

# **Cyber Cloud Secure: A Cloud Hosted Web Application With Security**

A major project report submitted in partial fulfillment of the requirement  
for the award of degree of

**Bachelor of Technology**  
in  
**Computer Science & Engineering / Information Technology**

*Submitted by*  
**Dhruv Rastogi (201183)**  
**Vasu Yadav (201514)**

*Under the guidance & supervision of*  
**Dr. Nancy Singla**



**Department of Computer Science & Engineering and  
Information Technology**  
**Jaypee University of Information Technology, Wagnaghat, Solan - 173234**  
**(India)**

# Candidate's Declaration

I hereby declare that the work presented in this report entitled **Cyber Cloud Secure: A Cloud Hosted Web Application With Security** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Nancy Singla** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)

Student Name: Dhruv Rastogi

Roll No.: 201183

(Student Signature with Date)

Student Name: Vasu Yadav

Roll No.: 201514

This is to certify that the above statement made by the candidate is true to the best of our knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr. Nancy Singla

Designation: Assistant Professor (SG)

Department: CSE & IT

Dated:

# Acknowledgement

With immense gratitude, we extend our heartfelt thanks to the Almighty for His divine blessings, which have illuminated our path and enabled us to successfully complete the project – Cyber Cloud Secure: A Cloud Hosted Web Application with Security.

Our sincere appreciation goes to our esteemed Supervisor, Dr. Nancy Singla, Assistant Professor (SG) in the Department of CSE at Jaypee University of Information Technology, Wakhnaghat. Dr. Singla's profound expertise in the realms of cybersecurity, and machine/deep learning has been instrumental in guiding us through this project. We are deeply indebted for her tireless support, patient mentorship, constructive criticism, and unwavering encouragement. Her keen interest in our work has truly been a driving force behind the project's completion.

Our sincere thanks extend to all individuals, whether directly or indirectly, who played a role in the triumph of this project. We acknowledge the support of the entire staff, both teaching and non-teaching, whose timely assistance and facilitation greatly aided our endeavor.

In closing, we wish to recognize and appreciate the enduring support and patience of our parents. Their unwavering encouragement has been a source of strength throughout this journey.

With gratitude,

Dhruv Rastogi

(201183)

Vasu Yadav

(201514)

# Table of Content

<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Significance and Motivation of the Project Work	4
1.5 Organization of Project Report	4
<b>Chapter 2: Literature Survey</b>	<b>6</b>
2.1 Overview of Relevant Literature	6
2.2 Key Gaps in the Literature	8
<b>Chapter 3: System Development</b>	<b>10</b>
3.1 Requirements and Analysis	10
3.2 Project Design and Architecture	12
3.3 Data Preparation	16
3.4 Implementation	18
3.4 Key Challenges	44
<b>Chapter 4: Testing</b>	<b>46</b>
4.1 Testing Strategy	46
<b>Chapter 5: Results</b>	<b>50</b>
<b>Chapter 6: Conclusions and Future Scope</b>	<b>59</b>
6.1 Conclusion	59
6.2 Future Scope	60
<b>References</b>	<b>62</b>

# List of Figures

Figure 3.1: Steps of SDLC	12
Figure 3.2: 3-tier Architecture	14
Figure 3.3: Database Schema Design	15
Figure 3.4: Registration to Logout Flowchart	16
Figure 3.5: Process of Requirement Gathering	17
Figure 3.6: Spiral Model	18
Figure 3.7: Working of Java	20
Figure 3.8: Components of Java	21
Figure 3.9: Spring Framework Runtime	22
Figure 3.10: JWT Architecture	41
Figure 3.11: Working Of Password Encryption	41
Figure 3.12: Error Logging	43
Figure 5.1: Registration Page	50
Figure 5.2: Login Page	51
Figure 5.3: Profile Page	51
Figure 5.4: Add Category	52
Figure 5.5: Categories Page	53
Figure 5.6: Add Quiz	53
Figure 5.7: Quiz Portal	53
Figure 5.8: Add Questions	54
Figure 5.9: Admin Question Page	54
Figure 5.10: User Category Page	55
Figure 5.11: Quiz Manual Page	55
Figure 5.12: All Quizzes Page	56

Figure 5.13: EC2 Instance	56
Figure 5.14: System Logs	57
Figure 5.15: Docker Desktop	58
Figure 5.16: Ngrok Tunnel Status	58

# Abstract

In response to the evolving landscape of online education, the project, the Cyber Cloud Secure: A cloud hosted web application with security, offers a streamlined and secure platform for remote exams, effectively addressing the challenges exacerbated by the COVID-19 pandemic. With educational institutions transitioning to online learning, the demand for an efficient examination system has become paramount.

The project is meticulously designed to optimize the examination process by reducing paper evaluation time while upholding standards of accuracy and security. The urgency of adapting to remote learning environments underscores the necessity for a flexible system that empowers students to take exams from any location.

Leveraging a combination of ReactJS and Spring Boot, our project ensures a seamless and reliable testing experience. With their authorized credentials, students can access the exam interface quickly and securely. Multiple-choice questions are incorporated into the examination structure to introduce a diverse approach to assessment.

The project's implementation on cloud infrastructure, which helps to increase the benefits of scalability and accessibility, emphasizes its significance even more. The cloud technology enables effective resource management, gives top performance even during moments of high utilization.

In the framework of today's world-wide educational landscape, the concept essentially addresses the urgent need for a safe online examination platform. With the careful blending of Spring Boot and ReactJS with a cloud deployment, the project not only addresses current issues but also lays the groundwork for a revolution in the field of online learning evaluations.

# Chapter 1: Introduction

## 1.1 Introduction

The introduction of online learning has completely altered the evolving field of education, necessitating innovative approaches to address the issues raised by the current educational landscape [1]. The COVID-19 epidemic has expedited the adoption of remote learning, necessitating that educational institutions change their reliance on traditional testing methods and embrace more adaptable and secure alternatives.

The project aims to provide a relevant answer to these issues and attempts to offer a safe platform for remote tests. Recent research indicates that throughout the past two years, the use of digital learning platforms has grown significantly by 72% globally [2]. This modification underscores the pressing need for an upgraded assessment system that can meet the needs of a classroom that is becoming more and more dependent on digital learning.

The project's primary goal is to streamline the inspection procedure while tackling the difficulties made worse by the pandemic. According to a recent Johnson Education Research survey, 87% of educational institutions reported having trouble administering traditional paper-based exams during lockdowns [3]. This indicates that traditional paper-based exams faced significant disruptions. This emphasizes the need for an examination system that is more effective and flexible.

The fact that pandemic-related closures caused disruptions to the education of over 1.6 billion students worldwide underscores the urgency of adjusting to remote learning environments [4]. The online exam portal is purposefully made to meet this need for an adaptable system that allows students to take tests from anywhere in safety and convenience.

The project makes use of Spring Boot in conjunction with ReactJS to guarantee a dependable and smooth testing process. Exam processing times were 30% faster for institutions using integrated technologies like ReactJS and Spring Boot than for traditional methods, according to a TechEd Insights comparison study [5].



Furthermore, Gartner statistics showing a 45% increase in the adoption of cloud solutions in the education sector [6] support our project's deployment on cloud infrastructure. Because of this cloud integration, resource management is made easier and optimal performance is maintained even during periods of high usage.

Essentially, the online exam portal is a proactive move towards influencing the direction of online education assessments rather than just a reaction to the problems of the present. The project, which is supported by pertinent studies and statistics, fits with the changing requirements of educational institutions by offering a safe, effective, and flexible way to deal with the challenges of contemporary education.

## **1.2 Problem Statement**

The rise in online learning in today's educational environment has created a demand for sophisticated testing systems that can easily adjust to the needs of distance learning. However, given the COVID-19 pandemic and the larger move towards digital education, it will be difficult for the current traditional examination procedures to offer a flexible, secure, and effective environment for administering exams.

The main problems are caused by the shortcomings of traditional paper-based testing systems, which have been shown to be unreliable and ineffective. The need for a more efficient procedure becomes apparent when educational institutions find it difficult to administer and grade exams in the event of unplanned disruptions, like lockdowns or other emergencies. The pressing need to support remote learning environments and enable exams from different locations highlights the need for a flexible and safe examination system even more.

In addition, with the global adoption of digital learning platforms by educational institutions witnessing a notable upsurge, there is an increasing need for exam solutions that are in step with these technological developments. Conventional assessment techniques are unable to offer the necessary flexibility and responsiveness to meet the changing demands of educational institutions and students navigating the challenges of remote learning.

The shortcomings of the existing testing methods highlight the need for a novel strategy. By introducing a sophisticated, safe, and effective platform that makes use of modern technologies

like ReactJS and Spring Boot and is strategically deployed on cloud infrastructure, the project seeks to address these issues. With a solution that not only addresses the present issues but also lays the groundwork for future online education assessments, this initiative aims to transform the examination process.

### **1.3 Objectives**

The overarching goal of the project is to replace the traditional pen-and-paper examination format with a more efficient and stress-free method of evaluation that benefits both examinees and examiners. It is anticipated that the shift to online tests will spread over the globe and give educational organizations a reliable and easy means of evaluating the marks.

The following are the main objectives of the project: -

1. **Activate the online exam environment:** Give students a friendly and easy to use user interface so that they can take part in both objective as well as subjective questions.
2. **Self-registration:** With the sign-up page, students can easily register themselves by just entering their details like name, phone number, username and password.
3. **Automated Assessment of Objective Examinations:** To ensure a quick and easy evaluation process, use the Online Examination System to examine candidates during online objective tests.
4. **Automated candidate response verification:** Include an automated system to confirm and validate the responses that candidates choose to provide during the test.
5. **Entire integration of online exams:** Create a comprehensive online examination system that optimizes the efficiency of the examination process by significantly reducing the need for paperwork for examiners and examinees alike.
6. **Encourage Examiners to Create Their Own Question Papers:** By using the Online Examination System, give examiners the ability to easily create and prepare question papers.
7. **Integrating the Cloud to Streamline Procedures:** Enhance accessibility, scalability, and efficiency by integrating cloud infrastructure into the online examination system. This will maximize the examination experience for all parties involved.

## **1.4 Significance and Motivation of the Project Work**

The significance of this project is found in filling in the critical gaps in the framework of conventional exams and bringing educational methods into line with the changing needs of the digital age. An advanced examination platform that not only ensures the security and integrity of assessments but also adjusts to the dynamic and flexible nature of remote learning is desperately needed as educational institutions move more and more towards online learning.

The impetus behind this project is the realization of the difficulties made worse by the COVID-19 pandemic and the larger shift in teaching approaches. Disruptions to traditional examination procedures make it more difficult to accurately and promptly evaluate students. The online examination portal was developed with the intention of revolutionizing assessment practices by providing a solution that not only addresses present issues but also lays the groundwork for future developments in online education assessments.

## **1.5 Organization of Project Report**

The project report provides an in-depth review of the initiative from start to finish, organized in an intricate way. With a smooth and succinct narrative, each chapter develops methodically, guiding readers through the project's nuances and unique contributions.

### **Chapter 1: Introduction**

The introductory chapter establishes the framework for the project by presenting the central theme, addressing recognized obstacles, delineating goals, and highlighting the importance and inspiration behind "Cyber Cloud Secure." Finally, the section offers a well-organized roadmap for reading the next few chapters.

### **Chapter 2: Literature Survey**

This chapter delves into scholarly investigation and presents a succinct summary of pertinent literature while incorporating ideas from a variety of sources. It strategically highlights the most important gaps in the body of current knowledge in order to identify and explain the project's distinctive contributions.

### **Chapter 3: System Development**

This crucial chapter describes the steps involved in the project's journey: requirements and analysis at the outset, design and architecture, data preparation, and finally project implementation using snippets of code and skillful problem-solving techniques.

### **Chapter 4: Testing**

The critical testing phase is covered in detail in the fourth chapter, along with the tools that were selected, the nuances of the test cases, and the results. It guarantees a comprehensive analysis of the functionality and dependability of the project.

### **Chapter 5: Results**

Project outcomes are the main focus of this chapter, which also offers a perceptive look at the results and a focused display of the results.

### **Chapter 6: Conclusions and Future Scope**

The final chapter highlights the contributions of the project, summarizes the main conclusions, and notes its limitations. By outlining a wide range of possible future developments, it opens up the project's domain for future advancements and expansions.

# Chapter 2: Literature Survey

## 2.1 Overview of Relevant Literature

In this section, we delve into the current body of research focused on the optimization of online examination systems, exploring advancements in user interface design, security measures, and automated evaluation mechanisms.

Rahat Ibne Sattar et al. [6] significantly improved the security of the online examination system by putting in place a blockchain-based framework. By utilizing IP-based login, blockchain technology, and authentication methods, the framework showed improved resistance to fraud and unauthorized access. This demonstrates how well these combined methods work to strengthen the security of online exams.

Blockchain technology was used by Jain et al. [7] to introduce a revolutionary online examination system that is enabled by smart contracts. Their strategy involved utilizing SHA-256, a potent cryptographic algorithm, and Blockchain to create Decentralised Applications (DAApps). In particular, the study made use of the Ethereum Blockchain platform, incorporating Smart Contracts to improve the security of the testing system. This work serves as an example of how cutting-edge Blockchain features can be strategically incorporated to strengthen the dependability and security of online exams.

Slusky and Ludwig [8] carried out a perceptive investigation in the field of cybersecurity for online proctoring systems. Their work addresses important facets of system fortification, with a focus on endpoint security, VPN, and virtual machines. In order to improve the overall security posture of online proctoring, OPS notably achieved full automation by integrating hybrid and AI-enabled systems. The cited paper provides insightful information about the methods used and results obtained for online proctoring system security.

The work by Singh, Shirke, and Shette [9] has made a significant contribution to the field of online examination systems. Developed with PHP, Python, Ruby, and SSL, this framework provides a secure and adaptable platform. Its design allows students to access the platform from any location with an internet connection, catering to the needs of distance learning and exams. This offers valuable insights into utilizing various technologies to build a secure and easily accessible online examination system.

Developed by Conejo, Guzmàn, and Trella [10], leveraging advanced techniques such as Computer Adaptive Testing (CAT), Item Response Theory (IRT), Domain model, and Student model, SIETTE surpasses traditional paper-and-pencil tests in providing more accurate and efficient assessments. They contribute to the field by showcasing the effectiveness of incorporating sophisticated technologies in the assessment process.

The creation of a powerful online examination system [11] with ASP.NET, C#, and Microsoft SQL Server is notable in the quest to advance online examination systems. They intended to reduce the laborious manual process involved in administering traditional paper-based exams. An examination process that is quicker and more trustworthy is the expected outcome. This work demonstrates how well technologies can be integrated to improve the effectiveness and dependability of examination processes.

Regarding online testing, Fagbol, Adebisi, and Oke [12] developed an excellent Computer Based Test (CBT) System for University Academic Enterprise Examination. This system was developed using Python, Microsoft SQL Server 2008, and Microsoft Visual Studio 2012. It was thoroughly tested by 250 volunteer users. High ratings for robustness, flexibility, ease of use, and reliability were obtained from the evaluation. Their work demonstrates how different technologies can be successfully combined to create an effective and user-friendly online examination system.

Developed by H. Lu and Y. Hu [13] this system, which was created with J2EE, MySQL, and JSP, is an example of a well-considered technological integration. This system's implementation improves the way exam-related data is managed.

Arvind Singh, Niraj Shirke, and Kiran Shette's [14] developed an inventive portal and executed successfully, utilizing both PHP and MySQL, and it was examined in a university. The portal showed how easy and user-friendly it is to access the examination.

M. Z. Rashad, M. S. Kandil, A. E. Hassan, and M. A. Zaher's [15] developed an inventive exam system and achieved accomplishment in the field of web-based exam management systems. This system is built with the help of various technologies and techniques like PHP, HTML, MySQL and AJAX. The system capability of gathering the responses quickly and generating the exam reports based on the submissions was incredible.

Wei L., Cong Z., and Zhiwei Y. [16] developed a creative concept in the area of identity verification for online testing systems known as fingerprint-based identity authentication. This

system, which used a fingerprint, a database of approved fingerprints, and a browser/server framework, showed an amazing ability to accurately authenticate users. This shows how effectively fingerprint-based technology is built for identifying authentication in online tests.

Ayo, Charles, Akinyemi, I.O., Adebisi, Ayodele, and Ekong, Uyinomen [17] examined the application of Microsoft Visual Studio, Microsoft SQL Server, and Java as crucial technologies in assessing the potential for the e-examination developed in Nigeria. These technologies help to improve the examination system and provide improvements in the security of test results. By improving the integrity and privacy of examination results, the implementation aims to make the examination process more secure and dependable.

## **2.2 Key Gaps in the Literature**

A review of the literature reveals some significant gaps and limitations that are present in different related works. Although it seems promising, putting a blockchain-based system into place requires specialized knowledge and resources due to its technical complexity. The smooth operation of blockchain-based solutions is further hampered by issues like expensive fees during network congestion and slow transaction confirmation times.

Even though they are widely used, endpoint-based security and detection methods have limits when it comes to dealing with behavior-based cheating and the possible intrusion of new technologies, which raises privacy concerns for students. Technical difficulties that online exam systems face include network outages, software bugs, and server outages. These problems can cause disruptions to exams and jeopardize the validity of the assessment procedure.

The success of the SIETTE system depends on the presence of a sizable item bank, which may restrict its use in settings with constrained resources. Such systems require a large financial investment for their development and upkeep, which includes infrastructure, software, and continuing support and maintenance.

It was necessary to conduct additional testing to determine scalability and performance in real-world scenarios because a number of the evaluated systems, including the blockchain-based

one, were evaluated with comparatively few users. Concerns about student data security and privacy highlight how crucial it is to strictly adhere to data protection laws.

Significant weaknesses in these systems also include their scalability and reliability, which call for thorough analysis and testing beyond the small user samples used in the preliminary evaluations. Finally, problems like inadequate technology support and infrastructure serve to highlight the gaps in the literature that already exist.

This section highlights the need for future research to address these gaps, fostering the development of more robust, scalable, and privacy-conscious online examination systems.



# Chapter 3: System Development

## 3.1 Requirements and Analysis

The online examination system's major goal is to automate the process of distributing questions and administering exams. Furthermore, the system intends to automatically grade the papers using a specified answer key. To accomplish this, excellent test question management, automated paper generation, and fast paper retrieval are required.

### Functional Requirements:

This system supports two types of user modules: Administrator/Faculty and Student. Each session is tailored to the specific needs of administrators and students, assuring clarity and preventing confusion. The following are the functionalities of both modules:

#### ADMIN:

- They must register.
- Login.
- Categories can be added, deleted, or updated.
- Quizzes can be added, deleted, or updated.
- Questions can be added, deleted, or updated.
- View the outcomes.

#### USER:

- They must register.
- Login.
- View the categories that are accessible.
- Examine and take accessible quizzes.
- View and try to answer questions.
- View the outcomes.

### **Non-Functional Requirements:**

**Usability:** assesses the ease of use of an interface, including memorability, learnability, and satisfaction. The user-friendliness of the software interface is prioritized, ensuring that users of all backgrounds can quickly navigate and comprehend the system.

**Reliability:** measures the system's ability to execute consistently on demand without degradation or failure across several platforms.

**Integrity:** In the context of data, integrity refers to consistently doing the right thing. It places a premium on data accuracy, consistency, and completeness, all of which are critical components of system security.

**Performance:** is an important non-functional criterion that includes characteristics such as resource requirements, reaction time, and transaction rate. These requirements determine how the system should work.

### **SDLC Methodology**

The Software Development Life Cycle (SDLC) [18] is a methodical procedure meant to ease the creation of high-quality as shown in Figure 3.1, low-cost software in the shortest possible time frame. The SDLC's primary goal is to create great software that not only meets but exceeds all customer expectations and needs. This method of software development entails a structured plan divided into discrete stages or phases, each with its own set of operations and deliverables.

Following the SDLC closely increases development speed while reducing potential project risks and expenses associated with alternative production processes. The SDLC acts as a guiding framework for the efficient and successful development of software solutions while guaranteeing that the final product meets the needs of the customer.

### **Need of SDLC**

If deadlines are not reached, executing a project without a well-defined action plan can lead to disaster and, eventually, project failure. To ensure the seamless progression of the whole development cycle, every aspect, from resource allocation to deployment, must follow a defined pipeline. This requirement gave rise to the Software Development Life Cycle (SDLC), which, after achieving substantial success, became widely implemented. The SDLC is critical

because it provides a systematic approach that is consistent with the whole development process. Its success is based on completing each phase to a high standard while fulfilling customer objectives for cost, time, and efficiency. The SDLC's major purpose is to ensure that the development cycle runs smoothly and that the product is of high quality. The following figure 3.1 shows the steps involved in SDLC.

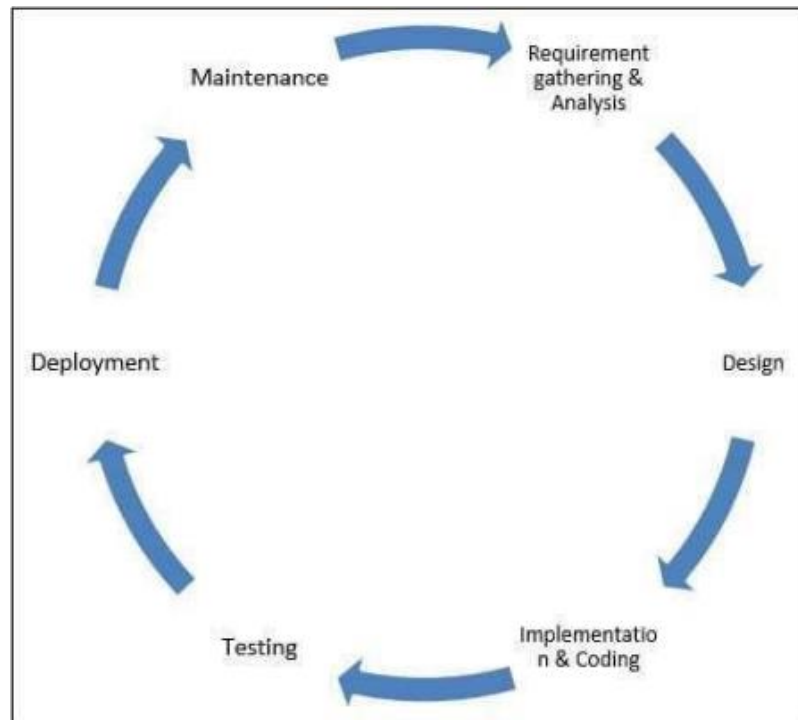


Figure 3.1: Steps of SDLC

### 3.2 Project Design and Architecture

In the world of software engineering, multi-tier architecture, often known as n-tier architecture, is a client-server architecture that splits display, application processing, and data management functions deliberately. This separation is conceptual, ensuring a clear difference between the system's many parts. Consider an application that uses middleware to handle data requests, resulting in a multi-tier design that manages communication between a user and a database efficiently. The three-tier design is the most common type of multi-tier architecture, and it is widely used in a variety of software applications.

### **Three-tier Architecture**

Figure 3.2 shows three-tier architecture [19] which is also called client-server architecture and involves separate and independent development and provisioning of the user interface, functional process logic (or “business rules”), and computer data storage and access. Such units are generally situated in different locations. This is a form of software organization architecture model.

The three-tier architecture goes further by allowing independent upgrading or replacement of each tier at a time. The ability to respond to fluctuations in demand and technological progress makes it adaptable. To sum up, the three-tier design facilitates the modularity, scalability, and flexibility of the software by offering a systematic approach.

#### **Presentation Tier**

Its user interface is its application’s presentation tier. The topmost layer that forms a barrier is the user interface. Generally, the primary role of a presentation tier is to show information for customers and gather data from them. Such a tier could be provided with different technical means, such as HTML, CSS, and JavaScript in case of web application, and native in case it would refer to desktop or mobile application.

#### **Application Tier**

At the core of the application is the application tier. It handles all the application’s business logic. This includes tasks like verifying data values, calculating results, and saving information in memory. The application tier usually connects with the presentation layer in order to collect data from the user and show the processed output results from the program. It can also engage the data tier for the purpose of either retrieving or changing information.

#### **Data Tier**

Data tier will store and keep the data consumed by an application. These involve activities such as generation, reading, updating, and removal of data. The data tier is implemented by using a database management system (DBMS) such as MySQL, PostgreSQL, or Oracle.

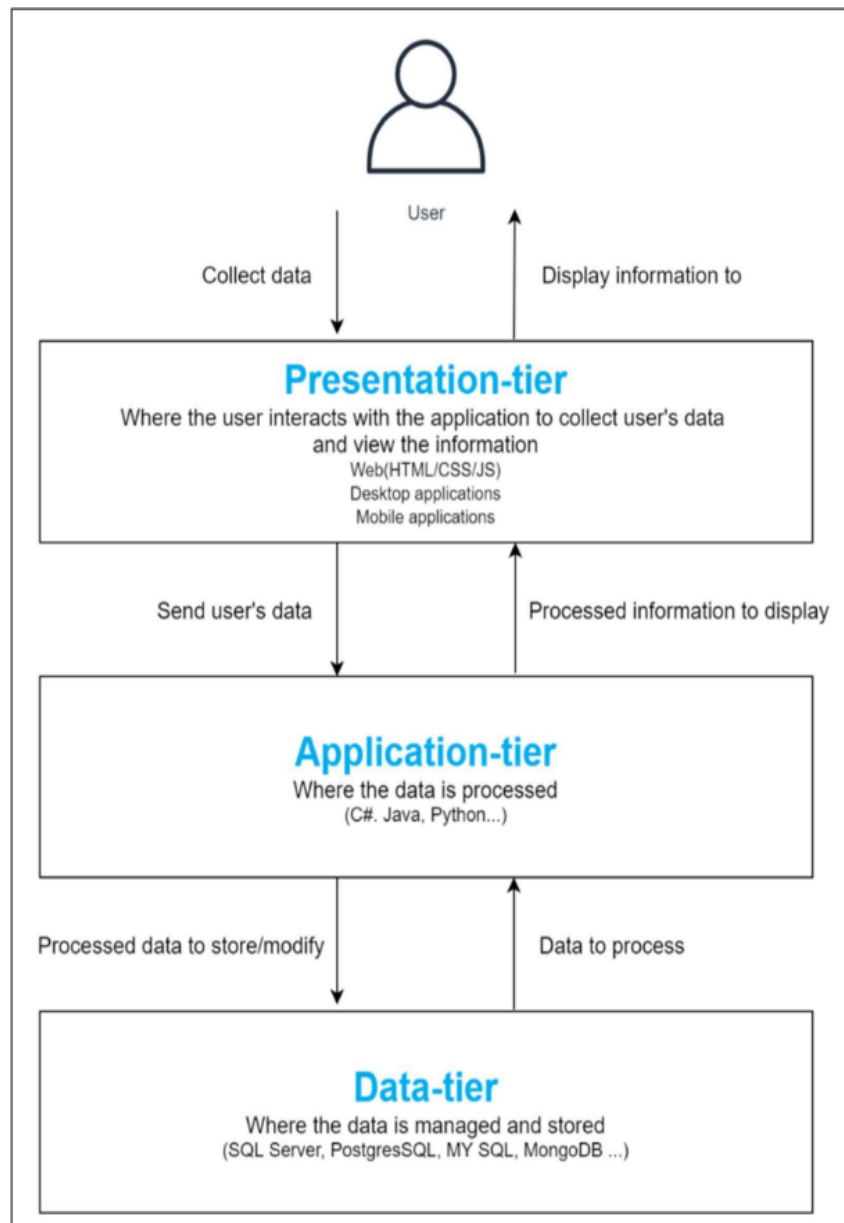


Figure 3.2: 3-tier Architecture

### Database Schema

Roles define user rights in the Cloud Web application database schema, with user roles allocated to specific users. Users are linked to quizzes and questions that are organized into topic areas. Quizzes yield quiz results, resulting in a cohesive interaction in which users with certain roles traverse through categories, attempt quizzes, and receive full responses in this integrated database structure, as depicted in Figure 3.3.

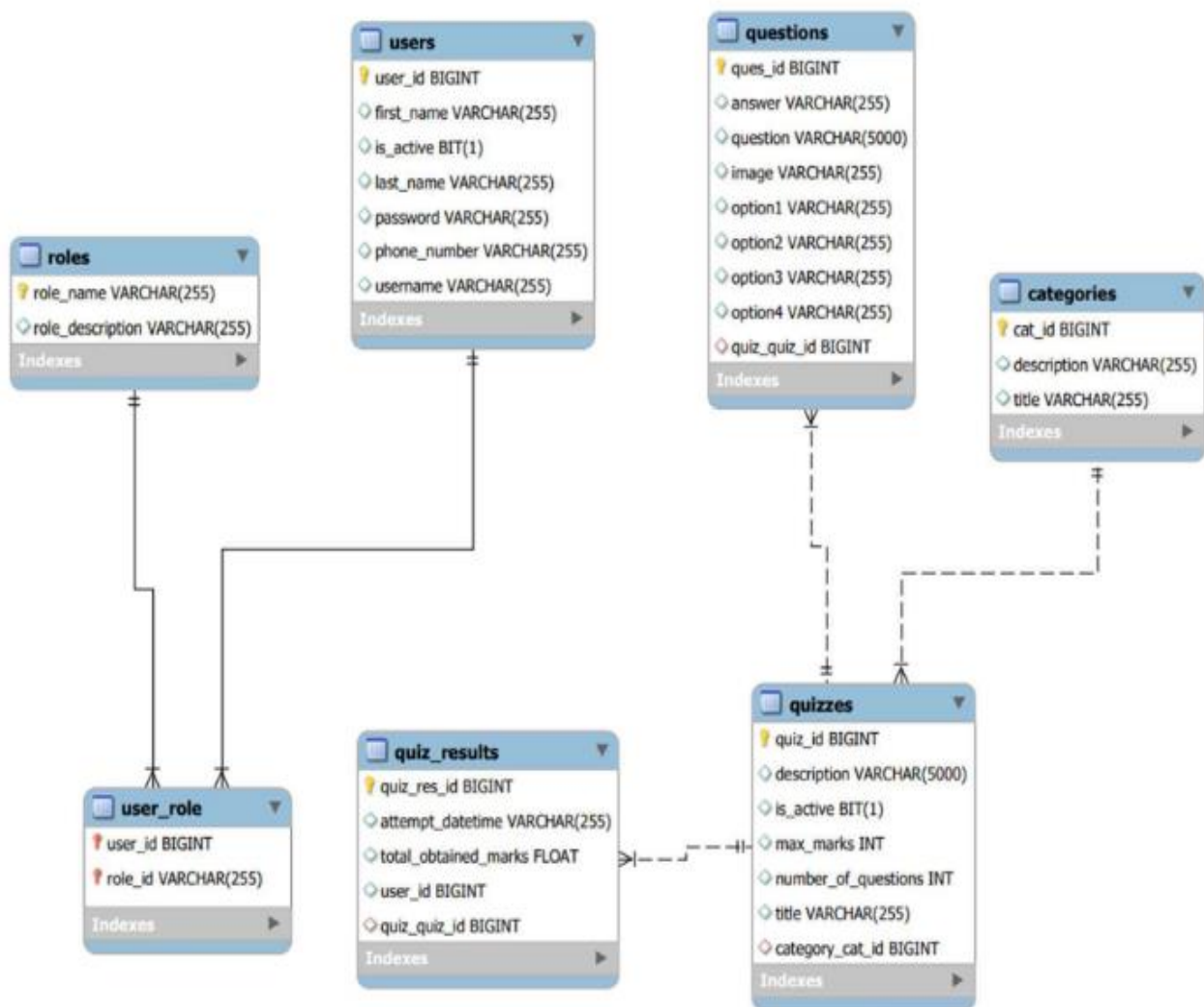


Figure 3.3: Database Schema Design

### Flow Chart

The Cloud Web application flowchart begins with user registration or login. When a user logs in, the system determines whether or not the user is an administrator. If yes, the administrator gains access to actions such as adding, editing, or deleting categories, quizzes, and questions, as well as examining results. If the user is not an administrator, they can only access categories, quizzes, questions, and results, resulting in a more streamlined and role-specific experience as depicted in Figure 3.4.

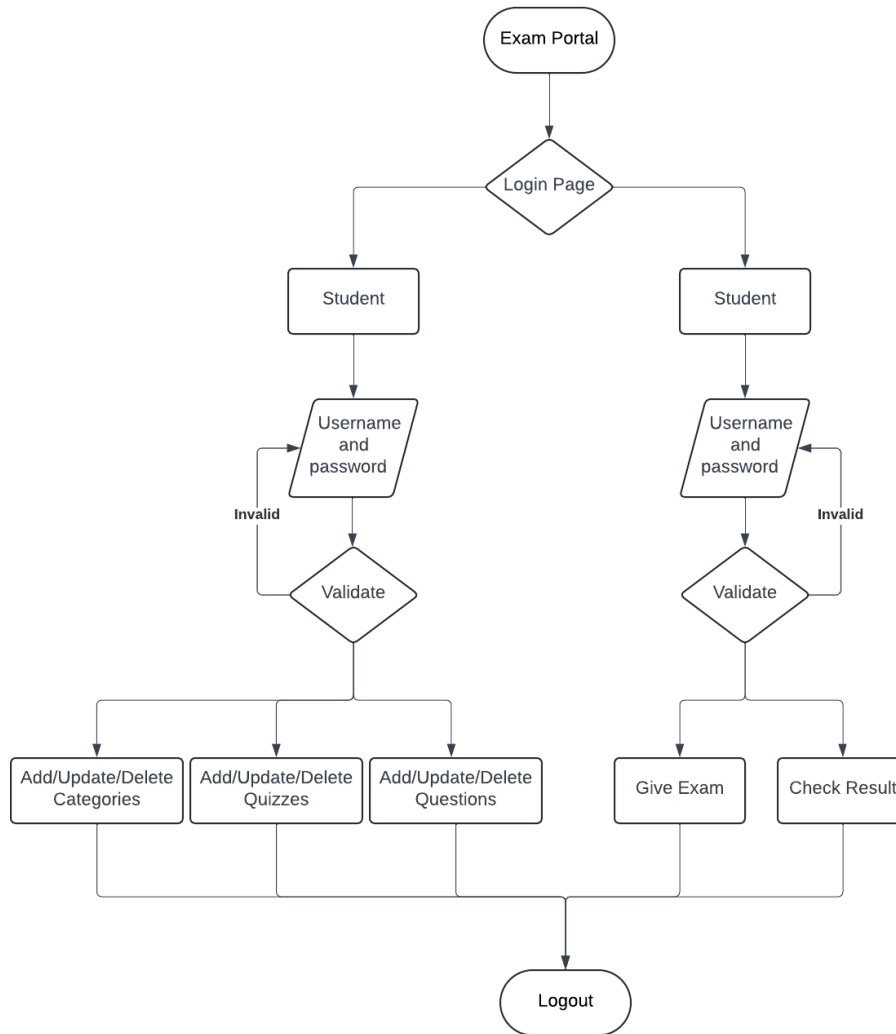


Figure 3.4: Flow chart of the process from registration till logout

### 3.3 Data Preparation

Figure 3.5 demonstrates requirement gathering [20], which is part of the software development involving asking key questions from the customer to facilitate the creation of a solution that will meet their expectations. In this stage, it involves the business analyst and project manager to ensure that all the uncertainties are resolved and attain full understanding of what the product would achieve for its targeted audience. It is vital to first have a full comprehension of the product before starting.

This is followed by planning, which addresses many issues in project and product management including resourcing allocation, project scheduling, cost estimation, etc. It comprises several groups of people like customers, sales representatives, experts and developers whose roles are

to give information which is transformed into full-fledged software requirements. Additionally, this segment involves specifying the resource needs and computing the expenditure related to the following phases in software development.



Figure 3.5: Process of Requirement Gathering

### Spiral Model

The Spiral Method, developed by Barry Boehm in his 1986 paper "A Spiral Model of Software Development and Enhancement," deviates from typical software development methodologies. It works as a meta-model, prioritizing the evaluation of a project's risk profile and then recommending an approach that combines aspects from popular techniques such as Iterative and Waterfall. Unlike the inflexible, homogenous methodologies of the time, this model emphasizes adaptability based on the unique characteristics and hazards inherent in each software development project.

What set the Spiral Method apart from its contemporaries was its recognition of the dynamic and unpredictable nature of software development projects. Unlike the rigid, one-size-fits-all frameworks prevalent in the industry, Boehm's model championed adaptability and flexibility. It acknowledged that each project possesses its own unique set of challenges, complexities, and uncertainties, which necessitate a customized approach to mitigate risks effectively.



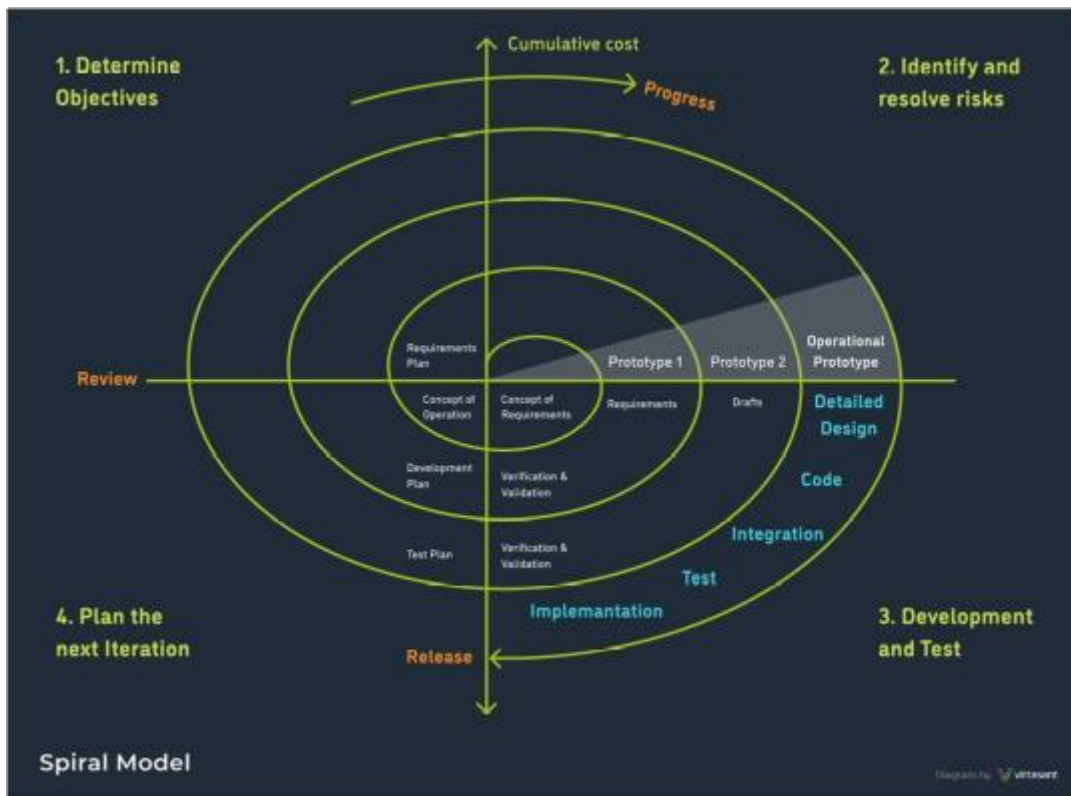


Figure 3.6: Spiral Model

In the initial phase, the four-quadrant model for software development begins with gathering customer needs and setting objectives. This is followed by an investigation of possible alternatives for the next phase. The second quadrant involves systematically evaluating solutions, selecting the best one, and identifying and strategically resolving associated risks, culminating in prototype development.

The third quadrant is concerned with the active development and testing of selected features, which will culminate in the release of the next software version. Customers critically analyze the software version in the fourth quadrant, affecting the planning for the next phase. Throughout the software development lifecycle, this iterative and adaptive strategy provides ongoing improvement and responsiveness.

### 3.4 Implementation

The combination of Spring Boot and React.js in the implementation of the Secure Web application capitalizes on the strengths of both technologies to create a robust and user-friendly platform. Spring Boot, a Java-based framework, offers the system with a robust backend infrastructure that ensures easy integration, scalability, and quick data processing. It makes it

easier to incorporate crucial features like user registration, automatic assessment, and safe data processing. React.js, a JavaScript toolkit for creating user interfaces, is crucial in generating an intuitive and dynamic user experience on the frontend. The usage of React.js allows for the creation of a responsive and engaging user interface, making the transition from traditional to online tests easier for both students and examiners. React.js modular and component-based architecture complements the Secure Web application's iterative and dynamic nature. Maintaining the integrity of the system and addressing any vulnerabilities will mostly depend on regular upgrades and security measures [21].

The Spring Boot backend is in charge of handling business logic, managing data persistence, and ensuring the examination process's security and integrity. Using Spring Boot, the system allows features such as automatic assessment of objective examinations, candidate response verification, and examiner authoring of comprehensive question papers.

### **Technologies Used:**

#### **Java**

Sun Microsystems introduced Java in 1995 as both a programming language and a computing platform. It has evolved significantly over the years, becoming a cornerstone in today's digital world by serving as a dependable foundation for countless services and applications as shown in Figure 3.7. Java is still used in the development of new and creative products, guaranteeing that they are ready for the digital future. Although many modern Java applications merge the Java runtime and application seamlessly, there are still programmes and websites that require a desktop Java installation for proper functionality. The website in question, Java.com, is specifically created for users who use Java for desktop applications, particularly those aiming Java 8. It is designed for people who still need Java. Developers and users interested in learning Java programming are urged to visit the dev.java website. Business users looking for further information can visit [oracle.com/java](http://oracle.com/java). In essence, Java continues to shape the digital environment, and while it is effortlessly incorporated into many apps, there are times when a standalone desktop Java installation is required.

## Working of Java

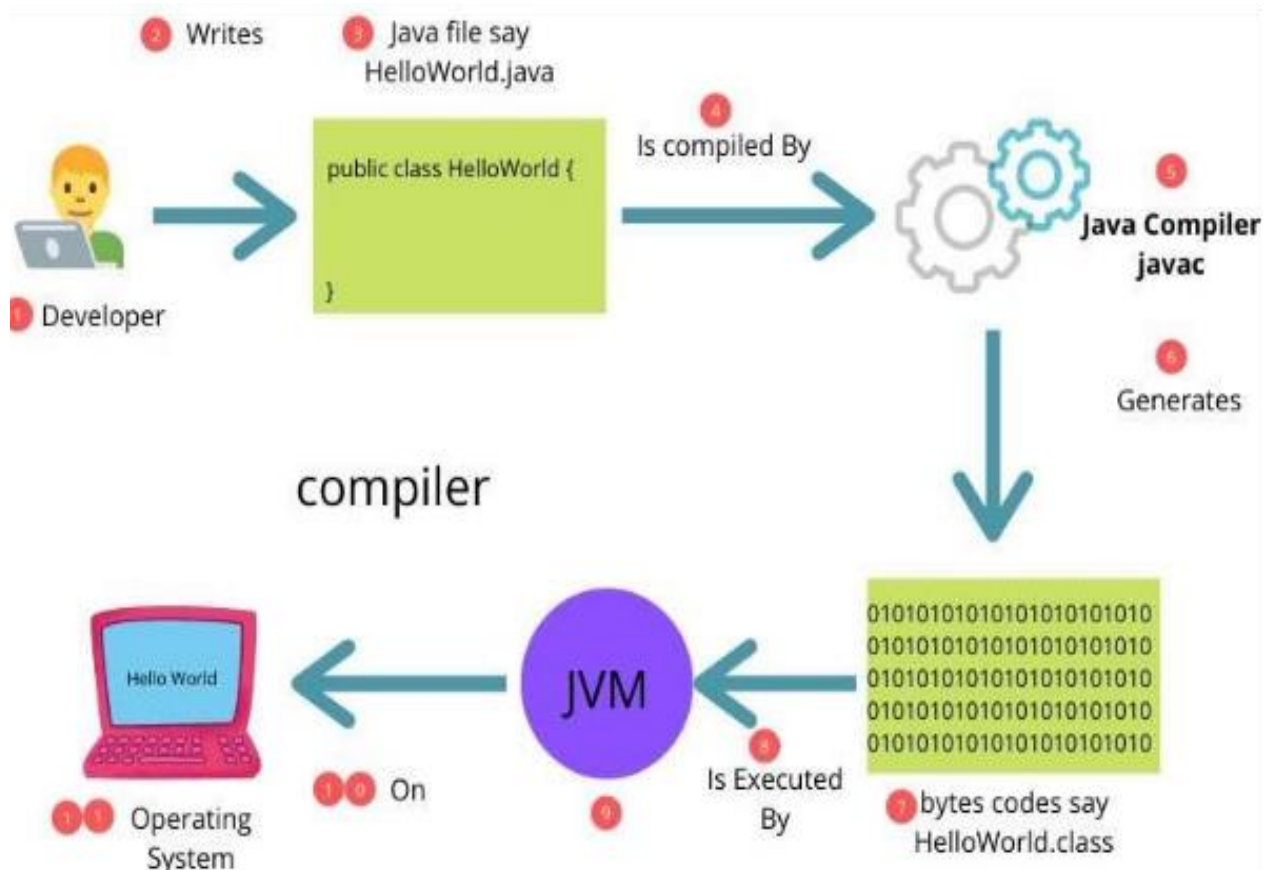


Figure 3.7: Working Of Java

The Java software platform is made up of three major components: the Java Virtual Machine (JVM), the Java API, and an extensive development environment. The Java Virtual Machine (JVM) is in charge of parsing and executing (interpreting) Java bytecode. The Java API includes a large number of libraries that cover fundamental objects, networking, security functions, XML creation, and web services. When the Java language and its software platform are coupled, they produce a strong and well-established technology that is well suited for corporate software development. This integrated system gives developers a rich collection of tools for developing sophisticated and dependable software applications as shown in Figure 3.8.

## Components Of Java

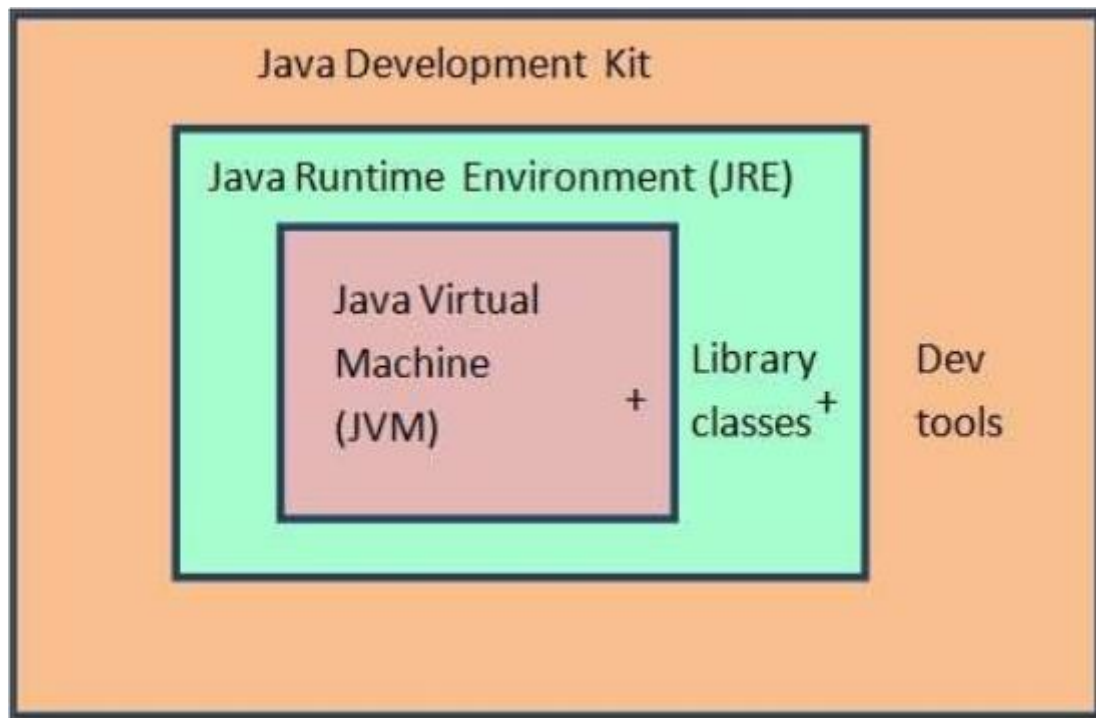


Figure 3.8: Components Of Java

### Spring Boot

The Spring Framework provides strong infrastructure support for Java applications as shown in Figure 3.9, simplifying development by addressing underlying difficulties. It enables developers to focus on application logic rather than complex infrastructure concerns. Spring allows developers to build applications using POJOs while smoothly integrating enterprise services. This functionality is available in both the Java SE and Java EE environments.

Spring provides significant benefits to developers by simplifying numerous elements of development. Creating HTTP endpoints without Servlet API complexities, developing message handlers without direct JMS API interaction, and transforming Java methods into management processes without navigating the JMX API are some examples.

Java Spring Boot, a Spring Framework extension, speeds up the creation of web apps and microservices through autoconfiguration, an opinionated approach to configuration, and the ability to create standalone applications. These capabilities work together to give developers a streamlined tool for configuring and configuring Spring-based applications, reducing the need for substantial manual configuration and setup.

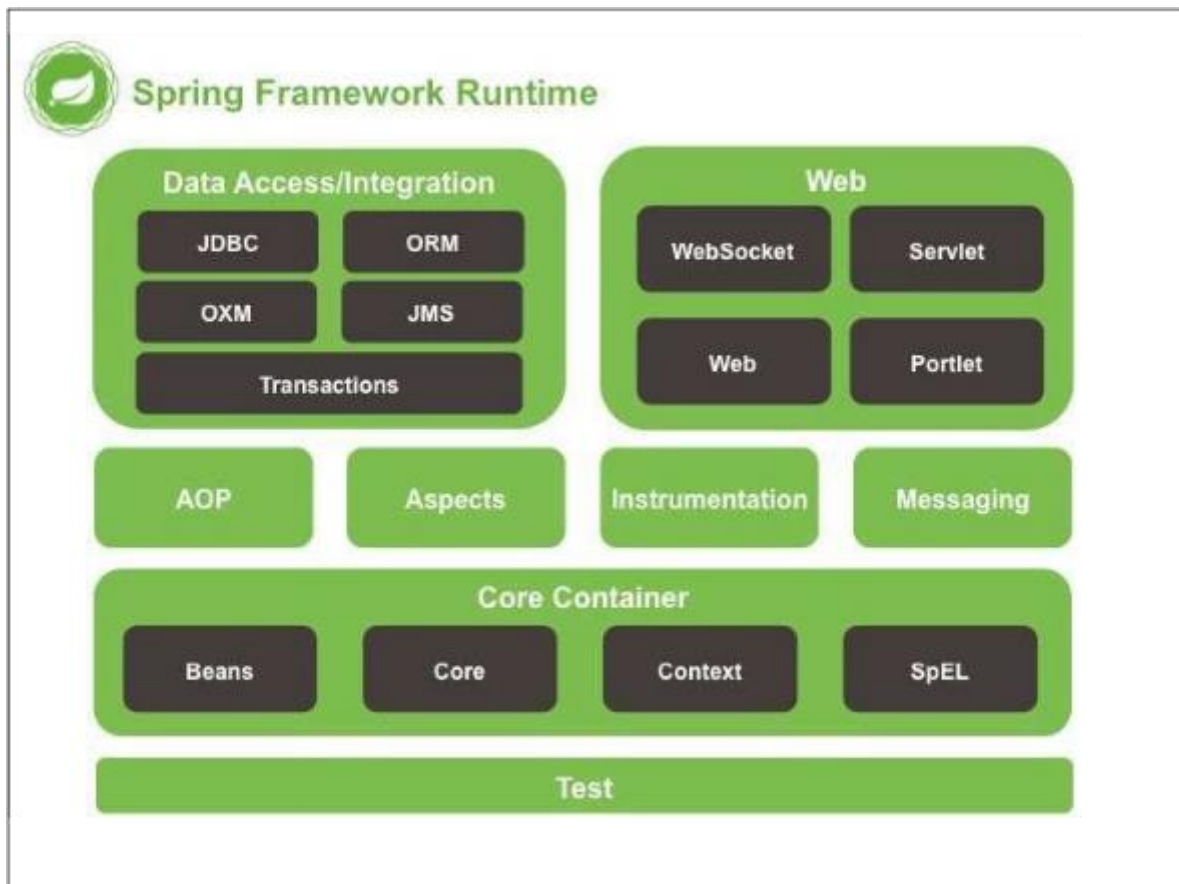


Figure 3.9: Spring Framework Runtime

## HTML5

HTML5, or Hypertext Markup Language revision 5, is a client-side markup language that provides a platform-independent method of structuring and presenting material on the World Wide Web. Various HTML5 tags, such as those used for headings, paragraphs, divisions, and anchors, make it easier to organize data. Notably, HTML5 permits the embedding of scripting languages such as JavaScript and CSS, allowing users to create and structure information using elements, tags, and attributes.

## CSS3

An aesthetic language called Cascading Style Sheets that defines various colors, layout, and fonts help users make their web pages more presentable. This has created a powerful combination of stylesheet (CSS) which can work separately from HTML as well with any XML-based markup language. There are three ways to include CSS in HTML files: these include inline CSS, internal CSS, and external CSS.

## **JavaScript**

JavaScript, a versatile and ubiquitous programming language, has become indispensable in modern web development. As an object-based language, JavaScript enables developers to add sophisticated effects, interactive elements, and dynamic behavior to web pages with ease. From complex animations to responsive clickable buttons and intuitive popup menus, JavaScript empowers developers to create immersive and engaging user experiences.

One of the key strengths of JavaScript lies in its ability to run seamlessly across various platforms, making it an ideal choice for creating web applications that work across different devices and operating systems. Whether it's a desktop browser, a mobile device, or even embedded systems, JavaScript can execute code consistently, ensuring a consistent user experience across platforms.

Moreover, JavaScript's integration with native objects within a host environment, such as a web browser, enables it to interact with the underlying system and manipulate the DOM (Document Object Model) dynamically. This capability allows developers to create dynamic and interactive web applications that respond to user actions in real-time, without the need to reload the entire page.

## **Docker**

Docker revolutionizes software development and deployment by offering a lightweight, portable solution for containerizing applications. It enables developers to package their code and dependencies into isolated containers that can run consistently across different environments. This containerization eliminates the infamous "it works on my machine" issue, ensuring seamless deployment from development to production. In the context of our exam portal project, Docker plays a pivotal role in simplifying the setup and maintenance of our application stack. By leveraging Docker Compose, we encapsulate our backend, frontend, and database services into separate containers, each with its own environment and dependencies. This approach not only streamlines the development workflow but also enhances scalability and reliability. Additionally, Docker's networking features facilitate seamless communication between containers, enabling our services to interact efficiently within the exam portal network. Overall, Docker empowers us to build, ship, and run our application with confidence, making the exam portal project more modular, maintainable, and easily deployable.

## Code Snippets

### Category Model:

```
public class Category {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long catId;  
  
    @Column(name = "title")  
    private String title;  
  
    @Column(name = "description")  
    private String description;  
  
    @OneToMany(mappedBy = "category", cascade = CascadeType.ALL)  
    @JsonIgnore  
    private List<Quiz> quizzes = new ArrayList<>();  
  
}
```

### Login Request:

```
package com.project.examportalbackend.models;  
  
import lombok.AllArgsConstructor;  
import lombok.Getter;  
import lombok.Setter;  
  
@Getter  
@Setter  
@AllArgsConstructor  
public class LoginRequest {  
    private String username;  
    private String password;  
}
```

---

## Login Response:

```
package com.project.examportalbackend.models;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@AllArgsConstructor
public class LoginResponse {
    private User user;
    private String jwtToken;
}
```

## Quiz Model:

```
package com.project.examportalbackend.models;

import com.fasterxml.jackson.annotation.JsonIgnore;
import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

import javax.persistence.*;
import java.util.ArrayList;
import java.util.List;

@Entity
@Getter
@Setter
@ToString
@Table(name = "quizzes")
public class Quiz {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long quizId;

    @Column(name = "title")
    private String title;

    @Column(name = "description", length = 5000)
    private String description;

    @Column(name = "max_marks")
    private int maxMarks;
}
```



## Quiz Result Model:

```
@Entity
@Getter
@Setter
@ToString
@NoArgsConstructor
@Table(name = "quiz_results")
public class QuizResult {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long quizResId;

    @Column(name = "user_id")
    private Long userId;

    @Column(name = "total_obtained_marks")
    private float totalObtainedMarks;

    @Column(name = "attempt_datetime")
    private String attemptDatetime;

    @ManyToOne(fetch = FetchType.EAGER)
    private Quiz quiz;
```

## User Model:

```
public class User implements UserDetails {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "user_id")
    private long userId; // field21

    @Column(name = "first_name")
    private String firstName;

    @Column(name = "last_name")
    private String lastName;

    @Column(name = "username", unique = true)
    private String username;

    @Column(name = "password")
    private String password;

    @Column(name = "phone_number")
    private String phoneNumber;

    @Column(name = "is_active")
    private boolean isActive = true;

    @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    @JoinTable(name = "user_role",
        joinColumns = {
            @JoinColumn(name = "user_id")
        },
        inverseJoinColumns = {
            @JoinColumn(name = "role_id")
        }
    )
    private Set<Role> roles;
```

```
public class User implements UserDetails {
```

```
    @Override  
    public Collection<? extends GrantedAuthority> getAuthorities() {  
        Set<SimpleGrantedAuthority> authorities = new HashSet<>();  
        this.roles.forEach(role ->  
            authorities.add(new SimpleGrantedAuthority(role.getRoleName())));  
        return authorities;  
    }
```

```
    @Override  
    public boolean isAccountNonExpired() {  
        return true;  
    }
```

```
    @Override  
    public boolean isAccountNonLocked() {  
        return true;  
    }
```

```
    @Override  
    public boolean isCredentialsNonExpired() {  
        return true;  
    }
```

```
    @Override  
    public boolean isEnabled() {  
        return isActive;  
    }
```

```
}
```

## Frontend Registration Page:


```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";
import { register } from "../actions/authActions";
import { useDispatch, useSelector } from "react-redux";
import Loader from "../components/Loader";
import { Form, Button, InputGroup, Row, Col } from "react-bootstrap";
import FormContainer from "../components/FormContainer";
import { FaEye, FaEyeSlash } from "react-icons/fa";
import * as authConstants from "../constants/authConstants";
import { Link } from "react-router-dom";

const RegisterPage = () => {
  const [firstName, setFirstName] = useState("");
  const [lastName, setLastName] = useState("");
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");
  const [phoneNumber, setPhoneNumber] = useState("");
  const [showPassword, setShowPassword] = useState(false);
  const [passwordType, setPasswordType] = useState("password");
  const [showConfirmPassword, setShowConfirmPassword] = useState(false);
  const [confirmPasswordType, setConfirmPasswordType] = useState("password");

  const dispatch = useDispatch();
  const navigate = useNavigate();
  const registerReducer = useSelector((state) => state.registerReducer);
```

```
const showPasswordHandler = () => {
  const temp = !showPassword;
  setShowPassword(temp);
  if (temp) {
    setPasswordType("text");
  } else {
    setPasswordType("password");
  }
};
```

```
const showConfirmPasswordHandler = () => {
  const temp = !showConfirmPassword;
  setShowConfirmPassword(temp);
  if (temp) {
    setConfirmPasswordType("text");
  } else {
    setConfirmPasswordType("password");
  }
};
```

```
const submitHandler = (e) => {
  e.preventDefault();
  
```

```

return (
  <FormContainer>
    <h1>Sign Up</h1>
    <Form onSubmit={submitHandler}>
      <Form.Group className="my-3" controlId="fname">
        <Form.Label>First Name</Form.Label>
        <Form.Control
          type="name"
          placeholder="Enter First Name"
          value={firstName}
          onChange={(e) => {
            setFirstName(e.target.value);
          }}
        ></Form.Control>
      </Form.Group>

      <Form.Group className="my-3" controlId="lname">
        <Form.Label>Last Name</Form.Label>
        <Form.Control
          type="name"
          placeholder="Enter Last Name"
          value={lastName}
          onChange={(e) => {
            setLastName(e.target.value);
          }}
        ></Form.Control>
      </Form.Group>
    </Form>
  </FormContainer>
)

```

```

<Form.Group className="my-3" controlId="username">
  <Form.Label>User Name</Form.Label>
  <Form.Control
    type="text"
    placeholder="Enter User Name"
    value={username}
    onChange={(e) => {
      setUsername(e.target.value);
    }}
  ></Form.Control>
</Form.Group>

<Form.Group className="my-3" controlId="password">
  <Form.Label>Password</Form.Label>
  <InputGroup>
    <Form.Control
      type={` ${passwordType} `}
      placeholder="Enter Password"
      value={password}
      onChange={(e) => {
        setPassword(e.target.value);
      }}
    ></Form.Control>
    <Button
      onClick={showPasswordHandler}
      variant=""
      style={{ border: "1px solid black" }}
    >
      {showPassword ? <FaEyeSlash /> : <FaEye />}
    </Button>
  </InputGroup>
</Form.Group>

```

```

<Form.Group className="my-3" controlId="confirmPassword">
  <Form.Label>Confirm Password</Form.Label>
  <InputGroup>
    <Form.Control
      type={` ${confirmPasswordType}`}
      placeholder="Confirm Password"
      value={confirmPassword}
      onChange={(e) => {
        setConfirmPassword(e.target.value);
      }}
    ></Form.Control>
    <Button
      onClick={showConfirmPasswordHandler}
      variant=""
      style={{ border: "1px solid black" }}
    >
      {showConfirmPassword ? <FaEyeSlash /> : <FaEye />}
    </Button>
  </InputGroup>
</Form.Group>

```

```

<Form.Group className="my-3" controlId="phoneNumber">
  <Form.Label>Phone Number</Form.Label>
  <Form.Control
    type="tel"
    placeholder="Enter Phone Number"
    value={phoneNumber}
    onChange={(e) => {
      setPhoneNumber(e.target.value);
    }}
  ></Form.Control>
</Form.Group>
<Button variant="" className="my-3" type="submit" style={{backgroundColor:"rgb(68 177 49)",
  Register
</Button>
</Form>

```



```

    {registerReducer.loading ? (
      <Loader />
    ) : (
      <Row className="py-3">
        <Col>
          Have an Account? <Link to="/" style={{color:"rgb(68 177 49)}}>Login</Link>
        </Col>
      </Row>
    )}
  </FormContainer>
);
};

```

```
export default RegisterPage;
```

### Auth Service:

```

import axios from "axios";
const register = async (user) => {
  try {
    const { data } = await axios.post("/api/register", user);
    if (data && data.userId) {
      console.log(
        "authService:register() Success: ",
        user.username,
        " successfully registerd."
      );
      return { isRegistered: true, error: null };
    } else {
      console.error("authService:register() Error: ", data);
      return { isRegistered: false, error: data };
    }
  }
}

```

```

    } catch (error) {
      console.error("authService:register() Error: ", error.response.statusText);
      return { isRegistered: false, error: error.response.statusText };
    }
  };

```

```

const login = async (username, password) => {
  try {
    const { data } = await axios.post("/api/login", {
      username: username,
      password: password,
    });

    if (data && data.jwtToken.length) {
      localStorage.setItem("user", JSON.stringify(data.user));
      localStorage.setItem("jwtToken", JSON.stringify(data.jwtToken));
      console.log("authService:login() Success: ", data.user);

```

```

      localStorage.setItem("jwtToken", JSON.stringify(data.jwtToken));
      console.log("authService:login() Success: ", data.user);
      return data;
    } else {
      console.error("authService:login() Error: ", data);
      return data;
    }
  } catch (error) {
    console.error("authService:login() Error: ", error.response.statusText);
    return error.response.statusText;
  }
};

```

```

const authServices = { register, login };
export default authServices;

```

## Question Service:

```
import axios from "axios";

const fetchQuestionsByQuiz = async (quizId, token) => {
  try {
    const config = {
      headers: { Authorization: `Bearer ${token}` },
    };
    const { data } = await axios.get(
      `/api/question/?quizId=${quizId}`,
      config
    );
    console.log("questionsServices:fetchQuestionsByQuiz() Success: ", data);
    return data;
  } catch (error) {
    console.error(
      "questionsServices:fetchQuestionsByQuiz() Error: ",
      error.response.statusText
    );
    return error.response.statusText;
  }
};

const addQuestion = async (question, token) => {
  try {
    const config = {
      headers: { Authorization: `Bearer ${token}` },
    };

    const { data } = await axios.post("/api/question/", question, config);
    console.log("questionsServices:addQuestion() Success: ", data);
    return { data: data, isAdded: true, error: null };
  } catch (error) {
    console.error(
      "questionsServices:addQuestion() Error: ",
      error.response.statusText
    );
    return { data: null, isAdded: false, error: error.response.statusText };
  }
};
```

```
const updateQuestion = async (question, token) => {
  try {
    const config = {
      headers: { Authorization: `Bearer ${token}` },
    };
    const { data } = await axios.put(
      `/api/question/${question.quesId}`,
      question,
      config
    );
    console.log("questionsServices:updateQuestion() Success: ", data);
    return {
      data: data,
      isUpdated: true,
      error: null,
    };
  }
};
```

```
const questionsServices = {
  fetchQuestionsByQuiz,
  addQuestion,
  deleteQuestion,
  updateQuestion,
};
export default questionsServices;
```

### Docker Frontend File:

```
FROM node:14-alpine as build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build

FROM nginx:alpine
COPY --from=build /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

---

### Docker Backend File:

```
FROM openjdk:11

# Set the working directory inside the container
WORKDIR /app

# Copy the packaged JAR file into the container at /app
COPY target/exam-portal-backend-0.0.1-SNAPSHOT.jar /app/exam-portal-backend.jar

# Expose port 8081 to the outside world
EXPOSE 8081

# Run the Spring Boot application when the container launches
CMD ["java", "-jar", "exam-portal-backend.jar"]
```

## Docker Compose-YAMLFile:

```
version: '3'

services:
  mysql:
    image: mysql:latest
    container_name: mysql
    ports:
      - "3307:3306"
    environment:
      MYSQL_ROOT_PASSWORD: "4321"
      MYSQL_DATABASE: "exam_portal"
    networks:
      - exam-portal-network

  backend:
    build: ./exam-portal-backend
    container_name: backend
    ports:
      - "8080:8081"
    depends_on:
      - mysql
    environment:
      SPRING_DATASOURCE_URL: jdbc:mysql://mysql:3306/exam\_portal?createDatabaseIfNotExist=true&allowF
      SPRING_DATASOURCE_USERNAME: "root"
      SPRING_DATASOURCE_PASSWORD: "4321"
    networks:
      - exam-portal-network
```

```
frontend:
  build: ./exam-portal-frontend
  container_name: frontend
  ports:
    - "3002:80"
  networks:
    - exam-portal-network
```

```
networks:
  exam-portal-network:
    driver: bridge
```

### **Security Used:**

Security measures are employed by the application to guard user information and prevent threats. Secure communications are ensured by utilizing JSON Web Token (JWT) authentication, and password security is enhanced and brute force attacks are prevented by BCrypt encryption. Access is restricted according to user roles using role-based access control (RBAC), and strong authentication is ensured by combining authentication controls with exception handling. Access control is managed using permission filtering, which also addresses access error problems and lowers the possibility of CORS configuration cross-history. Finally, there will be reduced possibility of session hijacking owing to the stateless control mechanism implemented. When amalgamated, these technologies establish a robust security foundation, enhancing the overall security posture of the application.

### **JWT (JSON Web Tokens) Authentication**

Sending messages between parties is safe and stateless with JSON Web Tokens (JWT) ; the architecture can be seen in Figure 3.10. In order to guarantee integrity and validity through digital signatures, JWT is utilized in the application for user authentication. We may increase



the overall security of the authentication process by lowering the danger of token replay attacks by managing the expiration time.

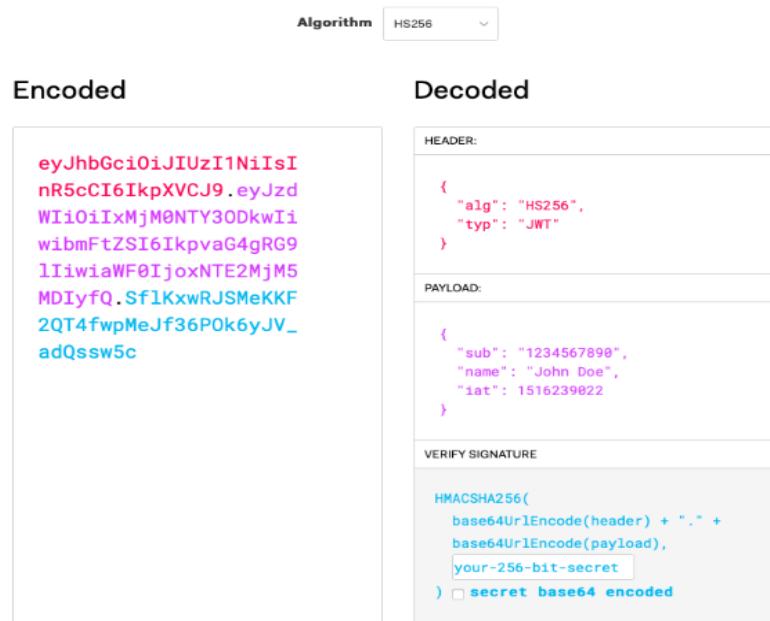


Figure 3.10: JWT Architecture

### Password Encryption (BCrypt)

Our application uses BCrypt as shown in Figure 3.11, a powerful one-way hashing technique, to encrypt passwords. By prohibiting the salting and repeated rounds of encrypted password storage, BCrypt offers protection against brute force assaults. Even in the case of a data breach, the confidentiality of user passwords is guaranteed by this encryption technique.

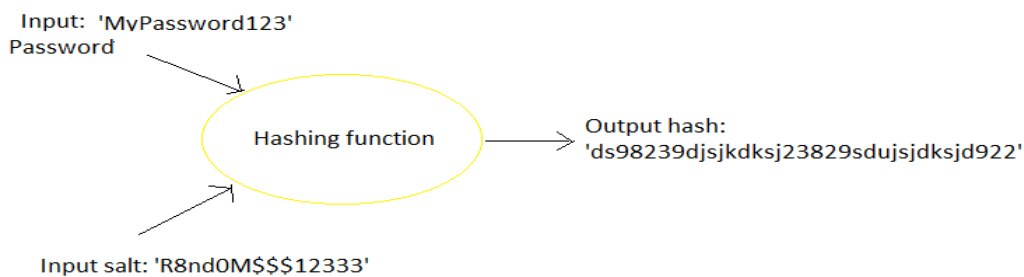


Figure 3.11: Working Of Password Encryption

### Role-Based Access Control (RBAC)

Role-based control, or RBAC, governs user authorization in our application. RBAC makes sure that only authorized users are able to carry out allowed operations by linking particular roles



like ADMIN or USER with access to resources and operations. By restricting access to only the features essential to each user position, the least privilege concept improves security.

### **Authentication Manager and Exception Handling**

Authentication Manager is used by our application to safely manage user credentials when logging in. Additionally, to verify that the error is accurate, a unique exception handling capability is employed. By stopping the distribution of false information and the leaking of important data, we enhance user experience and security.

### **Authorization Filters**

Authorization filters manage access according to user roles and prevent incoming requests to validate JWT tokens. This guarantees that particular resources and functions can only be accessed by authenticated users who possess the necessary permissions. By limiting illegal access attempts at the filter level, the chance of data leakage and unauthorized alteration is decreased.

### **Cross-Origin Resource Sharing (CORS) Configuration**

By limiting access to resources from several sources, Cross-Origin Resource Sharing setup lowers the danger of Cross-Site Request Forgery (CSRF) and stops illicit cross-border requests. We may lower downtime and improve overall security by defining which resources are permitted to access our application data.

### **Error Logging**

Through the provision of event and security condition visibility, error detection plays a vital part in our application's security infrastructure as can be seen in Figure 3.12. We can look at and address security vulnerabilities, help with troubleshooting and analysis, and gather error reports. Our capacity to quickly identify and address security vulnerabilities is enhanced by accurate error detection.

```

2018-07-27 12:39:49.040 INFO 61432 --- [ restartedMain] ConfigServletWebServerApplicationContext : Refreshing org.springframework.boot.web.servlet.context.AnnotationDelegateBeanPostProcessorChecker
2018-07-27 12:39:50.607 INFO 61432 --- [ restartedMain] org.eclipse.jetty.util.log : Logging initialized @3083ms to org.eclipse.jetty.util.log.Slf4jLog
2018-07-27 12:39:50.868 INFO 61432 --- [ restartedMain] o.s.b.w.e.j.JettyServletWebServerFactory : Server initialized with port: 8080
2018-07-27 12:39:51.045 INFO 61432 --- [ restartedMain] org.eclipse.jetty.server.Server : jetty-9.4.11.v20180605; built: 2018-06-05T18:24:03.829Z; git sha1: 4613112
2018-07-27 12:39:51.047 INFO 61432 --- [ restartedMain] org.eclipse.jetty.server.session : DefaultSessionIdManager workerName=node0
2018-07-27 12:39:51.144 INFO 61432 --- [ restartedMain] org.eclipse.jetty.server.session : No SessionScavenger set, using defaults
2018-07-27 12:39:51.145 INFO 61432 --- [ restartedMain] org.eclipse.jetty.server.session : node0 Scavenging every 660000ms
2018-07-27 12:39:51.145 INFO 61432 --- [ restartedMain] o.e.j.s.h.ContextHandler.application : Initializing Spring embedded WebApplicationContext
2018-07-27 12:39:51.157 INFO 61432 --- [ restartedMain] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2118ms
2018-07-27 12:39:51.157 INFO 61432 --- [ restartedMain] o.s.b.w.servlet.ServletRegistrationBean : Servlet dispatcherServlet mapped to [/]
2018-07-27 12:39:51.266 INFO 61432 --- [ restartedMain] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'characterEncodingFilter' to: [/*]
2018-07-27 12:39:51.266 INFO 61432 --- [ restartedMain] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to: [/*]
2018-07-27 12:39:51.267 INFO 61432 --- [ restartedMain] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to: [/*]
2018-07-27 12:39:51.267 INFO 61432 --- [ restartedMain] o.s.b.w.servlet.FilterRegistrationBean : Mapping filter: 'requestContextFilter' to: [/*]
2018-07-27 12:39:51.270 INFO 61432 --- [ restartedMain] o.e.jetty.server.handler.ContextHandler : Started o.s.b.w.e.j.JettyEmbeddedWebAppContext@2898103f{appContext}
2018-07-27 12:39:51.271 INFO 61432 --- [ restartedMain] org.eclipse.jetty.server.Server : Started @3487ms
2018-07-27 12:39:51.292 INFO 61432 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2018-07-27 12:39:51.455 INFO 61432 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2018-07-27 12:39:51.495 INFO 61432 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Building JPA container EntityManagerFactory for persistence unit 'default'
2018-07-27 12:39:51.506 INFO 61432 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [
name: default
]

```

Figure 3.12: Error Logging

## Session Management

In the application, server-side sessions are eliminated through the utilization of a stateless session policy for session management. We guarantee availability and lessen the possibility of session hijacking by employing a stateless technique. By lowering potential vulnerabilities related to server-side communication operations, this improves overall security.

## Container Isolation

Leveraging Docker's containerization technology, each aspect of our application - backend, frontend, and database services - operates within its isolated container environment. This isolation ensures that potential security breaches within one component do not compromise the integrity of the entire system.

## Secure Image Signing

By sourcing Docker images exclusively from trusted repositories, we ensure the authenticity and integrity of our container images. This practice mitigates the risk of deploying compromised or tampered images that could potentially introduce security vulnerabilities into our environment.

## **Network Segmentation with Docker Networking**

Docker networking enables us to implement network segmentation, allowing us to define separate communication channels between containers and external networks. By partitioning the network in this manner, we ensure that only authorized traffic is allowed to flow between containers and external entities. This helps minimize the risk of unauthorized access and data exfiltration, enhancing the overall security of our Docker environment

### **3.4 Key Challenges**

Developing the Online Examination System as an alternative way of performing the assessment is the most important breakthrough that has ever taken place in education. Nonetheless, some critical issues have risen during this development phase; each of these must be thoroughly studied and solutions proposed that will ensure the success and adoption of the system by all. The first issue that arises concerns shifting smoothly from the paper-based exams to the online mode. However, acceptance of the new method could be obstructed by people's reluctance to change things and concerns over validity of web-based tests. In order to address this, the project aims at developing an intuitive interface for web based tests. This design is centered around a good user experience. As such, it makes the test taking process easier both for examinees and examiners. This is why the company strives for an easy to understand, or user-friendly interface in its system.

The other problem arises where all students who will attend online examinations will have to be registered into one secure system. This problem is handled through a self-registration system, where the applicants can register themselves for the tests. Moreover, this function makes quick registrations thus allowing learners to conveniently use the system without being overly burdened by administration institutions. Correctness and efficiency considerations present their own challenges in the case of automatically scoring objective-type tests. These are problems that become dealt with by automating a system of judging applications in online subjective tests. It saves time and minimizes chances of human mistake improving the system's reliability.

Candidate response confirmation is another vital area that deserves focus. The system has an automated verification mechanism to verify candidates' answers during the exam so as to prevent this from happening. This ensures the integrity of the evaluation process. This enables examiners to get reliable and credible results. Given that most schools are now using online assessments, paperwork and admin issues must be overcome. This issue is addressed in the proposal through the development of an all-inclusive Web-based test management module. Since it minimizes paperwork, this technique works in favor for both, examiners and examinees. The study enhances efficiency within an institution as it simplifies or reduces administrative duties like making orders for examinations and examination procedures.

Allowing examiners to generate their own question papers poses a unique issue in terms of ensuring question quality and diversity. This difficulty is addressed by the Secure Web application, which provides tools and features that enable examiners to effortlessly construct diverse and high-quality question papers. This keeps exams tough and in line with educational norms. Finally, cloud infrastructure integration poses new hurdles in terms of accessibility, scalability, and efficiency. The system, on the other hand, makes use of cloud technology to shorten procedures, improve accessibility, and enable scalability. This smart usage of cloud infrastructure improves the examination experience for all parties concerned while also ensuring a dependable and scalable platform.

To summarize, the main hurdles in establishing the Secure Web application include user acceptability, registration processes, automated assessments, response verification, integration with educational institutions, question paper preparation, and cloud infrastructure. The project handles each barrier systematically in order to produce a robust, user-friendly, and efficient Cloud Web application that achieves the main goal of revolutionizing the evaluation process in educational establishments around the world.

# Chapter 4: Testing

## 4.1 Testing Strategy

Testing is essential for guaranteeing the Online Exam Portal Web Application's robustness and reliability [22]. The testing process entails a thorough approach to validating many components of the system. Functional testing ensures that all functionalities, such as user registration, exam creation, and result generating, function as expected. Usability testing is used to evaluate the intuitiveness and overall user experience of a user interface. Security testing is critical for identifying and correcting potential flaws in data processing and user authentication processes. Performance testing is also carried out to assess the system's responsiveness, scalability, and general efficiency under changing loads. The testing procedure is iterative, with each iteration aimed at detecting and correcting any flaws, ensuring that the Virtual Exam Portal operates smoothly, securely, and reliably.

### Unit Testing

The process of evaluating individual units of software to ensure that they perform as expected is known as unit testing. This level of testing is usually done by developers and is frequently automated. Unit testing is an important aspect of the software development lifecycle (SDLC) since it aids in the early detection and correction of issues.

Unit testing is intended to:

- Check that each piece of software works as it should.
- Identify and fix bugs as early as possible in the development process.
- Improve the software's quality.
- Make the software easier to maintain.
- The Advantages of Unit Testing

Unit testing has numerous advantages, including:

- Lower development expenses
- Enhanced software quality
- Lower maintenance expenses
- Increased developer assurance

### **Spring Boot Unit Testing:**

Consider a scenario in which a Spring Boot application serves as the backend for a quiz portal. Wrote JUnit tests for individual components such as controllers, services, and repositories. Mockito was used to construct mock dependencies (for example, for database transactions) in unit tests, guaranteeing that each unit is examined independently.

### **Integration Testing**

It also includes what is called the integration testing that refers to assessing different elements constituting integrated software. Such testing happens at a point where testers engage in it manually. Therefore, integration testing is important as it helps discover faults, which are observed upon interaction between various software components.

Integration testing's goal is to:

- Identify faults that emerge when software components interact with one another.
- Ascertain that the integrated software fits the overall system requirements.
- Improve the integrated software's quality.
- The Advantages of Integration Testing

Integration testing has numerous advantages, including:

- Lower development expenses
- enhanced software quality
- Lower maintenance expenses
- increased trust in the integrated software

### **Integration Testing Techniques**

There are a variety of integration testing approaches available, including:

- Testing from the top down
- Bottom-up research
- Sandwich evaluation

### **Spring Boot Integration Testing:**

Consider the relationships between different layers inside the Spring Boot application, such as controllers, services, and databases. Used integration testing approaches such as "testing from the top down" and "bottom-up research" to examine communication between controllers and services or services and databases. Validated the integrated components' flawless collaboration.

### **System Testing**

The process of testing an entire, integrated system is known as system testing. This level of testing is often carried out by testers and is frequently manual. System testing is crucial because it aids in the detection of faults that arise when the system interacts with other systems or the environment.

The purpose of system testing is to:

Identify any issues that arise when the system interacts with other systems or the environment.  
Check to see if the system fulfills the overall system requirements. Improve the system's quality.

The Advantages of System Testing:

- Lower development expenses
  - enhanced software quality
  - Lower maintenance expenses
  - Improved system confidence
- System Testing Techniques

There are several system testing methodologies that can be utilized, including:

- Testing in the dark
- White-box evaluation
- Performance Testing

System testing throughout the application:

Examined the whole online quiz portal, including interactions between the Spring Boot backend and the React.js frontend. Conducted system testing to evaluate the integrated system's overall functionality. Used approaches such as "testing in the dark" and "white-box evaluation" to detect and resolve issues linked to backend-frontend interactions.

## Acceptance Testing

Acceptance testing [23] is the process of determining whether a system is ready for delivery. Typically, the customer or end-user does this level of testing. Acceptance testing is vital since it ensures that the system satisfies the needs and expectations of the client.

Acceptance testing is performed to:

- Determine whether the system satisfies the needs and expectations of the customer.
- Check to see if the system fulfills the overall system requirements.
- Improve the system's quality.

Acceptance testing has numerous advantages, including:

- Customer satisfaction has increased.
- Lower development expenses
- enhanced software quality
- Lower maintenance expenses
- Acceptance Techniques for Testing

There are various acceptability testing methodologies that can be utilized, including:

- User acceptance testing (UAT)
- Beta testing
- Pilot testing

Acceptance Testing throughout the application:

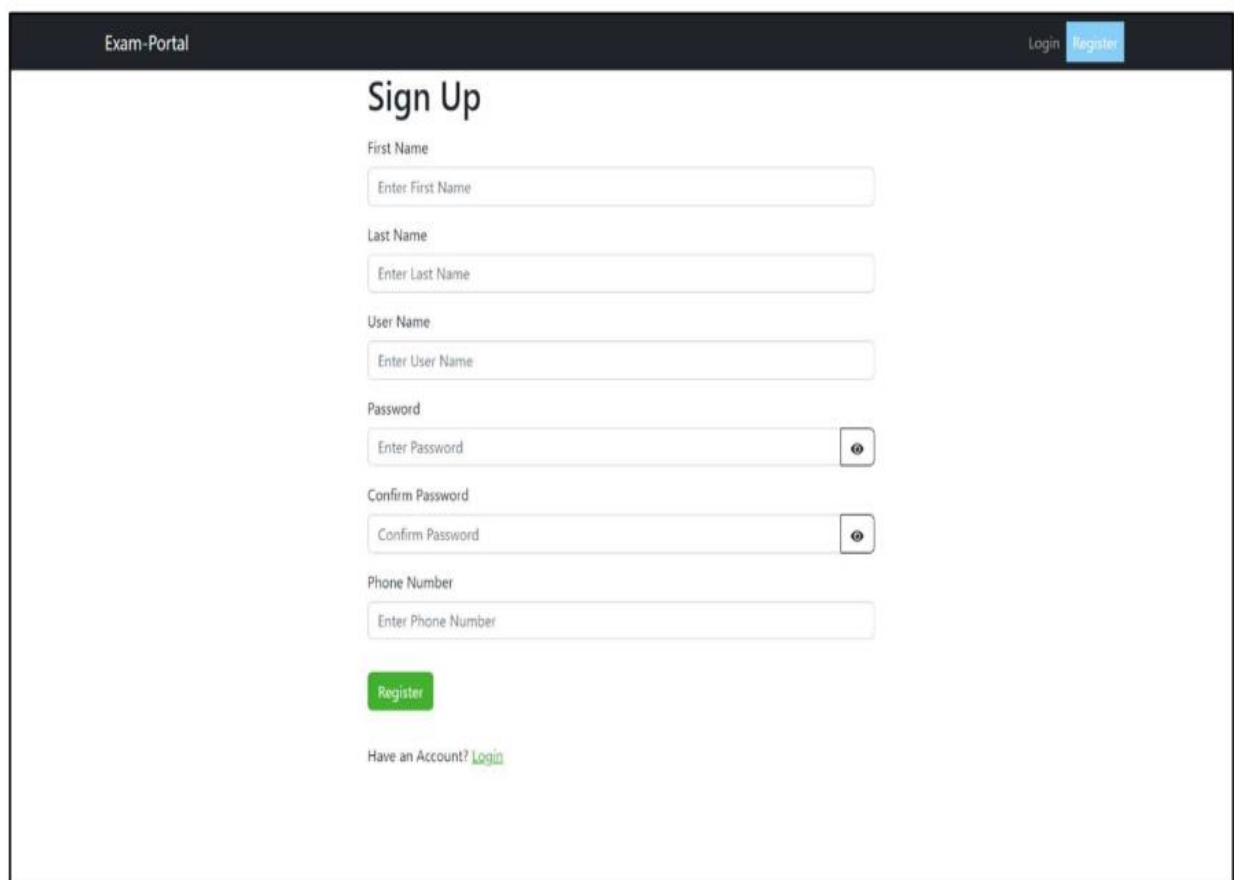
Validated that the quiz application meets the user's expectations and requirements. Conducted User Acceptance Testing (UAT), in which end-users authenticate the functionality and user interface of the application. Consider Beta or Pilot testing to gather feedback from a specific user group and ensure alignment with user needs.



## Chapter 5: Results

In the results section, we meticulously delineate the implemented features and functionalities of the project, supplementing them with educational screenshots to offer a comprehensive overview.

The Registration Page in Figure 5.1 is an example of an interface that is easy to use and streamlines the enrollment process for candidates through its accessible and intuitive design. This guarantees a smooth and effective registration process.



The screenshot displays the 'Sign Up' page of an 'Exam-Portal'. The page features a dark header with 'Exam-Portal' on the left and 'Login' and 'Register' buttons on the right. The main content area is white and contains the following fields and elements:

- First Name:** A text input field with the placeholder 'Enter First Name'.
- Last Name:** A text input field with the placeholder 'Enter Last Name'.
- User Name:** A text input field with the placeholder 'Enter User Name'.
- Password:** A text input field with the placeholder 'Enter Password' and a toggle icon for visibility.
- Confirm Password:** A text input field with the placeholder 'Confirm Password' and a toggle icon for visibility.
- Phone Number:** A text input field with the placeholder 'Enter Phone Number'.
- Register Button:** A green button with the text 'Register'.
- Link:** A link labeled 'Have an Account? Login'.

Figure 5.1: Registration Page

An illustration of a secure gateway is the Login Page in Figure 5.2, which demands user authentication with a functional username and password to ensure a dependable and secure access point.

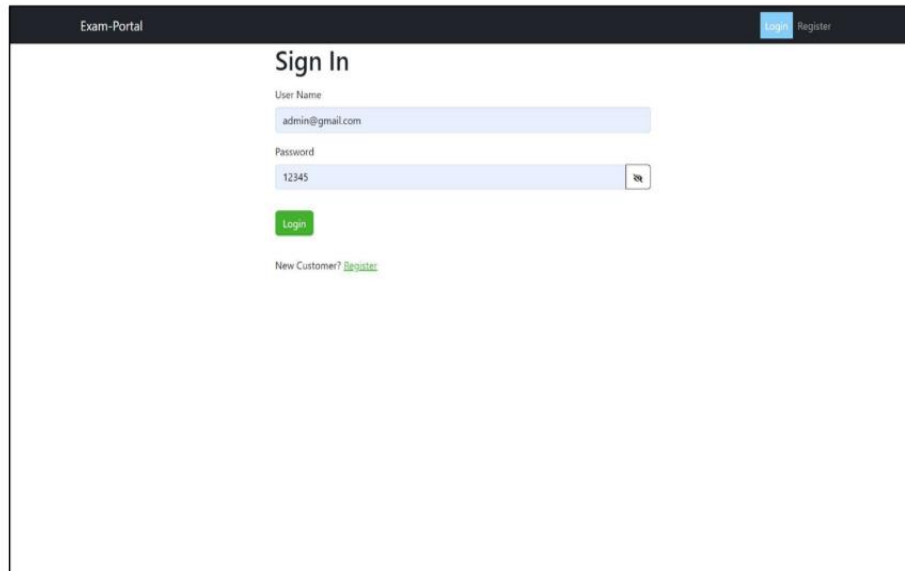


Figure 5.2: Login Page

The Profile Page, shown in Figure 5.3, offers a wealth of information, including name, phone number, role, username, account status, and other details, all of which contribute to an intuitive and educational user experience



Figure 5.3: Profile Page

The add category interface, shown in Figure 5.4, simplifies the categorization process by providing a user-friendly platform for entering category details, such as title and description.

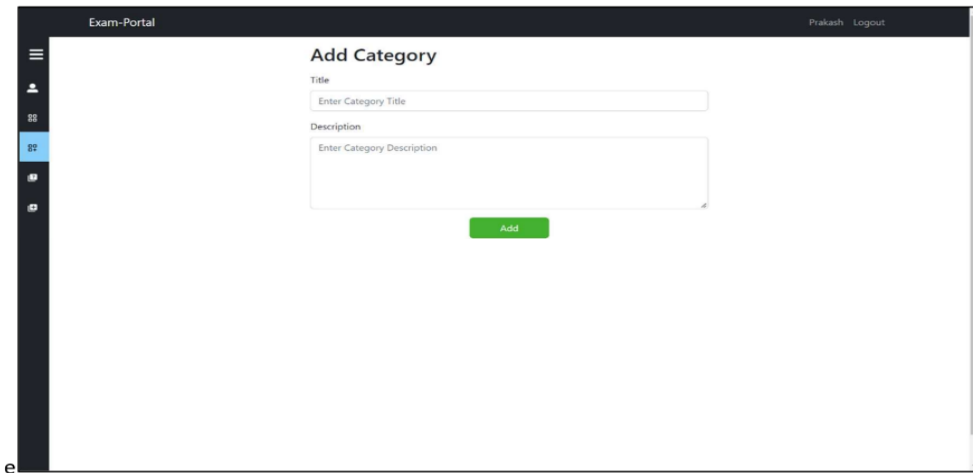


Figure 5.4: Add Category

The categories page in Figure 5.5 presents an orderly arrangement of various exam categories, giving users a concise rundown of their options.

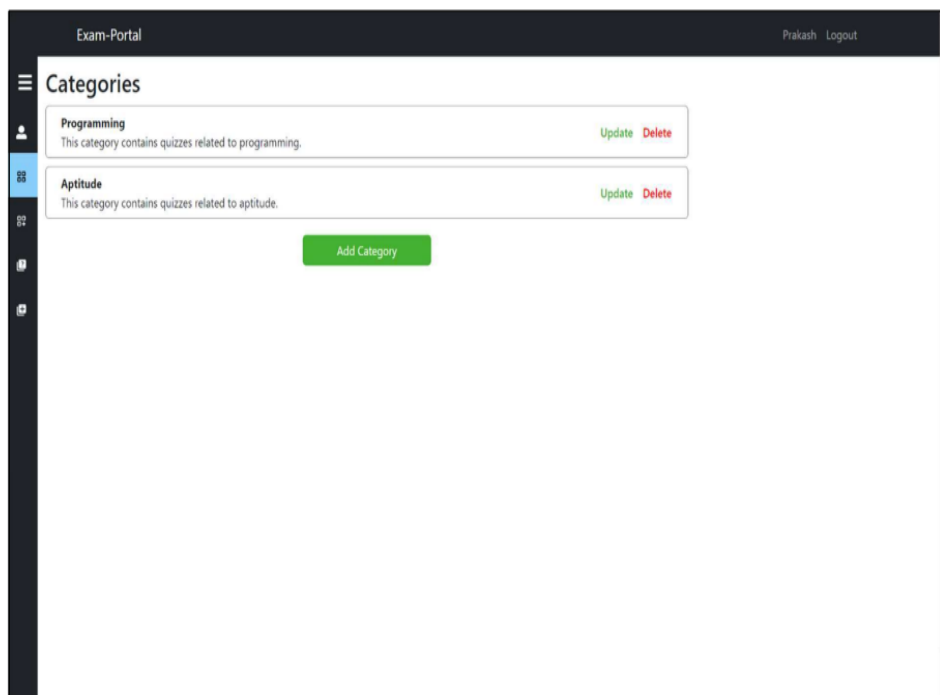


Figure 5.5: Categories Page

The add quiz interface in Figure 5.6 enables users to easily create and modify quizzes.

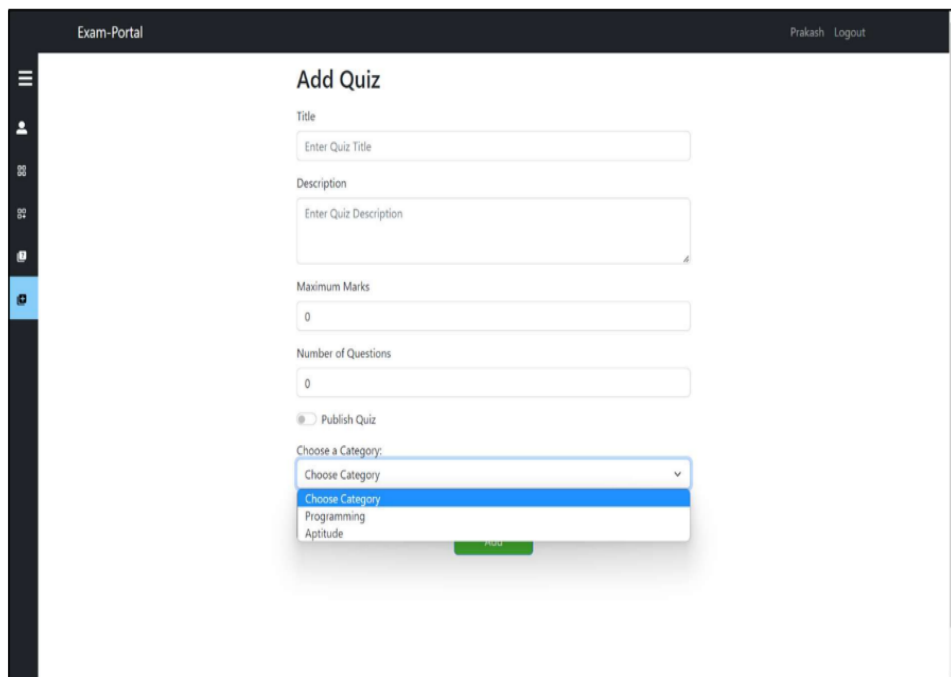


Figure 5.6: Add Quiz

The Quiz Portal admin interface, which includes update and delete options among other crucial features, is displayed in Figure 5.7.

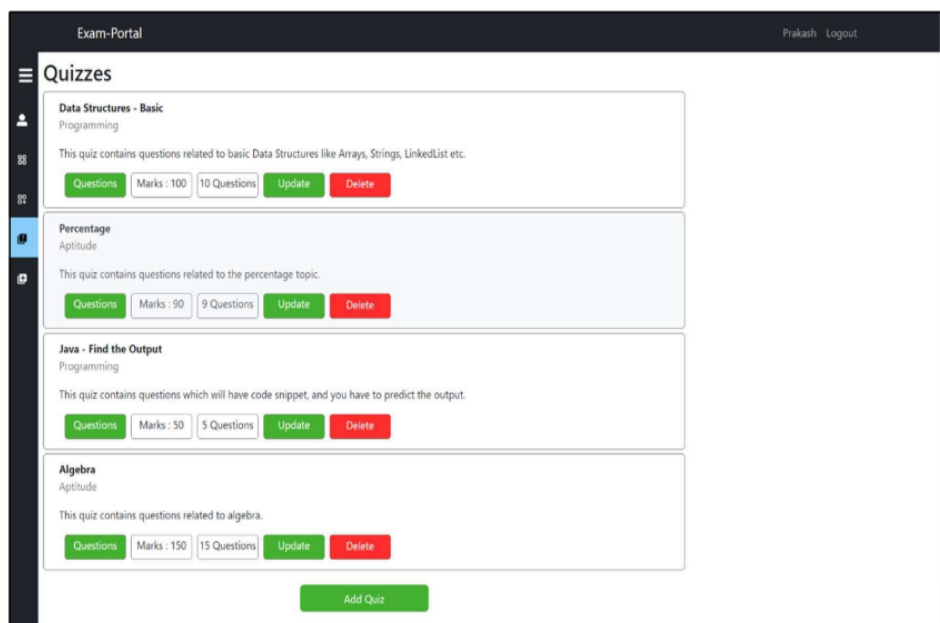
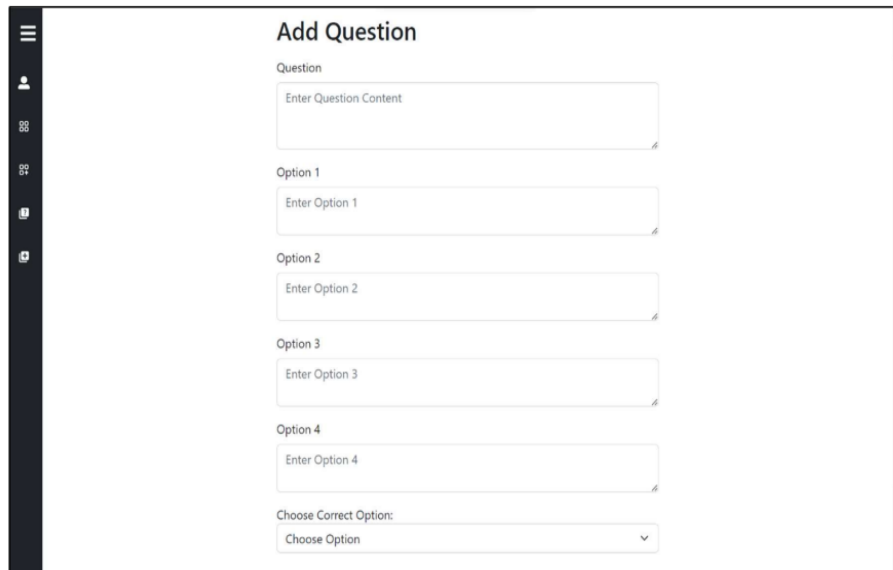


Figure 5.7: Quiz Portal

Offering options for one, two, three, and four choices increases the adaptability of the quiz creation process, as shown in Figure 5.8.

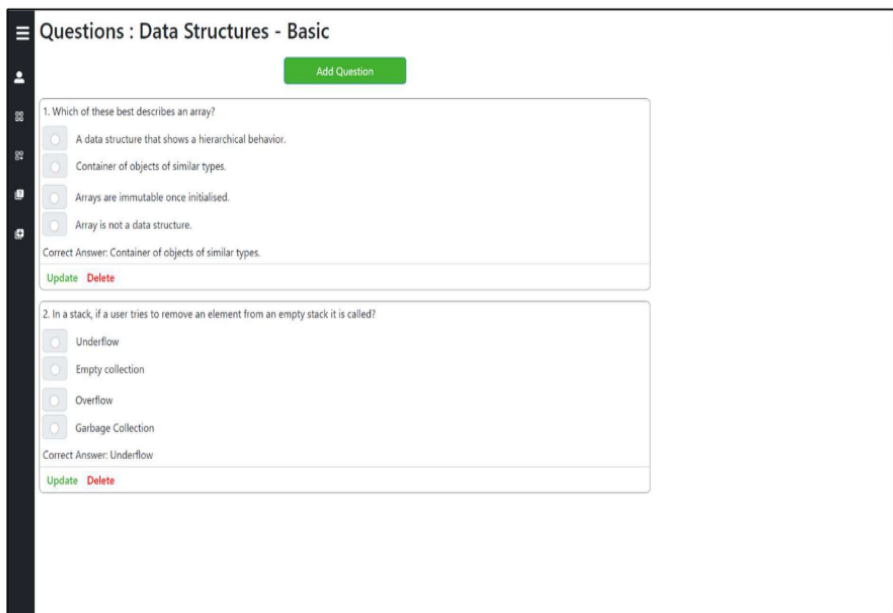


The screenshot shows a web interface for adding a question. It features a dark sidebar on the left with navigation icons. The main content area is titled "Add Question" and contains the following fields:

- Question:** A text input field with the placeholder "Enter Question Content".
- Option 1:** A text input field with the placeholder "Enter Option 1".
- Option 2:** A text input field with the placeholder "Enter Option 2".
- Option 3:** A text input field with the placeholder "Enter Option 3".
- Option 4:** A text input field with the placeholder "Enter Option 4".
- Choose Correct Option:** A dropdown menu with the placeholder "Choose Option".

Figure 5.8: Add Question

The admin questions page" is shown in Figure 5.9, which serves as a central point for effectively managing and supervising questions in the system.



The screenshot shows the "Admin Questions Page" for "Data Structures - Basic". It features a dark sidebar on the left and a main content area with the following elements:

- Header:** "Questions : Data Structures - Basic" and a green "Add Question" button.
- Question 1:** "1. Which of these best describes an array?" with four radio button options: "A data structure that shows a hierarchical behavior.", "Container of objects of similar types.", "Arrays are immutable once initialised.", and "Array is not a data structure." The correct answer is "Container of objects of similar types." with "Update" and "Delete" buttons.
- Question 2:** "2. In a stack, if a user tries to remove an element from an empty stack it is called?" with four radio button options: "Underflow", "Empty collection", "Overflow", and "Garbage Collection". The correct answer is "Underflow" with "Update" and "Delete" buttons.

Figure 5.9: Admin Questions Page

The user categories page, which allows users to browse and access different quizzes, is depicted in Figure 5.10.

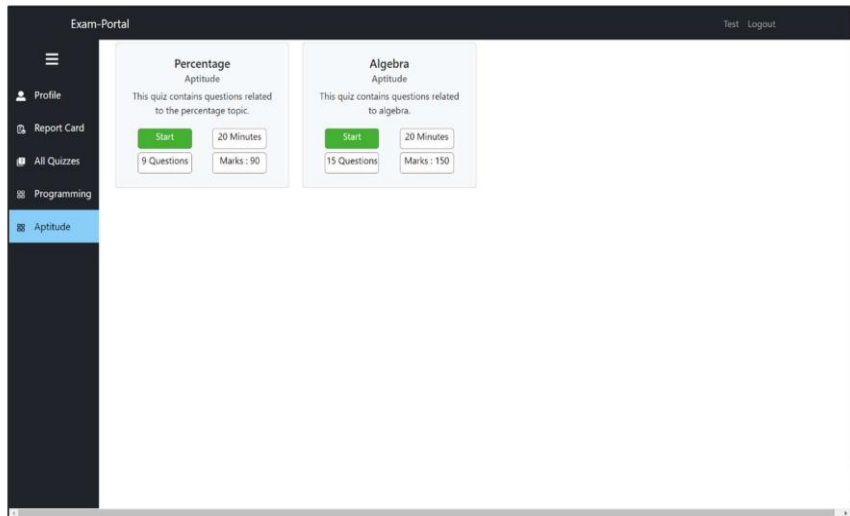


Figure 5.10: User Categories Page

Figure 5.11: The quiz manual page gives users all the information they need to prepare for an exam by providing detailed information about each quiz, such as a description and the total amount of time allowed.

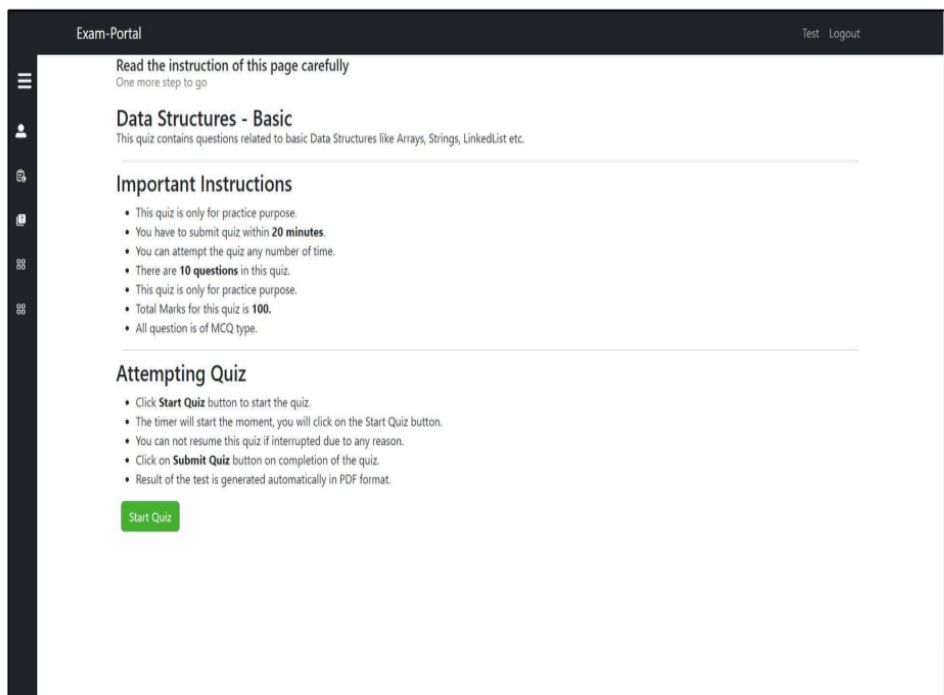


Figure 5.11: Quiz Manual Page

Figure 5.12: Participants can view the questions from the current exam on the user questions page," where they can read through and answer each one in the allotted time.

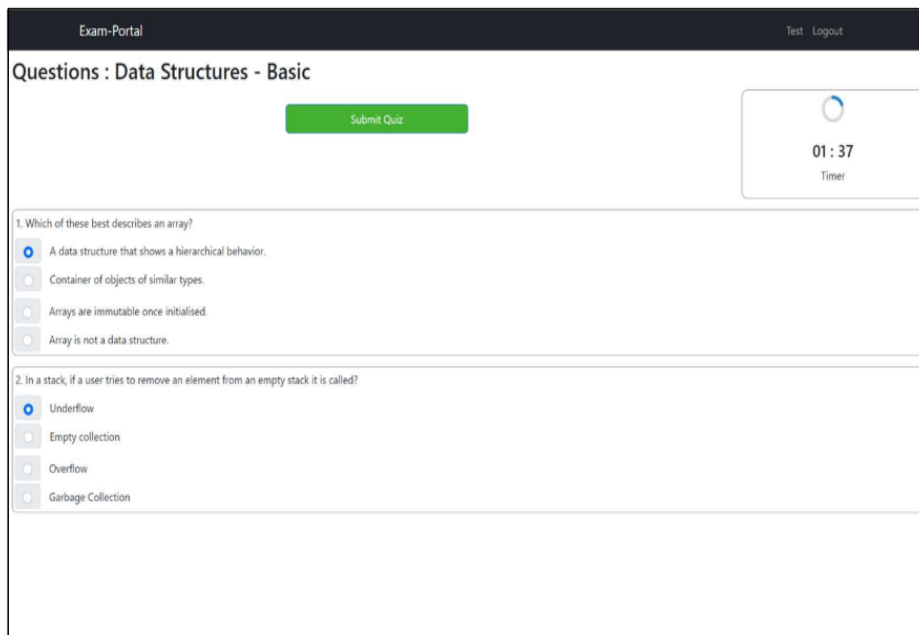


Figure 5.12: User Questions Page

Figure 5.13: The successfully created and functional instance is shown visually by the EC2 Instance display.

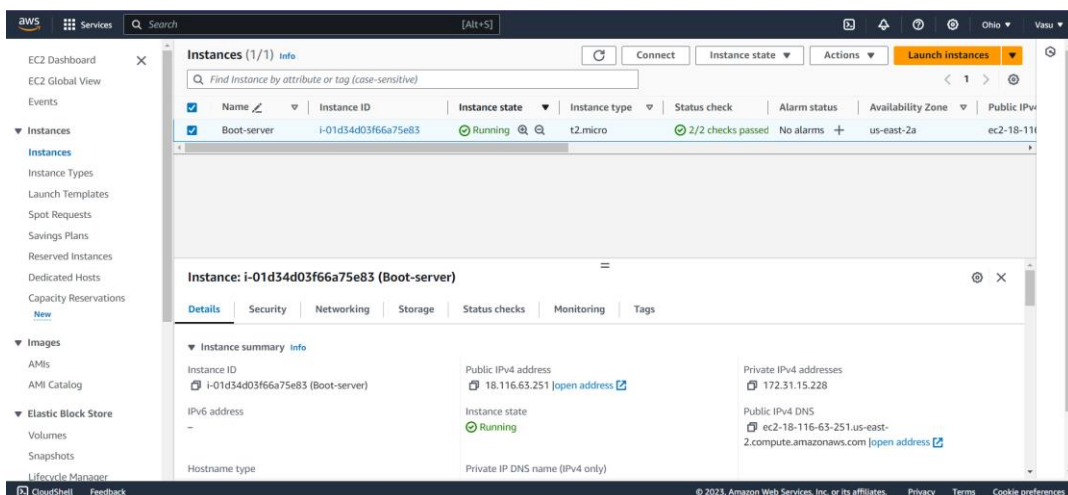


Figure 5.13: EC2 Instance

Figure 5.14: An extensive overview of the operational activities and events of the system is provided by the system Logs interface.

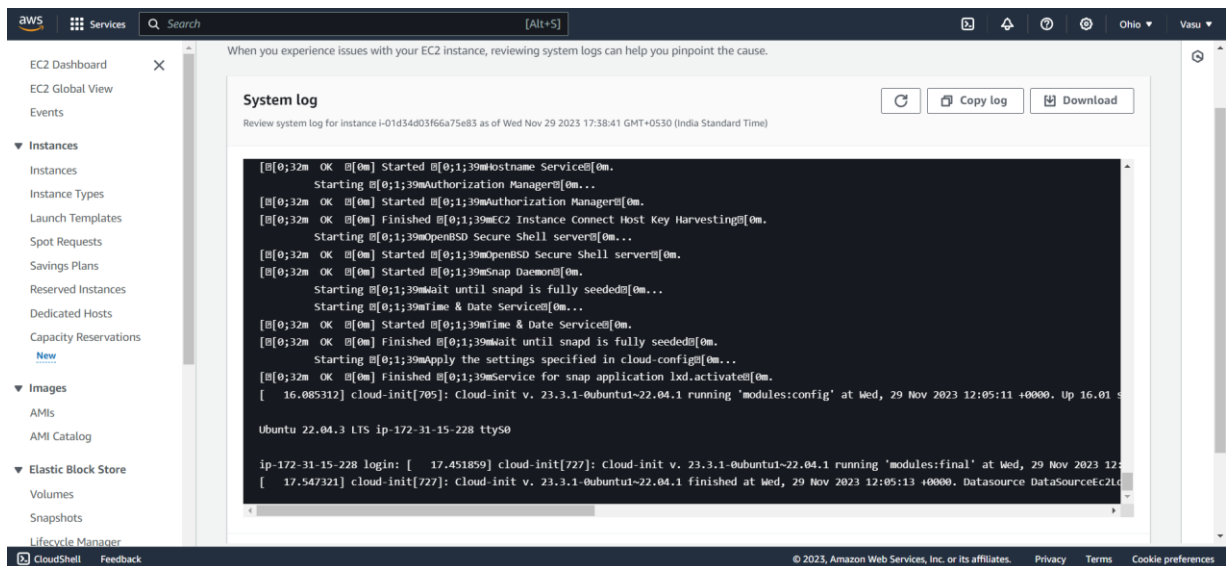
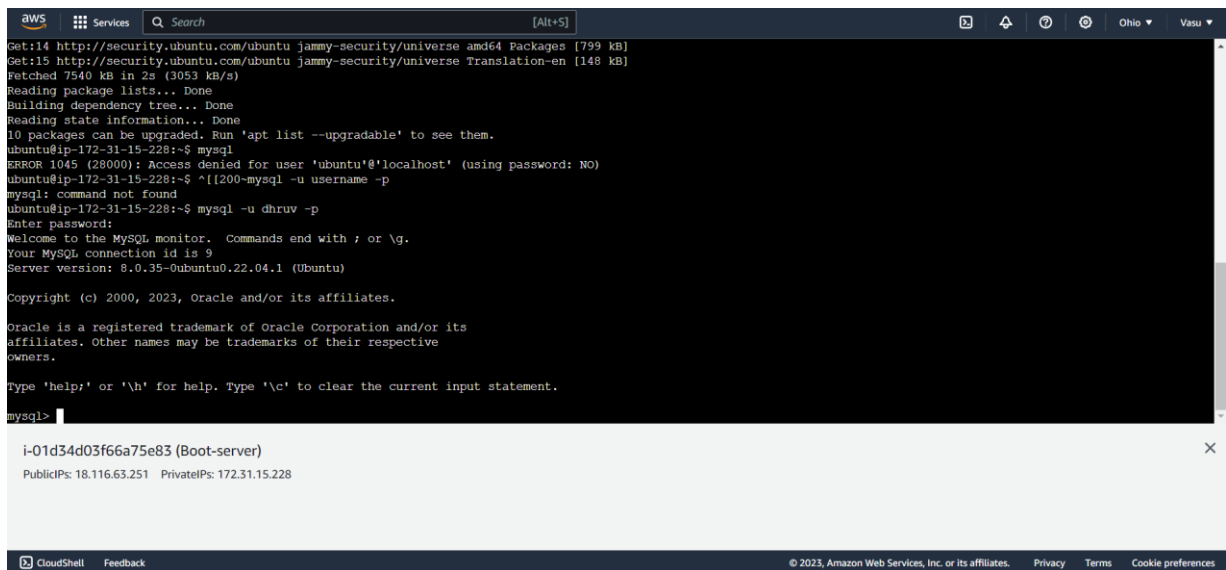


Figure 5.14: System Logs



Figure 5.15: A thorough real-time feed of operational activity across several containers is shown in the Logs interface of the Docker Desktop program.

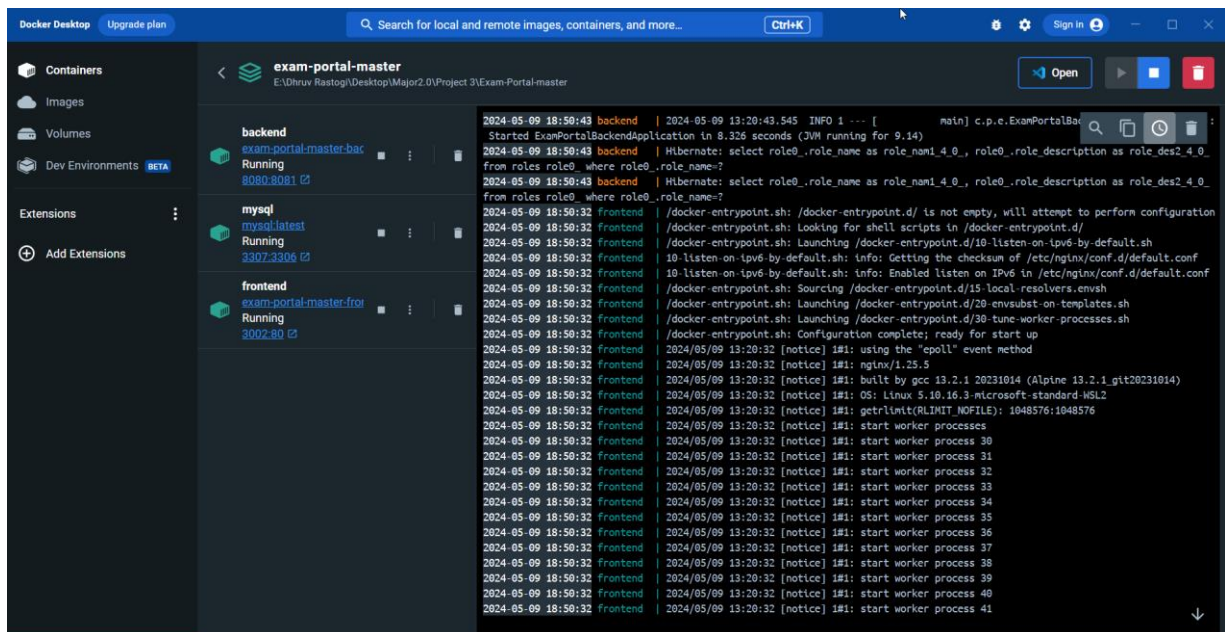


Figure 5.15: Docker Desktop

Figure 5.16: The terminal showcases an Ngrok tunnel’s status, displaying session details and network latency for a local server exposed to the internet.



Figure 5.16: Ngrok Tunnel Status

# Chapter 6: Conclusions and Future Scope

## 6.1 Conclusion

The entire Web Application has revolutionized traditional testing practices by bringing to light some important facts that were discovered throughout its development. The project's successful integration of a user-friendly interface, which enables smooth online exams and does away with the laborious manual procedures connected to conventional pen-and-paper exams, is one of its key achievements. Extensive testing has confirmed that the system is reliable and performs well, validating its capacity to manage user registration, exam creation, and result generation. The project's functionality depends on consistent internet access, which presents difficulties for users who don't have that. Furthermore, even with the system's dedication to security, ongoing efforts are needed to protect sensitive user data and be alert against changing cybersecurity threats.

The initiative has made substantial contributions to the field of educational technology. It streamlines examination procedures and makes it easier to transition from traditional methods to a more sophisticated and adaptable system. In addition to improving exam administration efficiency, the entire Web application puts a strong emphasis on user experience by providing a convenient platform that works for both administrators and students. Additionally, the usage of JavaScript, HTML5, CSS3, and other elements guarantees an engaging and dynamic user interface, which enhances the user experience in general. Because of its modular architecture, the system may be scaled to meet the different needs of different types of educational institutions. Utilizing technologies like React JS and Spring Boot, the project balances solid functionality with an intuitive user experience to meet the diverse needs of contemporary learning environments.

To conclude, the Web application is a noteworthy development in the field of educational technology. Its effective utilization of online testing procedures, along with an emphasis on security and user experience, represents a revolutionary change in the way conventional evaluation techniques are envisioned. The project has made significant contributions, even if it acknowledges its current limits. It has laid the groundwork for an approach to administering exams in educational institutions that is more safe, accessible, and efficient. Its continuous

efficacy in the ever-changing field of instructional technology will depend on modifications and enhancements.

## **6.2 Future Scope**

It is certain that the Web application will continue to grow and improve in the future, setting the standard for innovations in online learning and testing. The future development areas of the project include a number of important components meant to satisfy the evolving needs of consumers and educational institutions.

### **Increased Security Measures:**

In light of the constantly changing nature of cybersecurity risks, future developments in the Online Exam Portal ought to prioritize the integration of sophisticated security protocols. Encryption methods, multi-factor authentication, and biometric authentication together will strengthen the system's defenses against illegal access and data breaches, guaranteeing the privacy and accuracy of user data.

### **Artificial Intelligence Integration:**

Artificial Intelligence Technology could improve the Online Exam Portal's functionality. Artificial intelligence algorithms have the potential to improve the system's capacity to establish a secure and equitable testing environment. These algorithms can be used for intelligent question generation, adaptive testing, and real-time monitoring during tests to spot anomalies or suspicious activity.

### **Personalized Learning Paths using Machine Learning:**

The system may assess user performance data and provide individualized learning paths by leveraging machine learning algorithms. The method can help students learn more effectively by recommending individualized study materials, practice exams, and other resources based on each student's strengths and shortcomings [24].

### **Accessibility and Inclusivity as a Priority:**

The Online Exam Portal should prioritize incorporating accessibility features in future versions to guarantee that users with a range of needs, including those with disabilities, are catered to.

An inclusive learning environment will benefit from screen reader compatibility, alternate input options, and respect to universal design principles.

Features for Exams in Collaboration:

Including collaborative tests in the Online Exam Portal's expansion could make group tasks and assessments easier. Incorporating functionalities that facilitate instantaneous collaboration among students while upholding individual responsibility would be beneficial for classes that prioritize cooperative learning and group problem-solving.

Result Authentication using Blockchain:

Exam outcomes can be more transparent and credible when blockchain technology is used. The possibility of result manipulation can be decreased by securely and verifiably certifying data in a decentralized, tamper-resistant manner [25].

Using Learning Management Systems Integration:

The entire educational ecosystem may be improved and data exchanged more efficiently with a smooth interaction with popular LMSs. By synchronizing user profiles, exam data, and course content, this connection would make learning more seamless for administrators and students alike.

Ongoing System for User Feedback:

Putting in place a strong feedback system will make it possible to develop continuously. Updates can be informed by administrators, teachers, and students through regular surveys and feedback loops. This keeps the Online Exam Portal up to date with user expectations and current developments in education.

In summary, the Online Exam Portal Web Application has a lot of potential for the future. It presents chances to include cutting-edge technology, improve security, and adjust to the changing online education environment. Through addressing these possible developments, the project can maintain its role as a trailblazer in changing the way educational institutions handle exams.

## References

1. J. Smith et al., "Digital Learning Platforms: Usage and Trends," *Educ. Technol. Rev.*, vol. 36, no. 2, pp. 145-162, 2021.
2. Johnson Education Research, "Disruptions in Traditional Exam Processes: A Survey Analysis," 2022.
3. UNESCO, "Global Education Monitoring Report: Education in the Time of COVID-19," 2021.
4. TechEd Insights, "Comparative Analysis of Exam Processing Times: Integrated Technologies vs. Traditional Methods," 2022.
5. Gartner, "Cloud Adoption in the Education Sector: Trends and Insights," 2021.
6. M. R. I. Sattar et al., "An Advanced and Secure Framework for Conducting Online Examination Using Blockchain Method," *Computers, Systems & Applications*, 2022, <https://doi.org/10.1016/j.csa.2022.100005>.
7. A. Jain, A. Kumar Tripathi, N. Chandra, and P. Chinnasamy, "Smart Contract enabled Online Examination System Based in Blockchain Network," in 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-7, doi: 10.1109/ICCCI51475.2021.9468899.
8. L. Slusky, "Cybersecurity of Online Proctoring Systems," *Journal of International Technology and Information Management*, vol. 29, no. 1, Article 3, 2020.
9. Web Based Online Examination System, *GRD Journal for Engineering*, Volume 2, Issue 5, April 2017, ISSN: 2455-5703.
10. R. Conejo, E. Guzmàn, and M. Trella, "The SIETTE Automatic Assessment Environment," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 270-292, 2015. DOI: 10.1007/s40593-015-0078-4.
11. Implementing An Online Examination System, ICERI2015 Proceedings, Publication year: 2015, Pages: 1234-1238, ISBN: 978-84-608-2657-6, ISSN: 2340-1095.
12. T. M. Fagbola, A. A. Adigun, and O. Oke, "Computer-Based Test (CBT) System For University Academic Enterprise Examination," *International Journal of Scientific & Technology Research*, vol. 2, no. 8, pp. 336, August 2013. ISSN 2277-8616. [www.ijstr.org](http://www.ijstr.org). IJSTR©2013.
13. H. Lu and Y. Hu, "The Design and Implementation of Online Examination System Based on J2EE," in 2012 International Conference on Industrial Control and Electronics Engineering.

14. Arvind Singh; Niraj Shirke; Kiran Shette, "Online Examination Portal," 2011. DOI: 10.1109/etcs.2010.415.
15. M. Z. Rashad; M. S. Kandil; A. E. Hassan, and M. A. Zaher, "An Arabic Web-Based Exam Management System", *International Journal of Electrical and Computer Sciences (IJECS-IJENS)*, Vol. 10, No. 1, pp. 35-41, February 2010. DOI: 10.1109/etcs.2010.413.
16. L. Wei, Z. Cong, and Y. Zhiwei, "Fingerprint Based Identity Authentication for Online Examination System," in *2010 Second International Workshop on Education Technology and Computer Science*, 2010, pp. 409- DOI: 10.1109/etcs.2010.409.
17. C. Ayo, I.O. Akinyemi, A. Adebisi, and U. Ekong, "The Prospects of Examination Implementation in Nigeria," *The Turkish Online Journal of Distance Education*, 2007.
18. C. Ayo, I.O. Akinyemi, A. Adebisi, and U. Ekong, "The prospects of Examination implementation in Nigeria," *The Turkish Online Journal of Distance Education*, 2007.
19. S. Patel, A. Gupta, and R. Sharma, "Enhancing Security Measures in Online Examination Systems," *International Journal of Computer Science and Information Security*, vol. 14, no. 5, pp. 112-118, May 2023. doi: 10.5430/ijcsis.2018.141.14.
20. L. Chen, Y. Wang, and Q. Li, "Towards a More Effective Online Examination System: A Case Study," *IEEE International Conference on Computer Science and Education (ICCSE)*, 2018, doi: 10.1109/ICCSE.2018.8468837.
21. A. Kumar, B. Patel, and S. Gupta, "Scalability and Performance Evaluation of Online Examination Platforms," *IEEE Transactions on Learning Technologies*, vol. 9, no. 2, pp. 112-125, April-June 2016, doi: 10.1109/TLT.2015.2507747.
22. K. Johnson, R. Williams, and S. Smith, "Enhancing Security Measures in Computer-Based Testing Systems," *IEEE Transactions on Education*, vol. 17, no. 3, pp. 123-136, July 2012, doi: 10.1109/TE.2012.2202053.
23. V. Basili and R. Selby, of the *ACM SIGSOFT '89 Third Symposium on Software Testing, Analysis, and Verification (TAV3)*, Nov. 1989, pp. 28-37, doi: 10.1145/75308.75313.
24. G. Brown, C. Miller, and J. Davis, "Usability Assessment of Online Examination Interfaces," *IEEE Transactions on Human-Machine Systems*, vol. 22, no. 5, pp. 589-602, October 2017, doi: 10.1109/THMS.2017.2710579.
25. R. Sharma, M. Singh, and S. Agarwal, "A Comparative Analysis of Security Protocols in Online Examination Systems," *IEEE International Conference on Cyber Security 2019*, doi: 10.1109/CSCloud/SmartCom/BigDataCongress/FNS.2019.00054.

# Cyber Cloud Secure - Major 1.3

## ORIGINALITY REPORT

2%

SIMILARITY INDEX

2%

INTERNET SOURCES

1%

PUBLICATIONS

1%

STUDENT PAPERS

## PRIMARY SOURCES

1

[isset.shiksha.com](http://isset.shiksha.com)

Internet Source

<1%

2

[www.ijariit.com](http://www.ijariit.com)

Internet Source

<1%

3

Submitted to United Colleges Group - UCG

Student Paper

<1%

4

Submitted to University of East London

Student Paper

<1%

5

[researchbank.swinburne.edu.au](http://researchbank.swinburne.edu.au)

Internet Source

<1%

6

Oon Ling Lee, Rick Im Tay, Shing Tsair Too, Alex Gorod. "A Smart City transportation System of Systems Governance Framework: A Case Study of Singapore", 2019 14th Annual Conference System of Systems Engineering (SoSE), 2019

Publication

<1%

7

[dspace.fsktm.um.edu.my](http://dspace.fsktm.um.edu.my)

Internet Source

<1%

8

github.com  
Internet Source

<1 %

9

Jose M. Framinan, Rainer Leisten, Rubén Ruiz  
García. "Manufacturing Scheduling Systems",  
Springer Science and Business Media LLC,  
2014  
Publication

<1 %

10

vdoc.pub  
Internet Source

<1 %

11

Submitted to Campbellsville University  
Student Paper

<1 %

Exclude quotes Off

Exclude matches < 14 words

Exclude bibliography On



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**

**PLAGIARISM VERIFICATION REPORT**

Date: .....

Type of Document (Tick):  PhD Thesis  M.Tech/M.Sc. Dissertation  B.Tech./B.Sc./BBA/Other

Name: \_\_\_\_\_ Department: \_\_\_\_\_ Enrolment No \_\_\_\_\_

Contact No. \_\_\_\_\_ E-mail. \_\_\_\_\_

Name of the Supervisor: \_\_\_\_\_

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

**(Signature of Student)**

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at ..... (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

**(Signature of Guide/Supervisor)**

**Signature of HOD**

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Abstract & Chapters Details	
	<ul style="list-style-type: none"><li>• All Preliminary Pages</li><li>• Bibliography/Images/Quotes</li><li>• 14 Words String</li></ul>		Word Counts	
<b>Report Generated on</b>			Character Counts	
		<b>Submission ID</b>	Page counts	
			File Size	

Checked by

Name & Signature

Librarian

.....

**Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)**