Online Exam Portal

Project Report Submitted by

ANJU JOHN

Reg. No.: AJC21MCA-2031

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2022-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "ONLINE EXAM PORTAL" is the bonafide work of **ANJU JOHN (Regno: AJC21MCA-2031)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

Ms. Rini Kurian	**Ms. Meera Rose Mathew**
Internal Guide	**Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **“ONLINE EXAM PORTAL”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfillment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date:	**ANJU JOHN**
KANJIRAPPALLY	**Reg: AJC21MCA-2031**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms Rini Kurian** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ANJU JOHN

ABSTRACT

Online Exam Portal project in java is a web application that provides a platform for conducting online examinations. This Project is performing all the crud operations like z CREATE/RETRIEVE/UPDATE/DELETE functions. It is a web application through which examiners can conduct online examinations and students can attempt those examinations.It helps students in self assessing their results. Admin can add examination and enter an objective-type question-answers along with the correct answers.Students need to enroll and give the examination.Just after submitting, the students will be able to review their submissions and check the results.This project suggests integrating real-time object detection technology to improve the safety and validity of the online assessments. The system may actively monitor and identify unauthorised objects or behaviours during the testing process by utilising object detection algorithms. This ensures a fair testing environment and aids in preventing cheating. In order to identify and detect items in real-time, the real-time object identification module makes use of cutting-edge deep learning models like SSD or YOLO. It examines the pupils' camera or screen-sharing feed to look for signs of possible cheating, like mobile devices or unauthorised goods. When suspicious things are found, the proper responses can be made, such as alert generation or contacting the exam administrator.

CONTENT

List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Online Exam Portal Project in java is a web application that provides a platform for conducting online examinations.This project is performing all the crud operations like CREATE/RETRIEVE/ UPDATE/DELETE functions. It is a web application through which examiners can conduct online examinations and students can attempt those examinations. It helps in self assessing their results. This will permit to take and give online examination and maintaining master information and generating various reports of test. The main users of the project are Student or user ,Examiner and system Administrator. From an end-user perspective, the e-Examination System Project consists of giving examination, manage profile. And Examiner module for Create/Update/Retrieve questions,quiz and category. And Administration module for Mange members, examination, question bank.This proposal recommends the incorporation of real-time object detection technology in order to improve the security and integrity of the online tests. During the inspection process, the system can actively monitor and identify unauthorised objects or behaviours by utilising object detection algorithms. This promotes honesty throughout testing and helps to prevent cheating.

1.2 PROJECT SPECIFICATION

This project is a web application through which examiners can conduct online examinations and students can attempt those examinations.

There are three types of users in the application:

ADMIN :-
Admin can view the complete website

EXAMINER :-
Examiner can ADD examination and can VIEW/EDIT/DELETE the list of examinations . Also he can ADD questions and their respective answers

USER (STUDENT) :-
User can VIEW the examination list and can give examination .They can VIEW the result list and also can view his/her Profile.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Data collection and analysis, problem diagnosis, and system change recommendations are all steps in the process of system analysis. Throughout this problem-solving process, the system users and system developers must communicate often. Every system has an important analysis or study phase. process of creation. The system is painstakingly reviewed and evaluated. The system analysts take on the role of interrogators and probe thoroughly into how the current framework functions. The system's input is considered in its entirety and includes identified. It is possible to connect the various processes to the organisations' outputs. The aim of analysis is to identify the problem, identify its root cause, analyse and synthesise the many different parts, take into account important and deciding factors, and choose an ideal, or at the very least, an appropriate, solution or course of action. The procedure needs to be carefully examined using a range of methods, such as questionnaires and interviews. To make a choice, it is necessary to thoroughly study the facts from these sources. knowing how the system functions. The existing system must be terminated. Research identifies problem areas. The designer is currently working to find solutions to the problems the company is having. There are medusae offered as ideas. The current system is then analytically compared to the proposal. The offer is explained to the user, and they are prompted to agree.

2.2 EXISTING SYSTEM

The entire test-assignment process—including calculating the results—was carried out both manually and automatically till date. When the programme was not installed, processing the exam paper, including checking and distributing the results, took time.But in the case of automatic calculation,time consuming is less compared to manual calculation.

2.2.1 NATURAL SYSTEM STUDIED

In order to incorporate a natural system into an online exam portal, it is necessary to research and comprehend how people naturally interact and participate in exams. This entails investigating user behaviour, preferences, and cognitive processes in order to create user interfaces and functionalities that are consistent with human perception and learning styles. The online test portal seeks to develop an intuitive and user-friendly environment that increases user performance and happiness by studying the natural system. This environment will enhance the exam experience.

2.2.2 DESIGNED SYSTEM STUDIED

The system that is intended to be used for an online exam portal consists of user management for registration and authentication, exam management for creating and scheduling exams, a question bank for storing and organising questions, exam taking functionality for participants to attempt exams online, and result processing for grading and providing feedback to users.

2.3 DRAWBACKS OF EXISTING SYSTEM

- To take exams of more candidates, more invigilators are needed, but there is no need for an invigilator in the case of an online exam.
- The results are not precise because calculation and evaluation are done manually.
- The chances of paper leakage are higher in the current system than they are in the proposed system.
- The result processing takes longer because it is done manually.

- Result processing takes more time as it is done manually

2.4 PROPOSED SYSTEM

The suggested method for an online test portal entails thoroughly examining the shortcomings of the current system and finding potential areas for enhancement. New additions and improvements are suggested after thorough research and user feedback, including improved security measures, advanced question kinds, automatic grading, real-time feedback, and customised test experiences. The proposed system seeks to improve upon the weaknesses of the current system and offer a more reliable, effective, and user-focused online test platform.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- In comparison to the present system, the proposed system will take less time to complete tasks and is more effective.
- Analysis is made simple because it is automated.
- Results are precise and accurate and are announced quickly because the simulator performs calculations and evaluations itself.
- The proposed system is very secure because there is no chance of question paper leakage because it is solely dependent on the administrator.
- The records of applicants who appeared and their scores are kept and can be restored for use in the future.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study examines the effects that the creation of a system will have on an organisation.The effect could be favourable or unfavourable.It is done to see if the work, time, and effort put into the project will be worth it when it comes to achieving the organization's goals. It is a component of any project's or plan's early design phase. It is carried out to unbiasedly identify the advantages and disadvantages of a potential venture or an established company. The resources needed for the project, the likelihood of success, and the opportunities and hazards existing in the surrounding environment can all be identified and evaluated. The following questions are being investigated in order to provide answers:

1. Does the company have the necessary tools and technology?
2. Will the company's investment yield a high enough return?

3.1.1 Economical Feasibility

Analyses of costs and benefits are necessary to support the developing system. criteria to ensure that the project that would produce the best outcomes the quickest is given priority. One of the variables is the cost associated with establishing a new system. Following are a some of the main financial questions that were brought up during the original investigation:

- The cost of the software and hardware.
- the cost of doing a thorough system analysis.
- the benefits in terms of lower costs or fewer costly errors.

There are no manual costs involved with the suggested system because it was developed as part of a project. Additionally, the availability of all necessary resources suggests that the system might be implemented at a reasonable cost.

The system will use cutting-edge tools, thus the money spent on tools, system design, and development will be a wise investment.The advantages of adopting the system are not monetary; rather, they consist of more frequent communication between users and managers

3.1.2 Technical Feasibility

The system must go through a technical assessment first. An overview design of the system's requirements in terms of input, output, programmes, and procedures must serve as the foundation for the viability assessment. After identifying an outline system, the inquiry must next recommend the type of equipment, necessary steps for building the system, and methods of operating the system once it has been designed. Throughout the inquiry, the following technological issues were encountered:

- Does the proposed technology work with the existing technology?

- Can the system expand with improvements?

The project should be planned so that the required performance and functionality are met within the limits. The project calls for a high resolution scanning device and incorporates cryptographic techniques. The fact that a newer version of the same software still functions with an older version means that the system can still be used even though the technology may become outmoded with time.It establishes if a project can be created with the available technologies.The front-end and back-end necessary for the system are already installed and available, making it technically feasible. The system is created using Spring Boot as the back-end and Angular. It is technically feasible.

3.1.3 Behavioral Feasibility

People are naturally averse to change, yet computers have been shown to make changes easier.An assessment of the user's propensity to advance computerised systems should be done.To guarantee adequate authentication and the protection of sensitive data, there are different tiers of users. The suggested system includes the following questions:

- Is there enough user assistance?
- Will the proposed system cause any harm to anyone?

When created and implemented, the project would be advantageous since it will achieve the goals. The project is determined to be behavioural after carefully weighing every behavioural feature.

3.1.4 Questionnaire

Interview Q&A with Aparna (Student at Hindustan College of Pharmacy)

1.How online exam benefit students?

- No travelling difficulties
- Ensures safety of students from diseases

2.Did network issues create adverse effect on attending online exam?

- Yes,it creates tension,anxiety among students
- Creates a feeling that it is possible to attend the exam.

3.How much security do you feel in attending online exams?

- I think online examinations can detect cheating more easily

4.Is it possible you to self assess your results?

- Yes,I am able to self assess the results through marklist displayed on the website

5. Do you confused in how to take a online test?

- Yes

6. How will be the online exam pattern?

- It contains a set of questions and options.We need to click the correct answer submit it.The result will be published at the same time.

7. How many questions are there in an online exam?

- It contains a minimum of 10 questions and maximum of 20 questions or more

8. Is it possible to attend all questions in a specified timelimit?

- Sometimes

9. Is it possible to do malpractices in an online exam?

- No,it is highly secure

10. Which one do you prefer - Online or offline?

- Both have merits and demerits.I prefer online mode.Because it is very comfortable

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i5
RAM	-	8 G B
Hard disk	-	1 TB

3.2.2 Software Specification

Front End	-	ANGULAR
Backend	-	SPRING BOOT

Client on PC - Windows 7 and above.

Technologies used - Spring boot(java),Angular,Typescript,Hibernate

3.3 SOFTWARE DESCRIPTION

3.3.1 JAVA

Java is a set of computer software and specifications developed by James Gosling at Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is utilised in a wide range of computing systems, including supercomputers, embedded devices, and mobile phones. Less popular than standalone Java applications, Java applets were frequently used to provide many native application functionality by being embedded in HTML pages. They were typically executed in secure, sandboxed environments.

The most common technique to create code that will be deployed as byte code in a Java virtual machine (JVM) is to write it in the Java programming language; byte code compilers are also available for other languages, such as Ada, JavaScript, Python, and Ruby.In addition, a number of languages, such as Clojure, Groovy, and Scala, have been created to operate natively on the JVM. Although Java's object-oriented features are modelled after Smalltalk and Objective-C, its grammar significantly references C and C++.[10] While some implementations, like those all currently supported by Oracle, may use escape analysis optimisation to allocate on the stack in place of the heap, all variables of object types are references in Java, which avoids some low-level constructs like pointers. The JVM performs integrated automated garbage collection to handle memory management.

3.3.2 Hibernate

Hibernate is free software that is distributed under the GNU Lesser General Public License 2.1. The main capabilities of Hibernate are mapping Java classes to database tables and Java data types to SQL data types. Additionally, Hibernate offers tools for data retrieval and query. By generating SQL calls, it frees the developer from having to manually handle and convert the result set into objects.

An object-relational mapping tool for the Java programming language is called Hibernate ORM (or just Hibernate). A framework for converting an object-oriented domain model to a relational database is provided. By substituting high-level object handling operations for direct, permanent database accesses, Hibernate solves the problem of object-relational impedance.

3.3.3 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most wellknown Open Source SQL database management system. On the MySQL website, you may find the most latest information about the MySQL programme.

- MySQL is a database management system.

A planned collection of data is called a database. Anything might be it, including a straightforward shopping list, a photo gallery, or the enormous amount of data in a company network. A database management system, such as MySQL Server, must be used to add to, access, and handle the data contained in a computer database. Database management systems—whether used as independent programmes or as a component of other applications—are essential to computing because computers are so adept at processing massive volumes of data.

- MySQL databases are relational.

Instead of combining all the data into one huge warehouse, a relational database keeps it in individual tables. Physical files that have been speed-optimized are used to hold the database structures. The logical model, which consists of objects like databases, tables, views, rows, and columns, provides a flexible programming environment. One-to-one, one-to-many, unique, required or optional, and "pointers" between other tables are just a few examples of the rules you may make to control the relationships between various data fields. Your application won't ever encounter inconsistent, duplicate, orphan, out-of-date, or missing data since a well-designed database enforces these limitations. The term "Structured Query Language" is abbreviated "SQL" in MySQL. The most popular standard language for accessing databases is SQL. Depending on your programming environment, you might openly enter SQL (for example, to generate reports), embed SQL statements within other languages' code, or use a language-specific API that conceals the SQL syntax. The ANSI/ISO SQL Standard defines SQL. Since its inception in 1986, the SQL standard has seen a lot of modifications. "SQL92," a 1992 standard, is mentioned in this book. The standard's 1999 version is referred to as "SQL: 1999," and its most recent revision is "SQL: 2003." "The SQL standard" refers to the SQL Standard as it is at any given time.

- MySQL software is Open Source.

Considering that the software is open source, anyone can use and alter it. Anyone can use and download the MySQL software for free online. You have the right to look at the source code and make any necessary changes. The MySQL software adheres to the GPL (GNU General Public Licence), which outlines what you can and cannot do with the programme under specific conditions. You can purchase a commercially licenced version from us if the GPL bothers you or if you need to combine MySQL code into a for-profit application. For further details, see the MySQL Licencing Overview.

- The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If it is what you're after, you should try it. Along with your other apps, web servers, and other software, MySQL Server may run effortlessly on a desktop or laptop and requires little to no maintenance. You can adjust the parameters to fully utilise the RAM, CPU, and I/O power if you dedicate a whole system to MySQL.

- MySQL Server works in client/server or embedded systems.

The MySQL Database Software (APIs) is a client/server system that includes a multi-threaded SQL server, numerous client programmes and libraries, administrative tools, and a large range of application programming interfaces. Additionally, we provide MySQL Server as an integrated multi-threaded library that you can add to your applications to create a standalone solution that is smaller, quicker, and simpler to use

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered system or product's development process starts with design. Designing is a creative process. An efficient system depends on a good design. In order to fully detail a process or system so that it may be physically executed, "design" refers to the act of applying a variety of methodologies and concepts. It can be described as the process of using a variety of methodologies and concepts to precisely specify a part, a process, or a system to permit its physical realisation. Software design is the technical backbone of the software engineering process, regardless of the development paradigm used. The required architectural detail is produced by the system design.

4.2 UML DIAGRAM

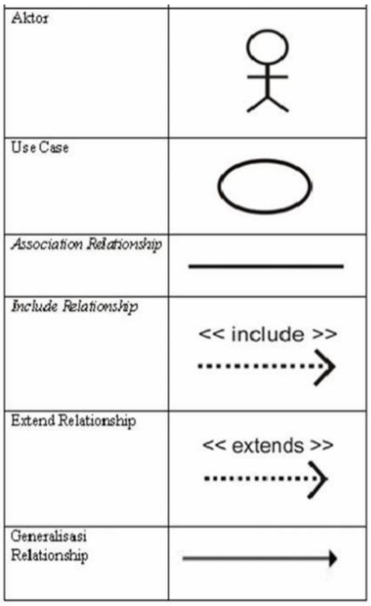
The Unified Modelling Language (UML), a language used in the field of software engineering, is used to represent the elements of the concepts of object-oriented programming. The complete software architecture or structure is defined by it as the industry standard. By treating them as objects or entities, complex algorithms are solved and interacted with in object-oriented programming. These objects can be anything. Either a bank manager or the actual bank could be to blame. It could be a machine, a creature, a car, etc. Even if they are capable of performing their responsibilities and should, the issue is how we interact with and manage them. Tasks include interacting with other things, communicating data across objects, controlling other items, etc. In a single piece of software, there could be hundreds or even thousands of objects.

UML includes the following nine diagrams.

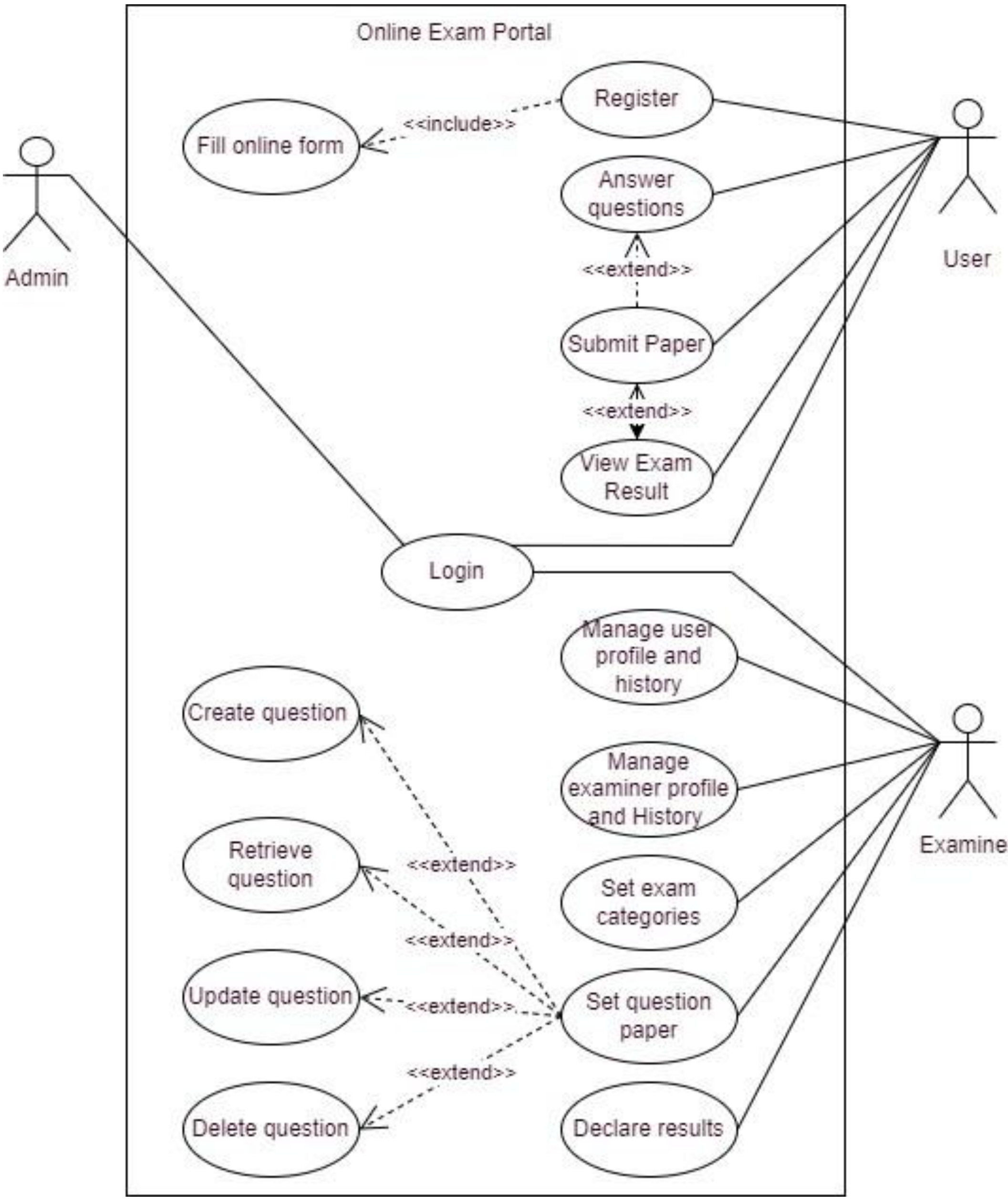
- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

An illustration of the interactions between system components is a use case diagram. A use case is a method for locating, defining, and organising system needs. In this context, the word "system" refers to a project or business that is currently being worked on or run, such a website for selling and providing services for mail-order goods. Use case diagrams are a tool that the Unified Modelling Language (UML) makes use of. a standard notation for replicating real-world systems and objects. One of the system goals is to plan for general needs.Verifying a hardware design, testing, and debugging a software product delivering a consumer assistance, creating, penning an online instruction manual, or undertaking a targeted activity For example, ordering of items, catalogue editing, transaction processing, and client are use cases in a context of product sales.



Aktor	
Use Case	⬭
Association Relationship	——
Include Relationship	<< include > ┈┈┈>
Extend Relationship	<< extends >> ┈┈┈>
Generalisasi Relationship	——→


Online Exam Portal
Admin
User
Examiner
Fill online form
<<include>>
Register
Answer questions
<<extend>>
Submit Paper
<<extend>>
View Exam Result
Login
Manage user profile and history
Manage examiner profile and History
Set exam categories
Set question paper
Declare results
Create question
Retrieve question
Update question
Delete question
<<extend>>
<<extend>>
<<extend>>
<<extend>>

4.2.2 SEQUENCE DIAGRAM

A sequence diagram fundamentally depicts the sequential order in which events occur or how they interact with one another. Event diagrams and event scenarios are other names for sequence diagrams. Sequence maps specify the actions and the sequential sequence in which the system's components occur. These schematics are commonly used by businesspeople and software developers to grasp specifications for both existing and upcoming systems.

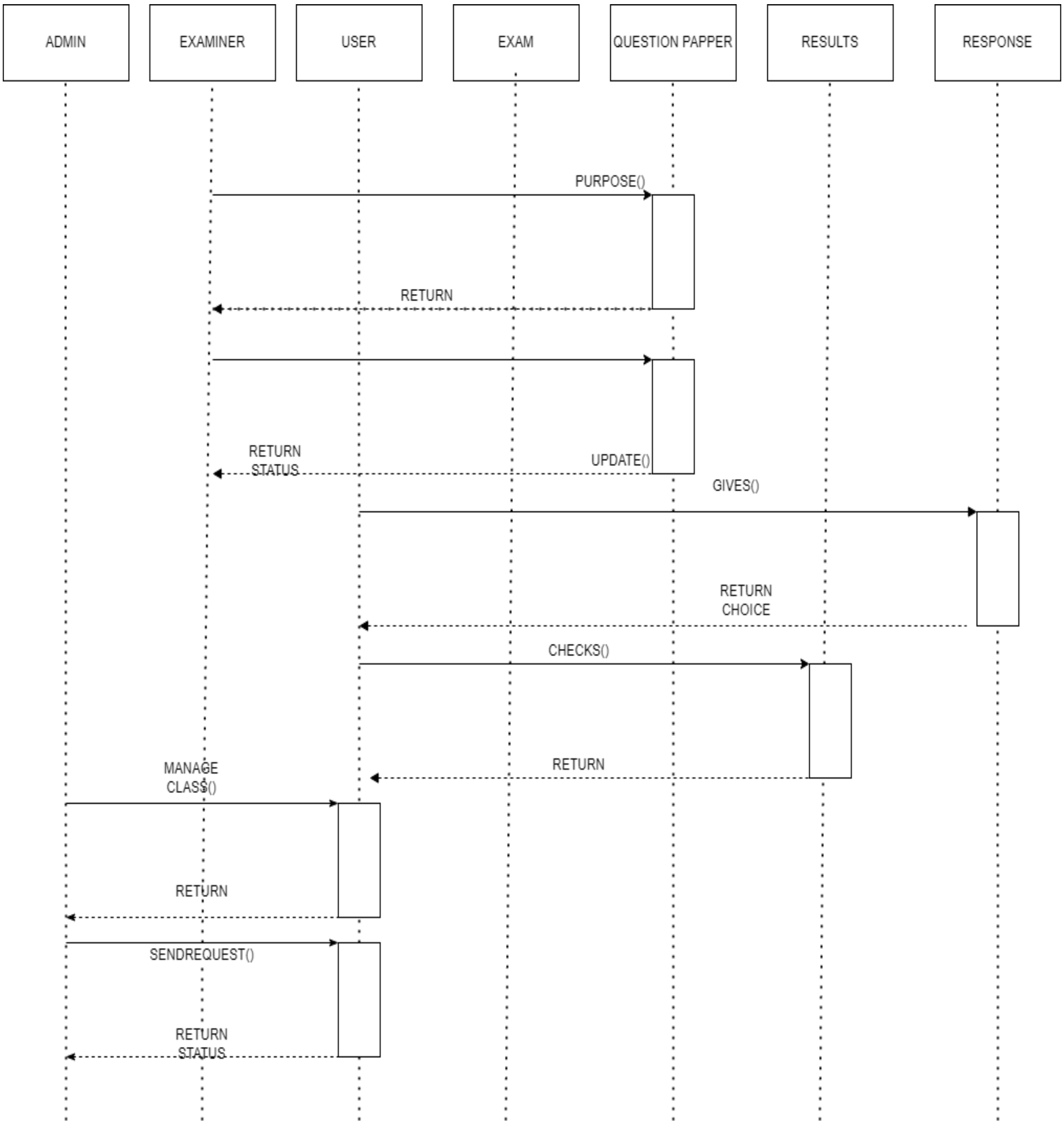
Sequence Diagram Notations –

Actors –In a UML diagram, an actor represents a particular kind of role in which it communicates with the system's objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

Lifelines – A named piece that shows a specific participant in a sequence diagram is called a lifeline. In essence, a lifeline represents each incident in a sequence diagram. The lifeline components in a sequence diagram are at the top.

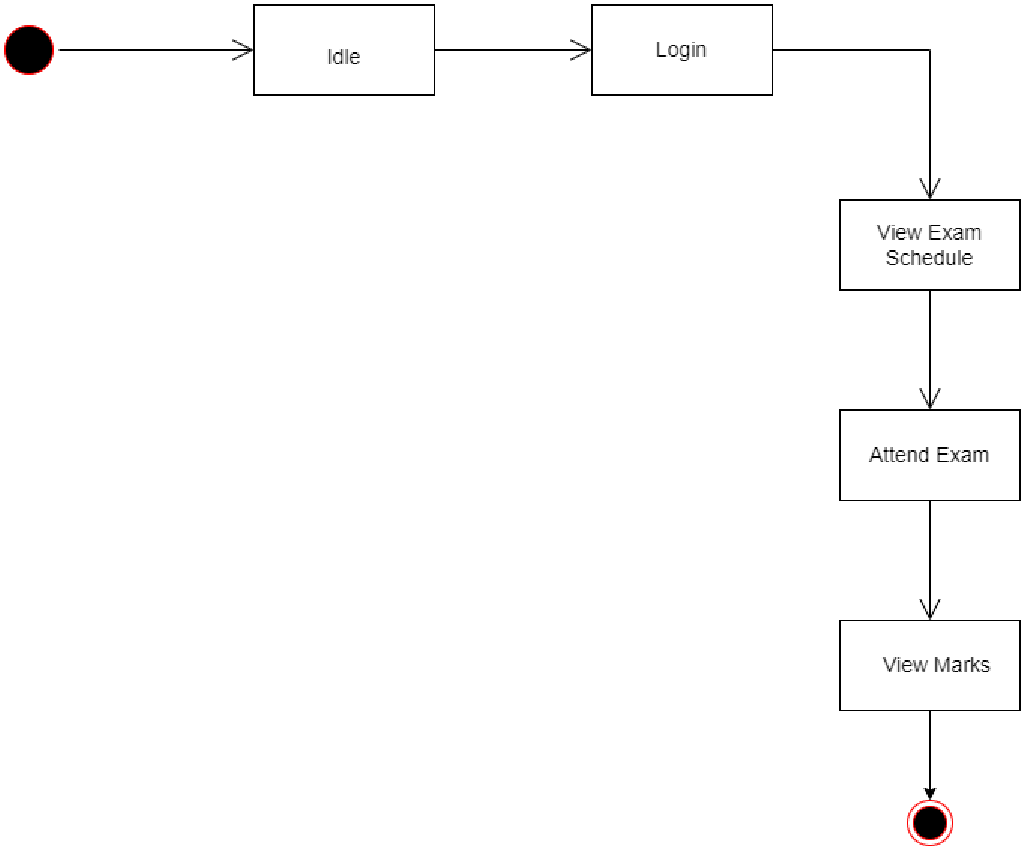
Messages –Messages are used to show how objects communicate with one another. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Guards –In the UML, we utilise guards to model circumstances. When we need to limit the flow of messages while pretending that a condition is met, we utilise them. Software engineers rely on guards to inform them of the limitations imposed by a system or specific process.


ADMIN
EXAMINER
USER
EXAM
QUESTION PAPPER
RESULTS
RESPONSE
PURPOSE()
RETURN
RETURN STATUS
UPDATE()
GIVES()
RETURN CHOICE
CHECKS()
RETURN
MANAGE CLASS()
RETURN
SENDREQUEST()
RETURN STATUS

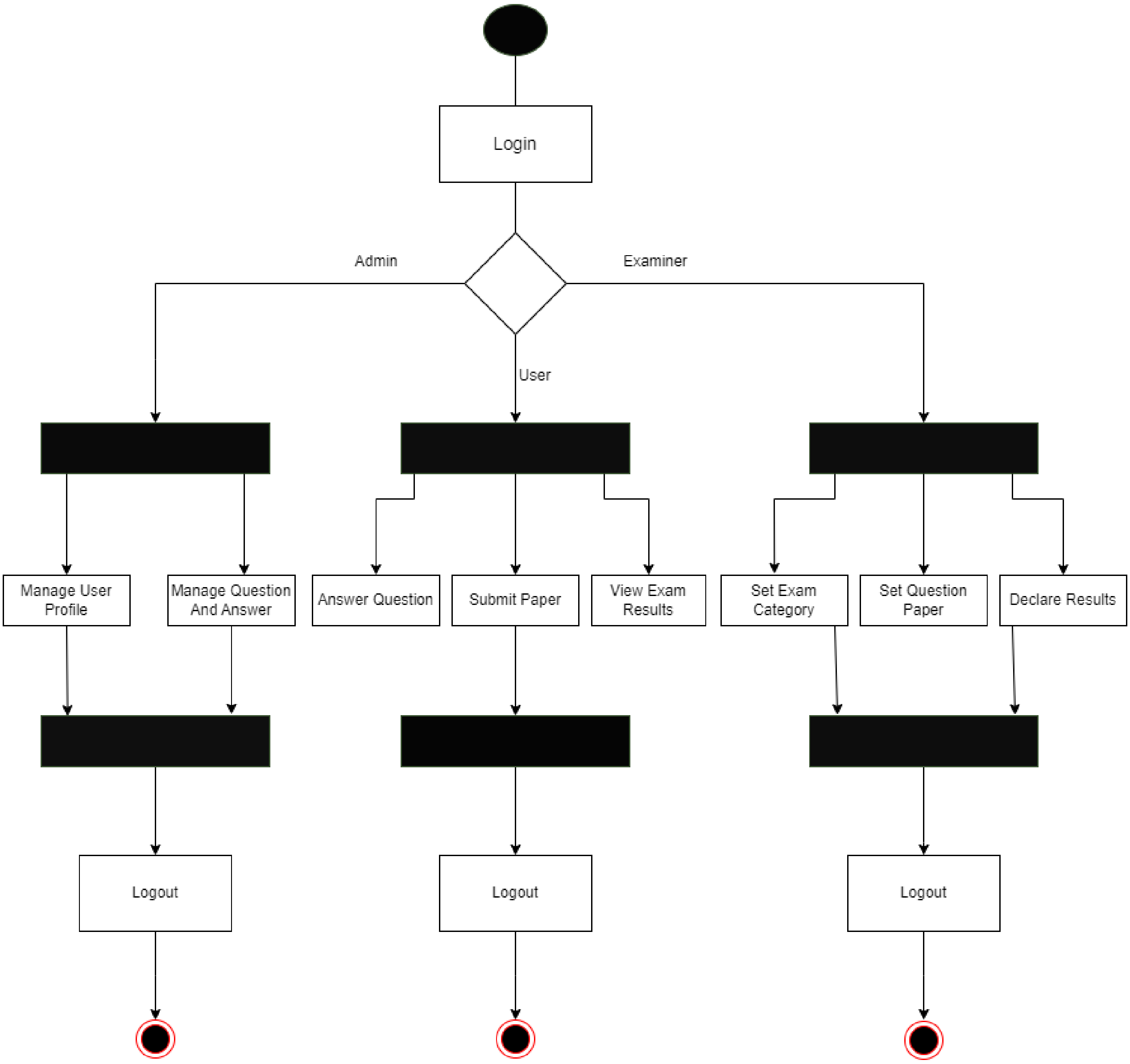
4.2.3 State Chart Diagram

The state chart diagrams are used to illustrate how the system behaves. The state chart diagram is a type of UML diagram that can be used to show the behaviour of a class, subsystem, package, or even the complete system. State transition diagram is another name for state cart diagram. Using state chart diagrams, we may investigate how a system interacts with outside entities. The event-based systems are represented using state charts. An event can be used to modify the state of an object. To depict the many states of an entity within an application system, state chart diagrams are utilised.



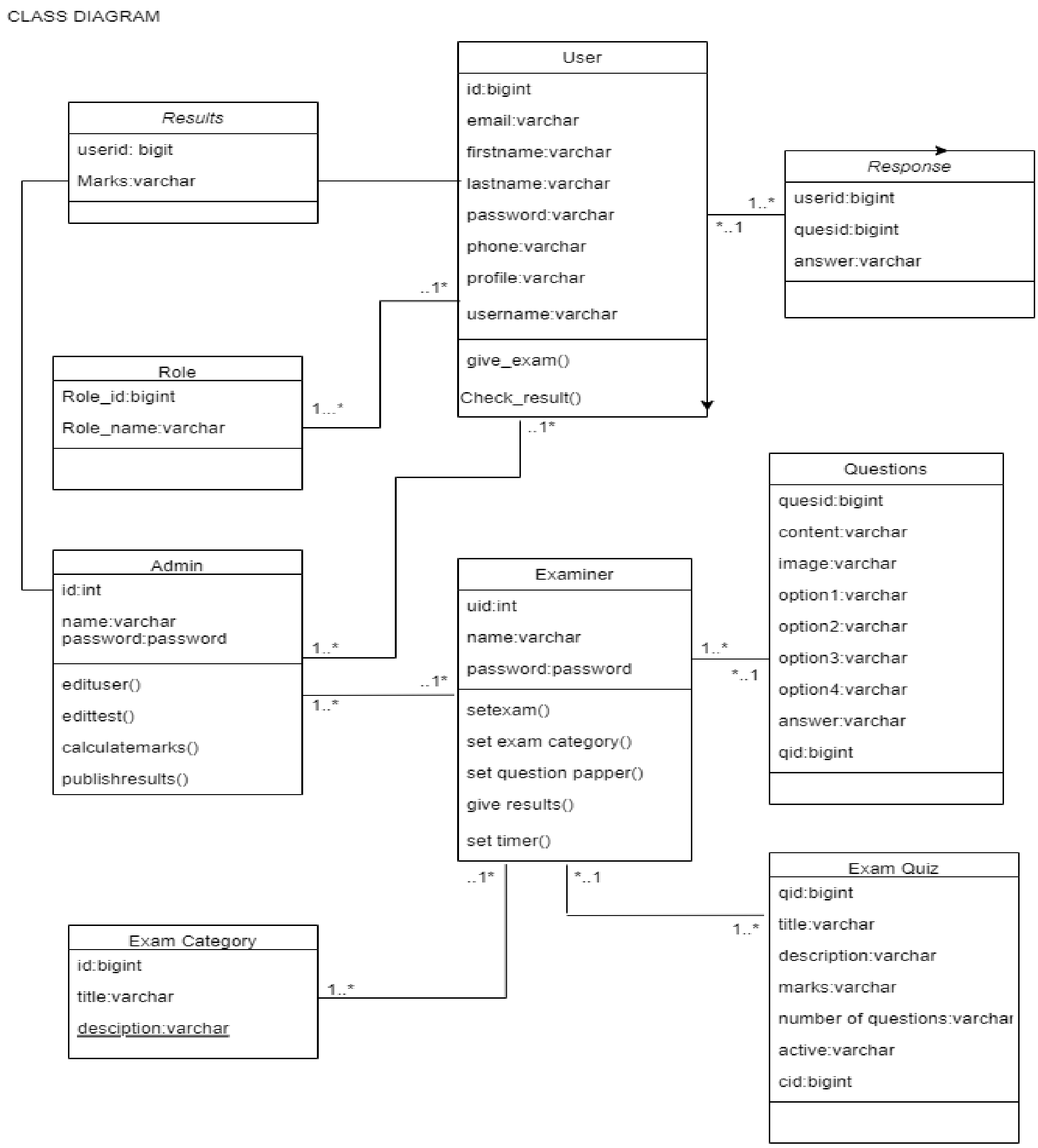
4.2.4 Activity Diagram

Activity diagrams show the connections between activities at different levels of abstraction to offer a service. An event is often completed by a few operations when events in a single use case are connected to one another and activities may overlap and require coordination. It can also be used to simulate how a group of use cases used to illustrate business processes coordinate one another. We may investigate the interplay between numerous activities in the system using activity diagrams.


Login
Admin
Examiner
User
Manage User Profile
Manage Question And Answer
Answer Question
Submit Paper
View Exam Results
Set Exam Category
Set Question Paper
Declare Results
Logout
Logout
Logout

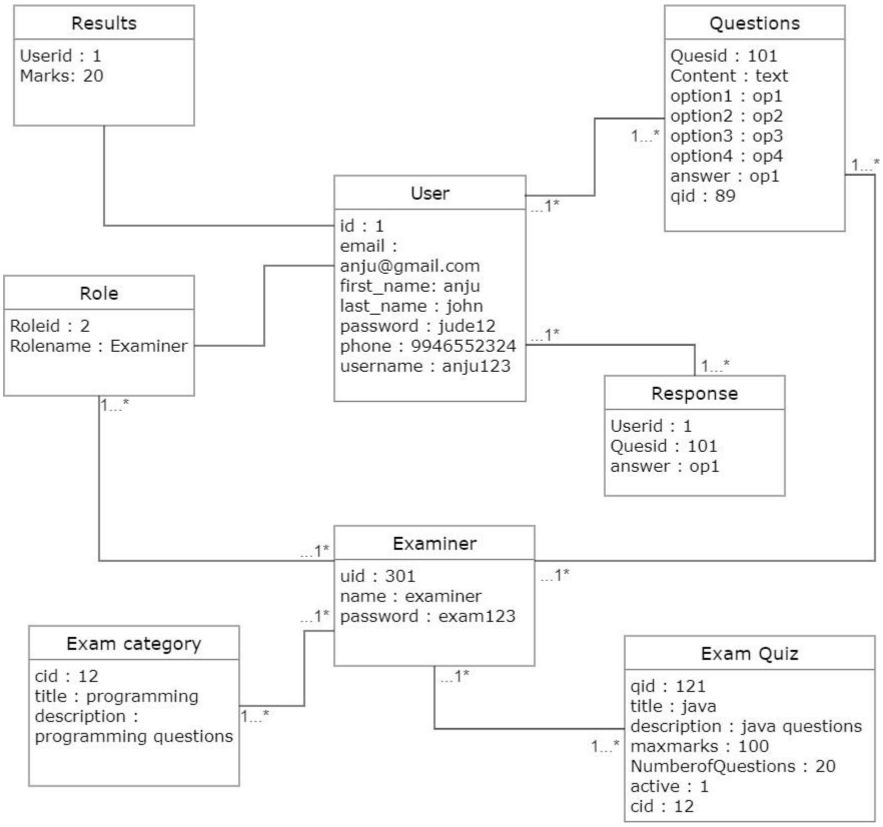
4.2.5 Class Diagram

A class diagram is a type of static diagram.These diagrams are used to illustrate the system's static nature or point of view. Class diagrams are used to produce the executable code for software programmes. Class diagrams can also be used for documentation, describing a variety of system components, and visualising data. Class diagrams show all the characteristics and constraints that apply to a class. Because it is the only UML diagram that can be linked to object-oriented languages, class diagrams are typically used for building object-oriented systems. A group of classes, interfaces, affiliations, collaborations, and limits are shown in a class diagram. Structural diagram is another name for a class diagram.



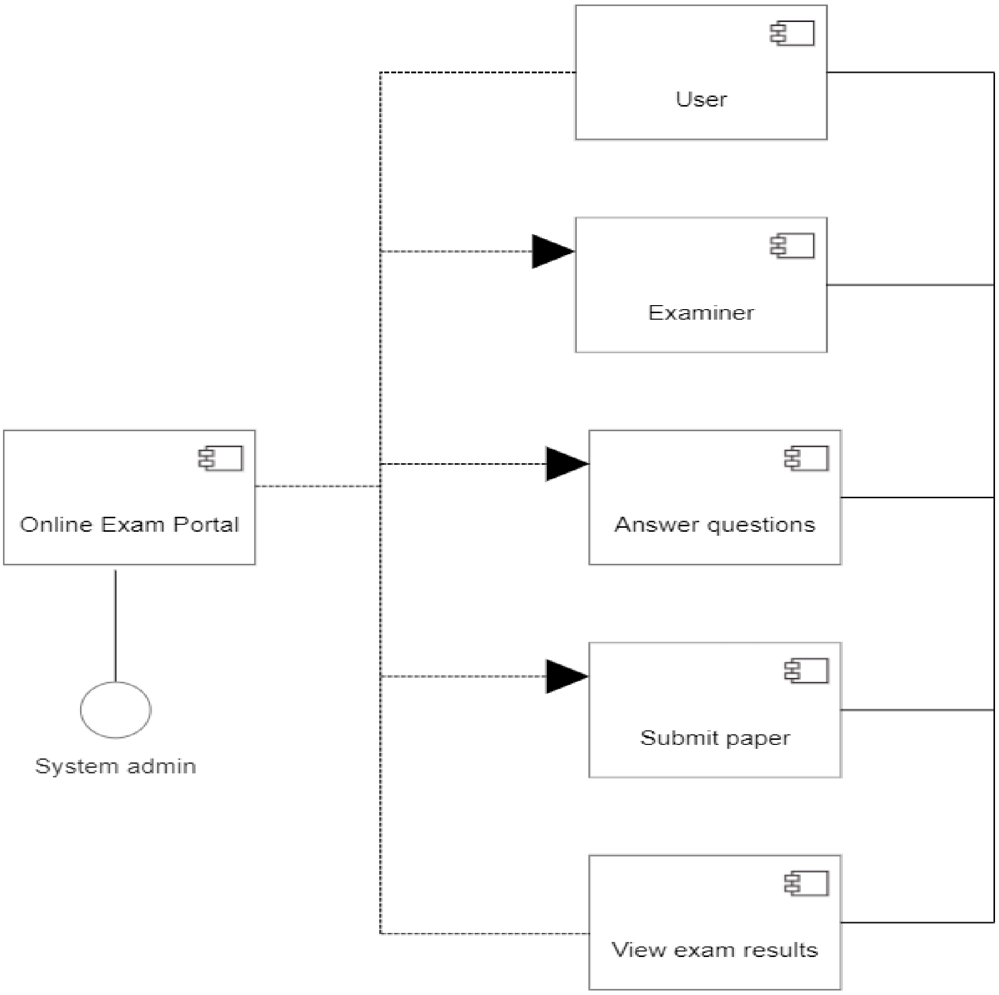
4.2.6 Object Diagram

These are the types of diagrams that can be created using a class diagram. Since objects are instances of classes, object diagrams are entirely dependent on class diagrams. For class diagrams and object diagrams, the fundamental ideas and approaches are nearly identical. Object diagrams also depict a system's static view, which is a snapshot of the system taken at a certain time. A group of items and their connections are typically represented using object diagrams.



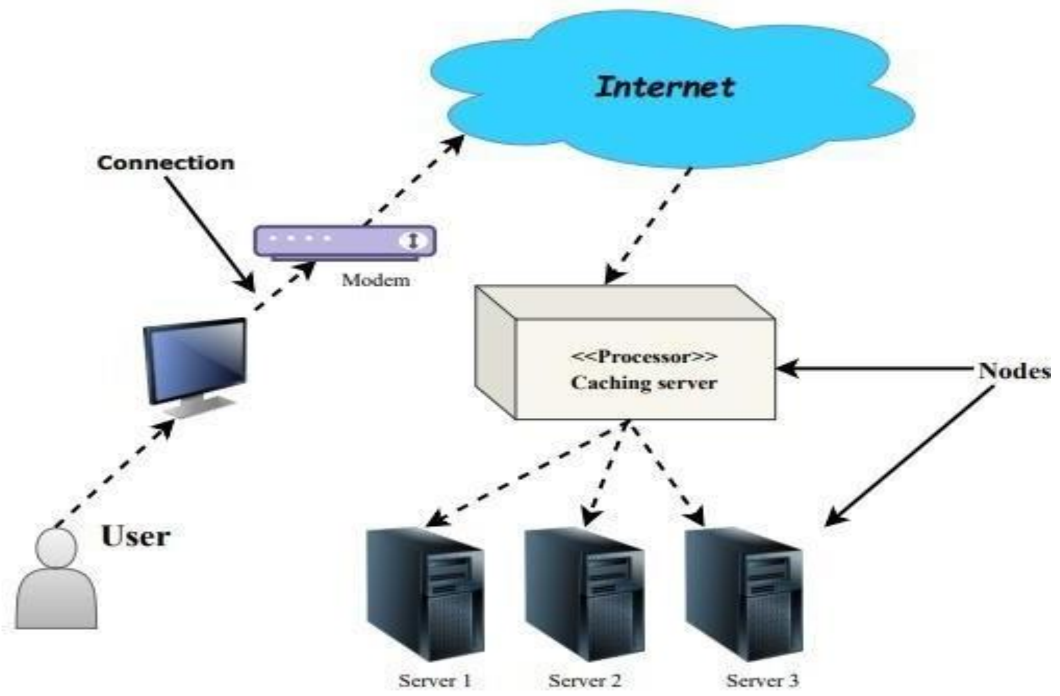
4.2.7 Component Diagram

Component diagrams differ in their appearance and function. To show the actual parts of a system, component diagrams are utilised. A node physically contains different kinds of applications, libraries, files, documents, and so on. The structure and interrelationships of a system's components are described in component diagrams. These designs are also used to create systems that are performable.



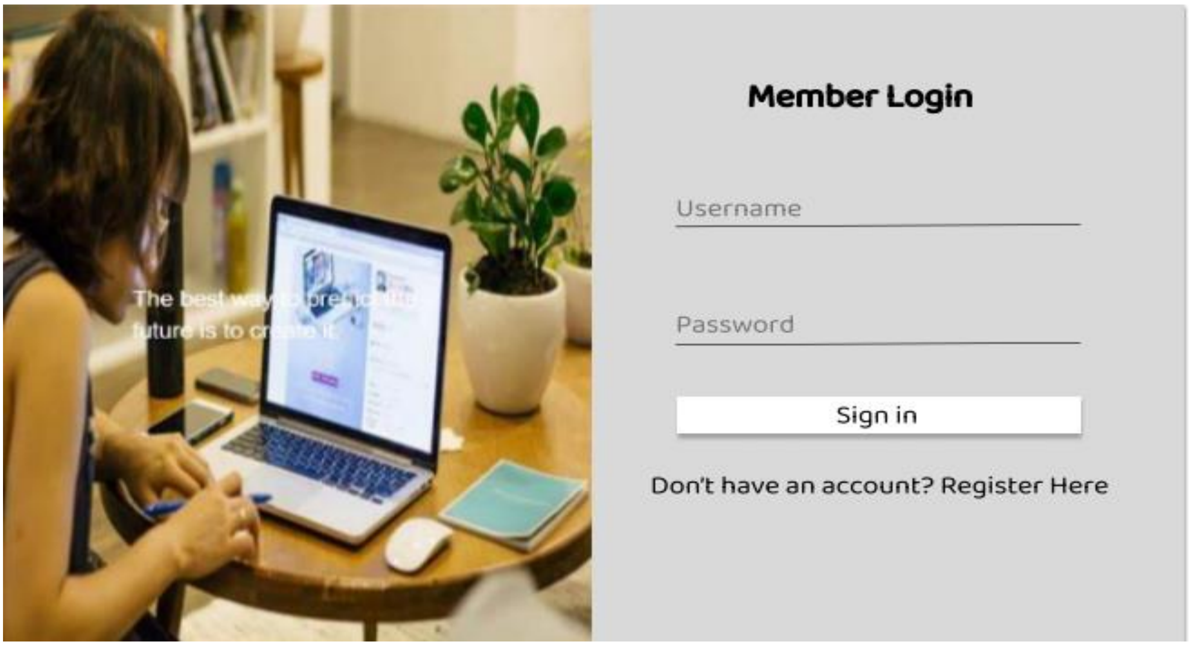
4.2.8 Deployment Diagram

Deployment diagram is type of UML diagram which is used to model the physical architecture of the system. It models the relationships between the software and hardware components. It depicts the execution architecture of the system. It describes about the static deployment view of the system. The component of the deployment diagram includes nodes, artifacts, communication association, device, and deployment specifications.



4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Login



Member Login

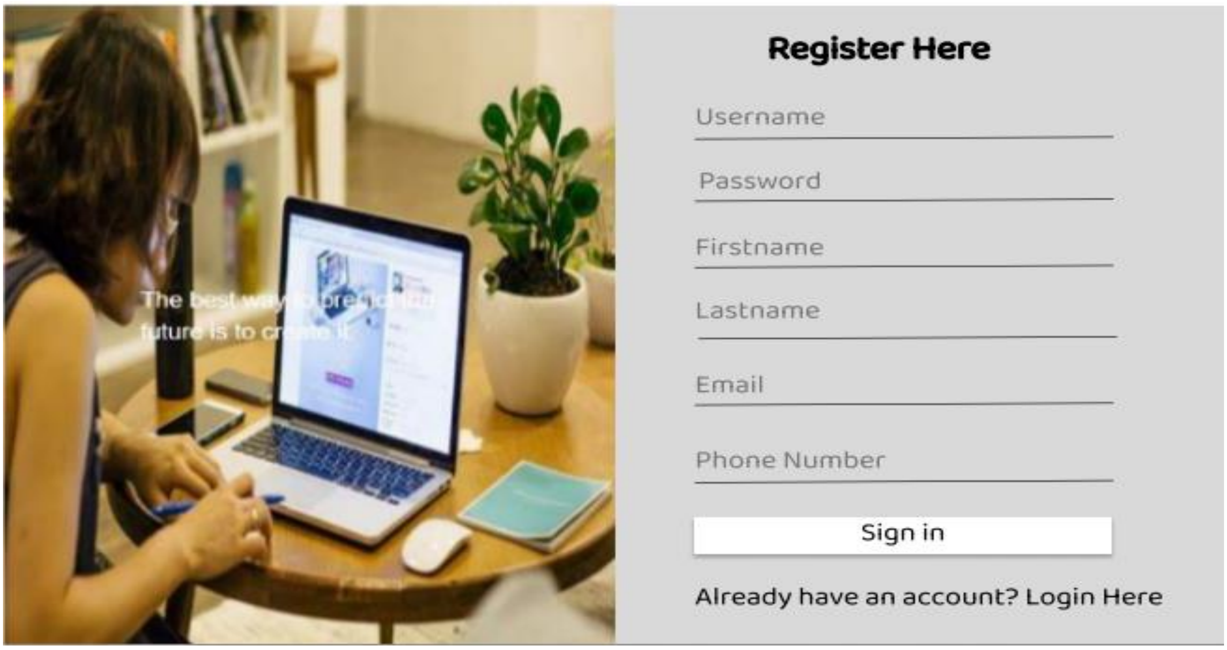
Username

Password

Sign in

Don't have an account? Register Here

Form Name: Register



Register Here

Username

Password

Firstname

Lastname

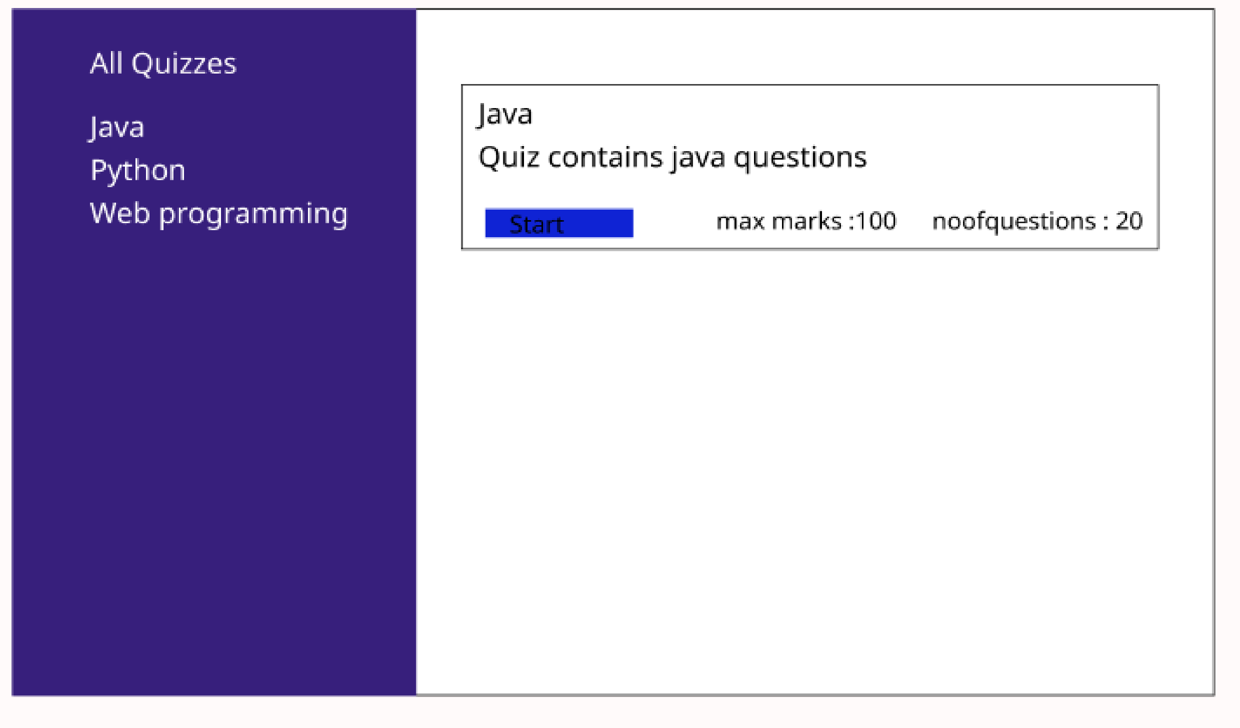
Email

Phone Number

Sign in

Already have an account? Login Here

Form Name : User-dashboard



All Quizzes

Java

Python

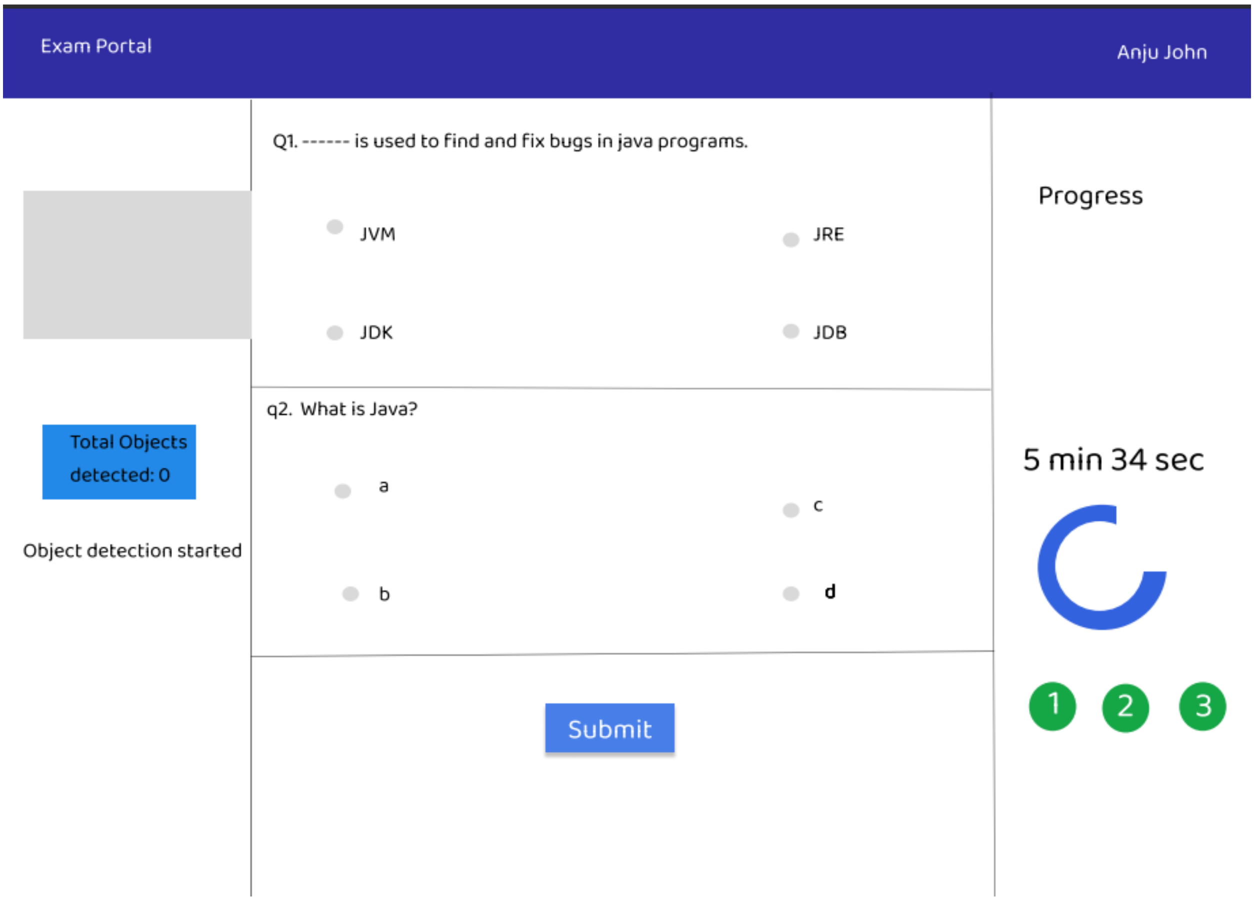
Web programming

Java

Quiz contains java questions

Start max marks :100 noofquestions : 20

Form Name : Start-Exam



Q1. ------ is used to find and fix bugs in java programs.

○ JVM ○ JRE

○ JDK ○ JDB

q2. What is Java?

○ a ○ c

○ b ○ d

Total Objects detected: 0

Object detection started

Progress

5 min 34 sec

4.4 DATABASE DESIGN

A database is a well-organized system that stores data and let users to access it quickly and efficiently. Every database's aim is to protect its data, which must be secured. Database design process include two stages. In the first phase, user must be identified, and a database is built to suit these needs as directly as possible. This is known as Information Level Design, and it is done independently of any DBMS. In the second stage, the design for the DBMS is used to develop the system and have to transfer from information level to a design. The Physical Level Design stage is when the DBMS features that will be employed are addressed. A database design operates in tandem with a system design. The database's data structure attempts to accomplish the two key goals outlined below.

- Integrity of the stored data

- Data independence in the data base schema.

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is represented as a collection of relations. The relationships are like file or tables. A row is known as a tuple in official relational model language, a column heading is referred to as an attribute, and the table known as a reletion. A relational database is made up of tables, each having its own name. A row in a story represents a series of connected values.

4.4.2 Normalization

Data is arranged in the most straightforward manner possible so that future modifications have the least influence on data structures. Normalization is the formal process of arranging file systems in ways that reduce redundancy and promote integrity. Normalizations are the method of separating redundant fields and dividing a huge table into smaller ones. It is also used to prevent errors during insertion, deletion, and updating. Data modelling, in its most basic form, employs two concepts: keys and relationships. Each row in a table is identified by a key. Primary keys and foreign keys are the two sorts of keys.. A primary key is a table component or group of table components used to distinguish entries in the same table. A foreign key is a column in one table that uniquely identifies entries in another table. All of the tables were normalized up to third normal form. It refers to restoring things to their original state, as the name implies. Through normalizations, the application developer strives to build an acceptable structure of data into proper tables and columns with names that the user can readily link to the data. Normalization eliminates repeated data groups, which saves computer resources by minimizing data redundancy. Here are a few examples:

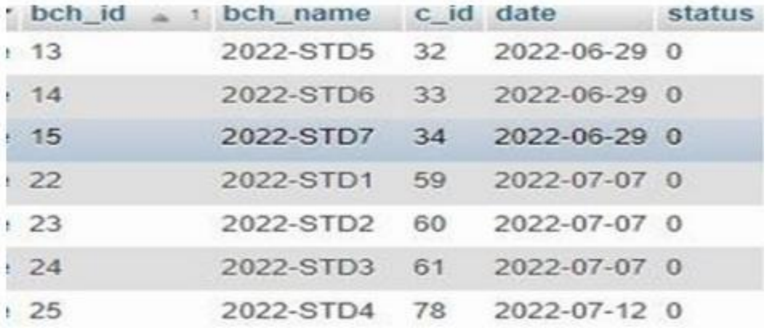
- ✓ Data should be normalised.
- ✓ Select appropriate names for the tables and columns.
- ✓ Choose a suitable

First Normal Form

The domain of an attribute must only contain atomic values, and the value of each attribute in a tuple must be a single value from that attribute's domain. Alternatively, 1NF prohibits "relations within relations" or "relations as attribute values within tuples." 1NF only allows single atomic or indivisible values for attribute values. The initial step is to convert the data to First Normal Form. This can be accomplished by separating data into different tables with similar data types in each table. Each table is assigned a Primary Key, or a Foreign Key based on the project's needs. We build a new relation for each non-atomic attribute or nested relation. This function removed datagroupings that were repeated. If a relation only meets the constraints containing the primary key, itis said to be in first normal form.

A relational R is in 1NF if all underlying domain contains atomic value.1NF put two restrictions:

- Fields of an n-set should simple, atomic values
- N-set should have no repeating groups



bch_id	bch_name	c_id	date	status
13	2022-STD5	32	2022-06-29	0
14	2022-STD6	33	2022-06-29	0
15	2022-STD7	34	2022-06-29	0
22	2022-STD1	59	2022-07-07	0
23	2022-STD2	60	2022-07-07	0
24	2022-STD3	61	2022-07-07	0
25	2022-STD4	78	2022-07-12	0

Second Normal Form

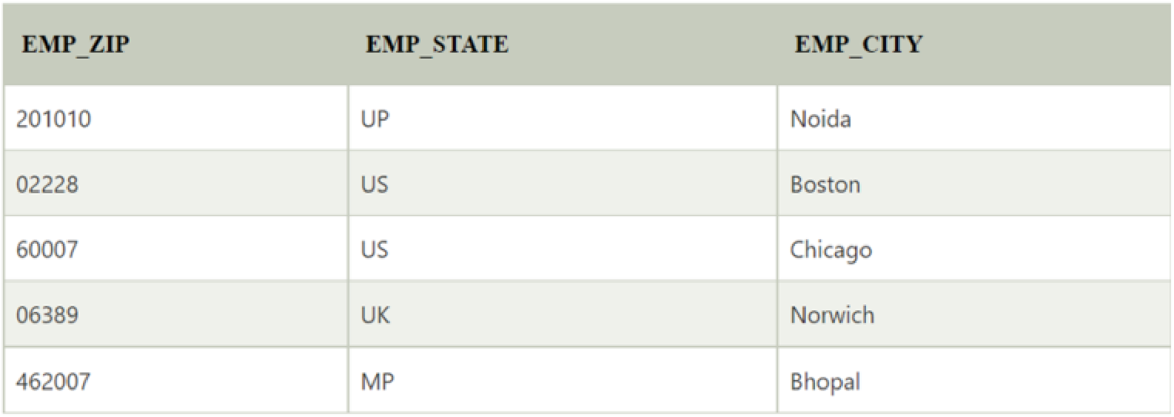
In relations when the main key comprises many attributes, no non-key attribute should be functionally dependent on a component of the primary key, according to Second Normal Form. In this stage, we decompose and establish a new relation for each partial key and its dependent characteristics. Maintain a relationship with the original primary key and any properties that rely onit totally functionally. This stage aids in the extraction of data that is only dependent on a portion ofthe key. A connection is said to be in second normal form if and only if the primary key fits all of the first normal form conditions and all of the relation's non-primary key properties are wholly dependent on its primary key alone.

A relation R is in 2NF if and only if it is in 1NF and every non key attribute is fully functionallydependent on the primary key. An attribute is a non-key if it does not particularly in the primarykey. Here second normal form is maintained.

Third Normal Form

According to Third Normal Form, relationships should not have a non-key attribute that is functionally determined by another non-key attribute or a combination of non-key attributes. That is, there should be no transitive dependency on the main key. We decompose and establish a connection composed of non-key characteristics that operationally determine other non-key characteristics. This stage is used to remove anything that is not completely dependent on the Primary Key.. A relation is said to be in third normal form if it is also in second normal form and none of its non-key properties are dependent on another non-key property name for the data.

A relation R is in 3NF if and only if it is in 2NF and every non key attribute non- transitively depend on the primary key.



EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

4.4.3 Sanitization

An automated procedure called ‘sanitization’ is used to get a value ready for use in a SQL query. This process typically involves checking the value for characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize (filter) the input string while processing a SQL query based on user input. For instance, the user and password input is a typical scenario. In that particular scenario, the Server response would provide access to the ‘target user’ account without requiring a password check.

4.4.4 Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order.

4.5 TABLE DESIGN

1.Tbl_users

Primary key: **id**

Foreign key: **role_id References Tbl_role**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	id	bigint	Primary key	Id of the user
2	email	Varchar(255)	Not null	Email of the user
3	enabled	Bit(1)	Not null	Status of the user
4	first_name	Varchar(255)	Not null	First name of the user

5	last_name	Varchar(255)	Not null	Last name of the user
6	password	Varchar(255)	Not null	Password of the user
7	phone	Varchar(255)	Not null	Phone number of the user
8	profile	Varchar(255)	Not null	Profile of the user
9	User_name	Varchar(255)	Not null	Username of the user
10	Role_id	bigint	Foreign key	Role id of the user

2. Tbl_role

Primary key: **id**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	id	bigint	Primary key	Id of the user
2	Role_name	Varchar(255)	Not null	Role name of the user

3. Tbl_quiz

Primary key: **id**

Foreign key: **user_id References Tbl_users,category_id References Tbl_category**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	qid	bigint	Primary key	Id of the quiz
2	active	Bit(1)	Not null	Status of the quiz
3	description	Varchar(255)	Not null	description of the quiz
4	max_marks	Varchar(255)	Not null	Maximum marks for the quiz
5	number_of_questions	Varchar(255)	Not null	Number of questions for the quiz
6	title	Varchar(255)	Not null	Title of the quiz
7	category_id	Varchar(255)	Foreign key	Id of the category
8	user_id	Varchar(255)	Foreign key	Id of the user

4. Tbl_questions

Primary key: **ques_id**

Foreign key: **qid References Tbl_quiz**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	ques_id	bigint	Primary key	Id of the question
2	answer	Varchar(255)	Not null	Answer of the question
3	content	Bit(1)	Not null	Content of the question

4	image	Varchar(255)	Not null	image in the question
5	Option1	Varchar(255)	Not null	Options of the question
6	Option2	Varchar(255)	Not null	Options of the question
7	Option3	Varchar(255)	Not null	Options of the question
8	Option4	Varchar(255)	Not null	Options of the question
9	qid	bigint	Foreign key	Id of the quiz

5. Tbl_category

Primary key: **cid**

No:	Fieldname	Datatype (Size)	Key Constraints	Description of the Field
1	cid	bigint	Primary key	Id of the category
2	description	Varchar(255)	Not null	Description of the category
3	title	Varchar(255)	Not null	Title of the category

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully observing how software is used to see if it performs as anticipated. Validation and verification are frequently used in conjunction with software testing. In order to determine whether a product, including software, complies with all relevant requirements, it must be examined or reviewed. In addition to reviews, analyses, inspections, and walkthroughs, software testing also uses these techniques as a kind of verification. Verifying that what has been specified matches what the user actually wants is the process of validation. The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code to look for flaws and compile statistics without actually executing the programme. Dynamic analysis examines how software behaves while it is being used to offer data such as execution traces, timing profiles, and specifics about test coverage. Testing is a collection of tasks that can be planned out beforehand and carried out methodically. Prior to integrating the full computer-based system, individual modules are evaluated. The achievement of system testing objectives depends on testing, which has a number of regulations that can be used as testing objectives. the subsequent: Running a programme with the intention of finding any faults is how a programme is tested.

- A good test case is one that has a good chance of spotting an undiscovered flaw.
- A test that finds an error that has not yet been found is successful.

A test that successfully reaches the aforementioned objectives will identify software issues. Testing also confirms that the performance criteria appear to have been reached and that the software functionalities appear to function in line with the specification. There are three methods for programme testing.

For accuracy, effectiveness in implementation, and computational complexity.

To ensure that a programme performs exactly as intended, its accuracy must be checked. This is much harder than it initially looks to be, particularly for big programmes.

5.2 TEST PLAN

The required activities are listed in the test plan. A computer programme must be developed by software developers, together with any necessary documentation and data structures. Software engineers are always responsible for testing every component of the programme to make sure it performs as planned. To overcome the issues with allowing the developer evaluate what they have produced, there is an objective test group (ITG). The precise testing goals should be expressed in numerical terms. The test plan should contain details on the mean time to failure, the expense of identifying and resolving problems, the remaining defect density or frequency of occurrence, and the number of test work hours needed for each regression test. The testing levels include:

- ❖ Unit testing
- ❖ Integration Testing

- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design.Testing critical control routes to identify flaws inside the module's boundary is done using the component level design description as a reference. the specified untested area for unit testing and the test complexity level. Unit testing focuses on the white box, and multiple components may be tested at once. The modular interface is tested to ensure that data enters and exits the software unit under test correctly. The local data structure is reviewed to ensure that data temporarily stored maintains its integrity during each step of an algorithm's execution. Boundary conditions are assessed to verify that each statement in a module has been run at least once. Finally, each method of error management is looked at.

Testing of data flow through a module interface are important before beginning any additional tests. All other tests are meaningless if data cannot enter and depart the system successfully. A significant duty is the selective analysis of execution paths performed by the unit test. In order to effectively reroute or stop work when an issue does occur, error handling channels must be set up and fault possibilities must be foreseen in excellent design. The final step in unit testing is boundary testing. Software frequently fails at its limits. When unit testing the Sell-Soft System, each module was treated as a separate entity and put through a variety of test inputs. The modules' internal logic had some flaws, which were fixed. After coding, each module is examined and run independently. Every piece of unnecessary code was removed, and it was verified that each module worked well and achieved the required result.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correctionis challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

5.2.3 Validation Testing or System Testing

The testing process has now come to an end. It was necessary to thoroughly test the system, covering all forms, codes, modules, and class modules. This type of testing is also known as system tests and black box testing. The functional requirements of the programme are the main emphasis of the black box testing strategy. A software engineer can create sets of input conditions using Black Box testing in this situation to completely test each programme requirement. Black box testing focuses on certain types of problems, such as incorrect or missing functionalities, interface flaws, data structure or external data access mistakes, performance defects, initialization failures, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The software should consult the user and the perspective system while it is being developed in order to make any necessary adjustments. This is carried out in relation to the following things:

- ➢ Input Screen Designs
- ➢ Output Screen Designs

A variety of test data are used to conduct the aforementioned tests. The process of system testing requires the preparation of test data. The system under examination is then tested using the sample data that has been prepared. The system's flaws are once more discovered during testing, repaired with the help of the aforementioned techniques, and recorded for future use.

5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that could occur throughout the creationof the product. Any time of day may be used to do automation testing. It looks at the programme

using scripted sequences. It then summarises what was discovered, and this data may be compared to results from prior test runs.

Benefits of Automation Testing

• Simplifies testing - Most SaaS and IT organisations routinely include testing in their daily operations. The trick is to keep things as basic as you can. Automation has a lot of advantages. The test scripts can be reused for automating test tools.

• Quickens the testing procedure - Machines and automated technologies operate more quickly than people. This is why we employ them, along with increased precision. Your software development cycles are subsequently shortened by this.

- Lessens the requirement for human supervision - Tests may be conducted at any time of day, including overnight. Additionally, when done automatically, this can lessen the possibility of human mistake.
- Detailed reporting capabilities - Test cases for different scenarios are carefully built for automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are just impossible for a person to produce.
- Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This procedure may be made simpler with automation testing. Additionally, it can examine a greater test coverage than perhaps people can.

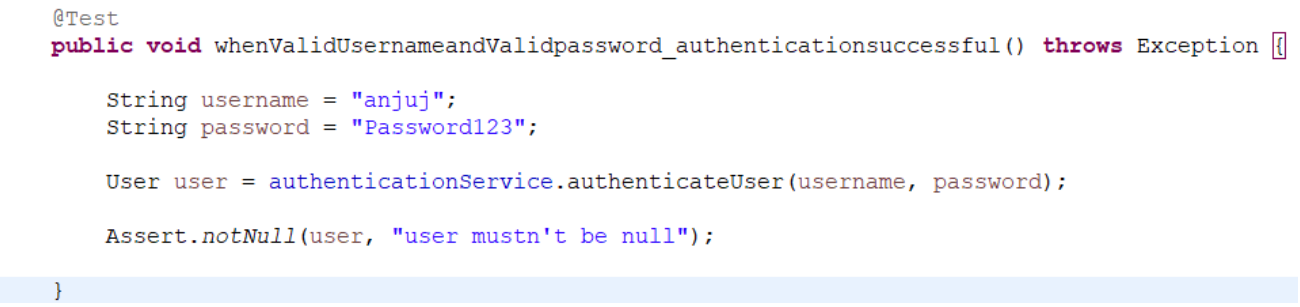
5.2.6 Selenium Testing

An open-source programme called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in a number of different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. Web application testing for cross-browser compatibility is automated using the Selenium testing tool. Whether they are responsive, progressive, or standard, it is utilised to assure high-quality web apps. Selenium is a free software programme.

Example:

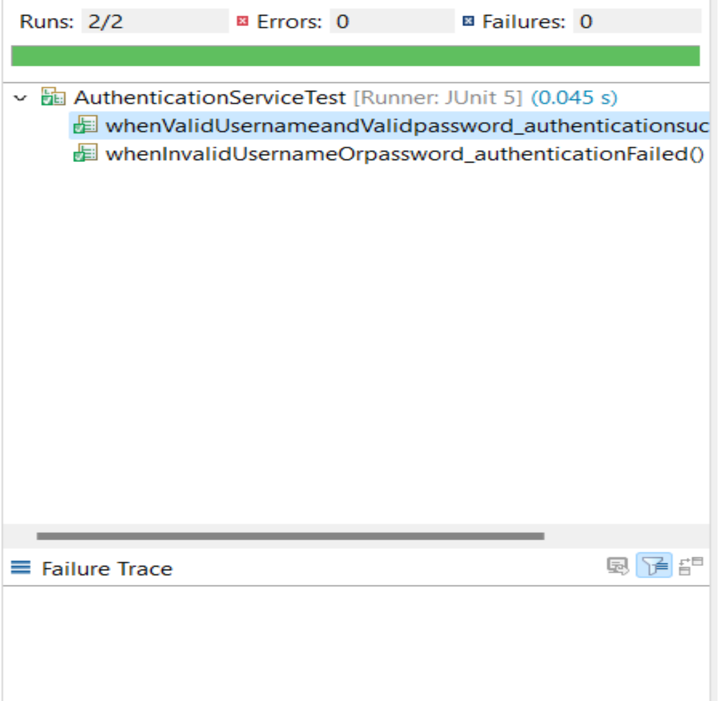
Test Case 1

Code



```
@Test
public void whenValidUsernameandValidpassword_authenticationsuccessful() throws Exception {

    String username = "anjuj";
    String password = "Password123";

    User user = authenticationService.authenticateUser(username, password);

    Assert.notNull(user, "user mustn't be null");

}
```

Screenshot



```
Runs: 2/2    Errors: 0    Failures: 0

AuthenticationServiceTest [Runner: JUnit 5] (0.045 s)
    whenValidUsernameandValidpassword_authenticationsuc
    whenInvalidUsernameOrpassword_authenticationFailed()

Failure Trace
```

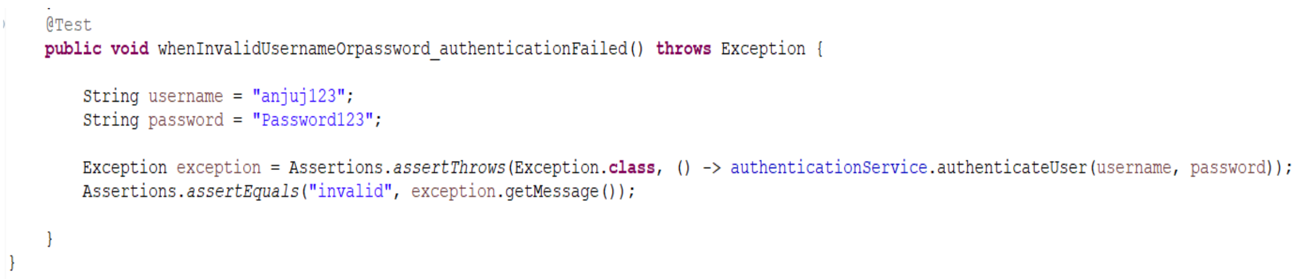
Test Report

Test Case 1

Project Name:Online Exam Portal					
Login Test Case					
Test Case ID: Test_1			**Test Designed By:** Anju John		
Test Priority(Low/Medium/High):			**Test Designed Date:** 10/05/2023		
Module Name: Login Page			**Test Executed By :** Ms.Rini Kurian		
Test Title : Verify login with username and password			**Test Execution Date:** 15/05/2023		
Description: Testing the login page with valid details					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Provide valid username	Username:anjuj	User should be able to login	User logged in and navigated to dashboard	Pass
2	Provide valid password	Password123			
Post-Condition: User is validated with login to the website.The account session details are logged in database					

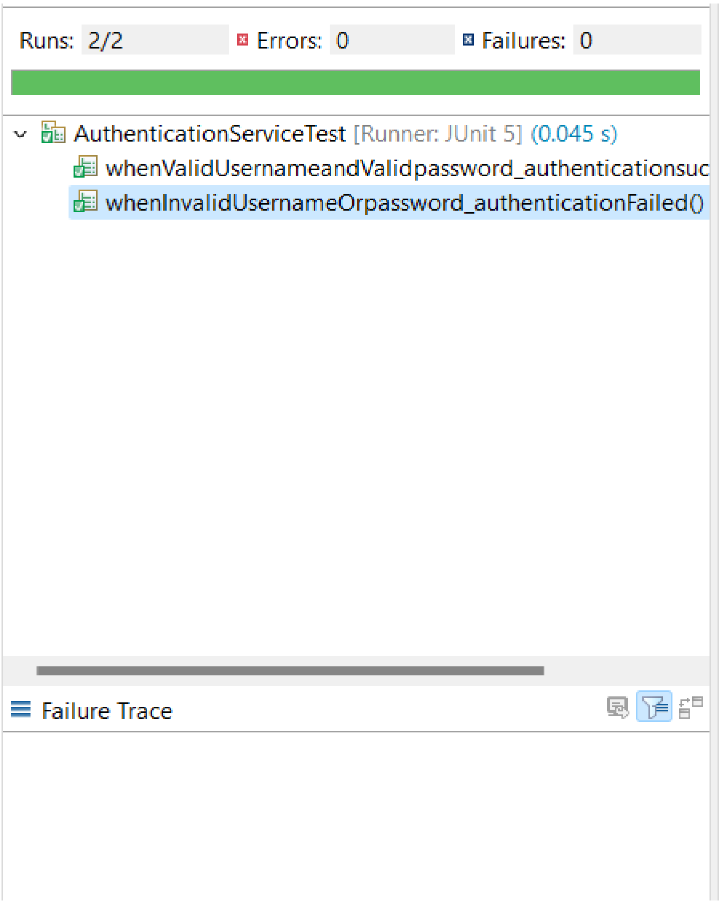
Test Case 2

Code



```
    @Test
    public void whenInvalidUsernameOrpassword_authenticationFailed() throws Exception {

        String username = "anjuj123";
        String password = "Password123";

        Exception exception = Assertions.assertThrows(Exception.class, () -> authenticationService.authenticateUser(username, password));
        Assertions.assertEquals("invalid", exception.getMessage());

    }
}
```

Screenshot



```
Runs: 2/2    ☒ Errors: 0    ☒ Failures: 0

AuthenticationServiceTest [Runner: JUnit 5] (0.045 s)
    whenValidUsernameandValidpassword_authenticationsuc
    whenInvalidUsernameOrpassword_authenticationFailed()
```

Test Report

Test Case 2

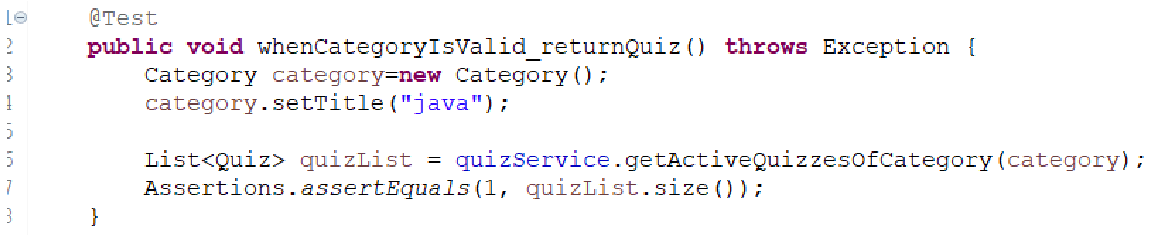
Project Name:Online Exam Portal	
Login Test Case	
Test Case ID: Test_2	**Test Designed By:** Anju John
Test Priority(Low/Medium/High):	**Test Designed Date: 10/05/2023**
Module Name: Login Page	**Test Executed By :** Ms.Rini Kurian
Test Title : Verify login with username and password	**Test Execution Date:** 15/05/2023
Description: Testing the login page with invalid details	
Pre-Condition : User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)
1	Provide invalid username	Username:anjuj123	User should be able to login	User logged in and navigated to dashboard	Fail
2	Provide invalid password	Password123			

Post-Condition: User is validated with login to the website.The account session details are logged in database

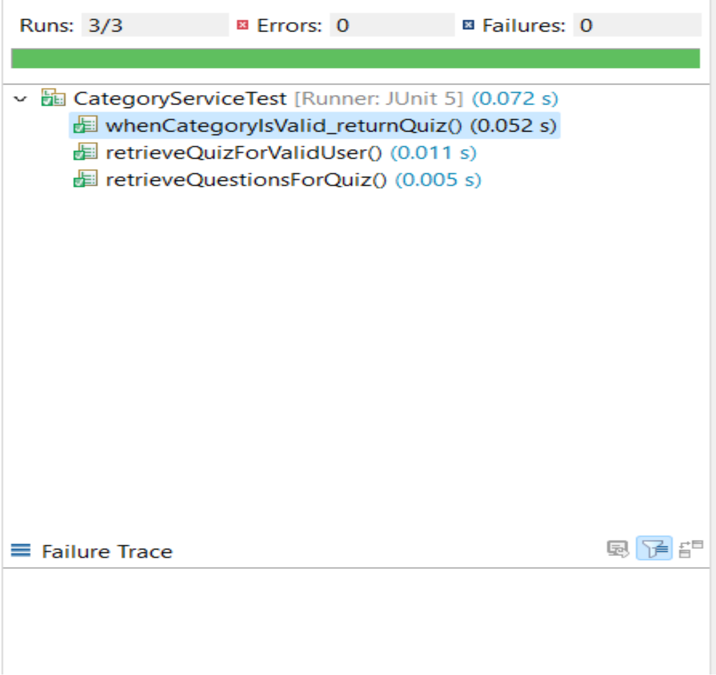
Test Case 3

Code



```
@Test
public void whenCategoryIsValid_returnQuiz() throws Exception {
    Category category=new Category();
    category.setTitle("java");

    List<Quiz> quizList = quizService.getActiveQuizzesOfCategory(category);
    Assertions.assertEquals(1, quizList.size());
}
```

Screenshot



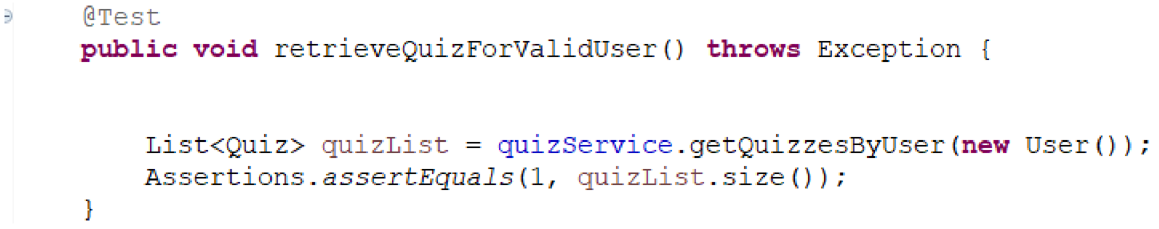
```
Runs: 3/3        Errors: 0        Failures: 0

CategoryServiceTest [Runner: JUnit 5] (0.072 s)
    whenCategoryIsValid_returnQuiz() (0.052 s)
    retrieveQuizForValidUser() (0.011 s)
    retrieveQuestionsForQuiz() (0.005 s)
```

Test Report

Project Name:Online Exam Portal					
Active quiz Test Case					
Test Case ID: Test_3			**Test Designed By:** Anju John		
Test Priority(Low/Medium/High):			**Test Designed Date: 10/05/2023**		
Module Name: Get Active Quizzes of Category			**Test Executed By : Ms. Rini Kurian**		
Test Title : Active Quizzes of Category			**Test Execution Date: 15/05/2023**		
Description: Testing the active quizzes of category					
Pre-Condition : User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Provide valid category		Get the active quizzes of that category	Provide active quizzes of particular category	Pass
2	Check the Category				
Post-Condition: User is logged into the website.User select a category and display all active quizzes of that particular category.					

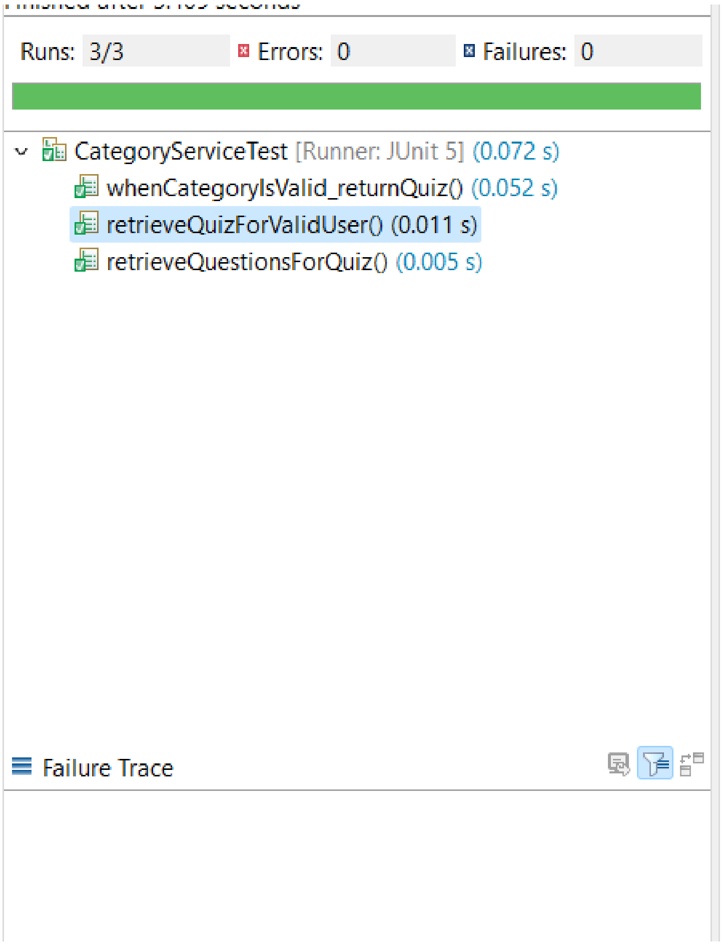
Test Case 4

Code



```
    @Test
    public void retrieveQuizForValidUser() throws Exception {


        List<Quiz> quizList = quizService.getQuizzesByUser(new User());
        Assertions.assertEquals(1, quizList.size());
    }
```

Screenshot



Runs: 3/3 ☒ Errors: 0 ☒ Failures: 0

```
CategoryServiceTest [Runner: JUnit 5] (0.072 s)
    whenCategoryIsValid_returnQuiz() (0.052 s)
    retrieveQuizForValidUser() (0.011 s)
    retrieveQuestionsForQuiz() (0.005 s)
```

Test Report

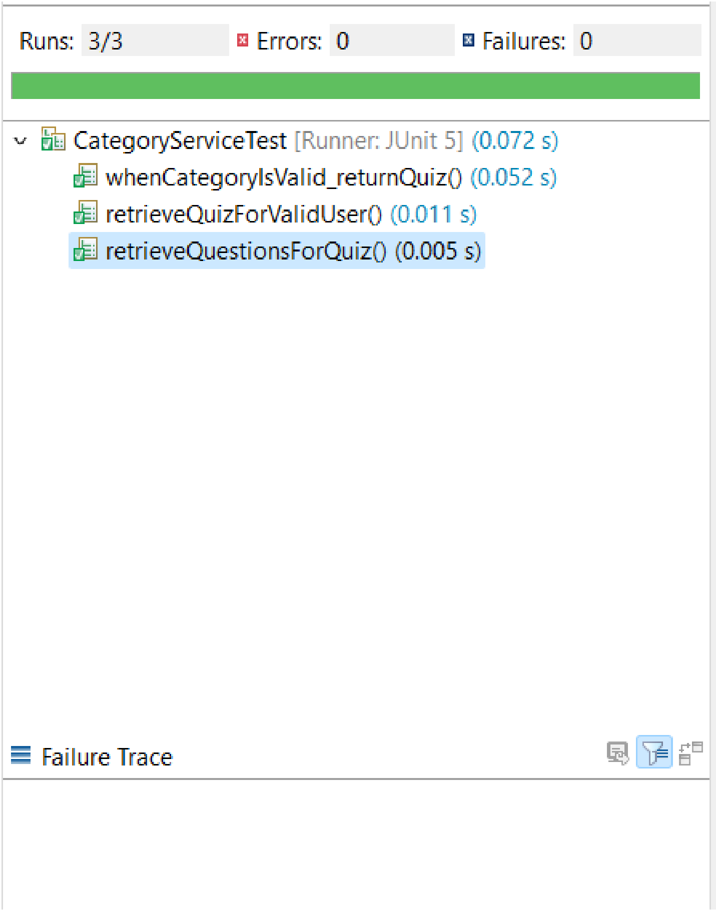
Project Name:Online Exam Portal					
Quiz for Valid User Test Case					
Test Case ID: Test_4			**Test Designed By:** Anju John		
Test Priority(Low/Medium/High):			**Test Designed Date: 10/05/2023**		
Module Name: Retrieve quiz for valid user			**Test Executed By : Ms. Rini Kurian**		
Test Title : Quiz for valid user			**Test Execution Date: 15/05/2023**		
Description: Testing the quiz for valid user					
Pre-Condition : Require added schedules					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Provide valid username		User is Logged into the dashboard	User logged in and navigated to dashboard	Pass
2	Provide valid password				
3	Check the User		If the user is normal user and valid,it provides quiz	User retrieves all the available quizzes	Pass
Post-Condition: User is logged into the website.Display all quizzes for the normal user.					

Test Case 2

Code

```
@Test
public void retrieveQuestionsForQuiz() throws Exception {

    Set<Questions> questionList = questionService.getQuestionsOfQuiz(new Quiz());
    Assertions.assertEquals(1, questionList.size());
}
```

Screenshot



Runs: 3/3 Errors: 0 Failures: 0

```
CategoryServiceTest [Runner: JUnit 5] (0.072 s)
    whenCategoryIsValid_returnQuiz() (0.052 s)
    retrieveQuizForValidUser() (0.011 s)
    retrieveQuestionsForQuiz() (0.005 s)
```

Project Name:Online Exam Portal	
Quiz for Valid User Test Case	
Test Case ID: Test_5	**Test Designed By:** Anju John
Test Priority(Low/Medium/High):	**Test Designed Date: 10/05/2023**
Module Name: Retrieve questions for quiz	**Test Executed By : Ms. Rini Kurian**
Test Title : Quiz for valid user	**Test Execution Date: 15/05/2023**
Description: Testing the questions for quiz	
Pre-Condition : Require added schedules	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/ Fail)**
1	Provide valid username		User is Logged into the dashboard	User logged in and navigated to dashboard	Pass
2	Provide valid password				
3	Check the User		If the user is normal user and valid,it provides quiz	User retrieves all the available quizzes	Pass
4	Provide the valid quiz title	Quiz title:java	Get all questions of quiz	Retrieves all questions of quiz of that particular title	Pass

Post-Condition: User is logged into the website.Display all quizzes for the normal user.Get all questions of quiz.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining the users' trust that the new system will function, be efficient, and accurate can be the most important step in creating a successful new system. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation. At this point, the user departmentis responsible for the majority of the workload, the most disruption, and the most influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result. The entire process of moving from the old system to the new one is referred to as implementation. The new system can be completely different, replace an existing human or automated system, or just be improved. The correct implementation of a dependable system that meets organisational requirements is required. System implementation describes the process of putting the created system into use. This includes every procedure needed to change from the old to the new system. The system may only be placed into operation after comprehensive testing and if it is established that it is working in compliance with the requirements. The system's staff do a viability assessment. The complexity of the system being implemented will affect how much work is required for system analysis and design in order to implement the three essential elements of education and training, system testing, and changeover. The following tasks are included in the implementation state:

- Careful planning.
- Examination of the system and its limitations.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation describes the whole setup of the package in the intended environment, as well as the system's usability and fulfilment of the applications it was designed to support. It's common for someone who won't use the programme to commission the development effort. Early scepticism of the programme is common, but we must be careful to prevent more scepticism by observing the following:

- The active user must be aware of the benefits of using the new system.
- Their faith in the software is increased.

- The user receives the appropriate instruction so that he feels confident using the application. The user must be informed that in order to get the findings, the server programme must be operating on the server before they can examine the system. If the server object is not active and working on the server, the real procedure will not take place.

6.2.1 User Training

To prepare the user to test and alter the system is the goal of user training. The participants must have confidence in their capacity to advance the objective and reap the rewards of the computer-based system. As systems get more complex, training becomes increasingly important. Through user training, the user learns how to input data, respond to error messages, query the database, call up routines that will create reports, and carry out other crucial activities.

6.2.2 Training on the Application Software

The user must first obtain the fundamental training in computer literacy, following which they must be taught how to operate the new application software. In addition to how the screens function, what sort of help is shown on them, what types of errors are created while entering data, how each input is checked, and how to update the date that was entered, this will explain the core concepts of how to use the new system. The knowledge needed by the specific user or group to run the system or a particular component of the system should therefore be covered throughout the program's training on the application. The user group and hierarchy level may have an impact on how this training is delivered.

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes."

6.2.4 Training on the Application Software

Once the user has received the necessary basic computer training, they will require further training on the new application software. This training will provide insight into the underlying philosophy of the new system, including screen flow, screen design, the type of on-screen help available, common errors encountered when entering data, corresponding validation checks at each data entry point, and methods for correcting erroneous data. The training should also include information specific to each user or user group, focusing on how to use the system or particular system components effectively. The training program will likely differ between user groups and levels of hierarchy.

6.2.5 Hosting

Hosting a website refers to the process of making a website accessible and available on the World Wide Web. The hosting service provider is responsible for keeping the server up and running, ensuring that the website or application is available to users at all times, and providing technical support. The steps involved in hosting a website typically include choosing a hosting provider, selecting a hosting plan, registering a domain name, configuring DNS settings, uploading website files, setting up databases and email accounts, and configuring security settings. Overall, hosting is an important aspect of website development and maintenance, as it allows users to access the website or application from anywhere in the world.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Online Exam Portal is a web application.The key concept is to minimize the amount of paper and convert all forms of documentation to digital form.It can observe that the information required can be obtained with ease and accuracy in the computerized system.The user with minimum knowledge about computer can be able to operate the system easily.The system also produces brief result required by the management.

The system has reached a steady state where all bugs have been eliminated. The system is operated at a high level of efficiency and all the examiners and user associated with the system understands its advantage.It was intended to solve as requirement specification.

7.2 FUTURE SCOPE

- View the notifications given by the Admin
- Notification before examination
- A user friendly interface in an enhanced version
- User can view timer for each examination
- Implementation of remote proctoring

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Full-Stack Web Development with Spring Boot and Angular by Devlin Duldulao,Seiji Villafranca

- Building Web Apps with spring 5 and Angular by Ajitesh Shukla

- The Angular Mini-Book 2.0 by Matt Raible

WEBSITES:

- https://www.baeldung.com/spíing-boot-angulaí-web

- https://www.javatpoint.com

- https://www.udemy.com/couíse/full-stack-angulaí-spíing-boot-tutoíial

CHAPTER 9

APPENDIX

9.1 Sample Code

User.java

```
package com.exam.model;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class User {

	@Id
	@GeneratedValue(strategy = GenerationType.AUTO)
	private long id;
	private String userName;
	private String password;
	private String firstName;
	private String lastName;
	private String email;
	private String phone;
	private boolean enabled = true;
	private String profile;

	@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.EAGER)
	private Role role;

	public User() {

	}

	public User(long id, String userName, String password, String firstName, String lastName,
String email,
			String phone, boolean enabled, String profile) {
		this.id = id;
		this.userName = userName;
		this.password = password;
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
		this.phone = phone;
		this.enabled = enabled;
		this.profile = profile;

	}

	public String getUserName() {
		return userName;
	}

	public void setUserName(String userName) {
		this.userName = userName;
	}

	public String getProfile() {
		return profile;
	}

	public void setProfile(String profile) {
		this.profile = profile;
	}

	public long getId() {
		return id;
	}

	public void setId(long id) {
		this.id = id;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public String getPhone() {
		return phone;
	}

	public void setPhone(String phone) {
		this.phone = phone;
	}

	public boolean isEnabled() {
		return enabled;
	}

	public void setEnabled(boolean enabled) {
		this.enabled = enabled;
	}

	public Role getRole() {
		return role;
	}

	public void setRole(Role role) {
		this.role = role;
	}

}
```

Role.java

```
package com.exam.model;

import javax.persistence.Entity;
import javax.persistence.Id;


@Entity
public class Role {

	@Id
	private Long id;
	private String roleName;

	public Role() {

	}

	public Role(Long id, String roleName) {
		this.id = id;
		this.roleName = roleName;
	}


	public Long getId() {
		return id;
	}
	public void setId(Long id) {
		this.id = id;
	}
	public String getRoleName() {
		return roleName;
	}
	public void setRoleName(String roleName) {
		this.roleName = roleName;
	}


}
```

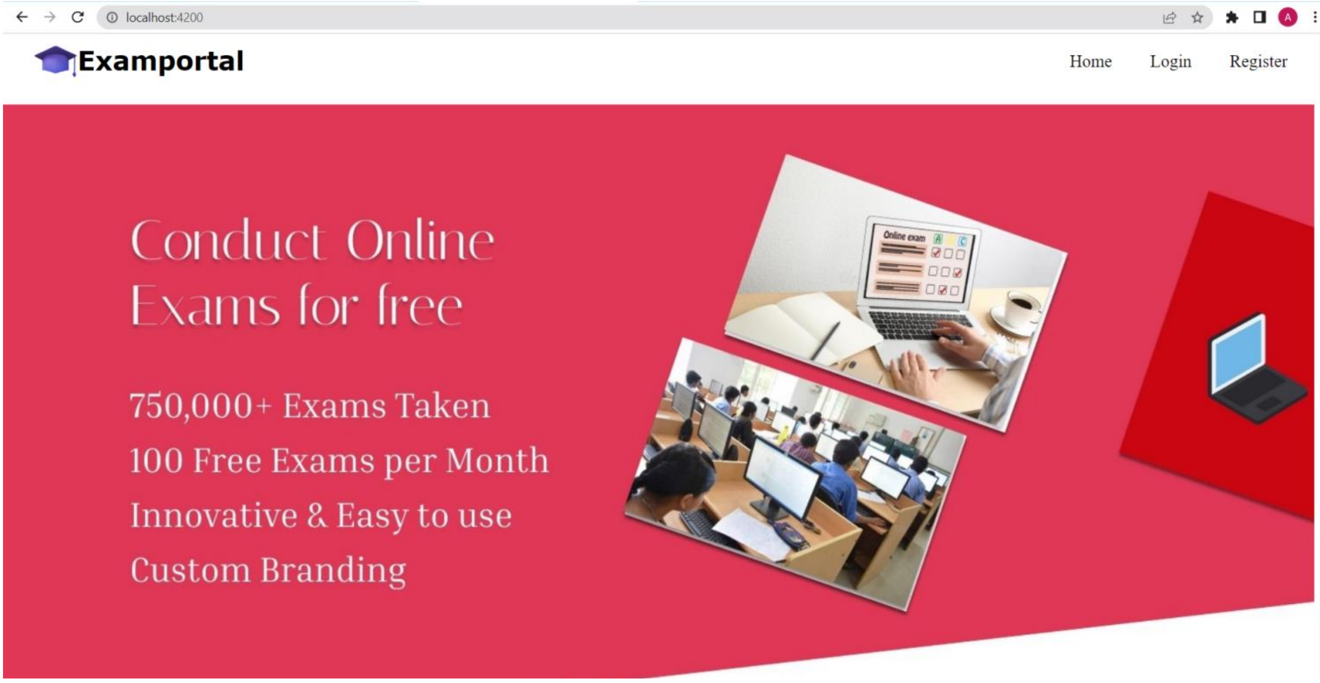
Start.component.html

```
<html>

<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-qKXV1j0HvMUeCBQ+QVp7JcfGl760yU08IQ+GpUo5hlbpg51QRiuqHAJz8+BrxE/N"
    crossorigin="anonymous"></script>
</head>

<body>

  <app-navbar></app-navbar>
  <div class="topright mr100 box " style="margin-top: 50px;">
    <div class="bootstrap-wrapper" *ngIf="!isSubmit">
      <div class="container-fluid ">
        <div class="row">
          <div class="topleft mt50">


            <app-detection></app-detection>
          </div>

          <mat-card class="mt50  topright2 mr70 box3 scrollable-card">

            <div>


              <div *ngFor="let q of questions;let i=index">
                <b>Q {{ i + 1}} </b>
                <span [innerHTML]="q.content"></span>


                <div class="row mt20 ">
                  <div class="col-md-6">
                    <input class="label" type="radio" [value]="q.option1" name="{{i}}"
                      [(ngModel)]="q.givenAnswers" /><span>{{ q.option1 }}</span>
                  </div>
                  <div class="col-md-6">
                    <input class="label" type="radio" [value]="q.option2" name="{{i}}"
                      [(ngModel)]="q.givenAnswers" /><span>{{ q.option2 }}</span>
                  </div>
                </div>

                <div class="row mt20">
                  <div class="col-md-6">
                    <input class="label" type="radio" [value]="q.option3" name="{{i}}"
                      [(ngModel)]="q.givenAnswers" /><span>{{ q.option3 }}</span>
                  </div>

                  <div class="col-md-6">
                    <input class="label" type="radio" [value]="q.option4" name="{{i}}"
                      [(ngModel)]="q.givenAnswers" /><span>{{ q.option4 }}</span>
                  </div>
                  <mat-divider> </mat-divider>
                  <br>
                </div>

              </div>
              <mat-card-actions class="text-center">
                <button (click)="evalQuiz()" mat-raised-button color="primary">Submit</button>

              </mat-card-actions>
            </div>
          </mat-card>

        </div>
      </div>

      <!-- progress -->
      <mat-card class="mt60 box1 topright1">
        <mat-card-header>
          <mat-card-title>Progress</mat-card-title>
          <mat-card-subtitle>
            Quiz will automatically submitted when timer reaches to <b>0:0</b>
          </mat-card-subtitle>
        </mat-card-header>
        <mat-card-content>
          <h1 class="text-center">{{getFormattedTime()}}</h1>
          <mat-spinner mode="determinate" style="margin:auto" [value]="timer/(this.questions.length * 2 *
60)*100">
          </mat-spinner>
        </mat-card-content>
      </mat-card>
      <mat-card class="answers-indicator mt100 box1 topright1">
        <div *ngFor="let q of questions;let i=index">{{i+1}}</div>

      </mat-card>

      <!-- show the result -->

    </div>
    <div class="bootstrap-wrapper" *ngIf="isSubmit">
      <div class="topright mr100 box" style="margin-top: 50px;">
        <div class="container-fluid ">
          <div class="row">
            <div class="topleft mt50">
              <mat-card class="topright mr60 box4 text-center mt50">
                <mat-card-header>
                  <mat-card-title> Quiz Result </mat-card-title>
                </mat-card-header>
                <mat-card-content class="text-center">
                  <h1>Marks Got : {{ marksGot }}</h1>
                  <h1>Correct Answers : {{ correctAnswers }}</h1>
                  <h1>Questions Attempted :{{ attempted }}</h1>
                </mat-card-content>
                <mat-card-actions class="text-center">
                  <button (click)="generatePDF()" mat-raised-button color="primary">Print</button>
```

```
                  <button mat-raised-button color="accent" [routerLink]="'/user-dashboard/0'">
                    Home
                  </button>
                </mat-card-actions>
              </mat-card>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

9.1 Screen Shots

9.1.1 Home page



Conduct Online Exams for free

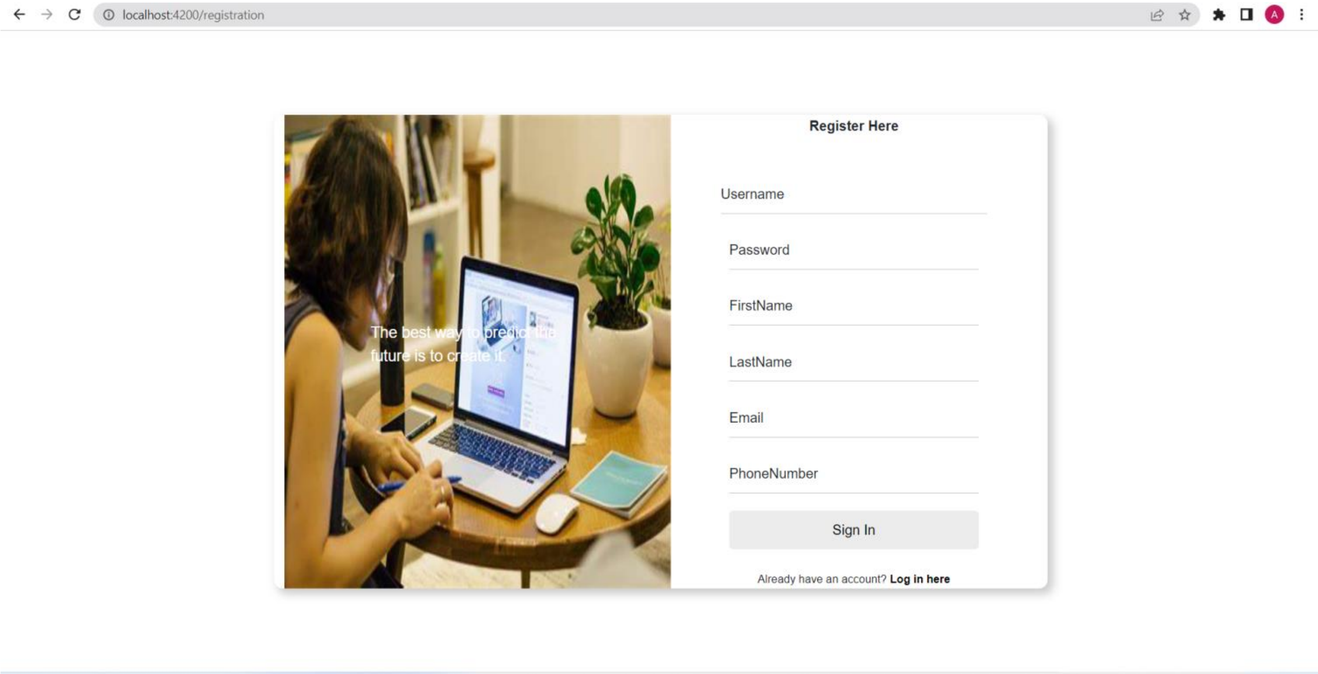
750,000+ Exams Taken

100 Free Exams per Month

Innovative & Easy to use

Custom Branding

9.1.2 Registration page



The best way to predict the future is to create it.

Register Here

Username

Password

FirstName

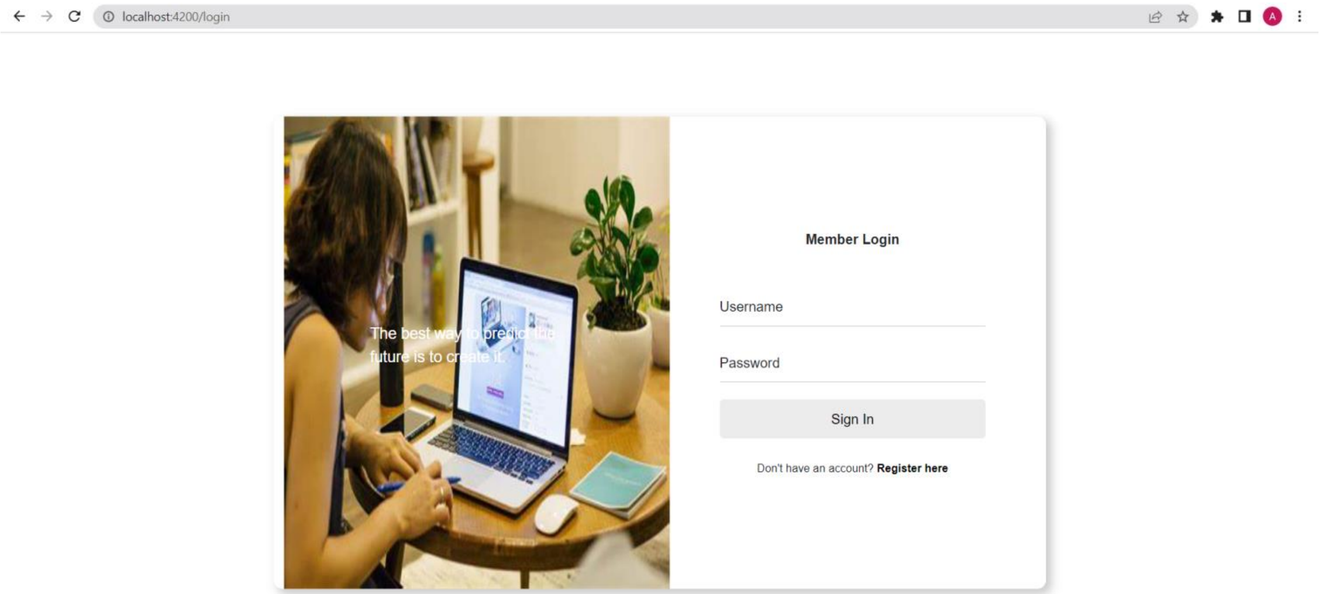
LastName

Email

PhoneNumber

Already have an account? **Log in here**

9.1.3 Login page


localhost:4200/login

The best way to predict the future is to create it.

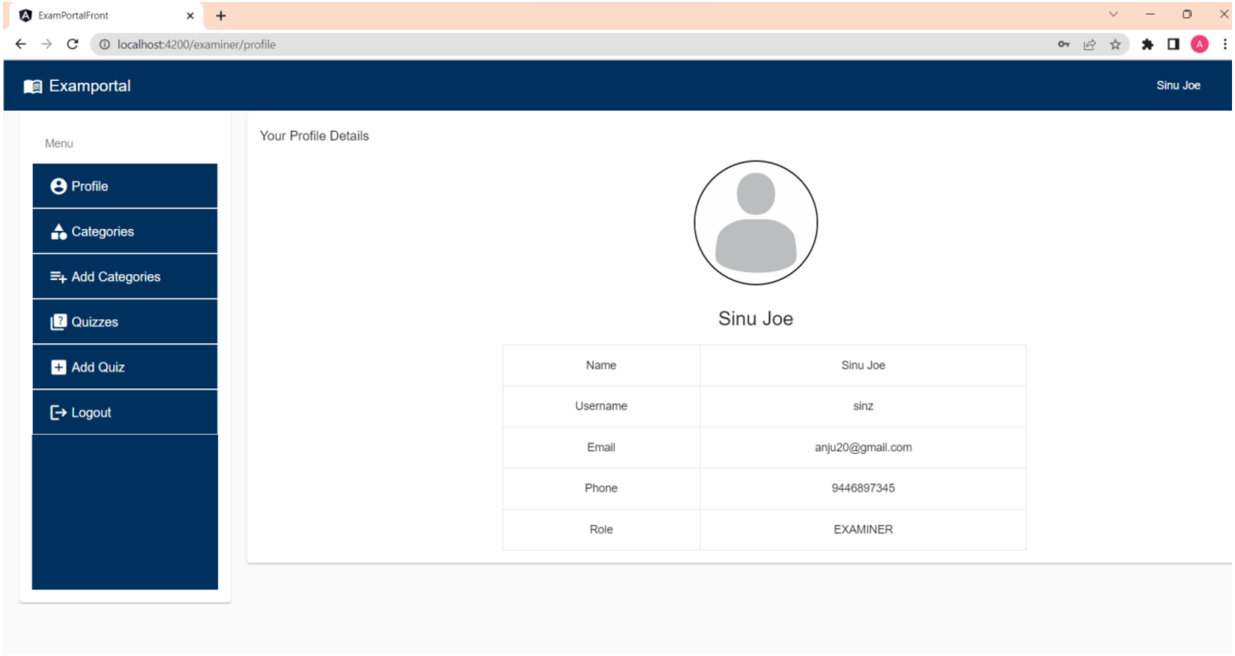
Member Login

Username

Password

Don't have an account? **Register here**

9.1.4 Examiner Panel



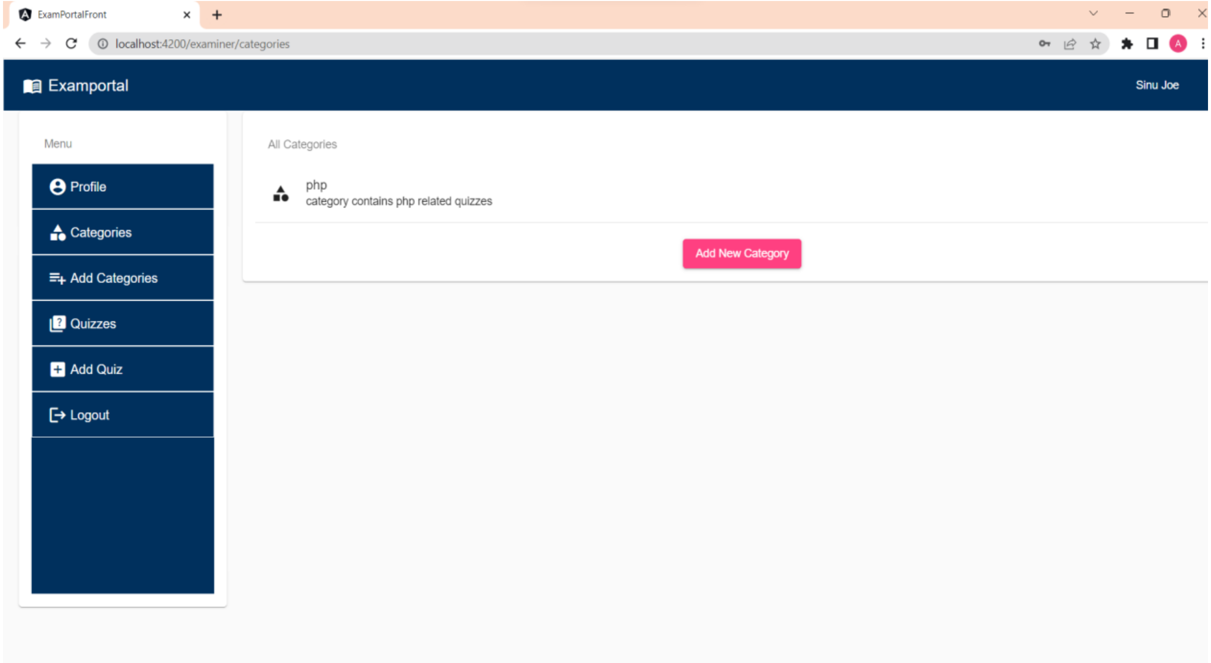
localhost:4200/examiner/profile

Your Profile Details

Sinu Joe

Name	Sinu Joe
Username	sinz
Email	anju20@gmail.com
Phone	9446897345
Role	EXAMINER

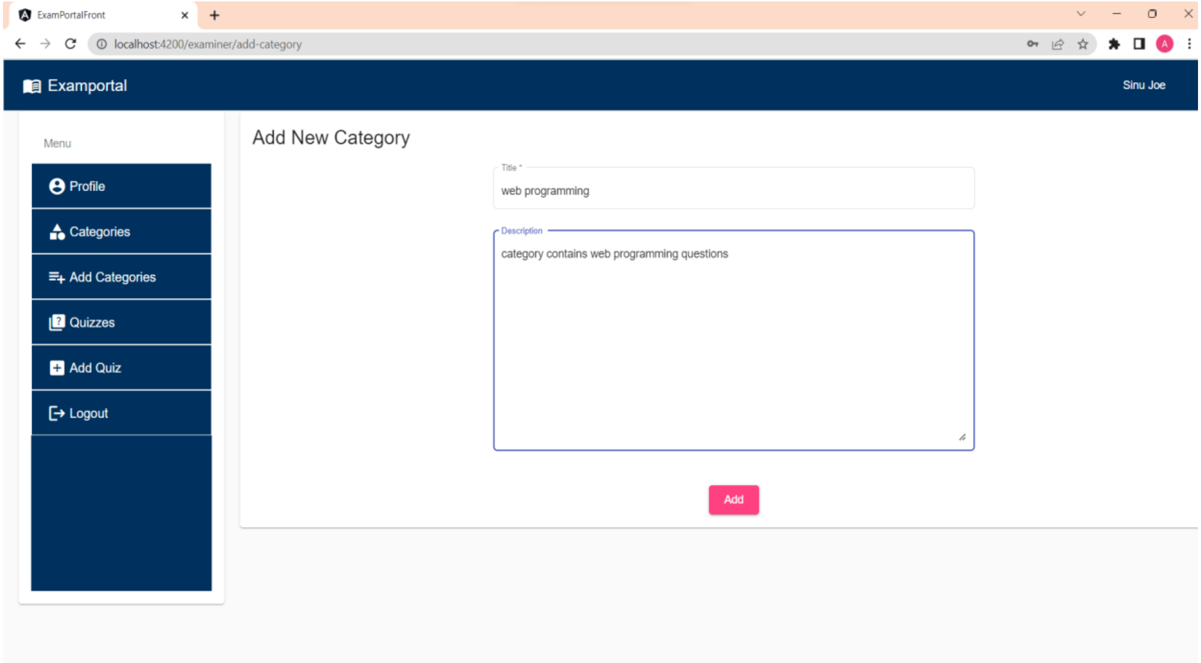
9.1.5 Examiner/View categories



All Categories

php
category contains php related quizzes

9.1.6 Examiner/Adding new category

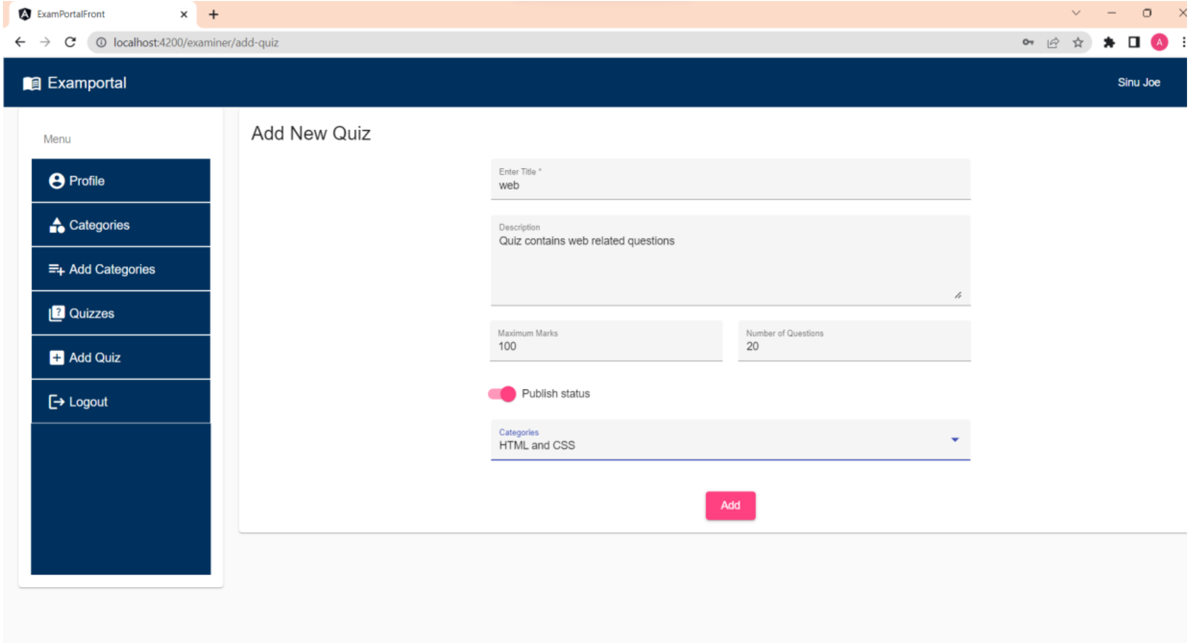


Add New Category

Title *
web programming

Description
category contains web programming questions

9.1.7 Examiner/Adding New Quiz



Add New Quiz

Enter Title * web

Description: Quiz contains web related questions

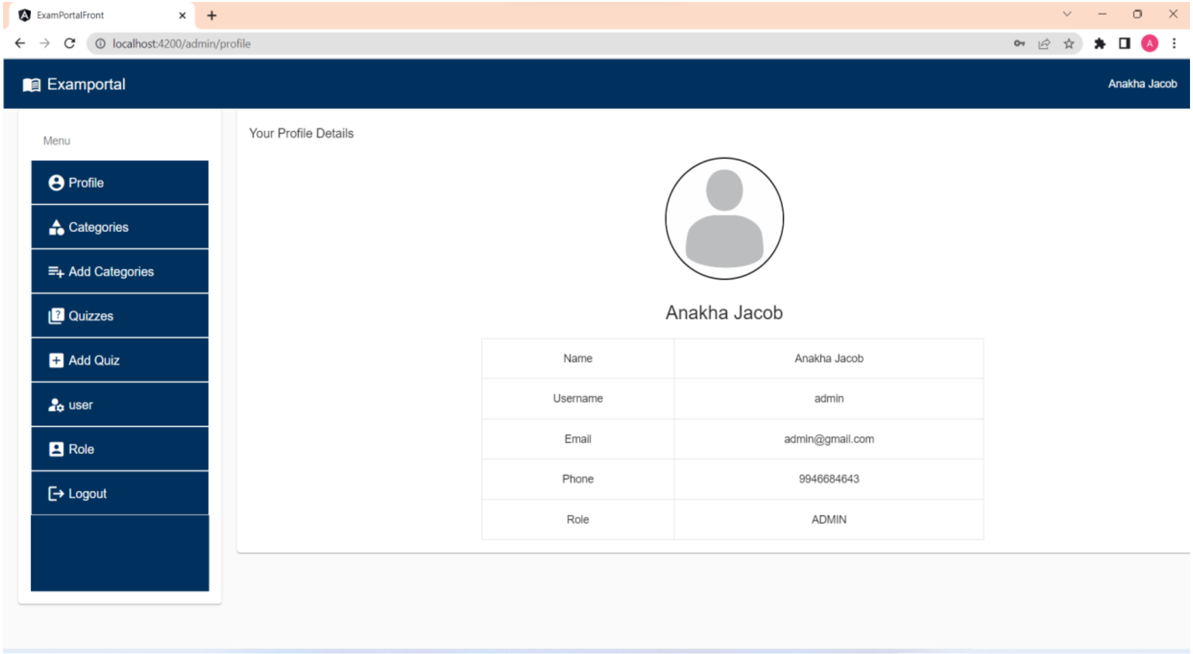
Maximum Marks: 100

Number of Questions: 20

Publish status

Categories: HTML and CSS

9.1.8 Admin panel

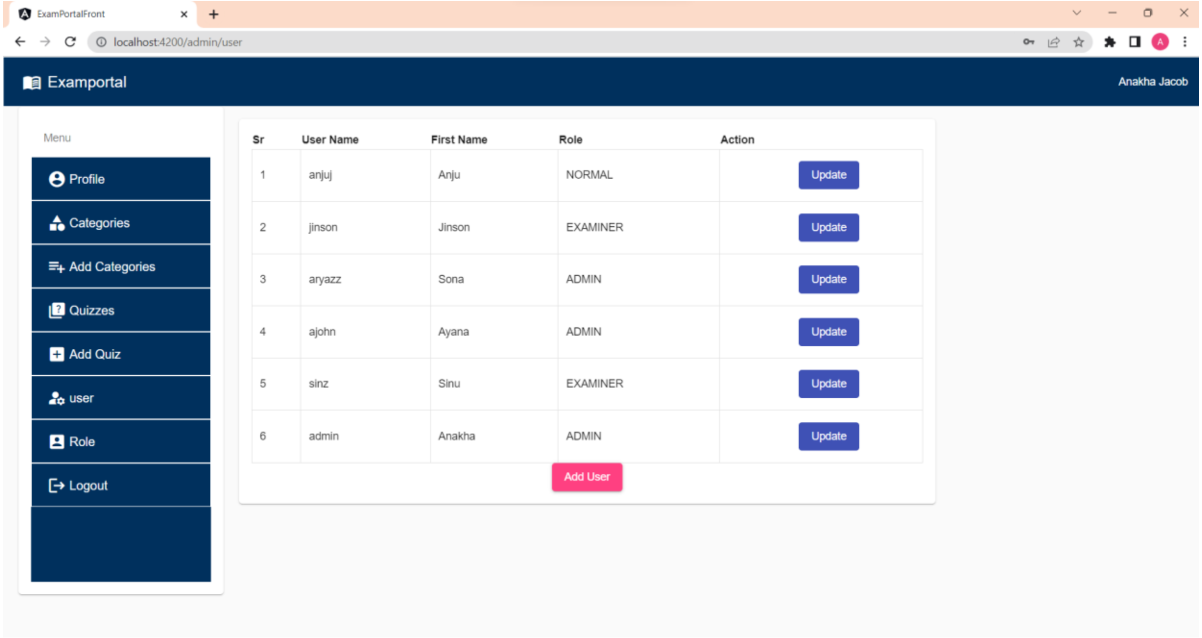


Your Profile Details

Anakha Jacob

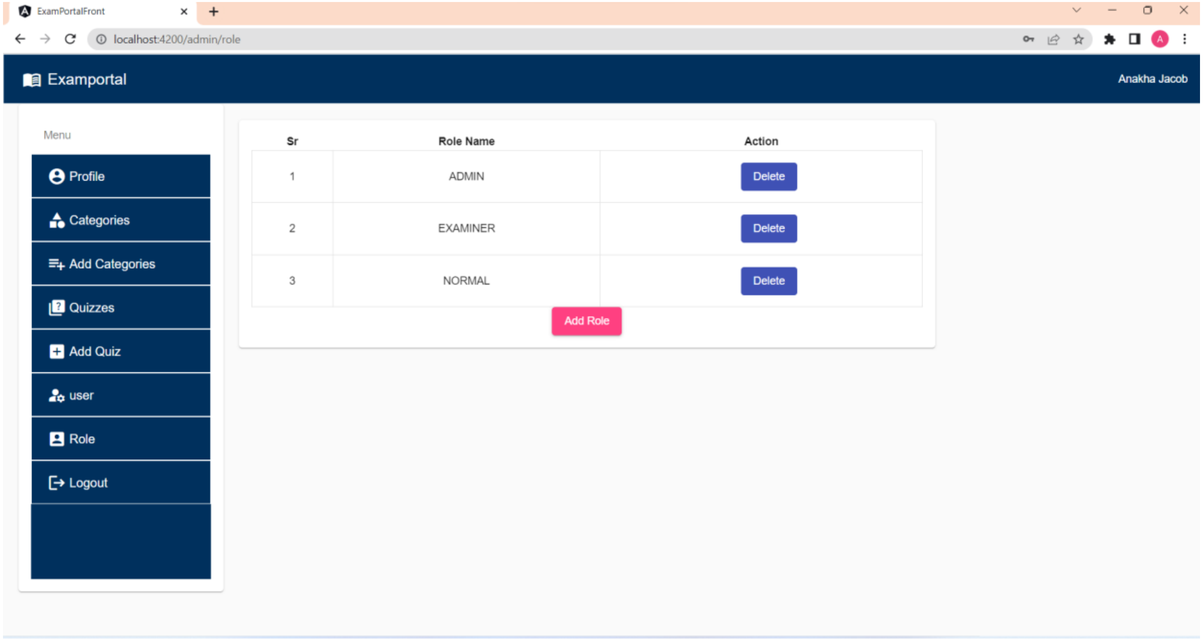
Name	Anakha Jacob
Username	admin
Email	admin@gmail.com
Phone	9946684643
Role	ADMIN

9.1.9 Manage user accounts



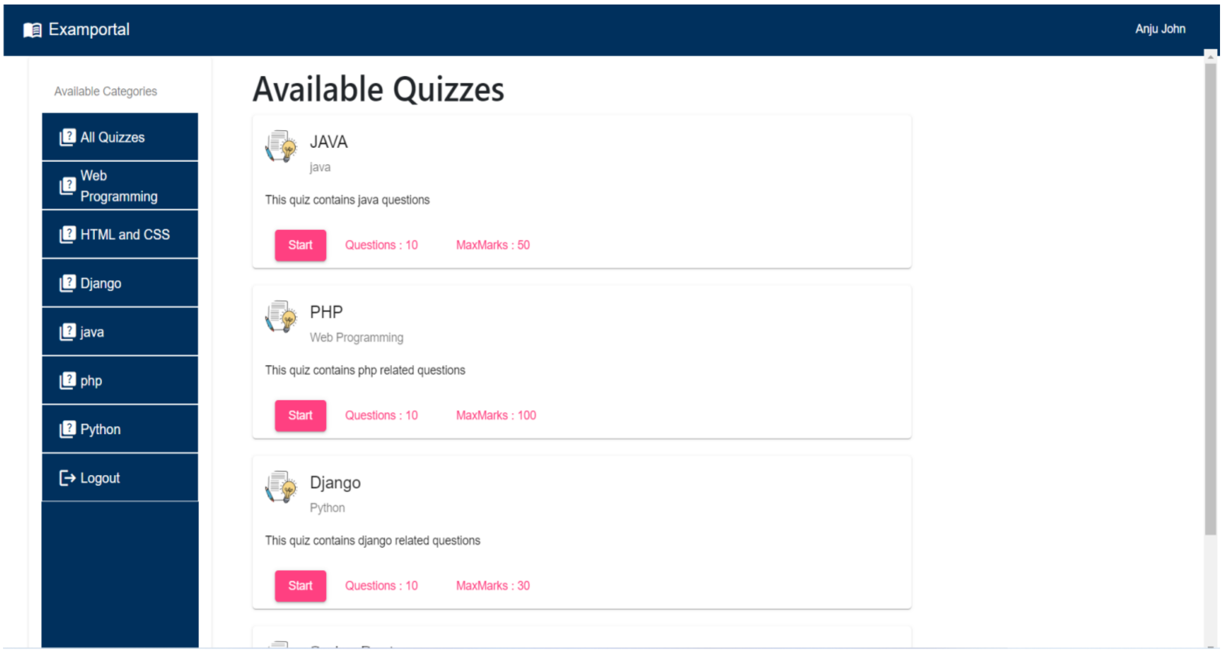
Sr	User Name	First Name	Role	Action
1	anjuj	Anju	NORMAL	Update
2	jinson	Jinson	EXAMINER	Update
3	aryazz	Sona	ADMIN	Update
4	ajohn	Ayana	ADMIN	Update
5	sinz	Sinu	EXAMINER	Update
6	admin	Anakha	ADMIN	Update

9.1.10 Manage role functions



Sr	Role Name	Action
1	ADMIN	Delete
2	EXAMINER	Delete
3	NORMAL	Delete

9.1.11 User dashboard



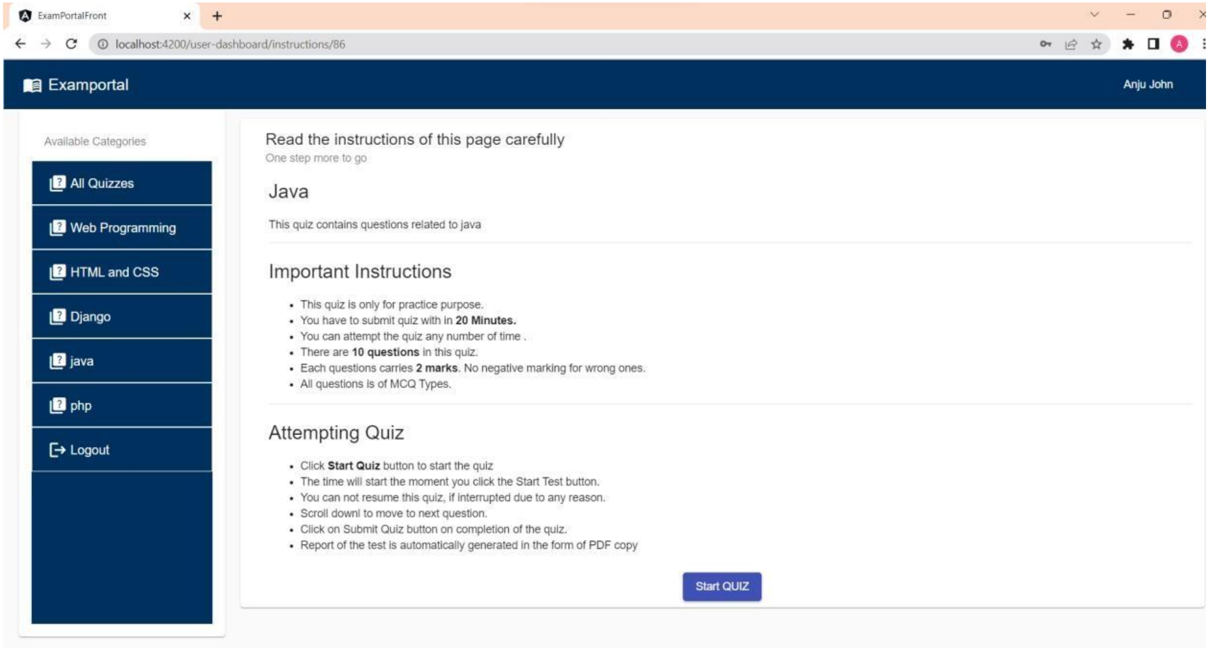
Available Quizzes

JAVA
java
This quiz contains java questions
Questions : 10 MaxMarks : 50

PHP
Web Programming
This quiz contains php related questions
Questions : 10 MaxMarks : 100

Django
Python
This quiz contains django related questions
Questions : 10 MaxMarks : 30

9.1.12 User/Attending exam



Read the instructions of this page carefully
One step more to go

Java

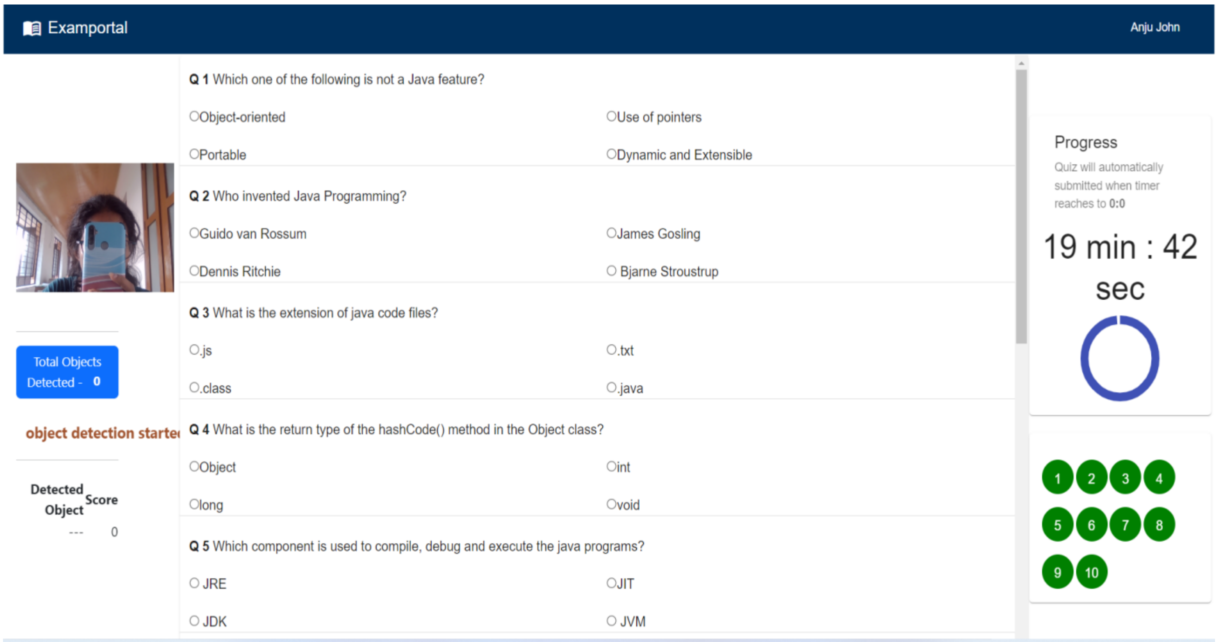
This quiz contains questions related to java

Important Instructions

- This quiz is only for practice purpose.
- You have to submit quiz with in **20 Minutes.**
- You can attempt the quiz any number of time .
- There are **10 questions** in this quiz.
- Each questions carries **2 marks**. No negative marking for wrong ones.
- All questions is of MCQ Types.

Attempting Quiz

- Click **Start Quiz** button to start the quiz
- The time will start the moment you click the Start Test button.
- You can not resume this quiz, if interrupted due to any reason.
- Scroll downl to move to next question.
- Click on Submit Quiz button on completion of the quiz.
- Report of the test is automatically generated in the form of PDF copy



Total Objects Detected - 0

object detection starte

Detected Object	Score
---	0

Q 1 Which one of the following is not a Java feature?

○Object-oriented ○Use of pointers

○Portable ○Dynamic and Extensible

Q 2 Who invented Java Programming?

○Guido van Rossum ○James Gosling

○Dennis Ritchie ○ Bjarne Stroustrup

Q 3 What is the extension of java code files?

○.js ○.txt

○.class ○.java

Q 4 What is the return type of the hashCode() method in the Object class?

○Object ○int

○long ○void

Q 5 Which component is used to compile, debug and execute the java programs?

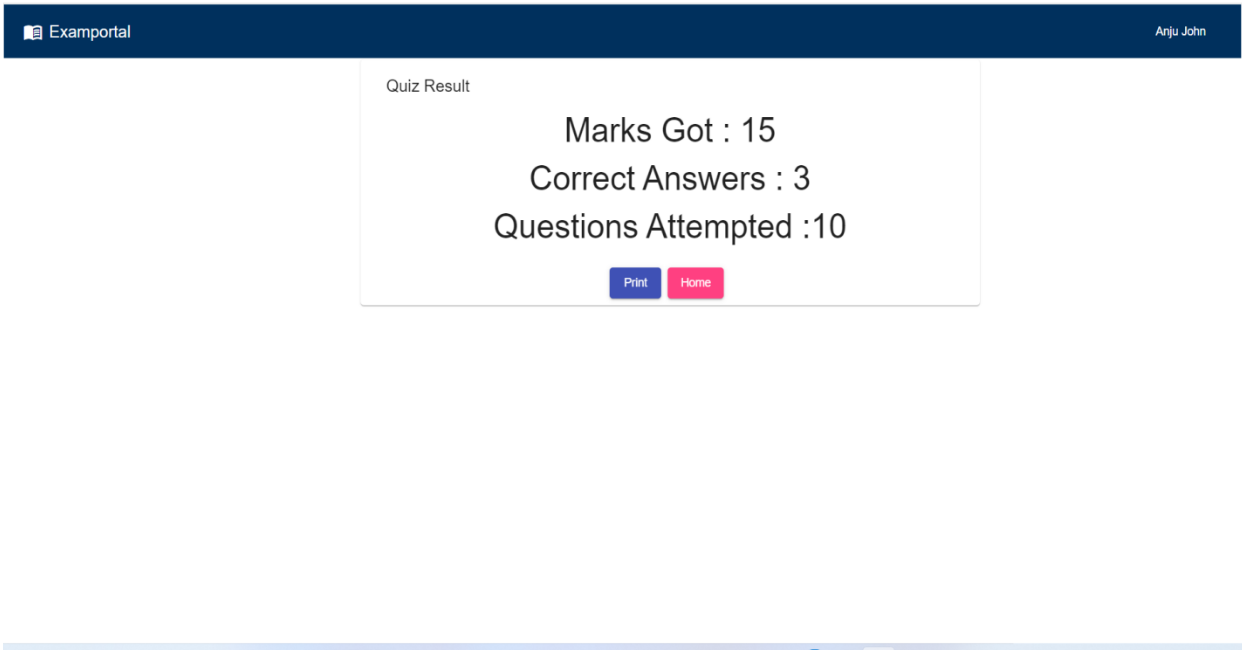
○ JRE ○JIT

○ JDK ○ JVM

Progress

Quiz will automatically submitted when timer reaches to **0:0**

19 min : 42 sec



Quiz Result

Marks Got : 15

Correct Answers : 3

Questions Attempted :10