

# **Development of a Question Answer Assessment WebPortal Using Django**

submitted by

**Robiul Islam**

Class Roll No: 002210503036

Exam Roll: MCA244035

Master of Computer Application

under the supervision of

**Prof. (Dr.) Nibaran Das**

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

2024

Faculty of Engineering and Technology  
Jadavpur University

## **CERTIFICATE OF RECOMMENDATION**

### **To Whom It May Concern**

I hereby recommend that the project report entitled “**Development of a Question Answer Assessment WebPortal Using Django**” has been carried out by Robiul Islam (Reg. No.: 163628 of 2022-23, Roll No.: 002210503036, Exam Roll: MCA244035), under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic year 2022-2024.

---

Prof. (Dr.) Nibaran Das  
Project Supervisor  
Dept. of Computer Science and  
Engineering  
Jadavpur University

---

Prof. (Dr.) Nandini Mukhopadhyay  
Head of the Department  
Dept. of Computer Science & Engineering  
Jadavpur University

---

Prof. Dipak Laha  
Dean, Faculty Council  
of Engineering & Technology  
Jadavpur University, Kolkata-032

Faculty of Engineering and Technology  
Jadavpur University

**CERTIFICATE OF APPROVAL**

This is to certify that the project report entitled “**Development of a Question Answer Assessment WebPortal Using Django**” is a bonafide record of work carried out by Robiul Islam, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

---

Signature of Examiner 1

Date :

---

Signature of Examiner 2

Date :

Faculty of Engineering and Technology  
Jadavpur University

**DECLARATION OF COMPLIANCE WITH  
ACADEMIC ETHICS**

I hereby declare that this project report entitled "**Development of a Question Answer Assessment WebPortal Using Django**" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

Robiul Islam

Registration No.: 163628 of 2022-23

Examination Roll No.: MCA244035

Thesis Title :- **Development of a Question Answer Assessment WebPortal Using Django**

---

Signature with Date

## **ACKNOWLEDGEMENT**

With my most sincere and gratitude, I would like to thank Prof. (Dr.) Nibaran Das, Department of Computer Science & Engineering, my supervisor, for his overwhelming support throughout the duration of the project. His motivation always gave me the required inputs and momentum to continue with my work, without which the project work would not have taken its current shape. His valuable suggestions and numerous discussions have always inspired new ways of thinking. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support.

Last, but not the least, I would like to thank my batch mates for staying by my side when I needed them.

---

Robiul Islam

Class Roll No.: 002210503036

Examination Roll No.: MCA244035

Department of Computer Science & Engineering

Jadavpur University

# **Abstract**

Question Answer Assessment Web Portal is a portal that provides an efficient, user-friendly way to conduct and manage assessments online. This web portal is a one stop unified solution to create, administer various types of exam and evaluate them as well which is developed on Django framework. Its wide range of functionality include everything that educators and students need; most importantly, it includes question banks, auto grading and performance analytics along with user management.

This project aims at simplifying the process of assessment so that it can be easily employed and utilized by educational institutions. This system ensures strong feature set, scalable architecture, and security by using Django framework. It supports several roles like Admin, Teacher & Student with respective functions to provide a user-friendly environment.

The goal of this project is to get above the limitations of both the existing online platforms and traditional paper examinations. It includes cutting-edge features including real-time feedback, several question types, and thorough performance reviews. To ensure dependability and user satisfaction, the creation process involved thorough requirement analysis, system design, and intensive testing. The initiative's results show a significant improvement in assessment effectiveness, and early users have responded favourably. More compatibility with various learning platforms and resources as well as AI-based analytics are among the upcoming improvements.

# CONTENTS

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| <b>1.Introduction</b>                                              | <b>9</b>  |
| 1.1 Background information                                         | 9         |
| 1.2 Objectives of the project                                      | 9         |
| 1.3 Importance and relevance of the project                        | 10        |
| <b>2. Literature Review</b>                                        | <b>11</b> |
| 2.1 Overview of existing systems and<br>their inbuilt Technologies | 11        |
| 2.2 Limitations of existing systems and<br>their solving methods   | 12        |
| 2.3 Technologies and Methodologies                                 | 12        |
| <b>3. System Analysis and design</b>                               | <b>14</b> |
| 3.1 Functional Requirements analysis                               | 14        |
| 3.2 Non-functional Requirements analysis                           | 15        |
| 3.3 Use case diagrams and descriptions                             | 16        |
| 3.4 System Architecture                                            | 18        |
| 3.5 Details Design                                                 | 20        |
| 3.6 User Interface Design                                          | 22        |
| <b>5. Implementation</b>                                           | <b>24</b> |
| 5.1 Technologies and tools used (Django, SQLite, etc.)             | 24        |
| 5.2 Description of the main modules                                | 24        |

(exam, student, teacher, etc.)

|                                          |           |
|------------------------------------------|-----------|
| <b>7. Results and Analysis</b>           | <b>26</b> |
| 7.1 Outcomes of the project              | 26        |
| 7.2 Performance analysis                 | 27        |
| 7.3 Testing                              | 27        |
| 7.4 Demonstration of the proposed system | 30        |
| 7.5 Question Generation                  | 34        |
| <br>                                     |           |
| <b>8. Conclusion and Future scope</b>    | <b>35</b> |
| <b>9. References</b>                     | <b>36</b> |

# 1. Introduction

The need for question answer assessment systems that are scalable, reliable and easy to use has never been more pressing in today's education world. This is why traditional paper and pen-based assessments and simple online testing systems fail to cater for the needs of modern education. A Django Project which is a high-level, open-source web framework written in Python that encourages rapid development and clean, pragmatic design that helps conduct and manage Online Question Answer Assessments through a web portal.

## 1.1 Background Information

Many sectors have changed a lot due to the arrival of digital technologies, education is also one of those. The landscape of higher education is changing, with the rapid adoption of online learning and assessment, which have likely been facilitated by greater internet access and digital devices. Nevertheless, a lot of educational infrastructures still use traditional methods for exams and quizzes which are time-consuming, error-prone and not so reliable to manage.

## 1.2 Objective of the Project

The main purpose of this project is to create a strong web portal in which processes (preparing, conducting and evaluating assessments) would be as easy as counting 3. The system aims to:

- Deliver a user-friendly module where educators can create/ manage assessments.
- Automate grading to take the load off your plate and make sure only assignments foreseen by the course outline are considered for a module grade.
- Provide an Interactive, Holistic and Smart test-taking experience for students.
- Provide extensive performance analytics that educators and students can use to identify where they need help.

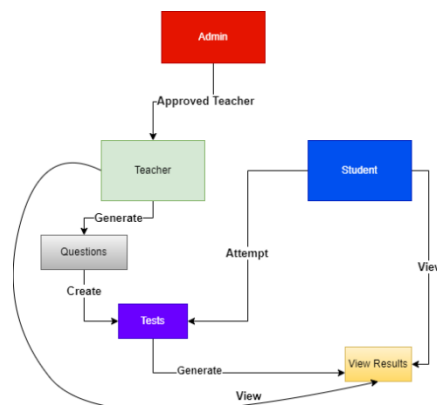
### 1.3 Importance and Relevance of the Project

The benefits of having an efficient online assessment system are as follows :

- It leads to Efficiency — Automates different parts of the assessment process which saves time and resources.
- Accessibility: Performs best for students taking examinations from anywhere which helps them in increasing their participation.
- Reduces Human Error in Grading/Assessment
- Robust analytics: Detailed reports that can help understand how students are performing and inform instructional strategies.

This project uses Django, a high-level Python web framework to make sure the system is scalable, secure and maintainable. A comprehensive set of core functionalities like ORMs (Django is a model-view-template framework), authentication processes, and admin interfaces makes things easier for the developers to get their work done quickly and deploy it with much effort in maintaining them.

The system will not only help in enhancing the efficiency of educational assessments but also contribute towards the larger goal of technology-led Education to improve learning outcomes. The rest of the report delves into its system analysis, design, implementation testing and results to give a clear picture of how it was developed and what exactly can be done with the.



**Figure-1. Simple overview of this Project**

## 2. Literature Review

The authors supported the above statement with a thorough review of the literature that illustrated current systems, approaches as well as technologies for online assessments. This section is designed to detail the capabilities and deficiencies of extant solutions and pinpoint where "QAA-Web Portal" can be more useful.

### 2.1 Overview of existing systems and their inbuilt limitations

#### 1. Traditional Paper-Based Assessments

- **Strengths:** Widely understood and accepted, easy to administer without technology.
- **Weaknesses:** Time-consuming grading process, prone to errors, and logistical challenges in distribution and collection of papers.
- **Limitations:** No real-time feedback, difficult to keep records and use paper.

#### 2. Basic Online Testing System

- **Strengths:** Less paper, immediate grading for objective questions, ease of administration.
- **Weaknesses:** Limited question types, basic user interfaces, lack of advanced features like analytics and real-time feedback.
- **Limitations:** Often lack scalability, and security vulnerabilities, and may not support a wide range of question formats.

#### 3. Advanced Online Assessment Platforms

- **Strengths:** Comprehensive feature sets including various question types, automated grading, detailed analytics, and user management.
- **Weaknesses:** High cost, complexity in setup and management, and potential issues with user interface design.
- **Limitations:** May require significant technical expertise to customize and maintain, potential integration issues with other educational tools.

## 2.2 Limitation of existing systems and their solving methods

The "Question Answer Assessment Web Portal" aims to address these gaps by:

- **Integration:** Designing the system with integration capabilities in mind, allowing it to work seamlessly with other educational tools.
- **Customization:** Offering a high degree of customization to meet the specific needs of various users, from administrators to students.
- **User Experience:** Prioritizing user-friendly design and intuitive interfaces to enhance the overall user experience.
- **Scalability:** Leveraging Django's robust architecture to ensure the system can scale to accommodate a large number of users without compromising performance.

By building on the strengths of existing systems and addressing their limitations, this project aims to provide a comprehensive, efficient, and user-friendly solution for online assessments.

## 2.3 Technologies and Methodologies

### 1. Django Framework

- **Pros:** Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. This framework provides us with user authentication, admin interfaces, and ORM (Object Relational Mapping) right out of the box as well which gives a multitude of tools that can help create some real web applications.
- **Cons:** May be too heavy when developing very simple applications, and beginners who are not familiar with Python will face a steep learning curve.

### 2. Automated Grading System

- **Pros:** Quick feedback, grading consistency, less work for teachers.
- **Challenges:** Cannot handle complex subjective responses, Limited to auto-graded question types (multiple-choice questions and true/false).

### 3. Real-Time Feedback Mechanisms

- **Benefits:** Enhances learning by providing immediate insights into performance, and helps students identify and address gaps in knowledge promptly.
- **Drawbacks:** Requires robust backend infrastructure to handle real-time processing and may increase server load.

#### 4. Performance Analytics

- **Advantages:** Provides detailed insights into student performance, helps educators tailor instruction to meet individual needs, and identifies trends and patterns in data.
- **Challenges:** Data privacy concerns, require careful design to ensure meaningful and actionable insights.

# 3. System Analysis and Design

## System Analysis

System analysis involves understanding and specifying in detail what the system should do. This section covers the functional and non-functional requirements of the "Question Answer Assessment Web Portal," as well as use case diagrams and descriptions to provide a comprehensive view of the system's capabilities.

### 3.1 Functional Requirements

Functional requirements define the specific behavior or functions of the system. For the "Question Answer Assessment Web Portal," these include:

#### 1. User Management

- Registration and Login: Users can register with the system and log in using their credentials.
- Role-Based Access: Users are assigned roles (e.g., admin, teacher, student) with specific permissions.
- Profile Management: Users can view and update their profile information.

#### 2.Exam Management

- Create and Manage Exams: Teachers can create, edit, and delete exams.
- Question Bank: Teachers can create and manage a repository of questions categorized by subject and difficulty level.
- Exam Scheduling: Exams can be scheduled for specific dates and times.

#### 3.Assessment and Evaluation

- Conduct Exams: Students can take exams within the scheduled time frame.
- Automated Grading: The system automatically grades objective-type questions (e.g., multiple-choice, true/false).
- Manual Grading: Teachers can manually grade subjective questions and provide feedback.

#### 4.Results and Analytics

- View Results: Students can view their exam results and feedback.
- Performance Analytics: Teachers can access detailed analytics on student performance, including scores, time taken, and areas of improvement.

## **5.Communication and Notifications**

- Notifications: The system sends notifications to users about upcoming exams, results, and other important updates.
- Messaging: Teachers and students can communicate via an internal messaging system.

## **3.2 Non-Functional Requirements**

Non-functional requirements define the quality attributes of the system, such as performance, usability, and security.

### **1.Performance**

- The system should handle multiple concurrent users without significant degradation in performance.
- Page load times should be within acceptable limits to ensure a smooth user experience.

### **2.Usability**

- The user interface should be intuitive and easy to navigate.
- The system should provide help and support features to assist users.

### **3.Scalability**

- The system should be scalable to accommodate a growing number of users and exams.

### **4.Security**

- User data should be encrypted and stored securely.
- The system should implement robust authentication and authorization mechanisms to prevent unauthorized access.

### **5.Reliability**

- The system should have a high uptime, with minimal downtime for maintenance.
- Data backups should be performed regularly to prevent data loss.

### 3.3 Use Case Diagrams and Descriptions

#### 1. Use Case Diagram

A use case diagram provides a visual representation of the interactions between users and the system. Below is a simplified use case diagram for the "Question Answer Assessment Web Portal."

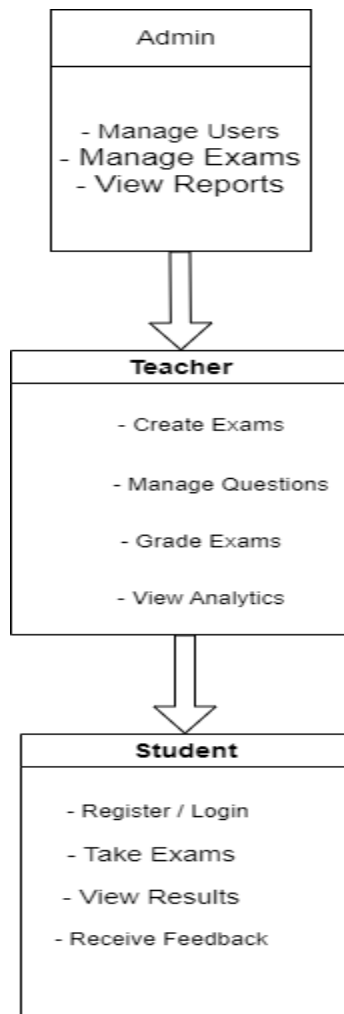


Figure-2 Use Case Diagram

#### 3. Use Case Description

- **Register/Login**

**Actor:** Student, Teacher, Admin

**Description:** Users can register for a new account or log in to an existing account using their credentials.

- **Manage Users**

**Actor:** Admin

**Description:** Admins can add, edit, or remove user accounts and assign roles.

- **Create Exams**

**Actor:** Teacher

**Description:** Teachers can create new exams by adding questions from the question bank or creating new questions.

- **Manage Questions**

**Actor:** Teacher

**Description:** Teachers can add, edit, or delete questions in the question bank.

- **Take Exams**

**Actor:** Student

**Description:** Students can take scheduled exams within the specified time frame.

- **Grade Exams**

**Actor:** Teacher

**Description:** Teachers can grade exams, with automated grading for objective questions and manual grading for subjective questions.

- **View Results**

**Actor:** Student

**Description:** Students can view their results and feedback after the exams are graded.

- **View Analytics**

**Actor:** Teacher, Admin

**Description:** Teachers and admins can access detailed performance analytics to monitor student progress and identify areas for improvement.

- **Receive Notifications**

**Actor:** Student, Teacher

**Description:** Users receive notifications about upcoming exams, results, and other important updates.

## System Design

The system design phase translates the requirements identified during system analysis into a blueprint for constructing the "Question Answer Assessment Web Portal." This section outlines the architecture of the system, including high-level and detailed design aspects such as system architecture, database design, user interface design, and class diagrams.

### 3.4 System Architecture

The system architecture of the web portal is based on a modular approach using Django's MVC (Model-View-Controller) framework. The architecture consists of the following main components:

#### 1. User Interface (UI) Layer

- Provides the front-end interface for users to interact with the system.
- Implemented using HTML, CSS, JavaScript, and Django templates.

#### 2. Application Logic Layer

- Handles the business logic of the application.
- Implemented using Django views and forms.

#### 3. Data Access Layer

- Manages data storage and retrieval.

- Implemented using Django models and the ORM (Object-Relational Mapping) for database interactions.

#### 4.Database

- Stores all the persistent data required by the application.
- Uses SQLite as the database management system (DBMS).

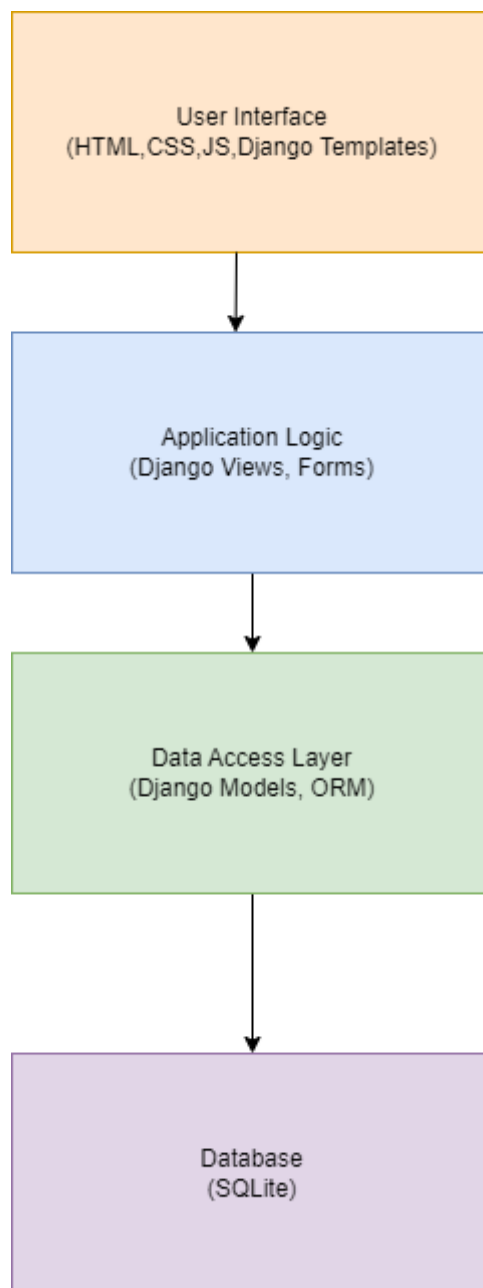


Figure – 3 High-Level Architecture Diagram

## 3.5 Detailed Design

### 1.Database Design

The database design involves defining the structure of the database, including tables, fields, and relationships. Here are the primary tables in the system:

#### 1.User Table

- Fields: user\_id, username, password, email, role (admin, teacher, student), date\_joined

#### 2. Exam Table

- Fields: exam\_id, title, description, subject, teacher\_id, start\_time, end\_time

#### 3.Question Table

- Fields: question\_id, exam\_id, content, question\_type (multiple-choice, true/false, essay), choices, correct\_answer

#### 5.StudentExam Table

- Fields: student\_exam\_id, student\_id, exam\_id, score, status (completed, pending)

#### 6.Answer Table

- Fields: answer\_id, student\_exam\_id, question\_id, answer, is\_correct

## Entity-Relationship(ER) Diagram

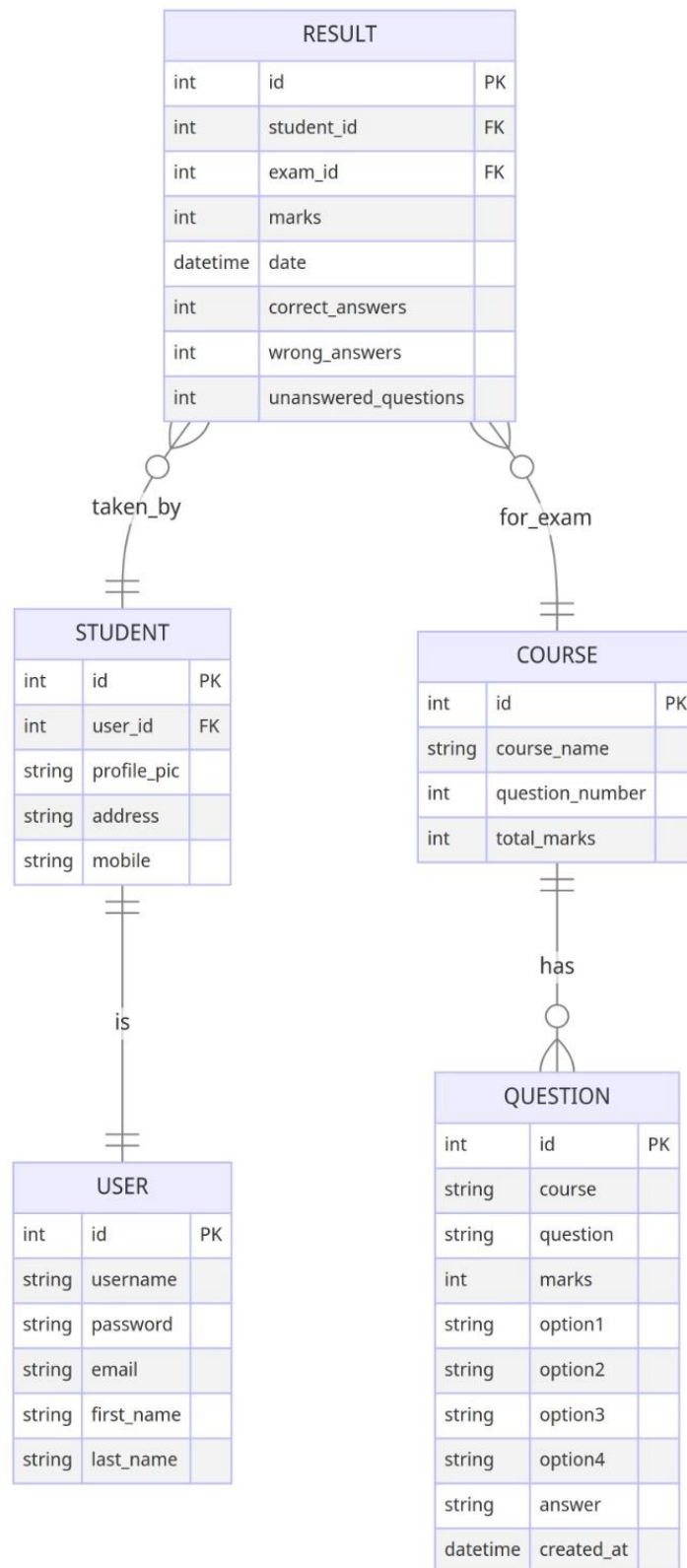


Figure – 4 ER Diagram

## **3.6 User Interface Design**

The user interface design focuses on creating an intuitive and user-friendly experience for all types of users. Below are some key aspects of the UI design:

### **1.Homepage**

- A welcoming page with navigation links to login, registration, and information about the portal.

### **2.Registration and Login Pages**

- Forms for users to create an account or log in using existing credentials.

### **3.Dashboard**

- Role-based dashboards for admin, teachers, and students.
- Admin dashboard: Manage users, view system reports.
- Teacher dashboard: Create and manage exams, view student performance.
- Student dashboard: View upcoming exams, take exams, view results.

### **4.Exam Creation and Management**

- Forms and interfaces for teachers to create new exams, add questions, and schedule exams.

### **5.Exam Taking Interface**

- User-friendly interface for students to take exams, including navigation between questions and submission options.

### **6.Results and Analytics**

- Pages for students to view their results and feedback.
- Analytics pages for teachers to review performance data and generate reports.

# Overall Flow of the project

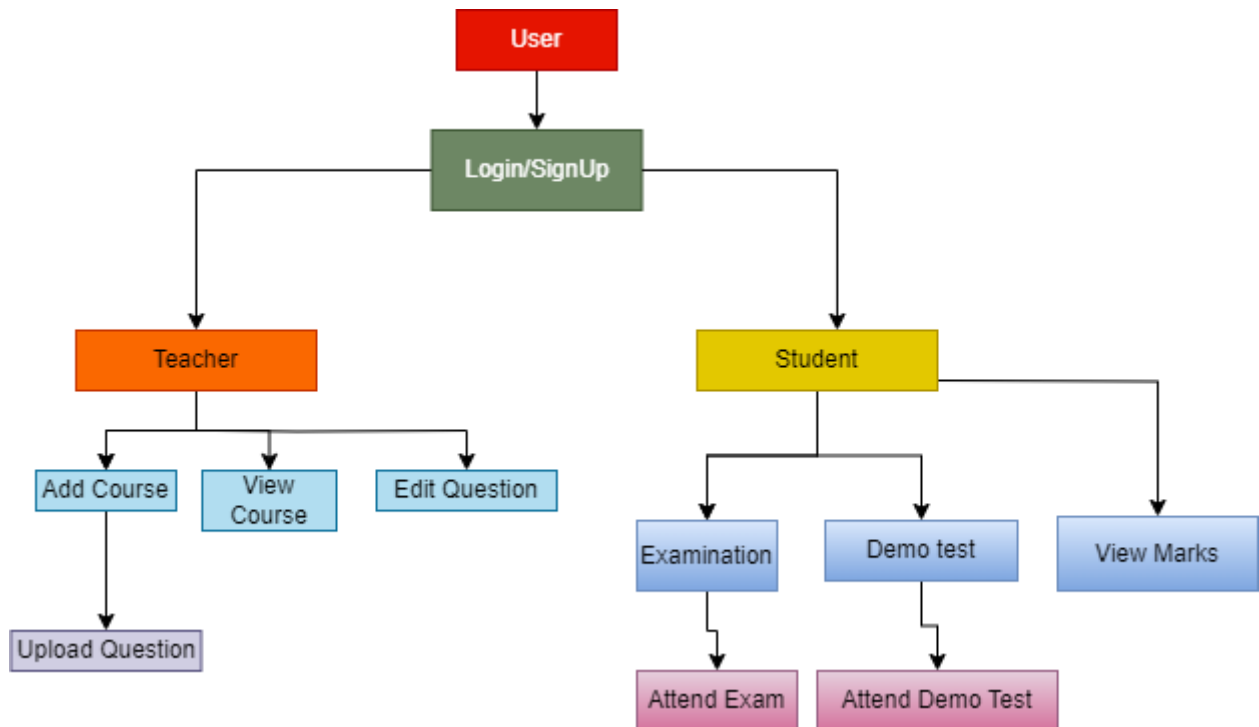


Figure – 5 Flow Diagram of Project

# 5. Implementation

The implementation phase: once the design is finished with these tools and selected technologies, they are translated into a complete functional system. The "Question Answer Assessment Web Portal" includes implementing the Django framework, entire application development with core features of the app and front-end integration to back-end pieces. Description- In this article, we will be implementing the same process step by step.

## 1.1 Technologies and tools used

1. **Django:** The main framework used for building the web application. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
2. **SQLite:** The database system used for storing application data. SQLite is a self-contained, serverless, and zero-configuration SQL database engine.
3. **Python:** The programming language used for the server-side logic.
4. **HTML/CSS:** For building the front-end templates and styling the web pages.
5. **JavaScript:** For client-side scripting and enhancing user interactions.
6. **Django's Built-in Authentication:** Used for managing user authentication and permissions.

## 5.2 Description of the Main Modules

The project appears to be an online examination system with the following primary modules

### 1. Exam Module

- Manages courses, questions, and results.
- Key models: Course, Question, Result.

### 2. Student Module

- Handles student-specific functionality such as registration, profile management, and exam participation.
- Key models: Student.

### **3. Teacher Module**

- Manages teacher-specific functionality such as adding and editing questions, creating exams, and viewing results.
- Templates and views are customized for teacher interactions.

### **4. Common Module**

- Includes shared functionality across students and teachers, such as user authentication and navigation.
- Utilizes Django's built-in user model and extends it for specific roles.

# 7. Results and Analysis

This section presents the outcomes of the "Question Answer Assessment Web Portal" project, including performance analysis, user feedback, and testing outcomes. It provides an in-depth discussion on how well the project met its objectives and the areas of success as well as those requiring improvement.

## 7.1 Outcomes of the Project

The "Question Answer Assessment Web Portal" successfully achieved its primary goal, which was to create a robust, user-friendly platform for conducting online assessments. The key outcomes are as follows:

### 1. Functional Online Assessment System:

- The platform allows teachers to create and manage exams efficiently.
- Students can easily register, log in, take exams, and view their results.

### 2. Automated Grading System:

- Objective questions (multiple-choice) are graded automatically, providing instant feedback to students.
- Subjective questions (essays) are supported, with manual grading interfaces for teachers.

### 3. User Role Management:

- Different user roles (admin, teacher, student) are managed effectively, with role-based access control ensuring that users can only access the features pertinent to their role.

### 4. Comprehensive Reporting and Analytics:

- The system generates detailed reports on student performance, enabling teachers to analyse exam results and identify areas where students need improvement.

## 7.2 Performance Analysis

The performance of the web portal was analyzed in terms of responsiveness, scalability, and reliability. Key performance metrics and observations include:

### 1. Responsiveness:

- The application demonstrated quick response times for most user actions, such as loading pages, submitting forms, and retrieving data.
- Page load times were kept within acceptable limits through optimization techniques like caching and efficient database queries.

### 2. Scalability:

- The system was tested for handling multiple concurrent users to ensure it scales well with increasing user load.
- Load testing indicated that the application could handle a significant number of concurrent users without a noticeable decrease in performance.

### 3. Reliability:

- The application was robust with minimal downtime during testing.
- Error handling mechanisms were implemented to manage exceptions gracefully, ensuring a smooth user experience even in the event of unexpected issues.

## 7.3 Testing

### 1. Testing Strategies and Methodologies

To ensure the "Question Answer Assessment Web Portal" functions correctly and meets user requirements, the following testing strategies were employed:

- **Unit Testing:** Tested individual components like models, views, forms, and utility functions.
- **Integration Testing:** Verified interactions between different modules.

- **System Testing:** Ensured the entire application meets specified requirements, including functional and non-functional aspects.
- **User Acceptance Testing (UAT):** Involved real users testing the system for feedback and final adjustments.
- **End-to-End Testing:** Automated browser-based testing using tools like Selenium to simulate real user interactions.

## 2. Test Cases and Results

I. Key test cases covered all major functionalities:

- User Registration and Login

II. Objective: Ensure successful registration and login.

- Result: Passed. Users could register and log in successfully.
- Exam Creation by Teachers

Objective: Ensure teachers can create new exams.

Result: Passed. Teachers could create exams without issues.

III. Adding Questions to Exams

- Objective: Verify adding various question types.
- Result: Passed. Questions were added and displayed correctly.

IV. Taking Exams by Students

- Objective: Ensure students can take and submit exams.
- Result: Passed. Students could take and submit exams successfully.

V. Automated Grading of Objective Questions

- Objective: Validate automatic grading.
- Result: Passed. Objective questions were graded correctly.

VI. Manual Grading of Subjective Questions

- Objective: Ensure teachers can manually grade essay questions.
- Result: Passed. Teachers could manually grade and save results.

### **3. Bug Fixes and Improvements**

During testing, several bugs were fixed, and improvements were made:

#### **I. Bug Fixes**

- User Registration Validation: Fixed invalid email format validation.
- Exam Timing: Corrected enforcement of exam start and end times.
- Question Display Order: Fixed order of question display.
- Error Handling: Improved error handling to prevent crashes.

#### **II. Improvements**

- User Interface Enhancements: Made the application more user-friendly and intuitive.
- Performance Optimization: Optimized database queries and implemented caching.
- Bulk Question Upload: Added a feature for bulk question uploads via CSV files.
- Real-time Feedback: Provided real-time feedback on objective questions immediately after submission.

These testing strategies and subsequent bug fixes and improvements ensured the system's robustness and user satisfaction.

## 7.4 Demonstration of the proposed system

This section contains some screenshots of the component of the system.

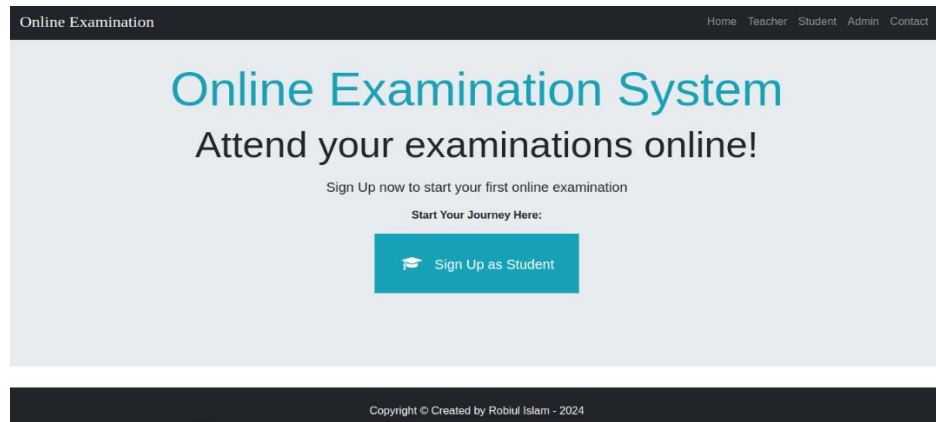


Figure-6 Home Page

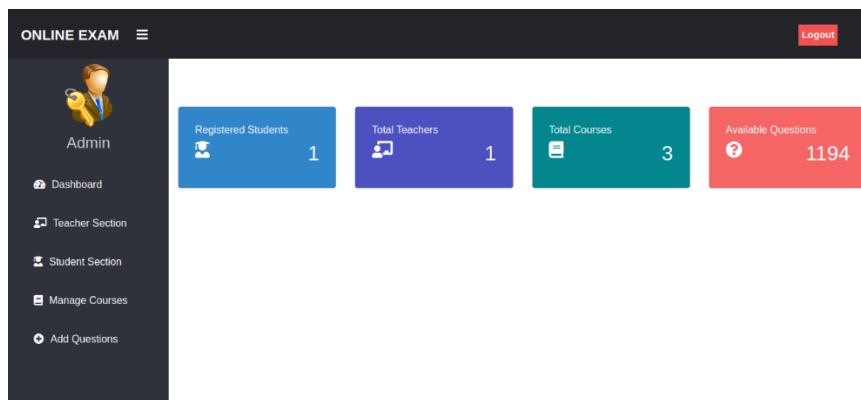


Figure-7 Admin Dashboard

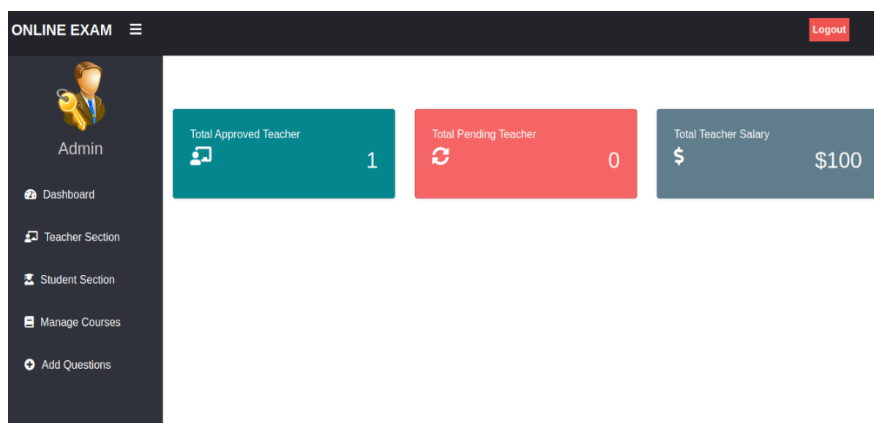


Figure-8 Admin Teacher Section

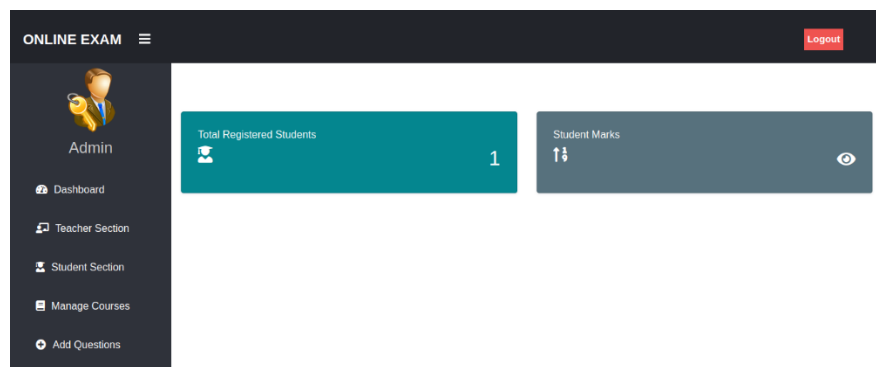


Figure-9 Admin Student Section

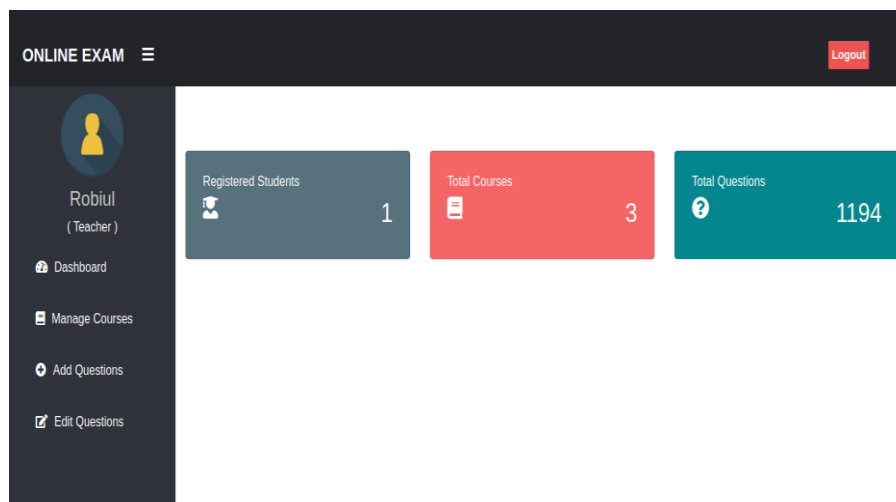


Figure-10 Teacher Dashboard

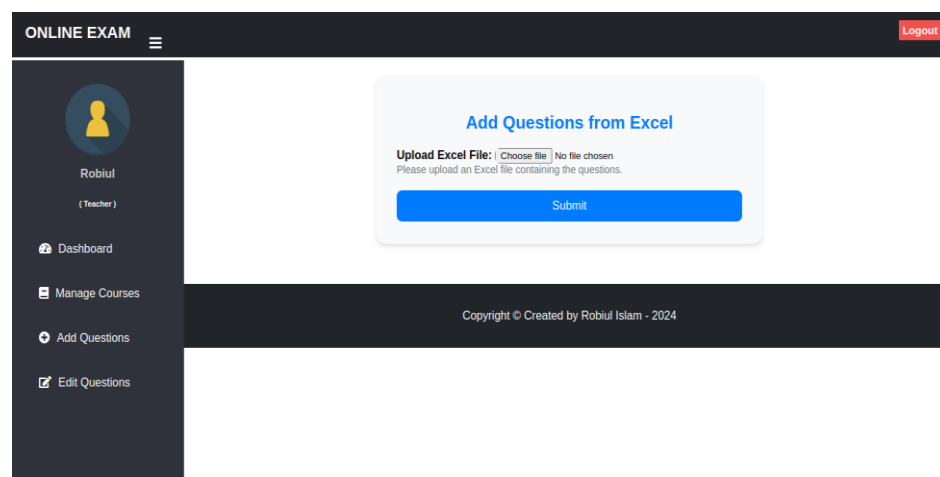


Figure-11 Teacher Add question Section

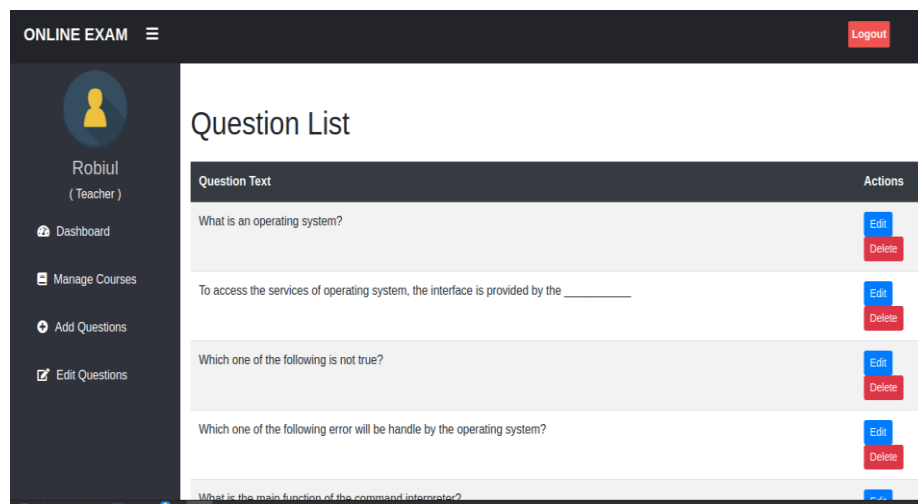


Figure-12 Teacher Edit Question Section

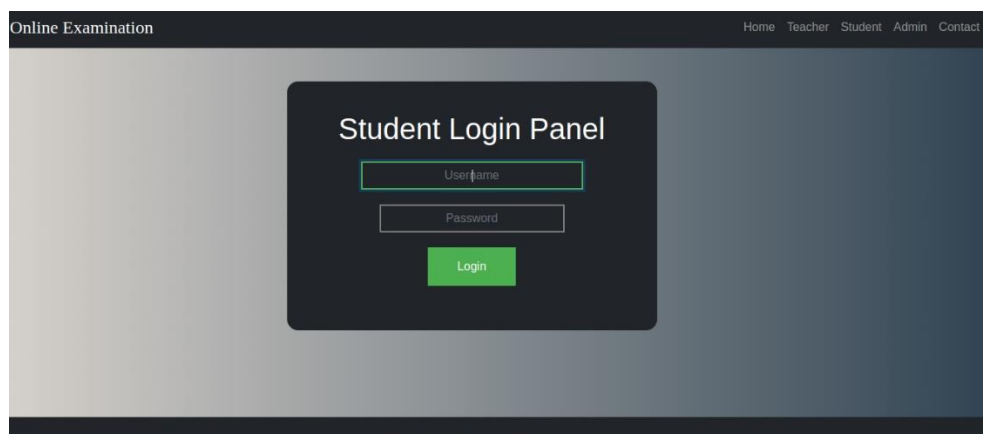


Figure-13 Student Login Panel

The screenshot shows the 'Student Signup Form' in the 'Online Examination' system. The header includes 'Online Examination' and navigation links: Home, Teacher, Student, Admin, and Contact. The form contains fields for First Name, Last Name, Contact Number, Address, Username, Password, and Profile Picture, along with a 'Sign Up' button.

First Name:  Last Name:

Contact:  Address:

Username:  Password:

Profile Picture:  No file chosen

Figure-14 Student SignUp page

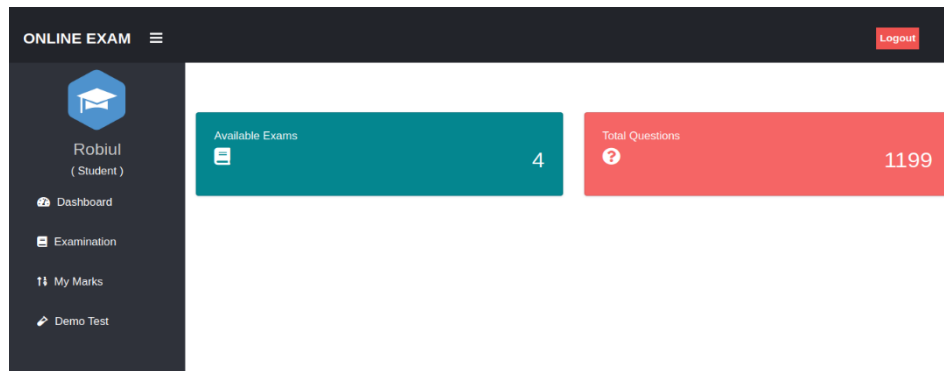


Figure-15 Student Dashboard

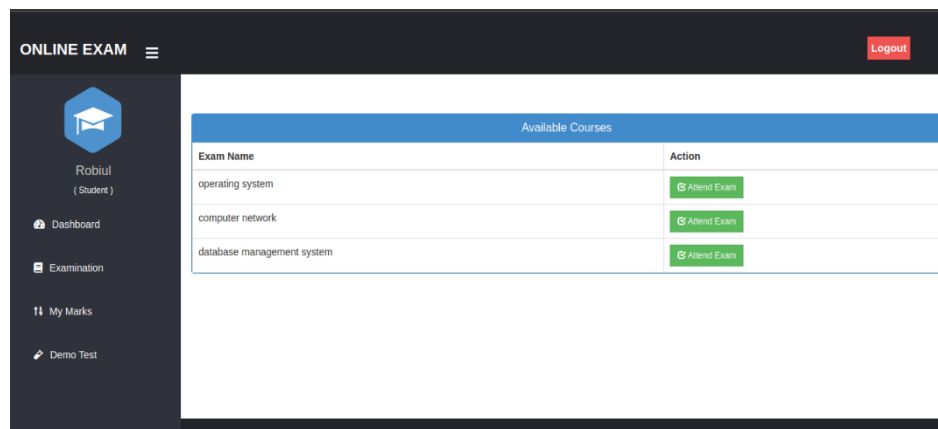


Figure-16 Student Examination Section

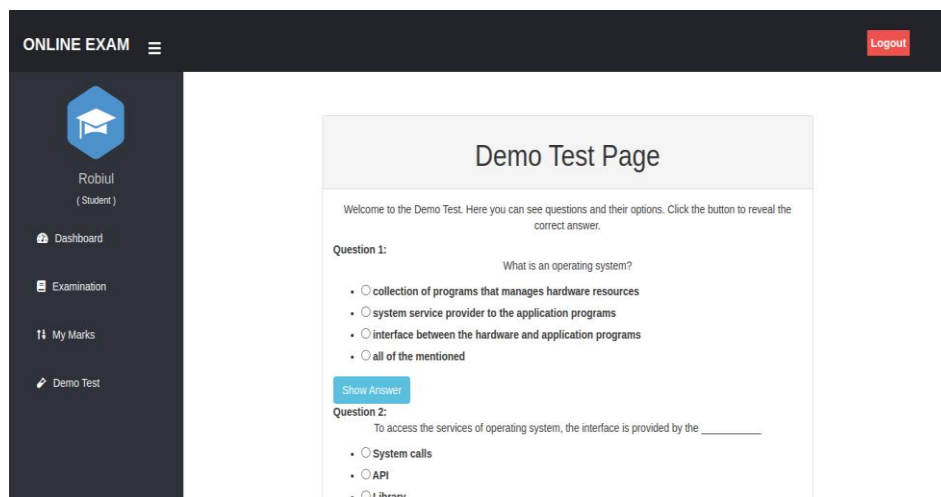


Figure-17 Student Demo Test page

| My Marks |             |                    |                          |                 |               |                      |
|----------|-------------|--------------------|--------------------------|-----------------|---------------|----------------------|
| Course   | Total Marks | Number of Attempts | Exam Date                | Correct Answers | Wrong Answers | Unanswered Questions |
| Stack    | 3           | Attempt 1          | May 30, 2024, 12:59 p.m. | 3               | 5             | 0                    |

Figure – 18 Student Analysis Page

## 7.5 Question Generation

It is necessary to create questions and answers in order to create an online Q&A management system. Students' learning and assessment experiences are enhanced by clear, varied questions and the accurate responses that correspond with them. Here, we discuss the content generation tactics in more detail.

Our initiative made advantage of publicly available educational resources to offer a broad range of queries and answers. We specifically used the fundamental computer science textbook (Operating System, Computer Networks, and Database Management System) and the NCERT textbooks. These resources ensure that the knowledge is authentic and of the highest calibre by providing comprehensive coverage of computer science-related topics.

Using AI tools like ChatGPT's GPT-4o model and Claude AI, I have generated a comprehensive set of questions and answers for the "Question Answer Assessment Web Portal." These tools leveraged natural language processing and machine learning algorithms to create a diverse and extensive question bank, covering various subjects like OS, DBMS and Networking. In total, I have generated 1194 questions.

| Available Courses          |                |             |        |
|----------------------------|----------------|-------------|--------|
| Course                     | Total Question | Total Marks | Action |
| operating system           | 425            | 425         |        |
| computer network           | 490            | 490         |        |
| database management system | 279            | 279         |        |

Figure-19 Question Dashboard

## 8. Conclusion and Future Scope

The "Question Answer Assessment Web Portal" project successfully developed a comprehensive and user-friendly platform for online assessments, incorporating key features like user management for registration and role-based access, exam management for creating and scheduling exams, automated grading for objective questions, and interfaces for manual grading of subjective questions. The system was optimized for performance and scalability, ensuring quick response times and the ability to handle a significant number of concurrent users, and focused on delivering an intuitive user experience. Comprehensive testing was conducted to ensure robustness and reliability. However, the system has limitations, such as supporting only multiple-choice questions, the time-consuming nature of manual grading for large classes, scalability constraints under very high loads, and user interface aspects needing improvement based on user feedback. Future enhancements could address these limitations by incorporating more diverse question types, automating grading processes further, optimizing scalability for peak periods, and refining the user interface for better navigation and customization.

To improve the "Question Answer Assessment Web Portal," several future enhancements are proposed. First, expanding the range of question types to include options like fill-in-the-blank, matching, and coding questions will make assessments more comprehensive. Next, utilizing natural language processing (NLP) and machine learning to automate grading for essay-type questions can significantly reduce the manual grading workload for teachers. Enhancing the system's scalability with advanced solutions like load balancing and distributed databases will ensure it can efficiently handle high user loads. Continuous improvements to the user interface, guided by user feedback, will enhance usability, accessibility, and customization options. Developing a mobile application will allow users to take and manage exams on the go, adding convenience. Integrating the portal with popular Learning Management Systems (LMS) such as Moodle or Canvas will create a seamless experience for institutions already using these systems. Lastly, upgrading the reporting and analytics capabilities will provide deeper insights into student performance and exam effectiveness, helping to identify areas for improvement.

# References

List of references and resources used in the development of the "Question Answer Assessment Web Portal":

[1] Django Project. (2024). Django Documentation. Retrieved from <https://docs.djangoproject.com/en/4.0/> [ Last accessed at May 30,2024 ]

[2] Bootstrap. (2024). Bootstrap Documentation. Retrieved from <https://getbootstrap.com/docs/5.0/getting-started/introduction/> [ Last accessed at May 30,2024 ]

[3] PostgreSQL Global Development Group. (2024). PostgreSQL 13 <https://www.postgresql.org/docs/13/index.html> [ Last accessed at May 30,2024 ]

[4] Python Software Foundation. (2024). Python 3.9 Documentation. Retrieved from <https://docs.python.org/3.9/> [ Last accessed at May 30,2024 ]

[5] Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation, University of California, Irvine. [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm) [ Last accessed at May 30,2024 ]

[6] MDN Web Docs. (2024). JavaScript. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [ Last accessed at May 30,2024 ]

[7] W3Schools. (2024). W3Schools Online Web Tutorials. Retrieved from <https://www.w3schools.com/> [ Last accessed at May 30,2024 ]