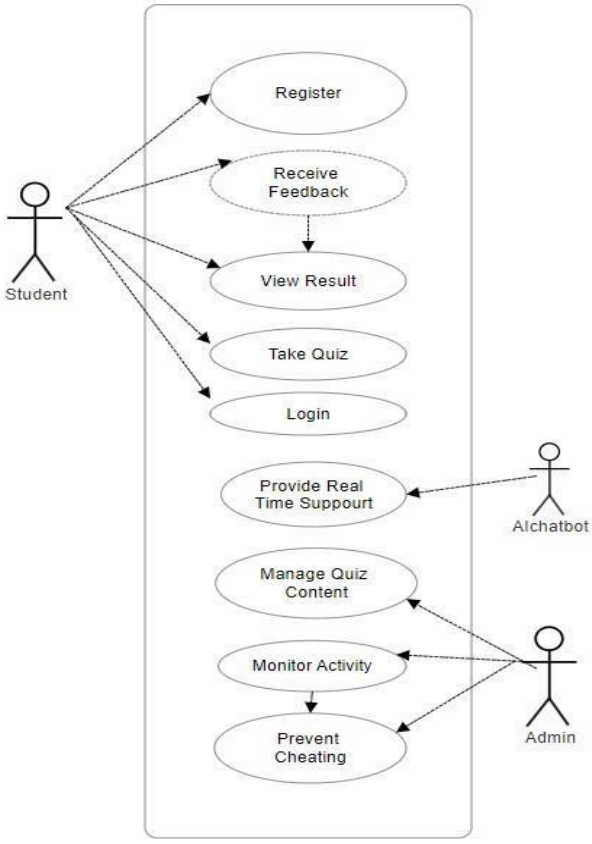


Fig 3.1.1 Use case diagram

This use case diagram illustrates the interactions between the system and users, detailing their activities and corresponding use cases within the placement preparation platform.

3.1.2 Class diagram:

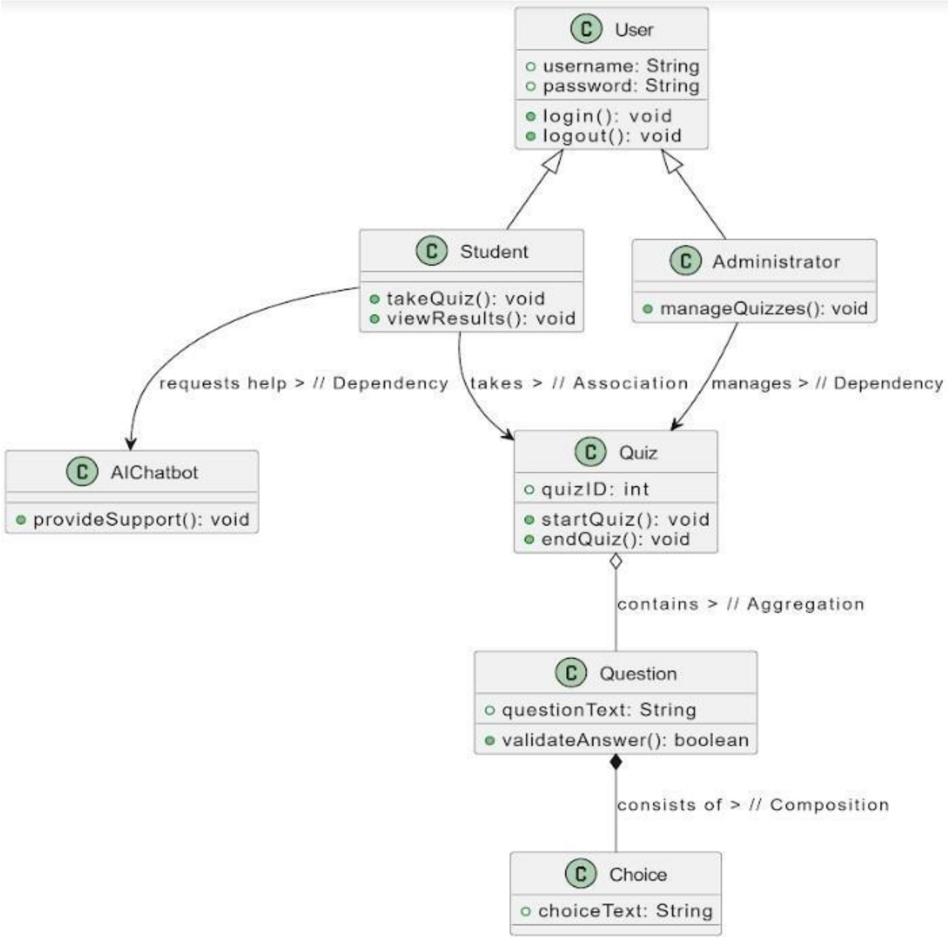


Fig 3.1.2 Class diagram

The class diagram illustrates the relationships and interactions between different classes within the system, providing a structured representation of the various components and their functionalities.

3.1.3 Sequence diagram:

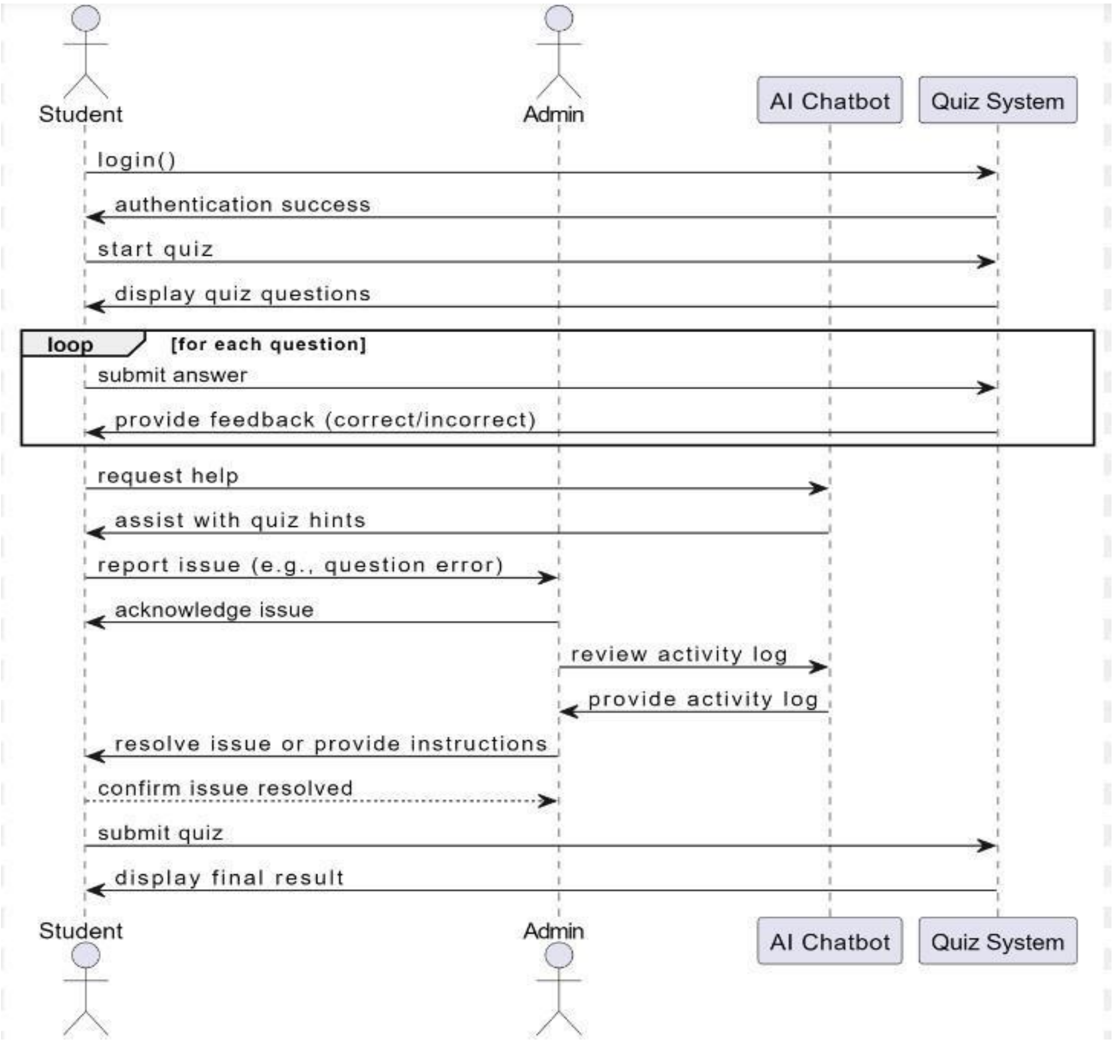


Fig3.1.3 Sequence dig

The sequence diagram of the placement preparation platform illustrates the sequence of activities performed by users while interacting with the application, detailing the flow of operations and the timing of events during their usage..

3.1.4 State chart diagram:

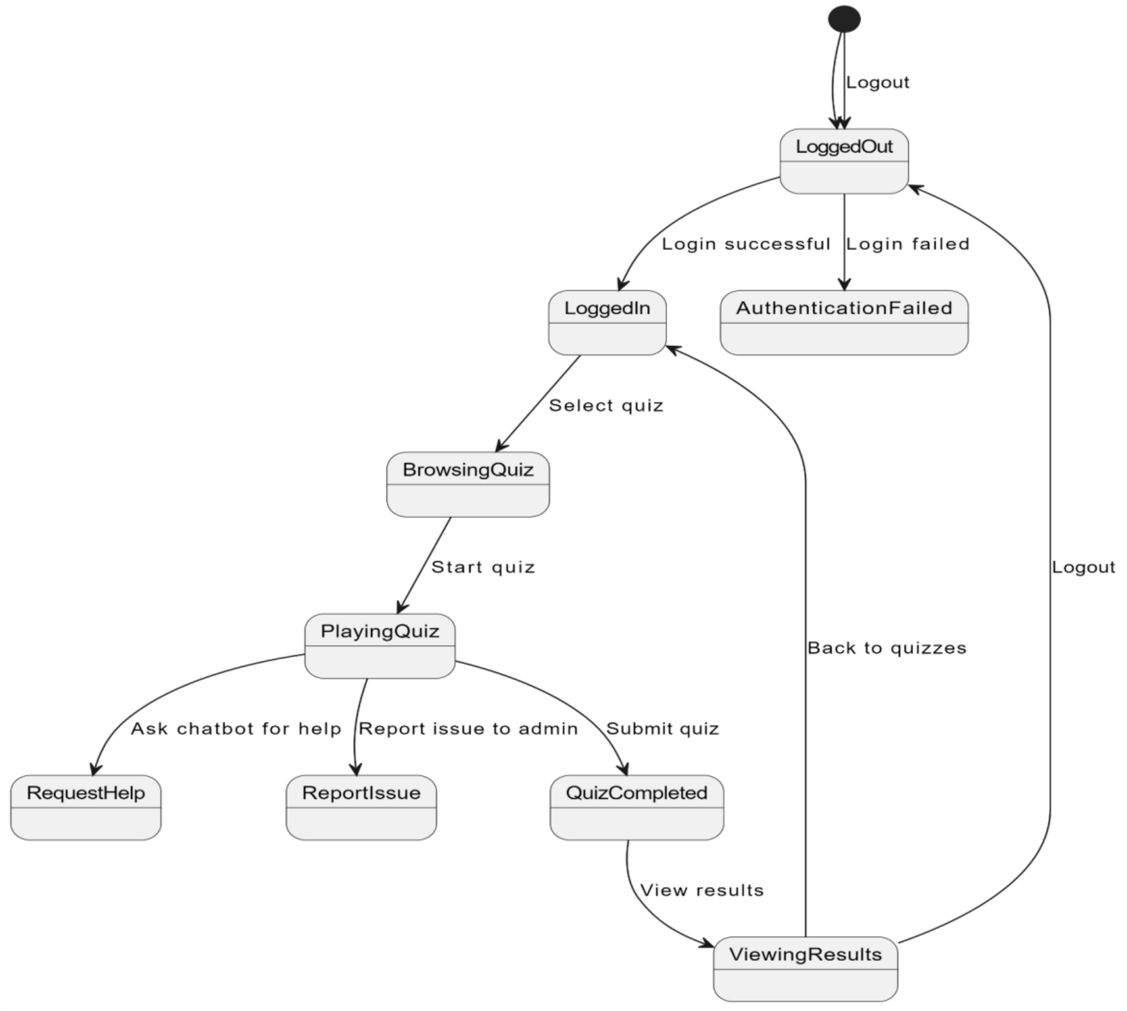


Fig 3.1.4 State chart diagram

The state chart diagram of the placement preparation platform depicts the entire workflow of the application, showcasing the various states from the initial installation stage through to user interactions and system responses.

3.1.5 Activity diagram:

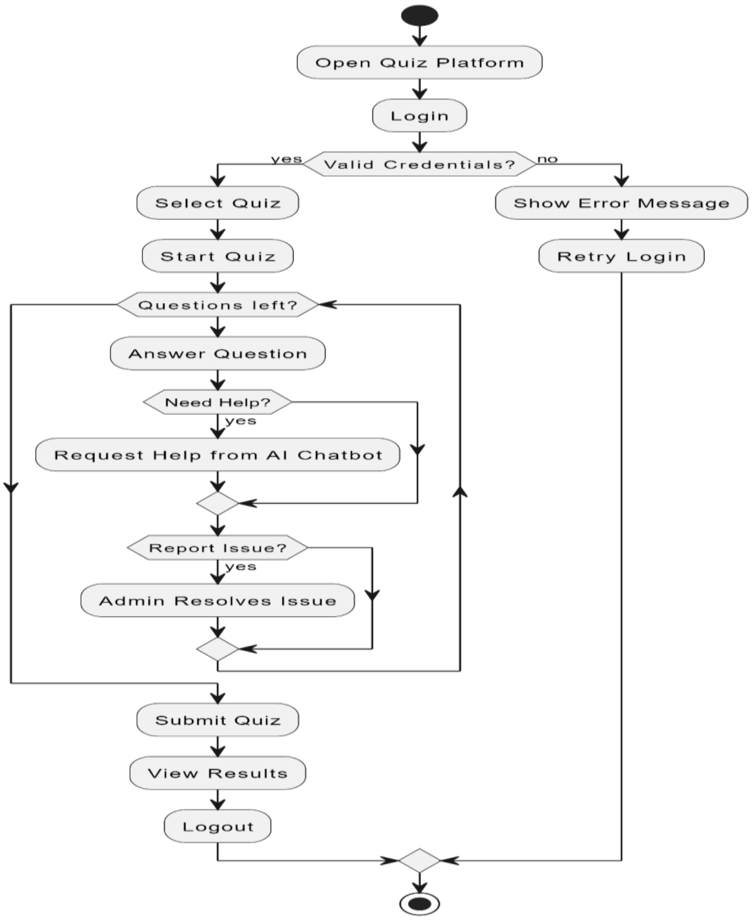


Fig 3.1.5 Activity diagram

The The activity diagram of the placement preparation platform illustrates the flow of activities involved in using the application, detailing the sequence of actions taken by users during their interaction with the system.

3.2 DATA DICTIONARY

The data dictionary is a central repository of metadata that provides detailed information about the data used within the placement preparation platform. It defines each data element encountered during the analysis and design of the system, ensuring clarity in the development and implementation phases. The rules for constructing data dictionary entries are as follows:

1. **Clear Definitions**: Data elements should be defined by their function and purpose, rather than the variable names used in the code.
2. **Unique Entries**: Each data term must be unique, avoiding duplication or ambiguity in definitions
3. **Simplified Terms**: Self-explanatory terms should not be decomposed further unless needed for deeper clarity or explanation.

In this project, the data dictionary includes key details such as:

- **User Data**: Information about users including their name, email, and role (student, admin).
- **Quiz Data**: Information about quiz content, including question banks, quiz IDs, and associated difficulty levels.
- **Performance Data**: Records of students' quiz results, scores, and completion time.
- **Security Data**: Passwords, access controls, and system logs to ensure the integrity of the quiz-taking process.

.

3.2.1 ADD USER TABLE

COLUMN NAME	DATA TYPE	SIZE	DESCRIPTION	CONSTRAINT
USERNAME	VARCHAR	20	Name of the user	NOT NULL
EMAIL	VARCHAR	50	Email address of the user	NOT NULL
PASSWORD	VARCHAR	10	Password for the account	NOT NULL
USER ROLE	VARCHAR	10	Role (Student/Admin)	NOT NULL

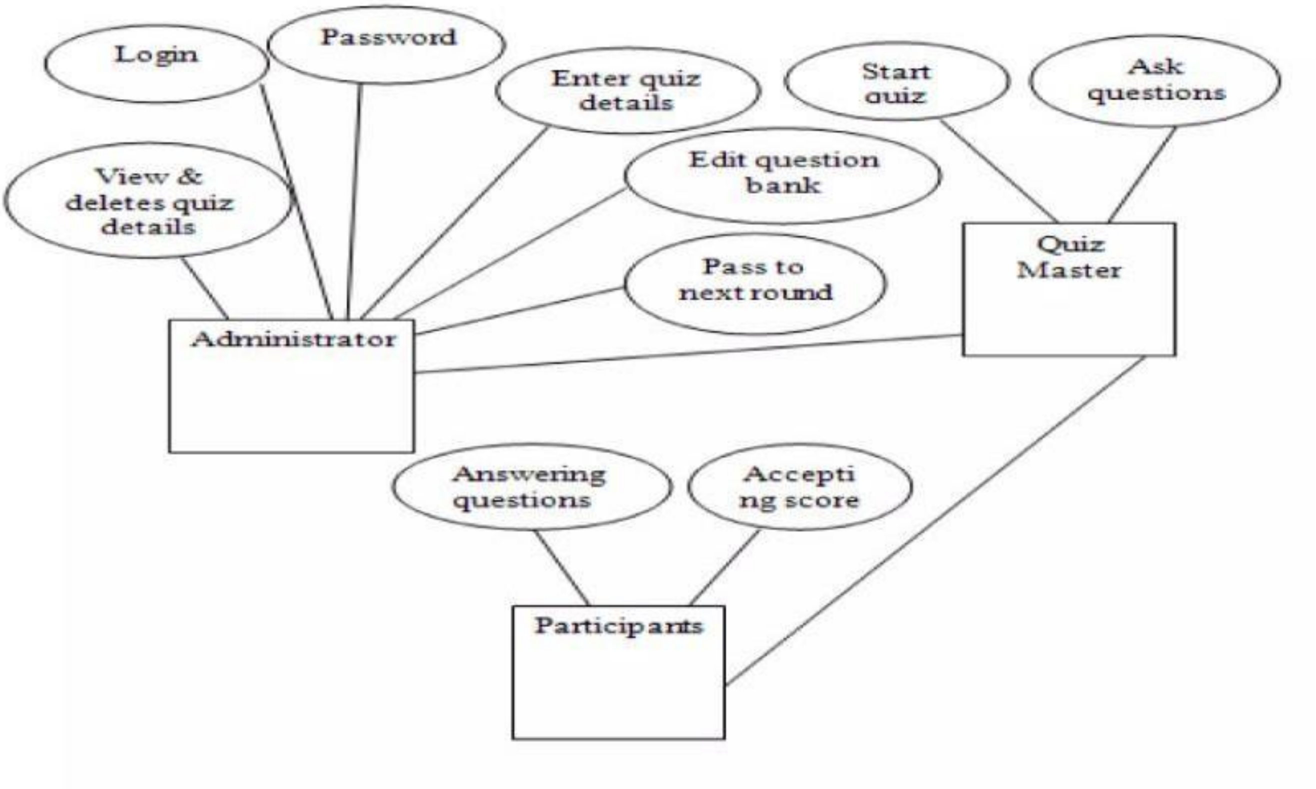
3.2.1 Add user table

3.2.2 DELETE CONTACTS TABLE

COLUMN NAME	DATA TYPE	SIZE	DESCRIPTION	CONSTRAINT
USERNAME	VARCHAR	20	Name of the user	NOT NULL
EMAIL	VARCHAR	50	Email address of the user	NOT NULL

3.2.2 Delete contacts table

3.3 ER DIAGRAM



The E-R diagram for this project represents key entities such as **User**, **Quiz**, and **Administrator**, along with their attributes like **quiz questions**, **answers**, and **user progress**. It showcases the relationships between these entities to manage secure and personalized quiz experiences.

3.4 DATAFLOW DIAGRAM3.4.1 0 LEVEL DFD

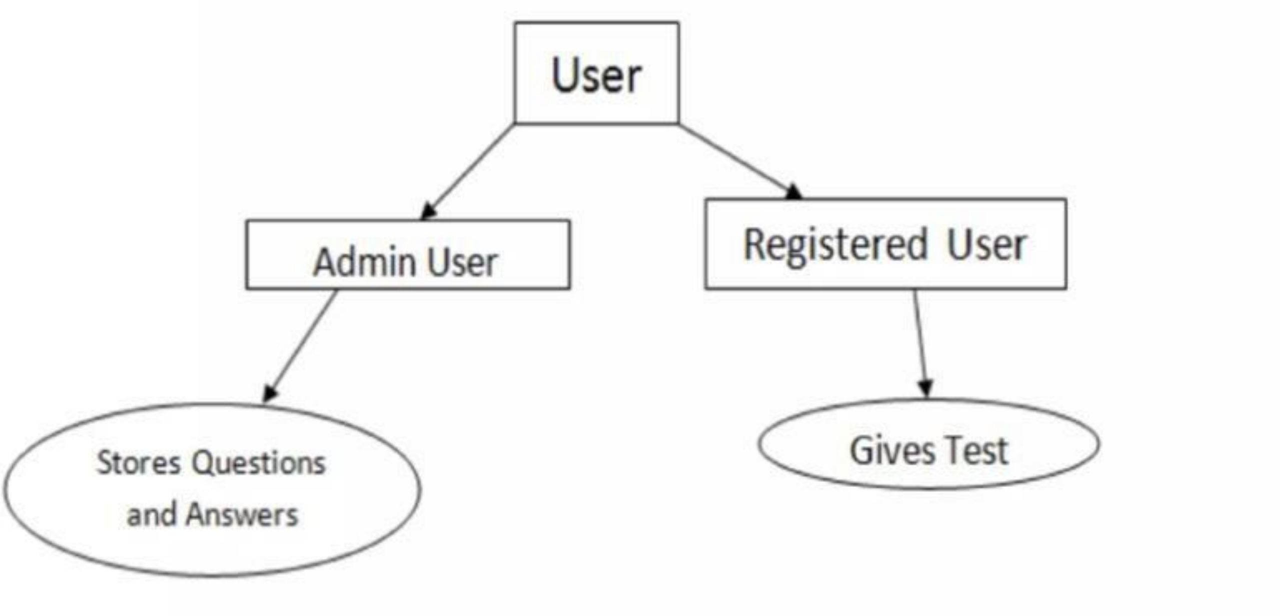


Fig 3.4.1 Data flow diagram level 0

The zero level of the data flow diagram for this project outlines the core management levels of the system, illustrating how data flows between users, quizzes, and the administrator. It provides a high-level overview of the interactions within the secure quiz platform.

3.4.2 FIRST LEVEL DFD

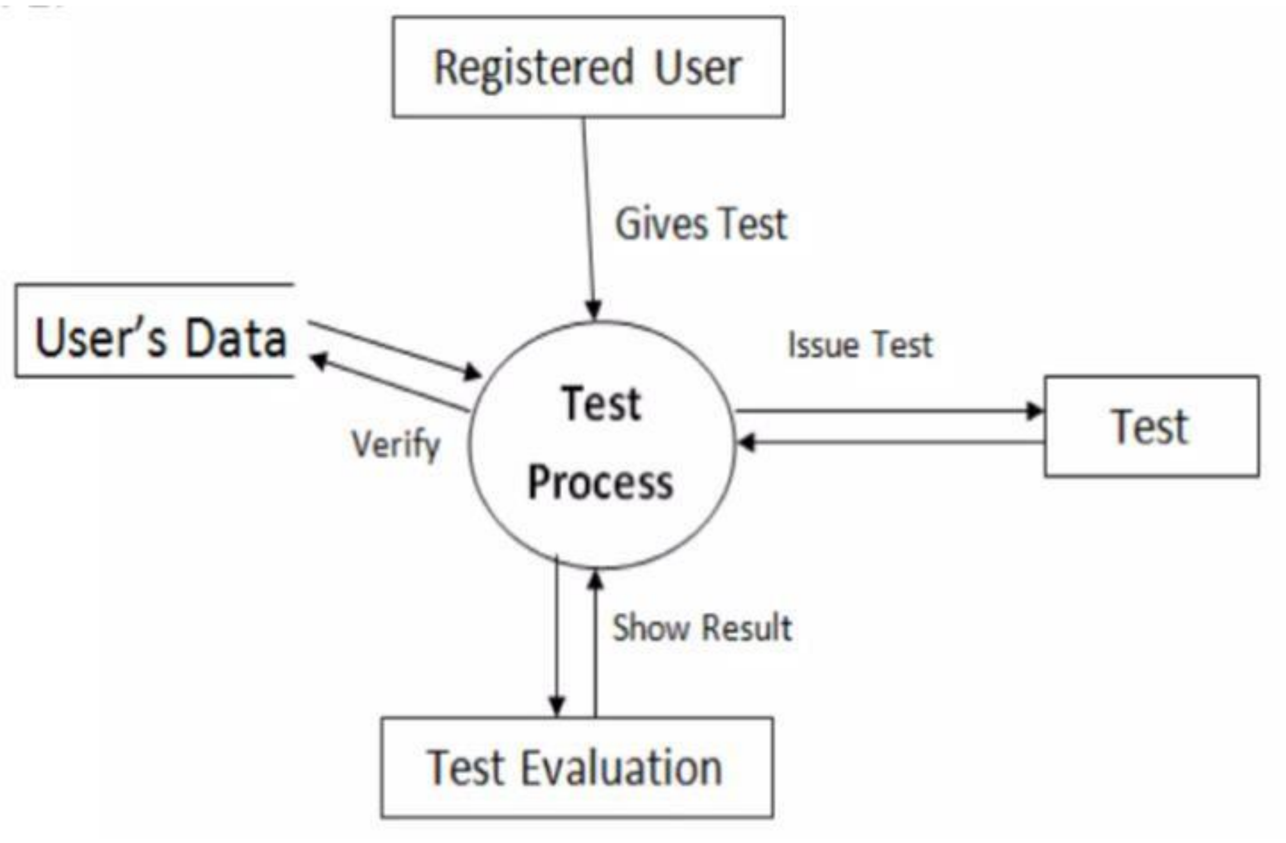


Fig 3.4.2 Dataflow diagram level 1

The first level of the data flow diagram for this project illustrates the different management levels and their corresponding interactions, detailing how reports and data are exchanged between users, the quiz system, and administrators. This level provides a deeper understanding of the operational flow within the platform.

3.4.3 SECOND LEVEL DFD

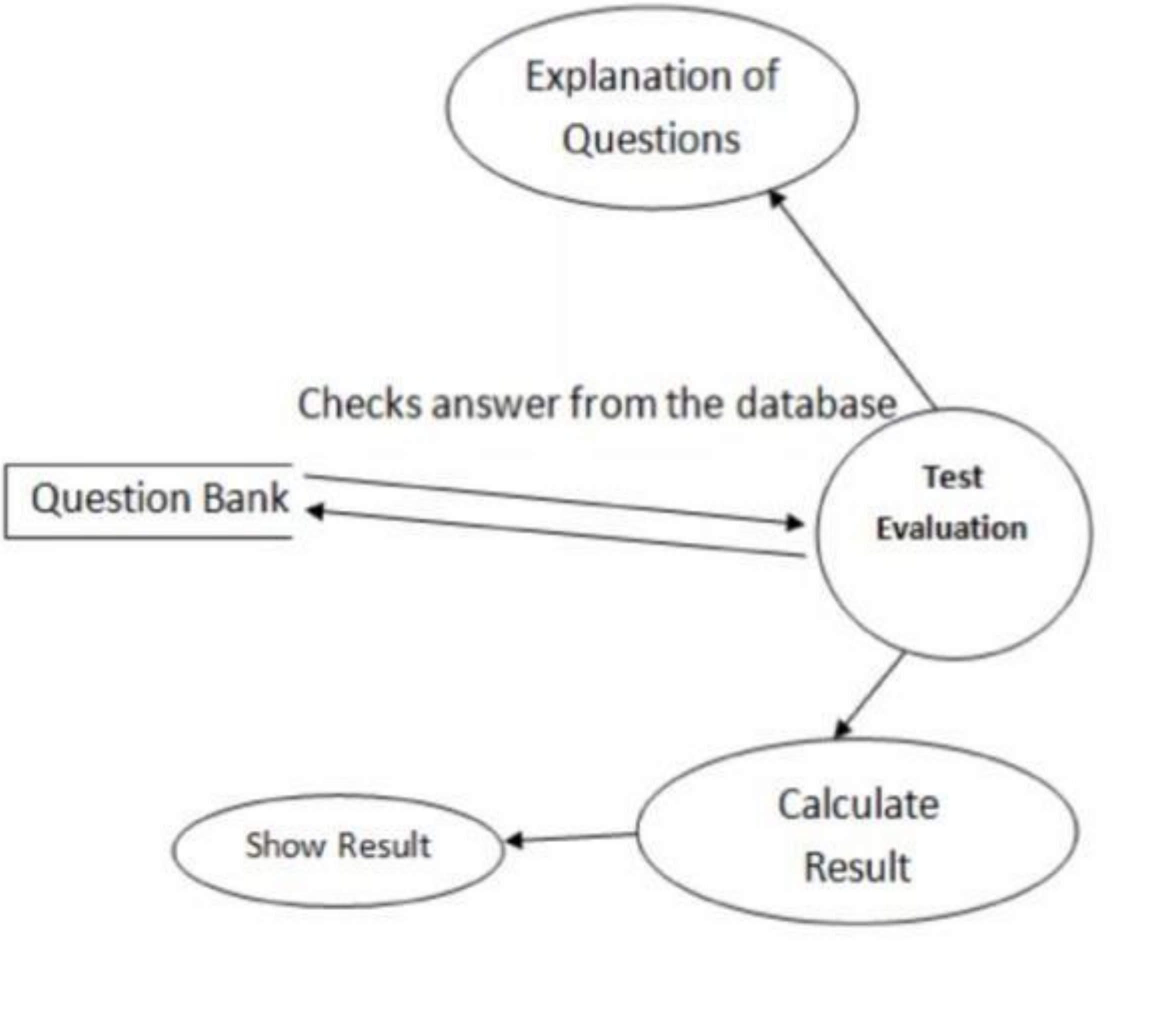


Fig 3.4.3 Dataflow diagram level 2

The second level of data flow diagram of Placement Preparation Platform shows the various details of actions.

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 ARCHITECTURE OVERVIEW

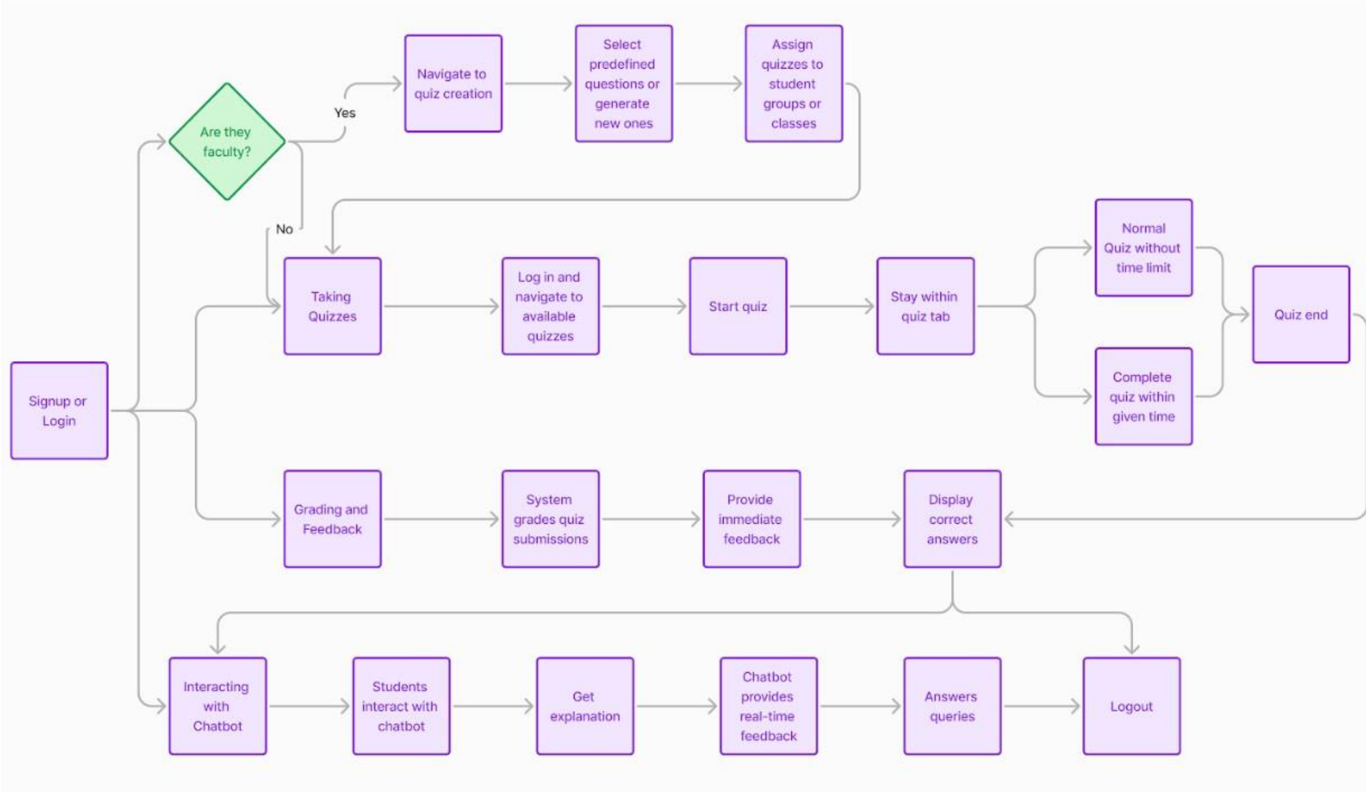

Navigate to quiz creation
Select predefined questions or generate new ones
Assign quizzes to student groups or classes
Are they faculty?
Yes
No
Taking Quizzes
Log in and navigate to available quizzes
Start quiz
Stay within quiz tab
Normal Quiz without time limit
Complete quiz within given time
Quiz end
Signup or Login
Grading and Feedback
System grades quiz submissions
Provide immediate feedback
Display correct answers
Interacting with Chatbot
Students interact with chatbot
Get explanation
Chatbot provides real-time feedback
Answers queries
Logout

Fig 4.1 Architecture diagram

The architecture diagram (Figure 4.1) illustrates the comprehensive structure of the AI- based placement preparation platform. Users begin by accessing the web application, where they are prompted to create an account and log in. Upon successful authentication, the platform utilizes an AI engine to analyze user profiles and generate personalized quiz content based on their strengths and weaknesses.Additionally, real-time feedback is provided through the user interface, allowing students to review their answers and receive guidance. The platform incorporates advanced security measures to prevent cheating, including monitoring for tab-switching and restricting access to external resources during quizzes. Overall, the architecture facilitates an interactive and efficient learning experience tailored to the needs of each user.

4.2 MODULE DESCRIPTION

The AI-based placement preparation platform consists of six main modules. They are:

- User Authentication Module
- Quiz Management Module
- AI-Driven Content Generation Module
- Real-Time Feedback Module
- Performance Analytics Module
- Admin Management Module

User Authentication Module:

- Allows users to create accounts and log in securely.
- Users can reset their passwords if forgotten.
- Supports authentication via email verification for added security.

Quiz Management Module:

- Facilitates the creation and customization of quizzes by faculty.
- Enables dynamic question generation from a pool of questions.
- Allows users to select quizzes based on topics and difficulty levels.
- Provides options for timed and untimed quizzes.

AI-Driven Content Generation Module:

- Analyzes user profiles and performance to tailor quiz content.

- Generates personalized quizzes that adapt to individual learning needs.
- Utilizes machine learning algorithms to improve question selection over time.

Real Time Feedback Module:

- Offers immediate feedback on quiz performance after completion.
- Provides detailed explanations for correct and incorrect answers.
- Suggests additional resources based on user performance.
- Allows users to review their answers and track progress.

Performance Analytics Module

- Displays user performance metrics, including scores and improvement over time.
- Generates reports on quiz attempts, strengths, and weaknesses.
- Provides visual analytics to help users understand their learning patterns.
- Enables users to set goals and track their progress towards placement readiness.

Admin Management Module:

- Allows administrators to manage user accounts and access levels.
- Provides tools for overseeing quiz submissions and tracking user engagement.
- Facilitates communication with users through announcements and updates.
- Supports data backup and security measures to protect user information.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 STAFF QUIZZ.js:

```
const mongoose = require('mongoose');

const questionSchema = new mongoose.Schema({
questionText: { type: String, required: true },
options: [{ type: String, required: true }], // Array of options
correctAnswer: { type: String, required: true } // Correct answer
});
const staffQuizSchema = new mongoose.Schema({
title: { type: String, required: true },
categoryName: { type: String, required: true }, // Only one category
topicName: { type: String, required: true }, // Only one topic
questions: [questionSchema], // Array of questions
createdBy: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true
}, //
Reference to the staff user
keyword: { type: String, required: true, unique: true }, // Unique keyword for
student
access
createdAt: { type: Date, default: Date.now }
});
const StaffQuiz = mongoose.model('StaffQuiz', staffQuizSchema);
module.exports = StaffQuiz;
```

APP.js :

```
// Initialize Express
const app = express();
// Middleware
app.use(cors());
app.use(express.json());
// Load environment variables
dotenv.config();
// MongoDB connection
mongoose.connect(process.env.MONGODB_URI, { useNewUrlParser: true,
useUnifiedTopology: true })
.then(() => console.log('MongoDB connected!'))
.catch(err => console.log('Error connecting to MongoDB:', err));
// Routes
app.use('/api/auth', authRoutes);
app.use('/api/quizzes', quizRoutes);
app.use('/api/quiz-results', quizResultsRoutes);
app.use('/api/staff-quizzes', staffQuizRoutes);
// Start server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(Server running on port ${PORT}));
```

QUIZRESULT.js

```
const mongoose = require('mongoose');
const quizResultSchema = new mongoose.Schema({
userId: {
type: mongoose.Schema.Types.ObjectId,
ref: 'User', // Reference to User model
required: true
},
score: {
type: Number,
required: true
},
 topicName: {
type: String, // Store the name of the quiz topic
required: true
},
totalQuestions: {
type: Number, // Total number of questions in the quiz
required: true
},
completedAt: {
type: Date,
default: Date.now,
required:true
}
});
const QuizResult = mongoose.model('QuizResult', quizResultSchema);
module.exports = QuizResult;
```

STUDENTDASHBOARD.js

```
    // src/components/StudentDashboard.js
  import React, { useState, useEffect } from 'react';
  import { fetchQuizzesByKeyword, getQuizResults } from '../services/quizService'; //
```

fetchQuizByKeyword is added for search functionality

```
 import { getUserDetails } from '../services/authService';
 import { Link, useNavigate } from 'react-router-dom';
 import '../styles/Dashboard.css';
const StudentDashboard = () => {
const [quizzes, setQuizzes] = useState([]);const
[results, setResults] = useState([]);
const { name: userName, role: userRole } = getUserDetails();const
[keyword, setKeyword] = useState(");
const [quizData, setQuizData] = useState(null);
const [errorMessage, setErrorMessage] = useState(");
const navigate = useNavigate(); // For navigating after successful search
// Handle search keyword input
const handleKeywordChange = (e) => setKeyword(e.target.value);

// Fetch the quiz by keyword
const handleSearchQuiz = async () => {try
{
if (keyword.trim() === ") { setErrorMessage('Please enter a
valid keyword');return;
}
const result = await fetchQuizzesByKeyword(keyword);
setQuizData(result); // Set quiz data if the quiz is found
setErrorMessage("); // Clear error message if successful
} catch (error) {
console.error('Error fetching quiz:', error); setErrorMessage('Invalid
keyword. Please try again.');setQuizData(null); // Clear quiz data if
the search fails
}
```

```
};
// Navigate to the start quiz page
const handleStartQuiz = () => {
if (quizData) {
navigate(/start-quiz/${quizData.keyword});
}
};
useEffect(() => {
// Fetch the student's quiz results const
fetchResults = async () => {
try {
const userId = localStorage.getItem('userId'); // Ensure userId is in localStorageconst
fetchedResults = await getQuizResults(userId); setResults(fetchedResults);
} catch (error) {
console.error('Error fetching quiz results:', error);
}
};
fetchResults();
}, []);

return (
<div    className="dashboard-container">
<div    className="dashboard-header">
 <h1>Welcome, {userName || 'Student'}!</h1>
<h2>Role: {userRole || 'Role not defined'}</h2>
</div>

{/* Search Bar for Quiz */}
<div className="search-quiz-container">
```

```
<h2>Search for a Quiz</h2>
<input
type="text"
placeholder="Enter quiz keyword"
value={keyword}
onChange={handleKeywordChange}
/>
<button onClick={handleSearchQuiz}>Search Quiz</button>
{errorMessage && <p className="error-message">{errorMessage}</p>}
</div>
{/* Display Quiz Data if Found */}
{quizData && (
<div className="quiz-data">
<h2>Quiz Found: {quizData.title}</h2>
<p>Category: {quizData.categoryName}</p>
<p>Topic: {quizData.topicName}</p>
<p>Keyword: {quizData.keyword}</p>
<button onClick={handleStartQuiz}>Start Quiz</button>
</div>
)}

{/* Quiz Results Section */}
<div   className="quiz-results">
<h2>Your Quiz Results</h2>
{results.length === 0 ? (
<p>No quiz results found.</p>
) : (
results.map(result => {
const quizDate = new Date(result.completedAt); // Create a Date objectfrom
const formattedDate = quizDate.toLocaleDateString('en-US'); // Format the date

return (
<div key={result._id} className="result-item">
<h3>Topic: {result.topicName}</h3>
 <p>Score: {result.score}/{result.totalQuestions}</p>
<p>Date: {formattedDate}</p>
</div>
);
})
)}
</div>
<div className="take-quiz-container">
<Link to="/take-quiz" className="take-quiz-button">
<button className="btn">Take Quiz</button>
</Link>
</div>
</div>
); };
export default StudentDashboard;
```

CHAPTER 6

SYSTEM TESTING

6.1 TEST CASES & REPORTS

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1	Selecting "Login" button on the homepage	Redirect to the user authentication page	Redirected to the user authentication page	Pass
2	Entering valid credentials and clicking "Submit"	Successful login and redirect to the dashboard	Successful login and redirected to the dashboard	Pass
3	Selecting "Take Quiz" option from the dashboard	Display available quizzes to attempt	Displayed available quizzes to attempt	Pass
4	Clicking on a specific quiz	Redirect to the quiz page with questions	Redirected to the quiz page with questions	Pass
5	Completing the quiz a submitting	Display final score and performance feedback	Displayed final score and performance feedback	Pass
6	Clicking on "View Per Analytics"	Show user performance metrics	Showed user performance metrics	Pass

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
7	Selecting "AI-driven Q option	Display personalized quiz based on user profile	Displayed personalized quiz based on user profile	Pass
8	Using "Get Help" featu a quiz	Provide real-time assistance for challenging questions	Provided real-time assistance for challenging questions	Pass
9	Clicking "Logout" button	Successful logout and redirect to the homepage	Successful logout and redirect to the homepage	Pass
10	Attempting to access t logging in	Redirect to the login page	Redirect to the login page	Pass
11	Checking "Admin Das functionality	Admin can manage users and quizzes	Admin can manage users and quizzes	Pass
12	Attempting to share an during a quiz	Display warning message and block sharing	Display warning message and block sharing	Pass

CHAPTER 7

7.1 CONCLUSION

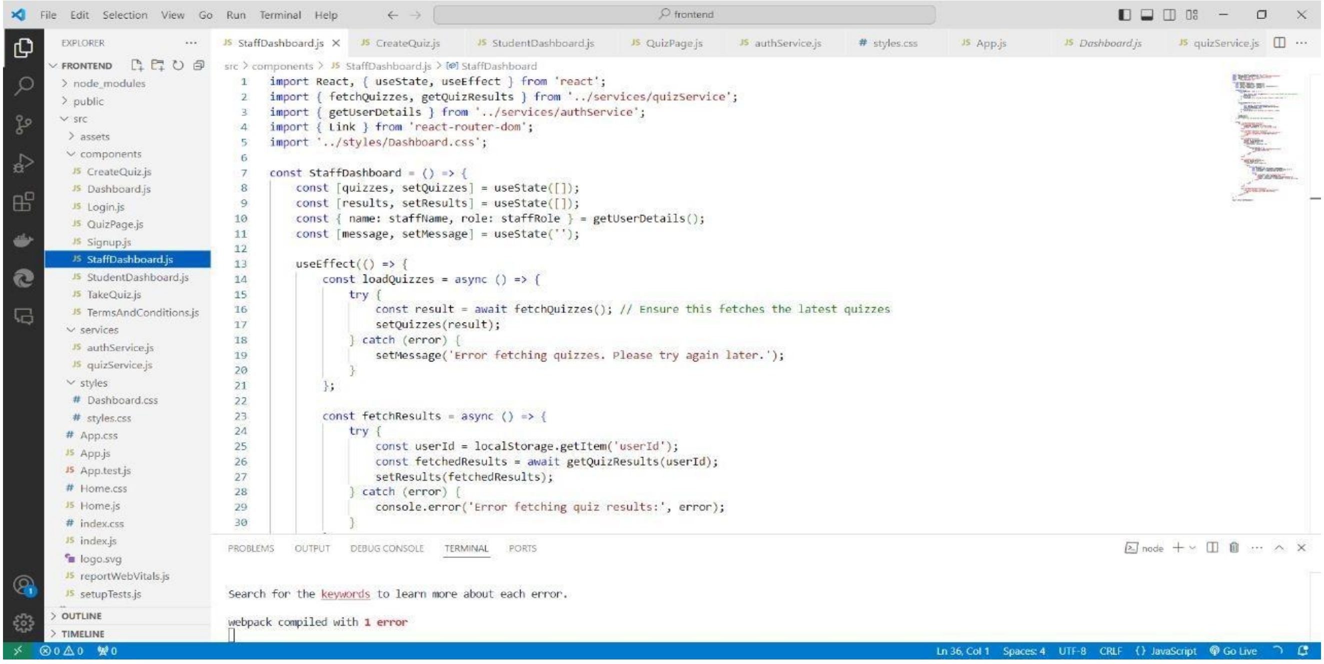
The AI-based placement preparation platform aims to create an inclusive and efficient learning environment for all students, enhancing their chances of success in competitive exams. By offering personalized quizzes, real-time assistance, and advanced security measures against cheating, the system empowers students to prepare effectively without the risk of academic dishonesty. The integration of dynamic content generation and feedback mechanisms ensures that students receive a tailored experience, addressing their unique learning needs and promoting academic integrity.

7.2 FUTURE ENHANCEMENTS

1. **Offline Functionality**: Future versions could enable offline access to quizzes and materials, allowing students to continue their preparation even without an internet connection.
2. **Cloud Storage**: Integrating cloud storage solutions would help in managing user data securely, ensuring that students can access their progress and resources from multiple devices.
3. **Voice-Activated Features**: Implementing voice recognition technology could enhance user experience by allowing hands-free navigation of the platform, making it more accessible for users on the go.

These enhancements would significantly improve the platform's usability and effectiveness, further supporting students in their placement preparation journey.

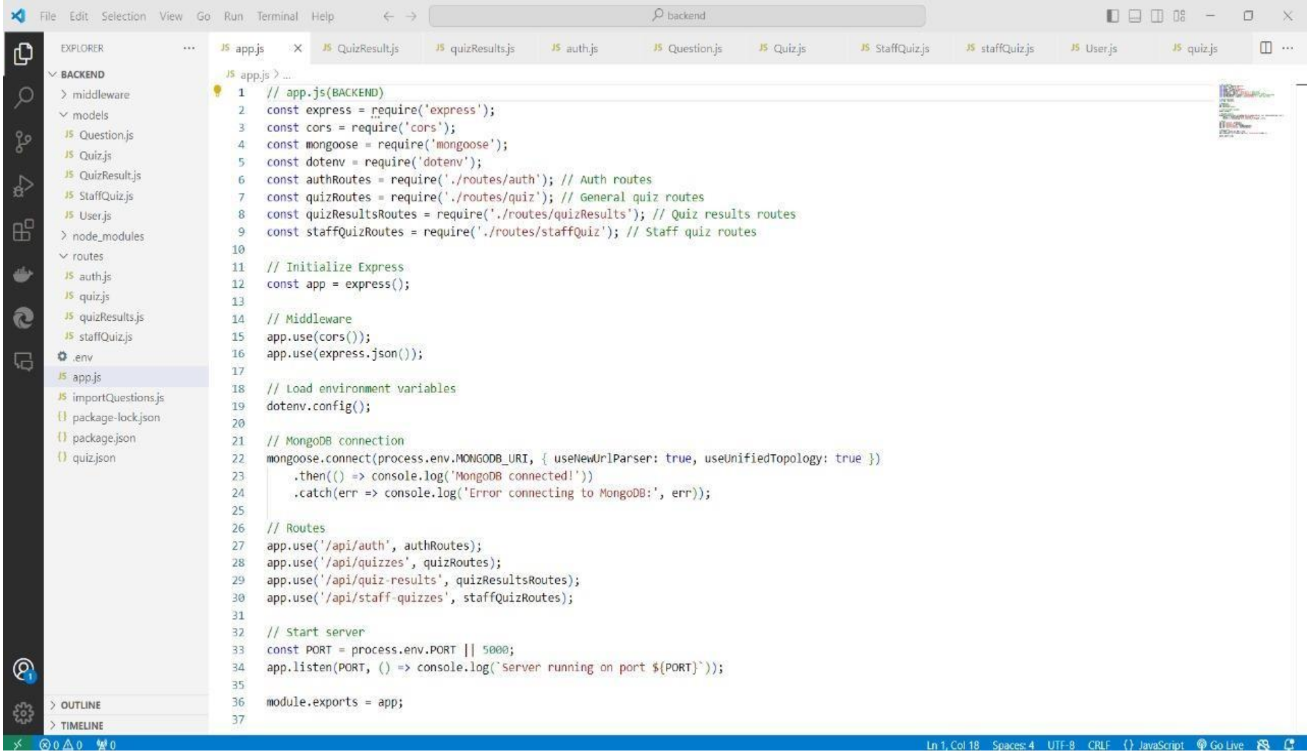
CHAPTER 8
APPENDICES



```
import React, { useState, useEffect } from 'react';
import { fetchQuizzes, getQuizResults } from '../services/quizService';
import { getUserDetails } from '../services/authService';
import { Link } from 'react-router-dom';
import '../styles/Dashboard.css';

const StaffDashboard = () => {
    const [quizzes, setQuizzes] = useState([]);
    const [results, setResults] = useState([]);
    const { name: staffName, role: staffRole } = getUserDetails();
    const [message, setMessage] = useState('');

    useEffect(() => {
        const loadQuizzes = async () => {
            try {
                const result = await fetchQuizzes(); // Ensure this fetches the latest quizzes
                setQuizzes(result);
            } catch (error) {
                setMessage('Error fetching quizzes. Please try again later.');
            }
        };

        const fetchResults = async () => {
            try {
                const userId = localStorage.getItem('userId');
                const fetchedResults = await getQuizResults(userId);
                setResults(fetchedResults);
            } catch (error) {
                console.error('Error fetching quiz results:', error);
            }
```

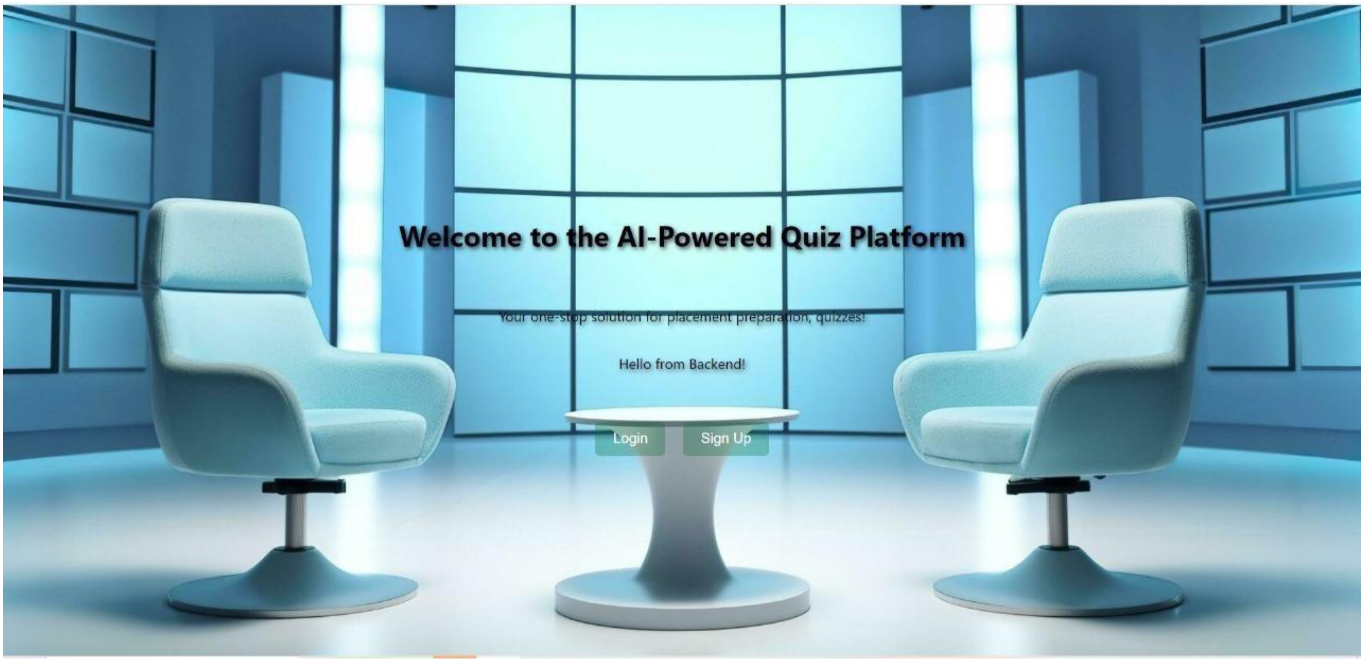
```
Search for the keywords to learn more about each error.

webpack compiled with 1 error
```

Fig 8.1 FRONTEND (VSC)



```
// app.js(BACKEND)
const express = require('express');
const cors = require('cors');
const mongoose = require('mongoose');
const dotenv = require('dotenv');
const authRoutes = require('./routes/auth'); // Auth routes
const quizRoutes = require('./routes/quiz'); // General quiz routes
const quizResultsRoutes = require('./routes/quizResults'); // Quiz results routes
const staffQuizRoutes = require('./routes/staffQuiz'); // Staff quiz routes

// Initialize Express
const app = express();

// Middleware
app.use(cors());
app.use(express.json());

// Load environment variables
dotenv.config();

// MongoDB connection
mongoose.connect(process.env.MONGODB_URI, { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => console.log('MongoDB connected!'))
    .catch(err => console.log('Error connecting to MongoDB:', err));

// Routes
app.use('/api/auth', authRoutes);
app.use('/api/quizzes', quizRoutes);
app.use('/api/quiz-results', quizResultsRoutes);
app.use('/api/staff-quizzes', staffQuizRoutes);

// Start server
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

module.exports = app;
```

Fig 8.2 BACKEND (VSC)

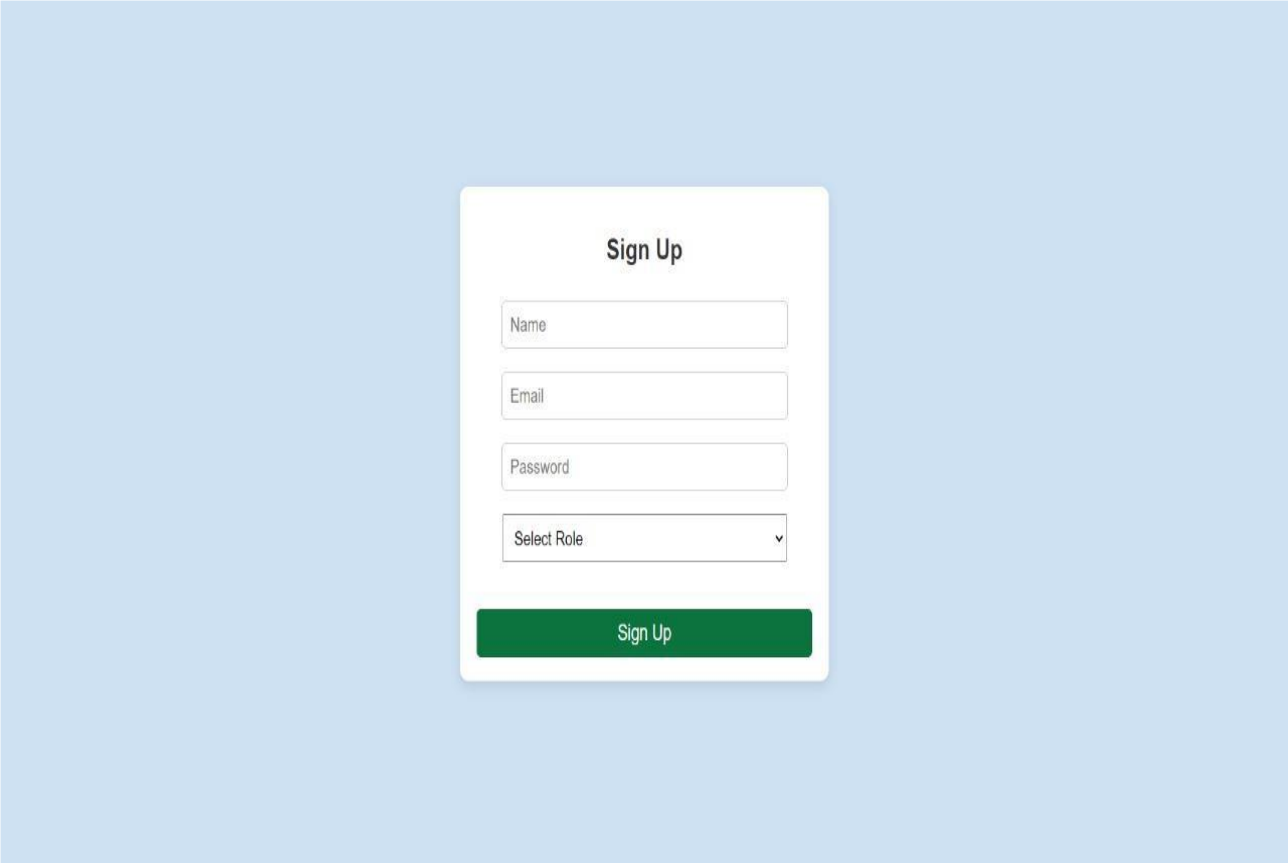


Welcome to the AI-Powered Quiz Platform

Your one-stop solution for placement preparation, quizzes!

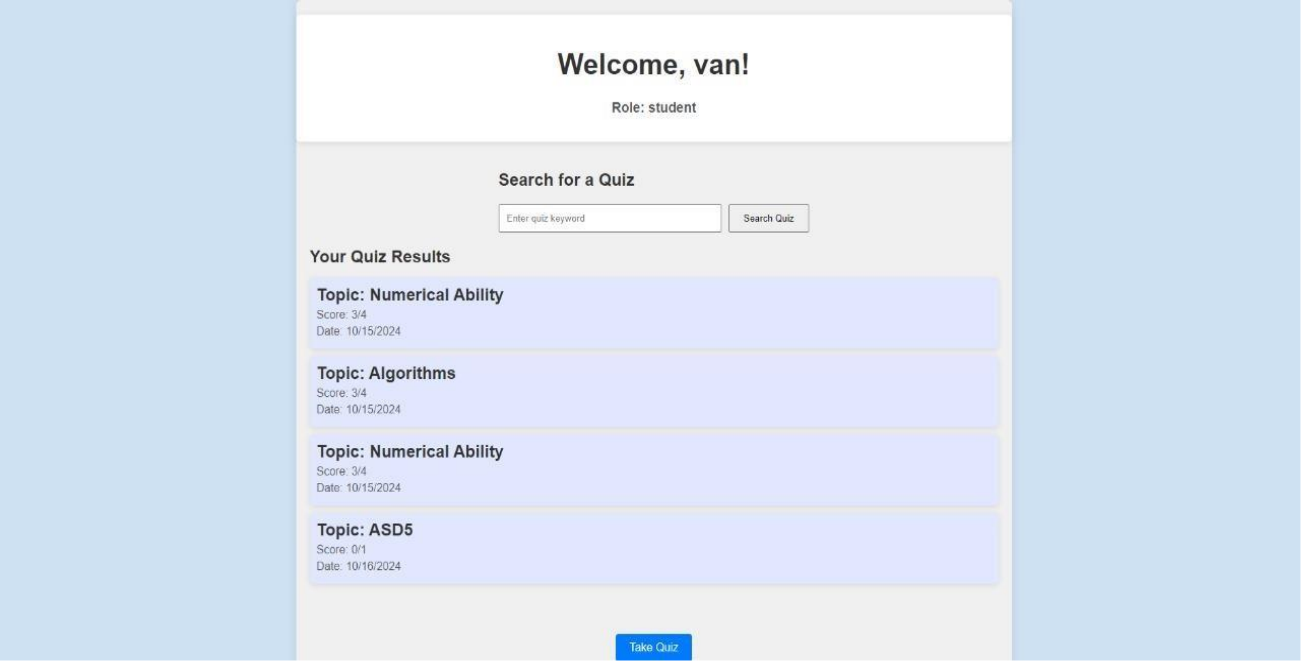
Hello from Backend!

Fig 8.3 HOME PAGE



Sign Up

Fig 8.4 SIGN UP PAGE



Welcome, van!

Role: student

Search for a Quiz

Your Quiz Results

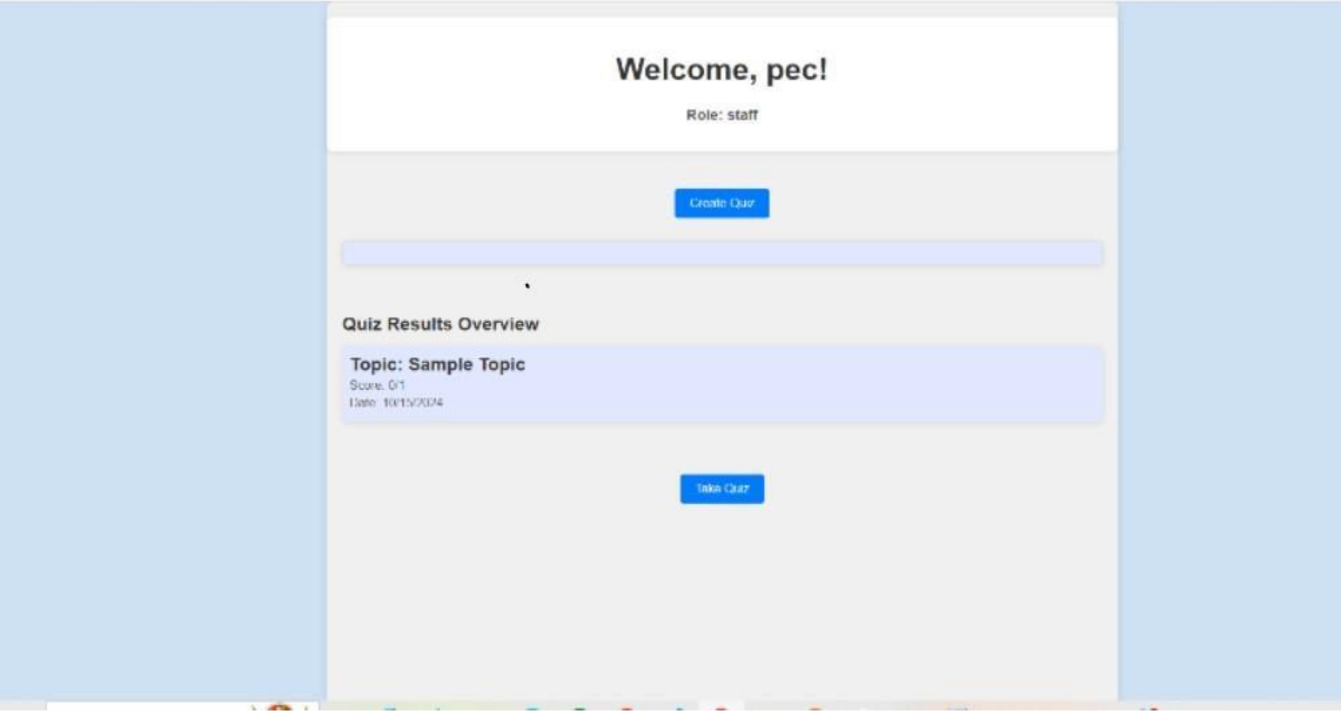
Topic: Numerical Ability
Score: 3/4
Date: 10/15/2024

Topic: Algorithms
Score: 3/4
Date: 10/15/2024

Topic: Numerical Ability
Score: 3/4
Date: 10/15/2024

Topic: ASD5
Score: 0/1
Date: 10/16/2024

FIG 8.5 STUDENT DASHBOARD



Welcome, pec!

Role: staff

Quiz Results Overview

Topic: Sample Topic
Score: 0/1
Date: 10/15/2024

FIG 8.6 STAFF DASHBOARD

CHAPTER 9

REFERENCES

- **Gupta, S., & Sharma, R.** (2021). *AI-Based Placement Preparation System:A Comprehensive Study*. International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 9(2), 34-40.
- **Kumar, A., & Singh, P.** (2020). *E-Learning Systems: AI in Education and Placement*. International Journal of Engineering and Advanced Technology (IJEAT), 9(1), 58-63.
- **Singh, R., & Kumar, S.** (2023). *Placement Preparation: A DynamicApproach with AI Integration*. Journal of Education and Learning Technologies, 11(3), 150-158.
- **Zhou, Y., & Zhang, L. (2022).** Intelligent Placement Preparation System Based on Machine Learning. *Journal of Educational Technology & Society*,25(4), 1-12.
- **Patel, D., & Desai, A. (2021).** Enhancing Student Engagement in OnlinePlacement Preparation through AI. *International Journal of Computer Applications*, 174(7), 17-22.
- **Mehta, A., & Joshi, P. (2022).** AI-Driven Personalized Learning for JobPlacement: A Review. *Journal of Intelligent Learning Systems and Applications*, 14(2), 67-75.
- **Srinivasan, S., & Kumari, S. (2020).** A Framework for AI-Based Assessment and Placement Solutions. *International Journal of AdvancedResearch in Computer Science*, 11(4), 34-39.
- **Ali, S., & Khurshid, H. (2023).** A Smart System for Placement PreparationUsing Artificial Intelligence Techniques. *Journal of Computer Science and Technology*, 38(1), 25-33.
- https://www.researchgate.net/publication/336115534_A_Project_on_Online_MCQ_Quiz_Application
- https://www.slideshare.net/harshverma164/minor-project-report-for-quiz- application