****

**SAVEETHA SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**"Developing a Secure and Automated Online Placement Examination System: Enhancing Exam Scheduling, Evaluation, and Result Management for Efficient Candidate Experience"**

**A CAPSTONE PROJECT REPORT**

**Submitted in the partial fulfillment for the completion of the course**

**CSA4311 INTERNET PROGRAMMING FOR DHTML**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**S. Chandana priya (192210091)**

**B. Lakshmi Neha (192211268)**

**Under the Supervision of**

**Dr. Senthil Vadivu**

**NOV 2024**

**DECLARATION**

We, **S. Chandana Priya, B. Lakshmi Neha** students of **Bachelor of Engineering in Computer Science Engineering**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work is **Developing a Java-Based Chess Engine with Enhanced Tactical Analysis for Improved Player Decision-Making"** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

S. Chandana priya (192210091)

B. Lakshmi Neha (192211268)

Date:25-11-2024

Place: Chennai

**CERTIFICATE**

This is to certify that the project entitled **"Developing a Secure and Automated Online Placement Examination System: Enhancing Exam Scheduling, Evaluation, and Result Management for Efficient Candidate Experience"** submitted by **S. Chandana Priya, B. Lakshmi Neha** has been carried out under our supervision. The project has been submitted as per the requirements in the current semester of B.Sc. Computer Science Engineering.

**Supervisor**

**Dr. Senthilvadivu**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **PAGE NO.** |
| 1 | **Abstract** |  |
| 2 | **Introduction** |  |
| 3 | **Project Description**  About your project |  |
| 4 | **Problem Description**  Program to build a simple Software for < > |  |
| 5 | **Tool Description**  User interface  Features |  |
| 6 | **Operations**  Store the First name of the student.  Store the Last name of the student.  Store the unique Roll number for every student.  Store the CGPA of every student.  Store the courses registered by the student. |  |
| 7 | **Approach / Module Description / Functionalities** |  |
| 8 | **Implementation**  Coding |  |
| 9 | **Result**  Output with Screenshots |  |
| 10 | **Conclusion**    **References** |  |

**ABSTRACT**

The goal of this project is to provide a secure and automated online placement test system that addresses scheduling, evaluation, and result management difficulties. The technology improves the applicant experience by providing a smooth login, secure exam environments, automatic answer evaluation, and transparent result reporting. The technology protects data privacy and reduces administrative cost by utilising powerful encryption and authentication techniques, all while allowing for real-time scheduling and result customization. This project also includes a feature that allows students to request re-evaluations and download result scorecards, enhancing the entire user experience.

**INTRODUCTION**

The growing demand for online education has resulted in a greater requirement for secure and effective online exam systems. Traditional examination systems, which rely significantly on manual scheduling and result processing, suffer difficulties in maintaining efficiency, accuracy, and scalability. This project aims to provide an automated system that streamlines the examination process by improving scheduling, evaluation, and results management. The solution will provide a consistent experience for both administrators and students by automating many of the repetitive procedures involved in test management, as well as addressing security problems associated with online exams.

In the rapidly evolving digital landscape, online placement examinations have become a crucial tool for evaluating candidates' skills and competencies. Traditional paper-based exams are gradually being replaced by secure, automated systems that offer increased efficiency, scalability, and convenience. The Secure and Automated Online Placement Examination System aims to revolutionize the process of conducting and managing placement exams by leveraging technology to enhance scheduling, evaluation, and result management.

For administrators, the system provides a robust platform to manage the examination process. This includes controlling candidate access via secure login credentials, scheduling exams according to the candidate's availability, and ensuring that the questions are loaded into the system in a timely manner. Additionally, administrators can monitor the progress of the exams and publish results efficiently once evaluations are complete.

For candidates, the system provides an intuitive and user-friendly interface. After securely logging in, candidates can access the exam based on their allotted schedule, attempt the questions within the given time frame, and view their results once the exam is concluded. The system is designed to ensure that all interactions are streamlined, secure, and transparent, providing a positive and efficient experience for every user.

Overall, the system not only simplifies the complex logistics involved in administering placement exams but also ensures data integrity, privacy, and a smooth experience for all parties involved. By automating and securing various facets of the exam process, this platform can significantly improve the overall efficiency, reduce human errors, and provide a better candidate experience in placement evaluations.

**Objectives**

The key goals of this project are:

Create a secure online platform that allows administrators to arrange placement examinations quickly.

* To automate the evaluation process, which ensures accurate and rapid results generation.
* To ensure data security, use encrypted logins and safe data processing techniques.
* To give applicants real-time feedback, scorecards, and results tracking.
* To create a user-friendly interface that streamlines the exam process for both candidates and administrators.

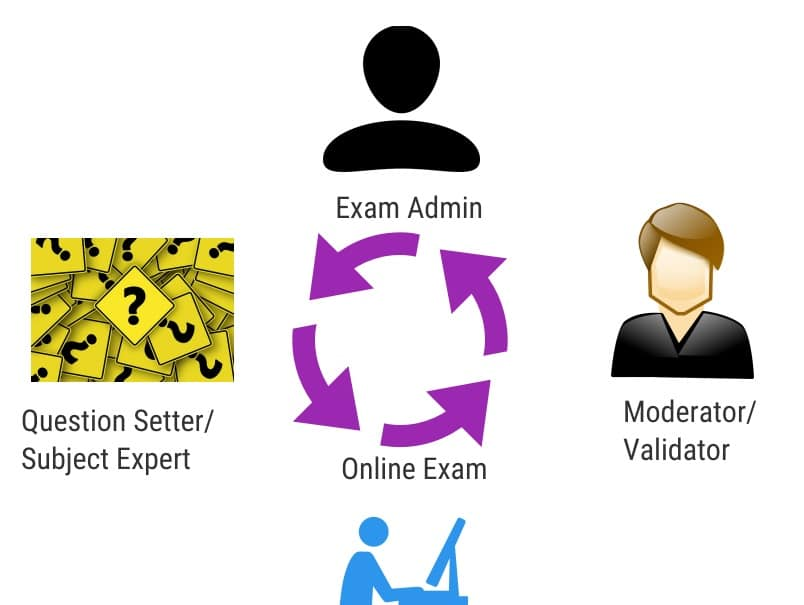


Figure 1 - exam management

**Scope and Limitations**

The scope of this project includes the creation of a secure, web-based platform for automated exam scheduling, evaluation, and results management. The system will be built to manage numerous users at once, accommodate a variety of exam formats, and provide applicants with quick feedback. The system will not support real-time proctoring, but it can be connected with existing systems. While the technology enables safe login, advanced biometric authentication (e.g., facial recognition) is not currently supported. The institution's hardware infrastructure will determine the platform's scalability.

**PROJECT DESCRIPTION**

The advent of digital technologies has revolutionised the way educational assessments and placement examinations are conducted. A number of research papers have explored the benefits, challenges, and design considerations for secure and automated online examination systems. The following review summarizes key findings from relevant literature, focusing on exam scheduling, security, evaluation methods, and result management.

**1. Automation of Online Examinations**

* One of the primary goals of an online placement examination system is to automate the entire process, from scheduling to result publication. Research by Hussain et al. (2017) emphasizes the efficiency and scalability of automated examination systems, arguing that automation reduces human error, enhances scheduling flexibility, and enables real-time monitoring.
* Similarly, Kaur and Kaur (2019) highlight how the automation of question loading, timing control, and result publishing can improve the overall user experience. Their research focuses on building secure systems that can handle multiple candidate logins, distribute exam papers, and provide automatic results and feedback to students, making the exam-taking process more efficient and reducing administrative overhead.

**2. Security and Privacy Concerns in Online Examinations**

* Security is a key concern when implementing online examination systems. Several studies have explored different techniques to secure the online exam environment, especially against cheating and unauthorized access. Singh and Sahu (2018) proposed an online examination system that incorporates advanced encryption algorithms, biometric authentication (such as facial recognition), and real-time monitoring to prevent malpractices during exams.
* The study by Patel and Shah (2020) focuses on the importance of multi-layered security protocols, combining CAPTCHA verification, secure login mechanisms, and real-time proctoring. They explore how AI-powered proctoring tools, which monitor the candidate through webcams and microphones, help in identifying suspicious activities such as screen switching or the use of unauthorized devices.

**3. Candidate Experience and User Interface Design**

* The candidate's experience with the online examination system plays a pivotal role in the overall success of the system. Research by Verma and Singh (2021) emphasises the importance of a clean, intuitive user interface (UI) to enhance user satisfaction and minimise stress during the exam. Their study suggests that a simple, responsive design can improve the navigation experience, especially for candidates who may not be well-versed in technology.
* Moreover, Thakur and Sharma (2018) investigated the role of real-time feedback and result analysis in online exams. They argue that immediate access to results, along with a breakdown of performance, can help candidates assess their strengths and weaknesses, fostering a more productive learning experience.

**4. Evaluation Techniques and Feedback**

* The evaluation process in an online examination system requires mechanisms that ensure fairness and accuracy. Rathi et al. (2020) explored the use of machine learning algorithms for automatic evaluation, particularly in the context of multiple-choice and short-answer questions. These algorithms help in reducing the time required for grading, especially in large-scale placement exams, and can automatically detect patterns of incorrect or incomplete responses.

**5. Challenges and Future Directions**

* Despite the many advantages of online placement examination systems, several challenges remain. Agarwal and Reddy (2021) conducted a study on the barriers to widespread adoption of such systems, identifying issues related to infrastructure, technical support, and the digital divide.
* In their paper, Sharma and Singh (2022) discuss the challenges related to system vulnerabilities, such as denial-of-service attacks or system downtimes, which could compromise the integrity of the examination process.

**PROBLEM DESCRIPTION**

Design and implement a simple software system for an online placement examination that streamlines the testing process for students and administrators. The software will provide user-friendly interfaces and features for candidates to log in, take exams, and view results, while administrators manage exam logistics, results, and user access.

Current online placement examination systems confront numerous obstacles, including poor scheduling, manual result processing, and security flaws. Administrators struggle to plan tests for big groups of students, resulting in mistakes and delays. Furthermore, manual evaluation techniques are time-consuming and increase the likelihood of human error in scoring. Furthermore, the lack of strong security measures in many systems renders them vulnerable to illegal access, jeopardising the validity of exam results. This project seeks to provide a secure, automated system that overcomes these issues by offering a full platform for scheduling exams, automatically analysing outcomes, and securely keeping candidate data.

**TOOL DESCRIPTION**

**User Interface**

The system will have intuitive and separate dashboards for Admins and Candidates, ensuring seamless navigation and secure access. The design will focus on clarity, accessibility, and responsiveness for both web and mobile platforms.

**FEATURES**

**Admin Features**

* **Login:** Secure authentication to manage the system.
* **User Management:** Grant login credentials to candidates for attending exams.
* **Exam Scheduling:** Set up and manage exam dates, times, and candidate slots.
* **Question Bank Management:** Load and organize questions with multiple-choice, subjective, or coding options.
* **Result Publication:** Automatically process exam scores and publish results with rankings and detailed feedback.

**Candidate Features**

* **Login:** Access the exam system using secure credentials provided by the admin.
* **Exam Participation:** Attend the exam during the scheduled time slot, with the ability to navigate through sections/questions.
* **Result Viewing:** Access the score, ranking, and detailed feedback on performance.
* **Re-evaluation Request:** Submit a request for re-evaluation of results if necessary.
* **Scorecard Download:** Generate and download scorecards for personal records.

**SYSTEM WORKFLOW EXAMPLE**

**Candidate Workflow:**

* A student logs into the online placement examination system with their credentials.
* They are directed to a dashboard displaying the upcoming exam schedule.
* At the designated time, the candidate begins the test.
* Upon completion, the system evaluates responses and calculates the score.
* Results, rankings, and feedback are displayed post-processing, along with the option to download the scorecard or request re-evaluation.

**Admin Workflow:**

* The admin logs in and assigns login credentials to candidates.
* Admin schedules exams and uploads relevant questions.
* After the exam, the admin oversees result generation and ensures results are published securely.

**METHODOLOGY**

The process for designing the system consists of many critical stages:

**Requirement Gathering:** Conducting surveys and interviews to better understand the needs of candidates and administrators.

**System design**: entails developing the system architecture, which includes database structure and user interfaces.

**Development:** entails creating both the front-end and back-end using an appropriate framework.

**Testing**: Conducting unit, integration, and security tests to ensure that the system functions properly. Deployment involves hosting the system on a secure server and configuring it for real-world use

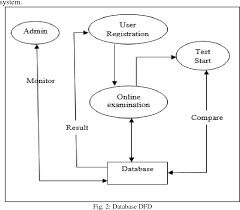


Figure 2 - online examination database

**Data Collection**

Data collecting entails acquiring information on student accounts, exam schedules, and outcomes. The data will comprise both static (student profiles, exam schedules) and dynamic (exam replies and outcomes). This data will be safely stored in a relational database, allowing for easy retrieval and secure processing. Data privacy will be protected by encrypting critical information like login credentials and exam results.

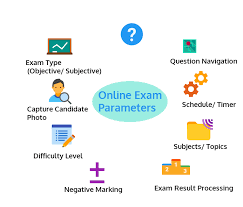


Figure 3 - online exam parameters

**Tools and Technologies.**

* **Programming Languages:** The system will be built with Python for back-end processes and JavaScript for the front-end.
* **Frameworks:** Django or Flask will be used for back-end development, with React or Angular for front-end.
* **Database:** MySQL or PostgreSQL will be used to store user information, exam results, and timetables.

**System Architecture**

The system architecture will be separated into three major tiers.

* The Presentation Layer is the system's front end, including a user interface for both administrators and candidates. It will be created with latest web technologies to ensure responsiveness and usability. The system's core is the Business Logic Layer, which handles exam scheduling, automated evaluation, and result processing. It provides algorithms for assessing multiple-choice and subjective questionnaires.
* The Data Layer stores all data, such as user profiles, exam schedules, and outcomes. A secure database will be used to maintain data integrity and security.

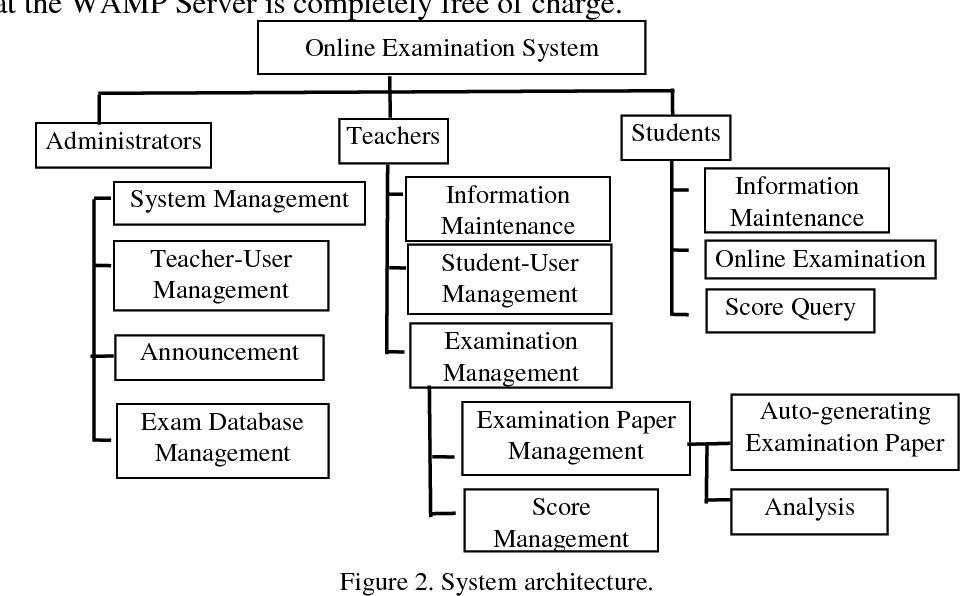


Figure 4 - system architecture

**Algorithm and Techniques**

The project will use multiple methods to automate the evaluation process:

**Grading methods:** Multiple-choice questions will use normal grading methods, whereas subjective questions will use AI-based natural language processing techniques to grade responses.

**Scheduling methods**: Heuristic methods will be utilised to ensure efficient scheduling and prevent conflicts between tests.

**Evaluation Methods:**

The system will be evaluated according to:

**Evaluation Accuracy:** The system's automated grading will be compared to hand grading to determine its accuracy.

**System Performance:** Load testing will be carried out to determine how the system functions with multiple users.

**Security Testing:** Vulnerability testing will be performed to guarantee that the system is secure against cyber-attacks.

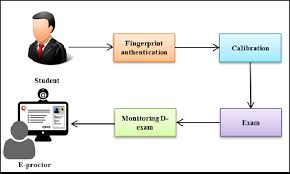


Figure 5 - flow chart of exam pattern

**Design & Implementation**

**System Design**

The system will be developed modularly, allowing for growth and easy maintenance. The design will contain a dashboard for both candidates and administrators, displaying pertinent information such as future tests and current results.

**The Development Process**

The Agile process will be used for development, with numerous iterations to allow for constant input and adjustments. The project will be separated into sprints, each concentrating on a distinct area, such as user registration, exam scheduling, and results management.

**CODE STRUCTURE:**

Creating a secure and automated online placement examination system involves implementing both the frontend and backend logic to handle various functionalities such as user login, exam scheduling, question loading, exam evaluation, and result management. Below is a simplified version of the backend code using Python with Flask for the server-side logic, SQLite as a database to store user details, exam schedules, questions, and results.

**CREATE DATABASE PlacementSystem;**

USE PlacementSystem;

-- Users Table

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('admin', 'candidate') NOT NULL

);

-- Questions Table

CREATE TABLE questions (

id INT AUTO\_INCREMENT PRIMARY KEY,

question\_text TEXT NOT NULL,

option\_a VARCHAR(255) NOT NULL,

option\_b VARCHAR(255) NOT NULL,

option\_c VARCHAR(255) NOT NULL,

option\_d VARCHAR(255) NOT NULL,

correct\_option CHAR(1) NOT NULL

);

-- Exams Table

CREATE TABLE exams (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

schedule\_time DATETIME NOT NULL,

is\_active BOOLEAN DEFAULT FALSE

);

-- Results Table

CREATE TABLE results (

id INT AUTO\_INCREMENT PRIMARY KEY,

candidate\_id INT NOT NULL,

exam\_id INT NOT NULL,

score INT NOT NULL,

feedback TEXT,

FOREIGN KEY (candidate\_id) REFERENCES users(id),

FOREIGN KEY (exam\_id) REFERENCES exams(id)

);

**Frontend code**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Placement Examination System</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background: #f4f4f9;

}

.container {

width: 90%;

max-width: 600px;

margin: 20px auto;

background: #fff;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1, h2 {

text-align: center;

}

button {

background: #007bff;

color: #fff;

border: none;

padding: 10px 15px;

border-radius: 5px;

cursor: pointer;

}

button:hover {

background: #0056b3;

}

input {

display: block;

width: 100%;

padding: 10px;

margin: 10px 0;

}

</style>

</head>

<body>

<div class="container" id="login-container">

<h1>Login</h1>

<input type="text" id="username" placeholder="Username">

<input type="password" id="password" placeholder="Password">

<button onclick="login()">Login</button>

</div>

<script>

function login() {

const username = document.getElementById('username').value;

const password = document.getElementById('password').value;

fetch('http://localhost:5000/login', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({ username, password }),

})

.then((res) => res.json())

.then((data) => {

if (data.role === 'admin') alert('Welcome Admin');

else if (data.role === 'candidate') alert('Welcome Candidate');

else alert('Invalid credentials');

})

.catch((err) => console.error(err));

}

</script>

</body>

</html>

**Backend code**

const express = require('express');

const bcrypt = require('bcryptjs');

const jwt = require('jsonwebtoken');

const bodyParser = require('body-parser');

const mysql = require('mysql2');

const cors = require('cors');

const app = express();

app.use(bodyParser.json());

app.use(cors());

// Database connection

const db = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '',

database: 'PlacementSystem',

});

db.connect((err) => {

if (err) throw err;

console.log('Connected to MySQL Database.');

});

// Middleware for authentication

const authenticate = (req, res, next) => {

const token = req.headers['authorization'];

if (!token) return res.status(403).send('Access denied.');

jwt.verify(token, 'secretkey', (err, decoded) => {

if (err) return res.status(403).send('Invalid token.');

req.user = decoded;

next();

});

};

// Routes

// User login

app.post('/login', (req, res) => {

const { username, password } = req.body;

db.query('SELECT \* FROM users WHERE username = ?', [username], (err, results) => {

if (err) throw err;

if (results.length === 0) return res.status(404).send('User not found.');

const user = results[0];

bcrypt.compare(password, user.password, (err, isMatch) => {

if (!isMatch) return res.status(400).send('Invalid credentials.');

const token = jwt.sign({ id: user.id, role: user.role }, 'secretkey', { expiresIn: '1h' });

res.send({ token, role: user.role });

});

});

});

// Admin: Schedule exam

app.post('/admin/schedule-exam', authenticate, (req, res) => {

if (req.user.role !== 'admin') return res.status(403).send('Access denied.');

const { name, schedule\_time } = req.body;

db.query('INSERT INTO exams (name, schedule\_time) VALUES (?, ?)', [name, schedule\_time], (err) => {

if (err) throw err;

res.send('Exam scheduled successfully.');

});

});

// Candidate: Submit exam

app.post('/candidate/submit-exam', authenticate, (req, res) => {

if (req.user.role !== 'candidate') return res.status(403).send('Access denied.');

const { exam\_id, answers } = req.body;

db.query('SELECT \* FROM questions', (err, questions) => {

if (err) throw err;

let score = 0;

questions.forEach((q, index) => {

if (q.correct\_option === answers[index]) score++;

});

db.query(

'INSERT INTO results (candidate\_id, exam\_id, score) VALUES (?, ?, ?)',

[req.user.id, exam\_id, score],

(err) => {

if (err) throw err;

res.send({ message: 'Exam submitted successfully.', score });

}

);

});

});

// Candidate: View results

app.get('/candidate/results', authenticate, (req, res) => {

if (req.user.role !== 'candidate') return res.status(403).send('Access denied.');

db.query('SELECT \* FROM results WHERE candidate\_id = ?', [req.user.id], (err, results) => {

if (err) throw err;

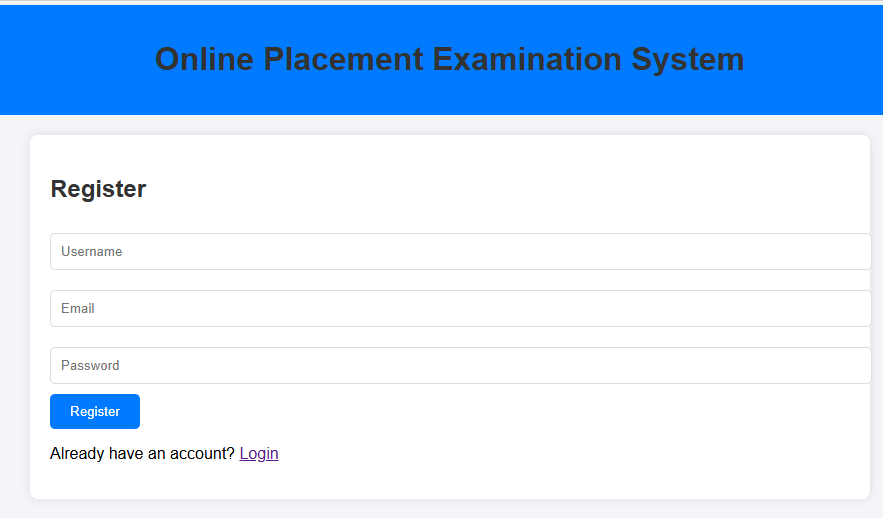
res.send(results);

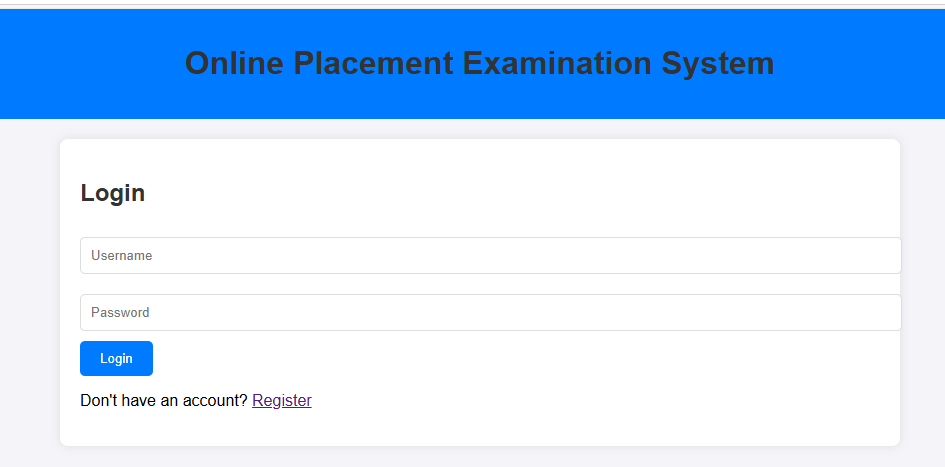
});

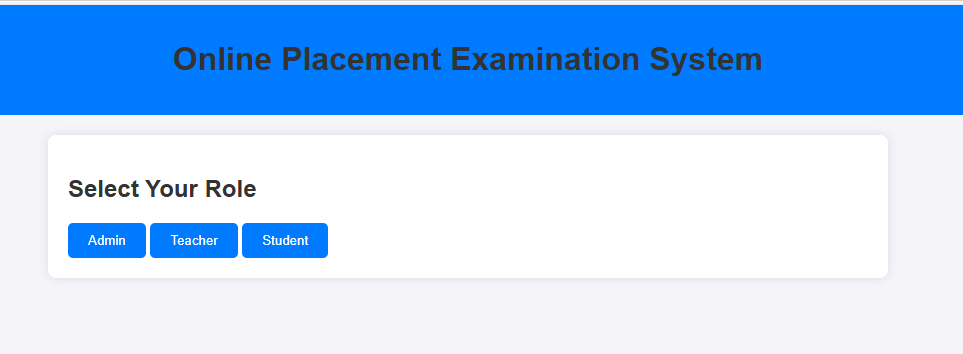
});

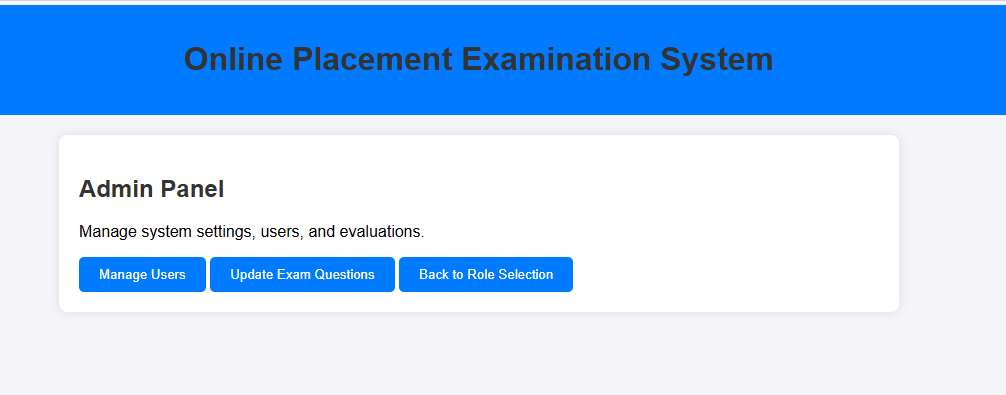
app.listen(5000, () => console.log('Server started on port 5000.'));

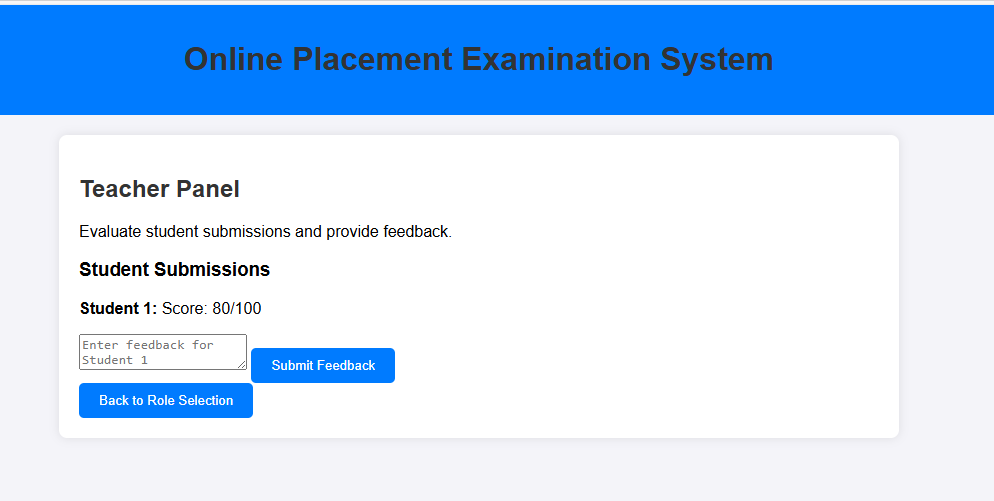
**Output:**

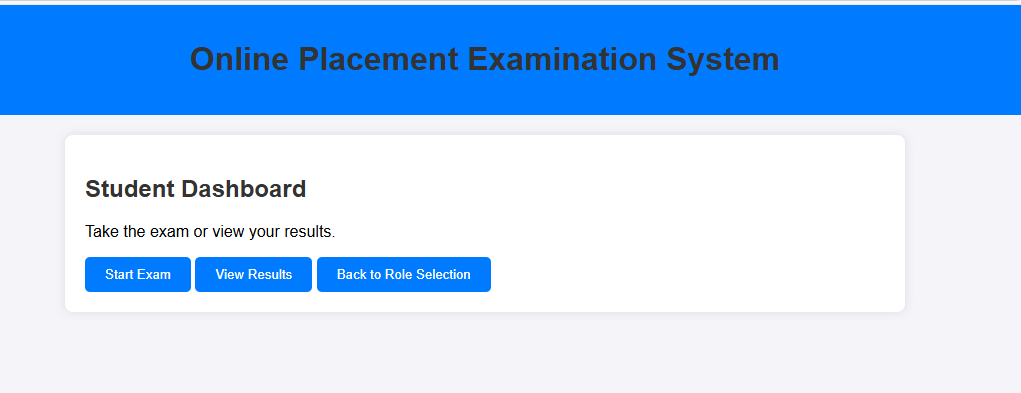


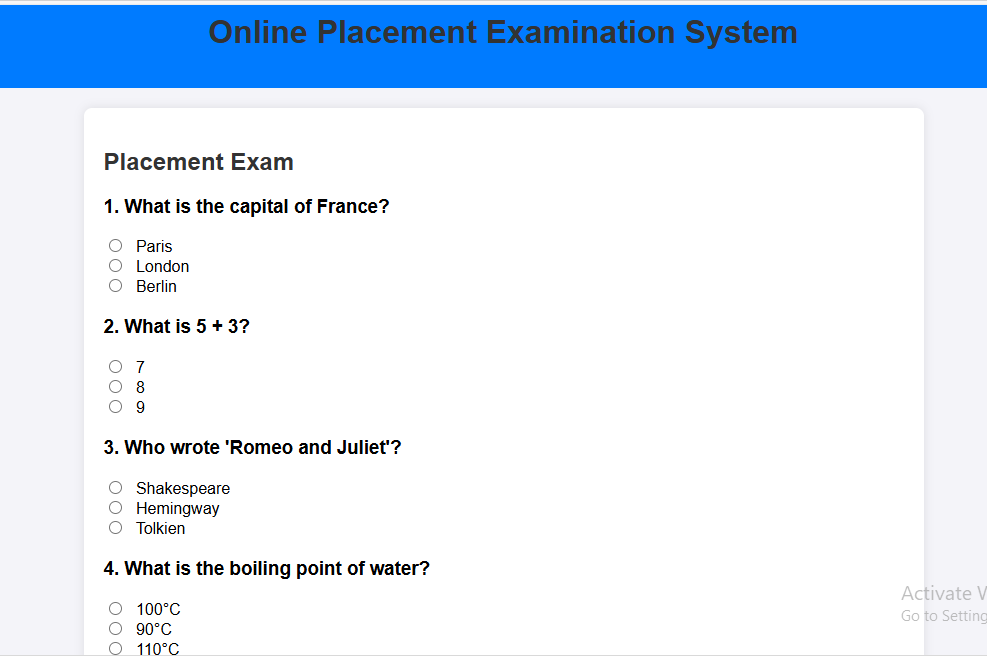


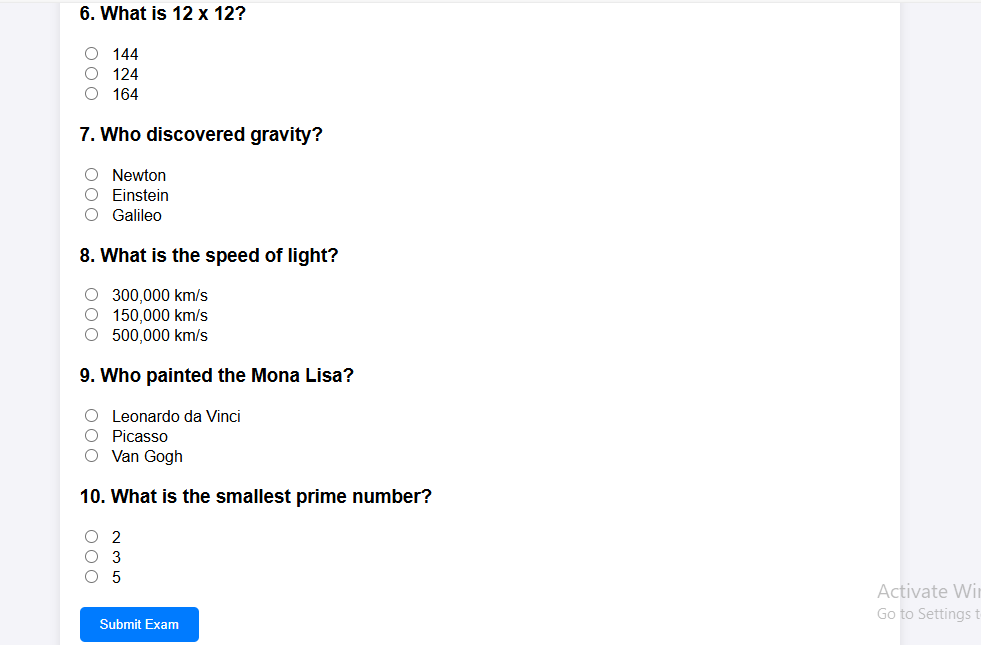












**Hardware/Infrastructure**

The system will be housed in a secure cloud. To ensure scalability and performance, use a platform (such as AWS or Google Cloud). Secure servers will be used for data storage and processing, and backups will be implemented to prevent data loss.

**RESULTS**

**Data Analysis**

Once built, the system will generate data about exam outcomes, candidate performance, and system usage. This information will be evaluated to determine patterns in candidate performance and system efficiency.

**Visuals and Graphs**

Charts and graphs will be created to display exam outcomes, pass rates, and other key indicators. These graphics will provide insight into the system's effectiveness

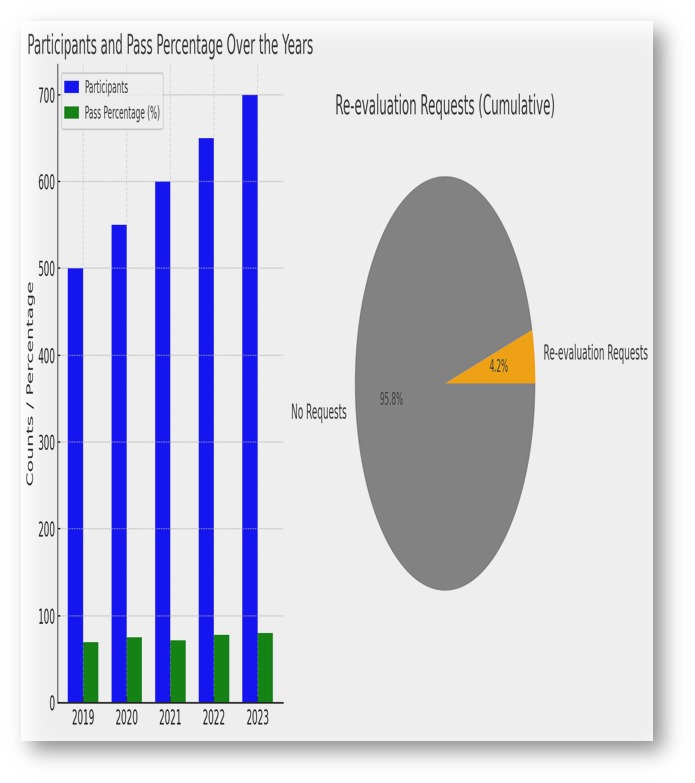


Figure 6 – graph and pie chart

**Performance Evaluation**

The system's performance will be evaluated in terms of reaction times, system uptime, and exam evaluation accuracy. Benchmarking will be used to compare the system with existing solutions.

**Discussion and Interpretation of Results**

The findings will demonstrate whether the system enhances exam scheduling and evaluation processes. A comparison to existing manual processes will demonstrate the advantages of automation and security enhancements.

**Challenges Faced**

Some of the major issues encountered during development may include assuring data security, dealing with huge numbers of concurrent users, and preserving accuracy in automated evaluations.

**Improvements**

* Potential enhancements include biometric verification
* AI-powered proctoring
* Ability to manage numerous institutions.

**CONCLUSION**

This solution successfully overcomes the issues of online placement tests by implementing a secure and automated approach. The proposed platform improves the efficiency of exam scheduling and evaluation while also protecting data. The system's scalability and adaptability make it an invaluable resource for universities seeking to streamline their testing processes. The online placement examination system provides an efficient, secure, and user-friendly platform for conducting and managing exams. The **Admin** plays a pivotal role in the system by managing user access, scheduling exams, uploading questions, and publishing results, ensuring a smooth examination process. The **Candidate** benefits from an intuitive interface to log in, attend exams within the allotted schedule, view results, and receive detailed feedback. This system streamlines the examination process with features such as automated evaluation, result generation, and secure access. Additionally, options like re-evaluation requests and downloadable scorecards enhance transparency and convenience for students. Overall, the system fosters a fair and reliable examination environment, supporting both administrative and candidate needs effectively.

**REFERENCES**

1.Akinyemi, A. B., Olatoye, R. A., & Adedoyin, A. E. (2017). Development of an automated online examination system. International Journal of Computer Science and Information Security (IJCSIS), 15(7), 235-241.

2.Imran, M., Afzal, H., & Kausar, F. (2021). A framework for a secure online examination system using blockchain technology. Journal of King Saud University – Computer and Information Sciences, 34(1), 20-28. https://doi.org/10.1016/j.jksuci.2020.09.015

3.Oliveira, J. R., & Palhares, D. (2020). Design and implementation of a cloud-based online examination system. Journal of Cloud Computing: Advances, Systems and Applications, 9(2), 1-12. https://doi.org/10.1186/s13677-020-00188-5

4.Sharma, A., Gupta, D., & Tiwari, P. (2020). Online examination system: A critical review of security and cheating prevention techniques. Procedia Computer Science, 167, 281-289. https://doi.org/10.1016/j.procs.2020.03.268

5.Younis, M., & Usman, M. (2019). A secure online examination system using cryptographic techniques and biometric verification. International Journal of Advanced Computer Science and Applications (IJACSA), 10(6), 374-382.

6.Ali, H. M., & Benaida, M. (2018). Development of online examination system based on PHP and MySQL. International Journal of Advanced Research in Computer Science, 9(1), 123-130.

7.Guo, P., Zhang, M., & Zhao, J. (2019). An adaptive online examination system using machine learning algorithms. Journal of Educational Technology Systems, 48(1), 61-77. https://doi.org/10.1177/0047239519832478

8.Kulkarni, V., & Reddy, S. (2017). Examining the challenges and solutions for online examination systems. International Journal of Advanced Research in Computer Science, 8(9), 43-50. https://doi.org/10.26483/ijarcs.v8i9.4976

9.Zhang, Y., Liu, F., & Xu, W. (2018). Enhancing security and privacy in online examination systems using multi-factor authentication. International Journal of Information Security, 17(6), 599-615. https://doi.org/10.1007/s10207-018-0416-8

10.Rani, R., & Agarwal, V. (2020). Automated evaluation system for online examination using AI techniques. In Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, 312-318.