# 



for Degree in

B.Sc.in Software Development

for the course in

**Submitted by :**

**Student Name : Adil**

**GMIT ID: g00376679**

**Under the guidance of**

**Professor Name: Joe Corr**

Professional Practice in IT (COMP07030)

**Exam Portal Application**

**A**

**Project Report**

**on**

# Acknowledgment

I like to share our sincere gratitude to all those who help us in completion of this project. During the work i faced many challenges due to our lack of knowledge and experience but these people help us to get over from all the difficulties and in final compilation of our idea to a shaped sculpture.

I would like to thank Joe Corr sir for his governance and guidance,

because of which i was able to learn the minute aspects of a project work.

I would also like to show our gratitude to our supervisors for this module :

Dominic Carr

Joe Corr

Damien Costello

Daniel Cregg

John French

Gerard Harrison

Martin Hynes

Martin Kenirons

I are also thankful to our whole class and most of all to our parents who have inspired us to face all the challenges and win all the hurdles in life.

Thank you All.

# Abstract

The aim of the project work was to build a minimum working model for an Online Exam Application, where teacher/supervisor can create quiz with number of multiple choice questions and students can give the those quiz and can view the score after submitting the quiz.

# Table of Contents

[Acknowledgment 1](#_Toc2406)

[Abstract 2](#_Toc5473)

[Table of Contents 3](#_Toc22648)

[Introduction 1](#_Toc19212)

[Application Development Life Cycle 2](#_Toc11508)

[SDLC Phases 2](#_Toc21187)

[Requirements Gathering 3](#_Toc21875)

[Requirements Analysis 4](#_Toc11514)

[Application Design 5](#_Toc5077)

[REST Architecture 6](#_Toc12899)

[What is a use case diagram? 8](#_Toc19209)

[What is JWT (JSON Web Token)? 9](#_Toc6934)

[Structure of JWT 9](#_Toc14376)

[What is hashing ? 10](#_Toc26551)

[What is Bcrypt password hashing algortihm ? 10](#_Toc31082)

[What is ER Diagram? 11](#_Toc1312)

[Why use ER Diagrams? 11](#_Toc31163)

[Project implementation 13](#_Toc1570)

[What is OOPS? 14](#_Toc19961)

[What is Relational Model? 15](#_Toc27716)

[Relational Model Concepts 15](#_Toc14626)

[Model 16](#_Toc32700)

[View 16](#_Toc30461)

[Controller 16](#_Toc30711)

[What is MVVM pattern? 17](#_Toc10502)

[Software Testing 18](#_Toc342)

[Testing Strategies in Software Engineering 18](#_Toc24474)

[Summary of Software Testing Basics: 18](#_Toc21026)

[Integration Testing 19](#_Toc18880)

[System Test 24](#_Toc20409)

[Conclusion 34](#_Toc10241)

# Introduction

**What is a Exam Portal?**

Exam Portal has been actively used by many schools, as well the IEB, to administer computer based tests. This creates a standardized testing environment that eliminates nepotism, and unfair advantages.

**Why create a Online Examination plat-from?**

Due to current pandemic, Offline exam are not being held. so online examination are conducted by many institutes and universities. So I though of creating a solution where online exam can conducted with ease.

**Our app provides these features**

* Multi-user.
* GUI based Interface
* Admin panel
* Can run on any web browser.
* Secure and fast.
* New Student can register.
* Existing Student can Login.
* Simple Dashboard with Navigation bar.
* JWT Authentication
* Bcrypt Password encoder
* View Quiz Scores
* Field validation
* Time based Quiz.

# 

# Application Development Life Cycle

**What is SDLC?**

SDLC is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the defined time frame and cost. SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software. Every phase of the SDLC life Cycle has its own process and deliverable that feed into the next phase. SDLC stands for Software Development Life Cycle and is also referred to as the Application Development life-cycle

**Why SDLC?**

Here, are prime reasons why SDLC is important for developing a software system.

It offers a basis for project planning, scheduling, and estimating

Provides a framework for a standard set of activities and deliverable

It is a mechanism for project tracking and control

Increases visibility of project planning to all involved stakeholders of the development process

Increased and enhance development speed

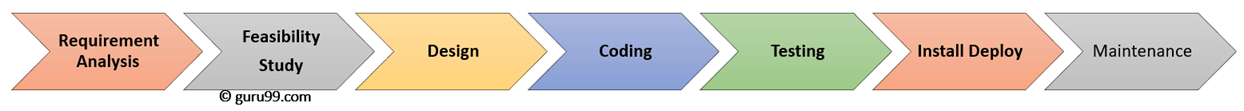
Improved client relations

Helps you to decrease project risk and project management plan overhead

## SDLC Phases

The entire SDLC process divided into the following stages:

* Phase 1: Requirement collection and analysis
* Phase 2: Feasibility study:
* Phase 3: Design:
* Phase 4: Coding:
* Phase 5: Testing:
* Phase 6: Installation/Deployment:
* Phase 7: Maintenance:



# 

# Requirements Gathering

**Requirements Gathering** is a fundamental part of any software development project. These are things like “User wants to do X. How is this achieved?” In effect, Requirements Gathering is the process of generating a list of requirements (functional, system, technical, etc.) from all the stakeholders (customers, users, vendors, IT staff) that will be used as the basis for the formal definition of what the project is.  
  
**What Techniques Can Be Used?**

**Interview**: turns out to be one of the most effective techniques for requirement gathering. In this method, the business analyst talks to the user and clients who are unable to give out detailed information as they are not aware of the system development and related functionalities

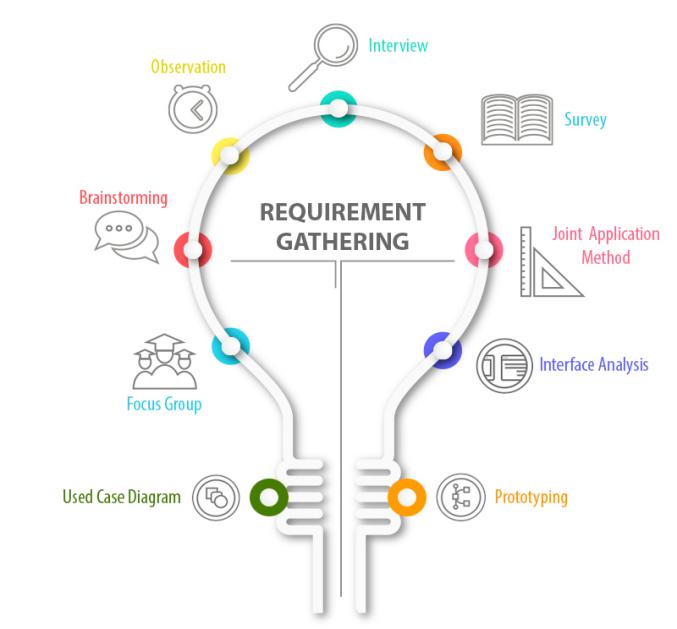
**Survey**: is another effective method to collect information and requirements within a short frame of time. Under this technique, it is advisable to first ascertain the goal of the survey and thereafter draft the questionnaire.

**Brainstorming**: SMEs or subject matter experts are the responsible people to conduct brainstorming sessions. They discuss and find out solutions to complex issues.

**Observation**: Under the observation method, the responsible person observes the team in working environment and gets ideas about the software and subsequently document the observation.

**Prototyping**: This is a technique of building a model of software which helps in uncovering and capturing software requirements from client. The output can be broad mockups or sketch formats of software.

**Use case diagram**: is a technique that shows how people interact with software. It shows what a system does.

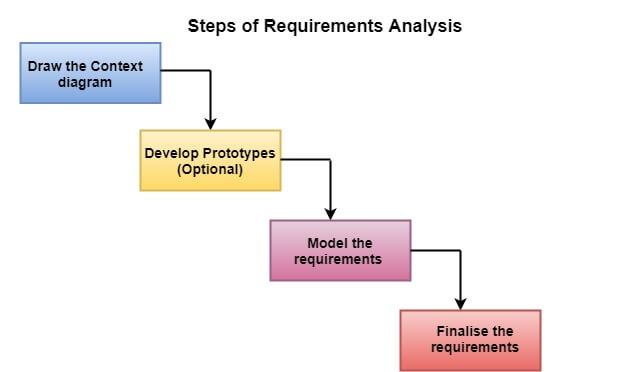


# Requirements Analysis

**Requirement analysis** is significant and essential activity after elicitation. We analyze, refine, and scrutinize the gathered requirements to make consistent and unambiguous requirements. This activity reviews all requirements and may provide a graphical view of the entire system. After the completion of the analysis, it is expected that the understand-ability of the project may improve significantly. Here, we may also use the interaction with the customer to clarify points of confusion and to understand which requirements are more important than others.  
  
**(i) Draw the context diagram:** The context diagram is a simple model that defines the boundaries and interfaces of the proposed systems with the external world. It identifies the entities outside the proposed system that interact with the system. The context diagram of student result management system is given below:

**(ii) Development of a Prototype (optional):** One effective way to find out what the customer wants is to construct a prototype, something that looks and preferably acts as part of the system they say they want.

**(iii) Model the requirements:** This process usually consists of various graphical representations of the functions, data entities, external entities, and the relationships between them. The graphical view may help to find incorrect, inconsistent, missing, and superfluous requirements. Such models include the Data Flow diagram, Entity-Relationship diagram, Data Dictionaries, State-transition diagrams, etc.

**(iv) Finalise the requirements:** After modeling the requirements, we will have a better understanding of the system behavior. The inconsistencies and ambiguities have been identified and corrected. The flow of data amongst various modules has been analyzed. Elicitation and analyze activities have provided better insight into the system. Now we finalize the analyzed requirements, and the next step is to document   


# Application Design

**Design** is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages.

Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfill the requirements mentioned in SRS.

There are two kinds of design documents developed in this phase:

**High-Level Design (HLD)**

* Brief description and name of each module
* An outline about the functionality of every module
* Interface relationship and dependencies between modules
* Database tables identified along with their key elements
* Complete architecture diagrams along with technology details

**Low-Level Design(LLD)**

* Functional logic of the modules
* Database tables, which include type and size
* Complete detail of the interface
* Addresses all types of dependency issues
* Listing of error messages
* Complete input and outputs for every module

## REST Architecture

REST stands for REpresentational State Transfer and API stands for Application Program Interface. REST is a software architectural style that defines the set of rules to be used for creating web services. Web services which follow the REST architectural style are known as RESTful web services. It allows requesting systems to access and manipulate web resources by using a uniform and predefined set of rules. Interaction in REST based systems happen through Internet’s Hypertext Transfer Protocol (HTTP).

Architectural Constraints of RESTful API: There are six architectural constraints which makes any web service are listed below:

* Uniform Interface
* Stateless
* Cacheable
* Client-Server
* Layered System
* Code on Demand

A Restful system consists of a:

client who requests for the resources.

server who has the resources.

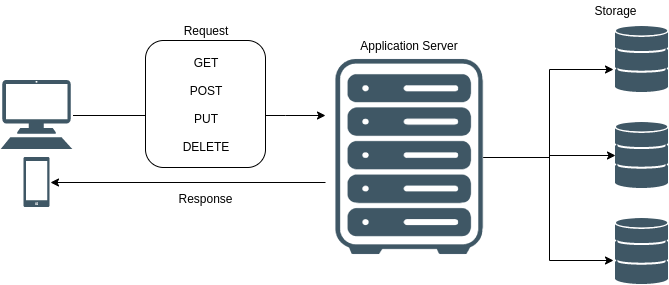
HTTP verbs:

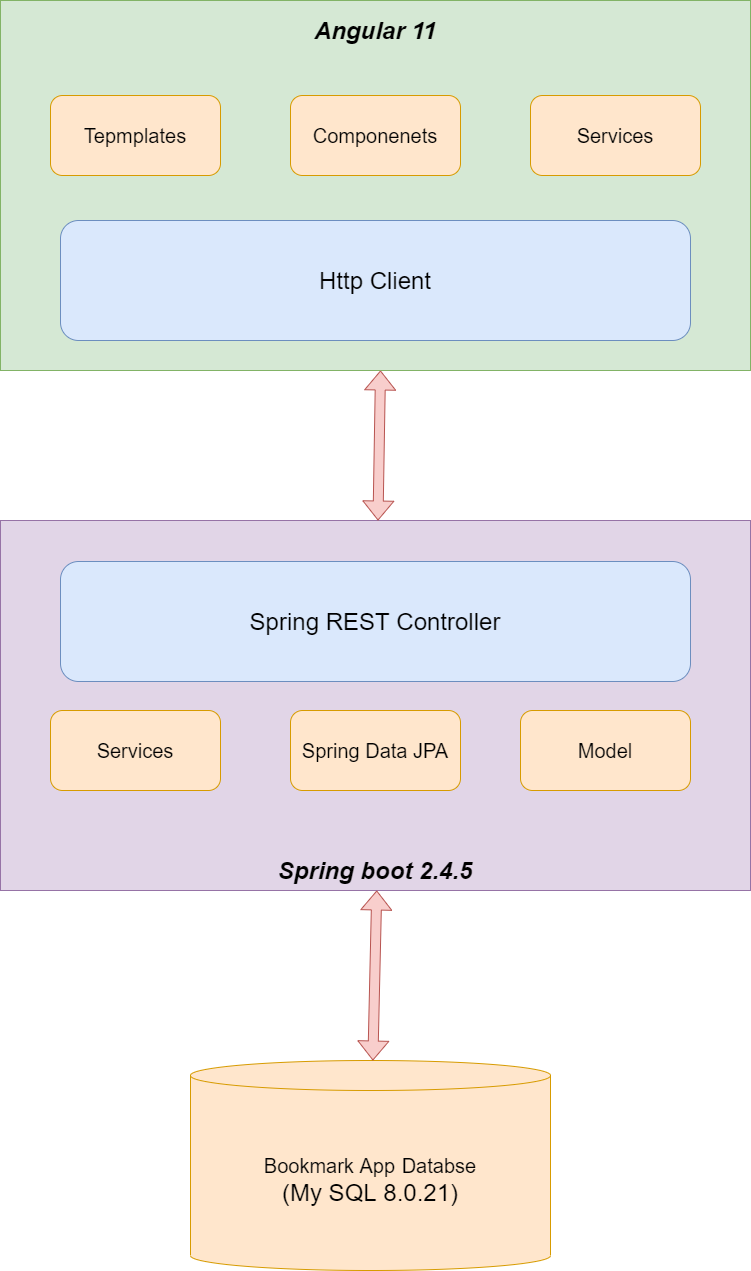
**GET**: Retrieves one or more resources identified by the request URI and it can cache the information receive.

**POST**: Create a resource from the submission of a request and response is not cacheable in this case. This method is unsafe if no security is applied to the endpoint as it would allow anyone to create a random resource by submission.

**PUT**: Update an existing resource on the server specified by the request URI.

**DELETE**: Delete an existing resource on the server specified by the request URI. It always return an appropriate HTTP status for every request.



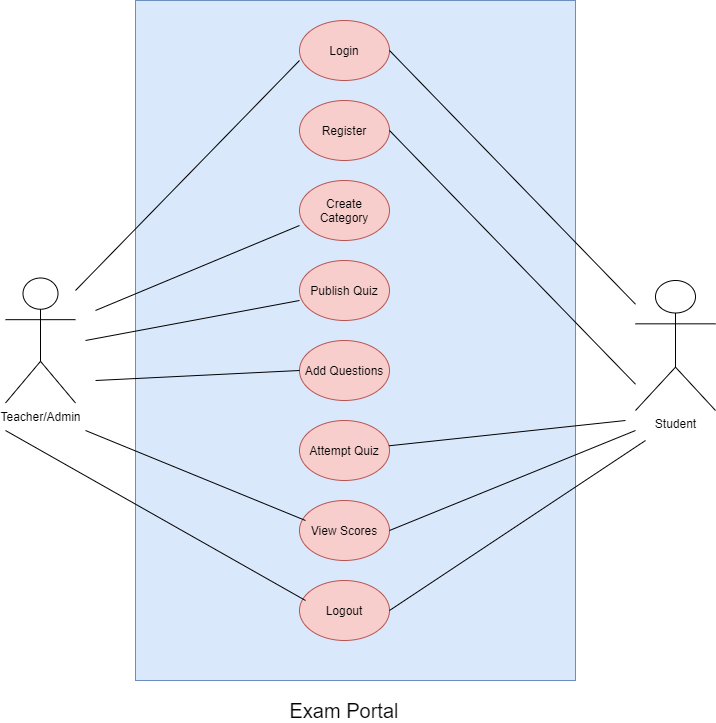
  
*Architectural design*

## What is a use case diagram?

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

Scenarios in which your system or application interacts with people, organizations, or external systems

Goals that your system or application helps those entities (known as actors) achieve



## What is JWT (JSON Web Token)?

JSON Web Token (JWT) is an open standard ([RFC 7519](https://tools.ietf.org/html/rfc7519)) for securely transmitting information between parties as JSON object.

It is compact, readable and digitally signed using a private key/ or a public key pair by the Identity Provider(IdP). So the integrity and authenticity of the token can be verified by other parties involved.

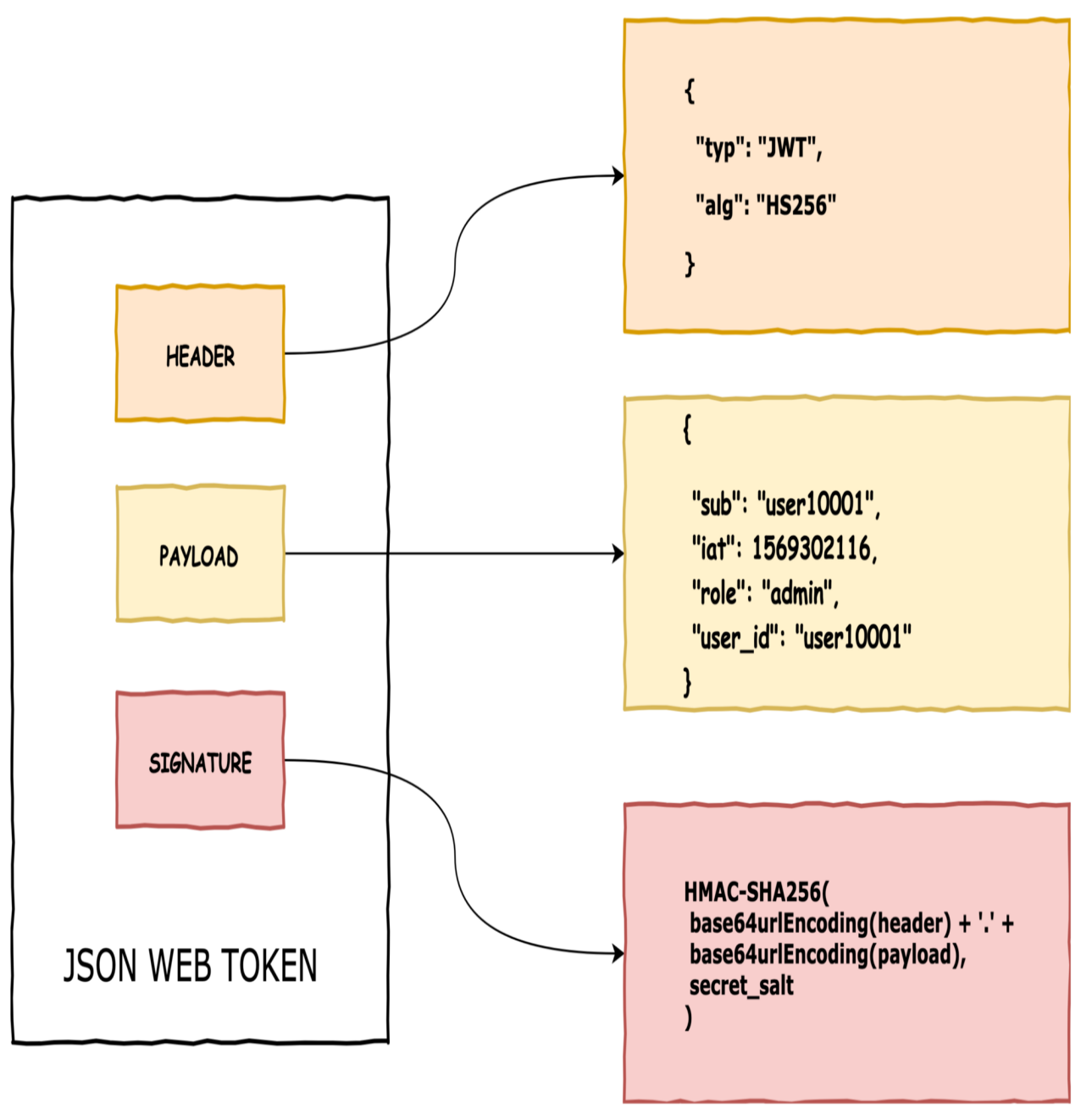
The purpose of using JWT is not to hide data but to ensure the authenticity of the data. JWT is signed and encoded, not encrypted.

JWT is a token based stateless authentication mechanism. Since it is a client-side based stateless session, server doesn’t have to completely rely on a datastore(database) to save session information.

## Structure of JWT

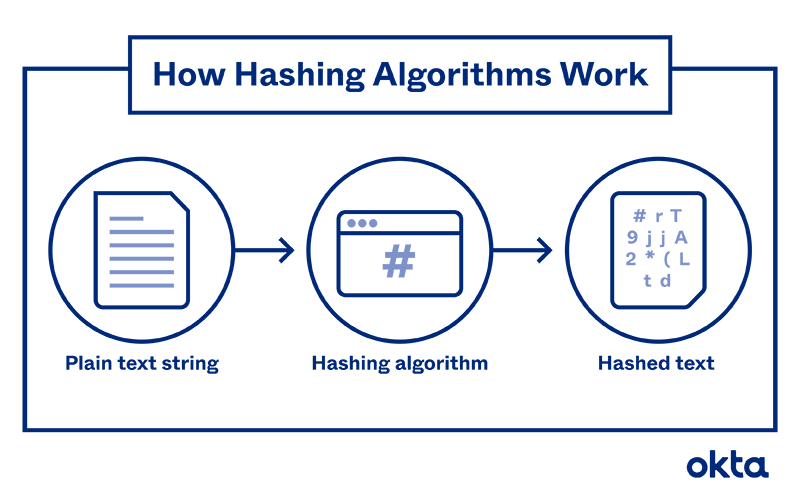
A JSON Web Token consists of 3 parts separated by a period.

header.payload.signature



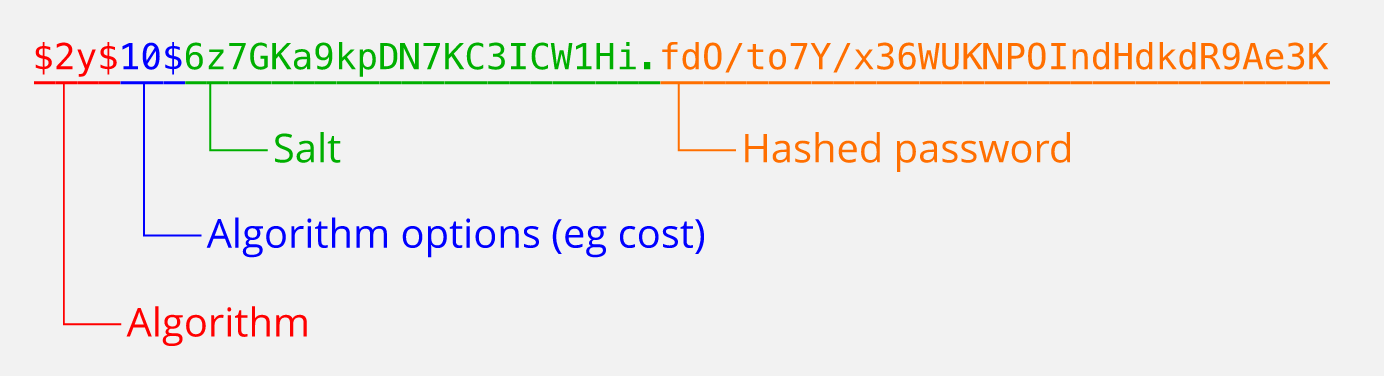
## What is hashing ?

Hashing is **simply passing some data through a formula that produces a result**, called a hash. That hash is usually a string of characters and the hashes generated by a formula are always the same length, regardless of how much data you feed into it

.

## What is Bcrypt password hashing algortihm ?

bcrypt is a password-hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher and presented at USENIX in 1999.Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.



## What is ER Diagram?

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

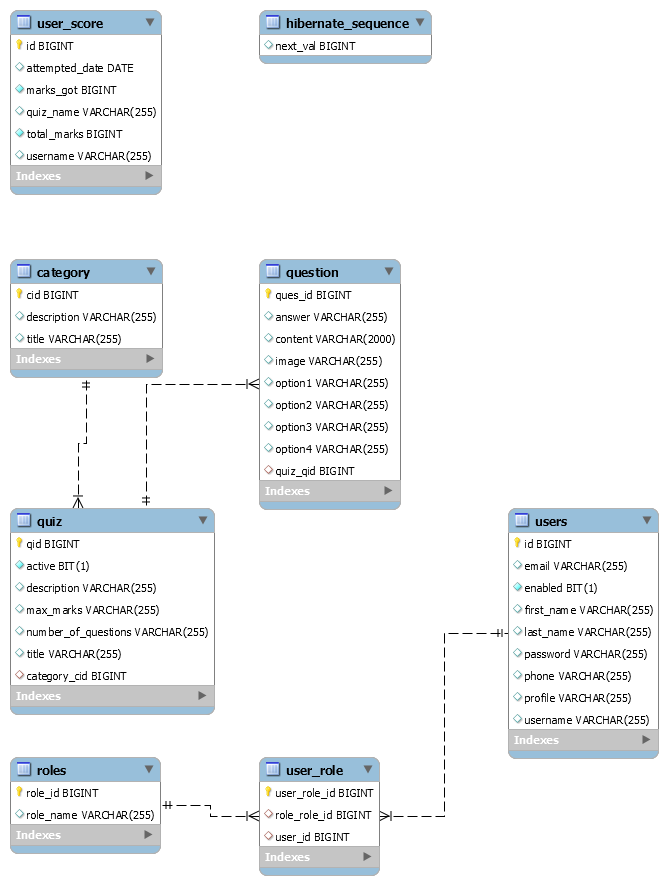
ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.

## Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

* Helps you to define terms related to entity relationship modeling
* Provide a preview of how all your tables should connect, what fields are going to be on each table
* Helps to describe entities, attributes, relationships
* ER diagrams are translatable into relational tables which allows you to build databases quickly
* ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
* The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
* ERD Diagram allows you to communicate with the logical structure of the database to users



# Project implementation

* Oops principle
* Relational pattern in database
* MVC pattern in Spring boot
* MVVM pattern in Angular

|  |  |
| --- | --- |
| Application/Layer | Technology/Tool used |
| Front-end | Angular 11 |
| Back-end | Spring boot 2.4.5 |
| Database | My SQL 8.0.21 |
| Java development kit | v1.8.0\_231 |
| JavaScript run-time environment | Node js v14.15.1 |
| Version control and Source code Management tool | git version 2.24.1.windows.2 |
| IDE | IntelliJ IDEA 2021.1 (Community Edition)  Visual Studio Code v1.55.2 |
| Web browser | Google Chrome v90.0.4430.93 (64-bit) |
| MySQL client | MySQL Workbench 8.0 |
| Rest Client | Postman v8.3.0 |
| Project build tool | Apache Maven 3.6.3 |

## 

## What is OOPS?

**Object-Oriented Programming System (OOPs)** is a programming concept that works on the principles of abstraction, encapsulation, inheritance, and polymorphism. It allows users to create objects they want and create methods to handle those objects. The basic concept of OOPs is to create objects, re-use them throughout the program, and manipulate these objects to get results.  
  
The following are general OOPs concepts in Java:

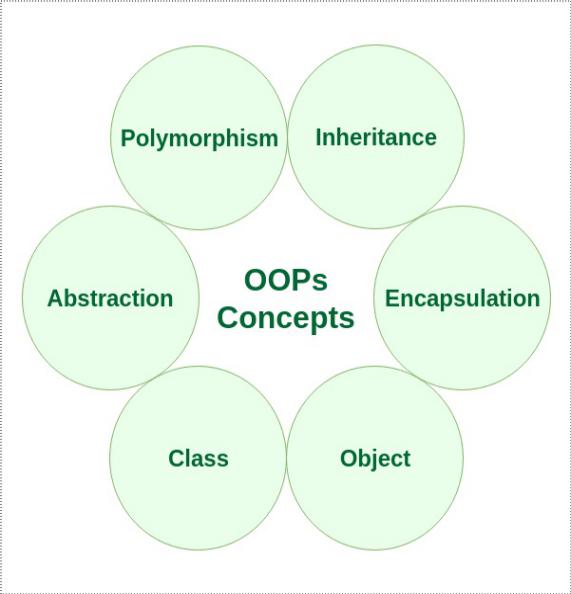
**Class:** The class is a like template or an object factory.  
**Object:** An object can be defined as an instance of a class, and there can be multiple instances of a class in a program.

**Inheritance**: one object acquires the properties and behaviors of the parent object. It’s creating a parent-child relationship between two classes. It offers robust and natural mechanism for organizing and structure of any software.

**Polymorphism:** which is the ability of a variable, object or function to take on multiple forms.

**Abstraction:** which is an act of representing essential features without including background details. It is a technique of creating a new data type that is suited for a specific application.

**Encapsulation:** wrapping the data and code. In this OOPs concept, the variables of a class are always hidden from other classes. It can only be accessed using the methods of their current class.



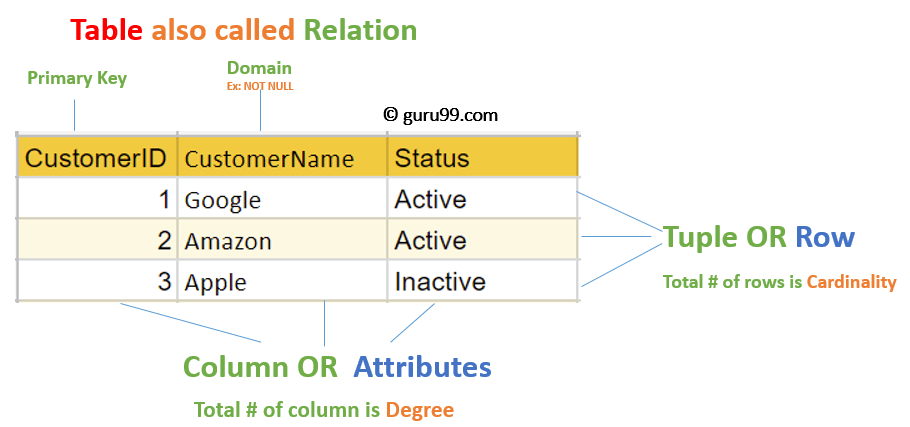
## What is Relational Model?

**Relational Model (RM)** represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized..

## Relational Model Concepts

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student\_Rollno, NAME,etc.
2. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
3. **Tuple** – It is nothing but a single row of a table, which contains a single record.
4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.
5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.
6. **Cardinality:**Total number of rows present in the Table.
7. **Column:** The column represents the set of values for a specific attribute.
8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain



What is MVC pattern?  
  
The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects

### Model

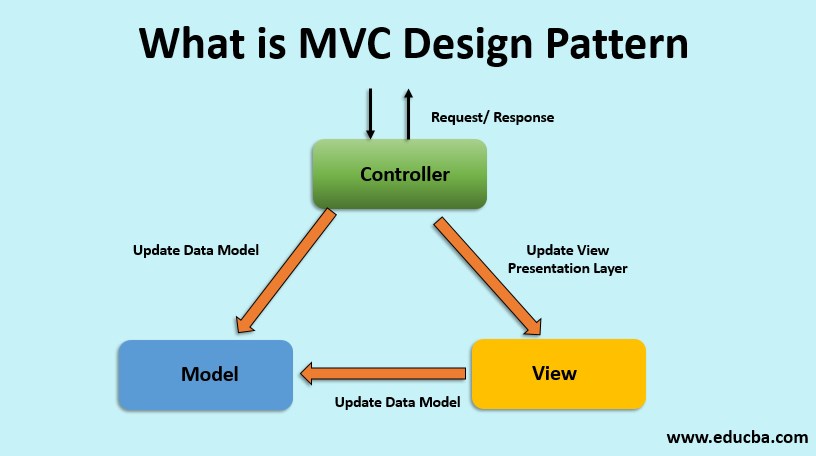
The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

### View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

### Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.



## What is MVVM pattern?

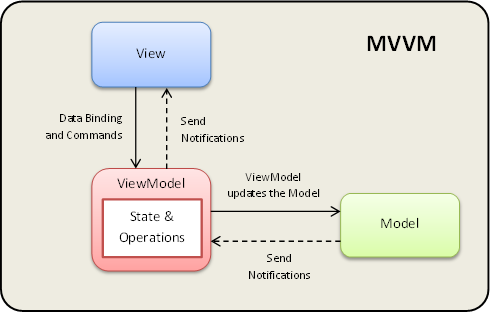
Developers always prefer a clean and structured code for the projects. By organizing the codes according to a design pattern helps in the maintenance of the software. By having knowledge of all crucial logic parts of the android application, it is easier to add and remove app features. Further, design patterns also assure that all the codes get covered in Unit Testing without the interference of other classes. Model — View — ViewModel (MVVM) is the industry-recognized software architecture pattern that overcomes all drawbacks of MVP and MVC design patterns. MVVM suggests separating the data presentation logic(Views or UI) from the core business logic part of the application.

The separate code layers of MVVM are:

**Model**: This layer is responsible for the abstraction of the data sources. Model and ViewModel work together to get and save the data.

**View**: The purpose of this layer is to inform the ViewModel about the user’s action. This layer observes the ViewModel and does not contain any kind of application logic.

**ViewModel**: It exposes those data streams which are relevant to the View. Moreover, it servers as a link between the Model and the View.



# Software Testing

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

## Testing Strategies in Software Engineering

**Unit Testing:**This software testing approach is followed by the programmer to test the unit of the program. It helps developers to know whether the individual unit of the code is working properly or not.

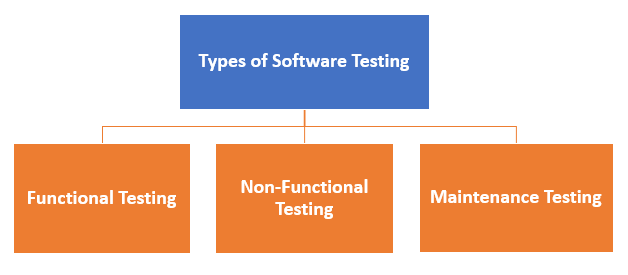
**Integration testing:**It focuses on the construction and design of the software. You need to see that the integrated units are working without errors or not.

**System testing:**In this method, your software is compiled as a whole and then tested as a whole. This testing strategy checks the functionality, security, portability, amongst others.

### Summary of Software Testing Basics:

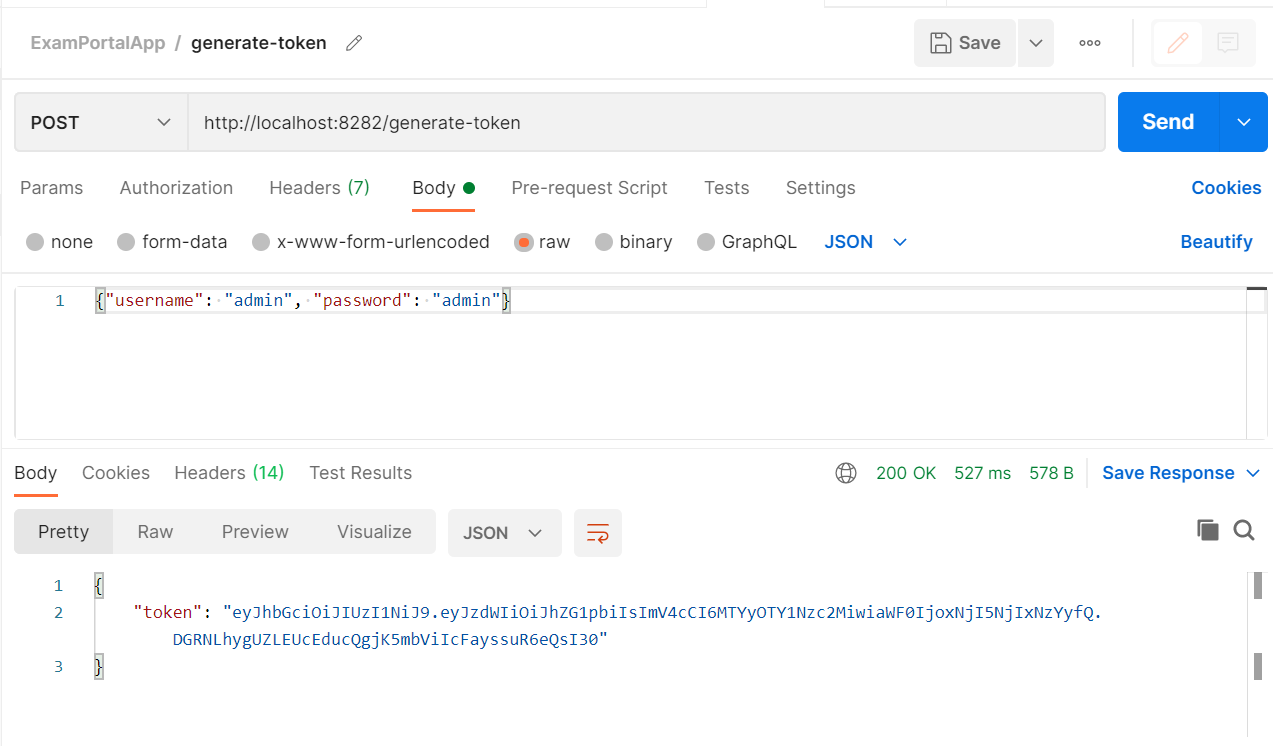
* Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free.
* Testing is important because software bugs could be expensive or even dangerous.
* The important are reasons for using software testing are: cost-effective, security, product quality, and customer satisfaction.

**Functional testing** is a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing, and it is not concerned about the source code of the application.  
  
**Non-functional testing** is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It is explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

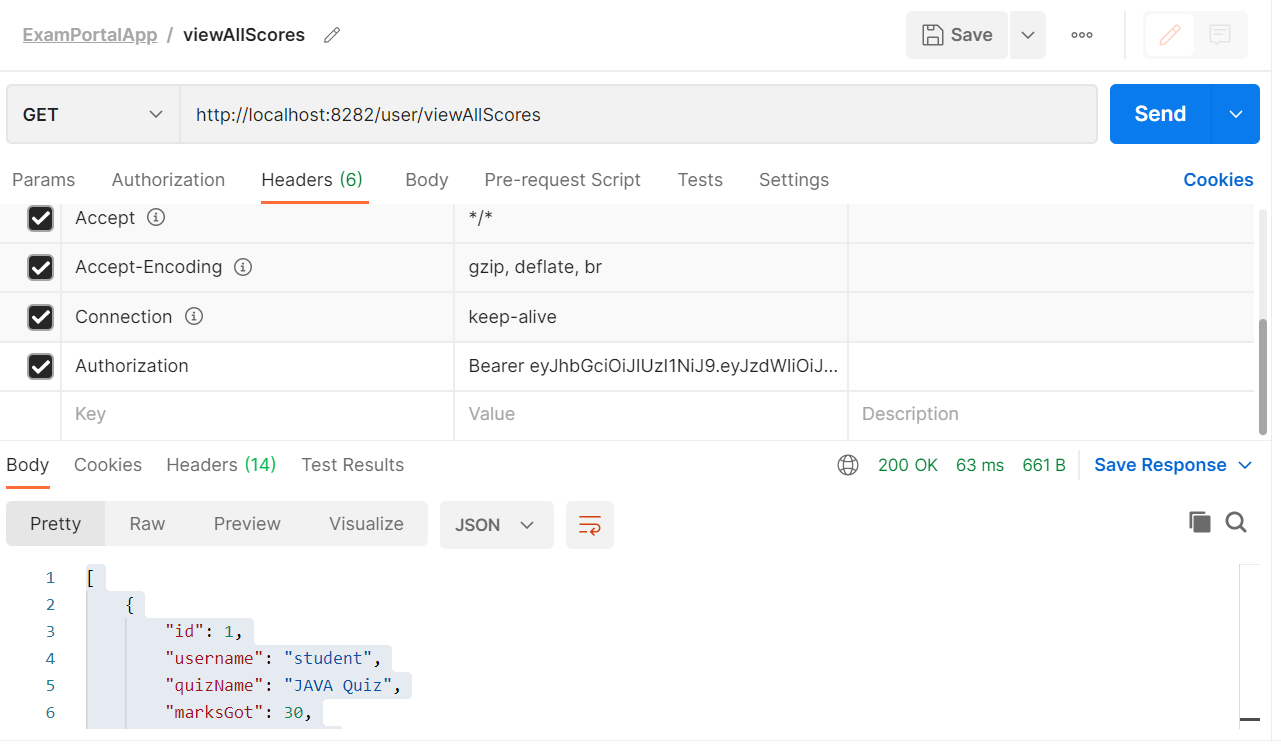


## Integration Testing

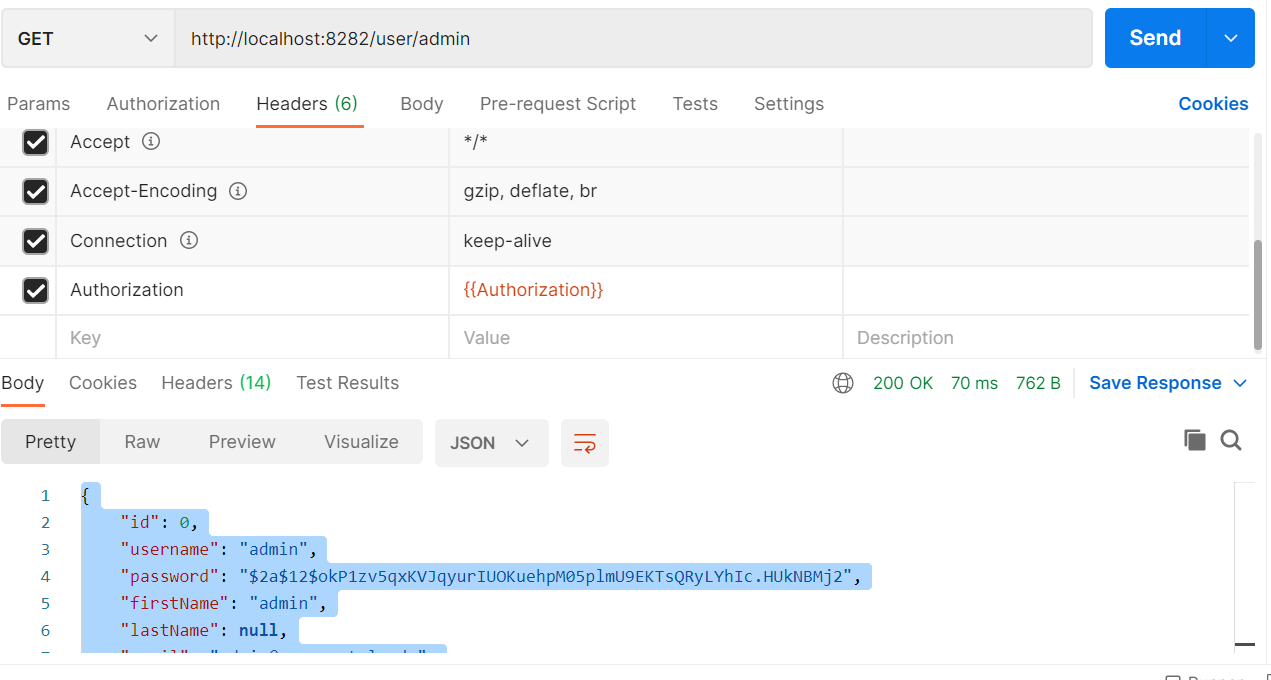
|  |  |
| --- | --- |
| Test Case 1: User Validation | |
| Action | Send POST Request |
| Endpoint URL | http://localhost:8282/generate-token |
| Request Body | {"username": "admin", "password": "admin"} |
| Excepted Response | Generated JWT token |
| Actual Response | {      "token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbiIsImV4cCI6MTYyOTY1Nzc2MiwiaWF0IjoxNjI5NjIxNzYyfQ.DGRNLhygUZLEUcEducQgjK5mbViIcFayssuR6eQsI30"  } |
| Status | PASS |



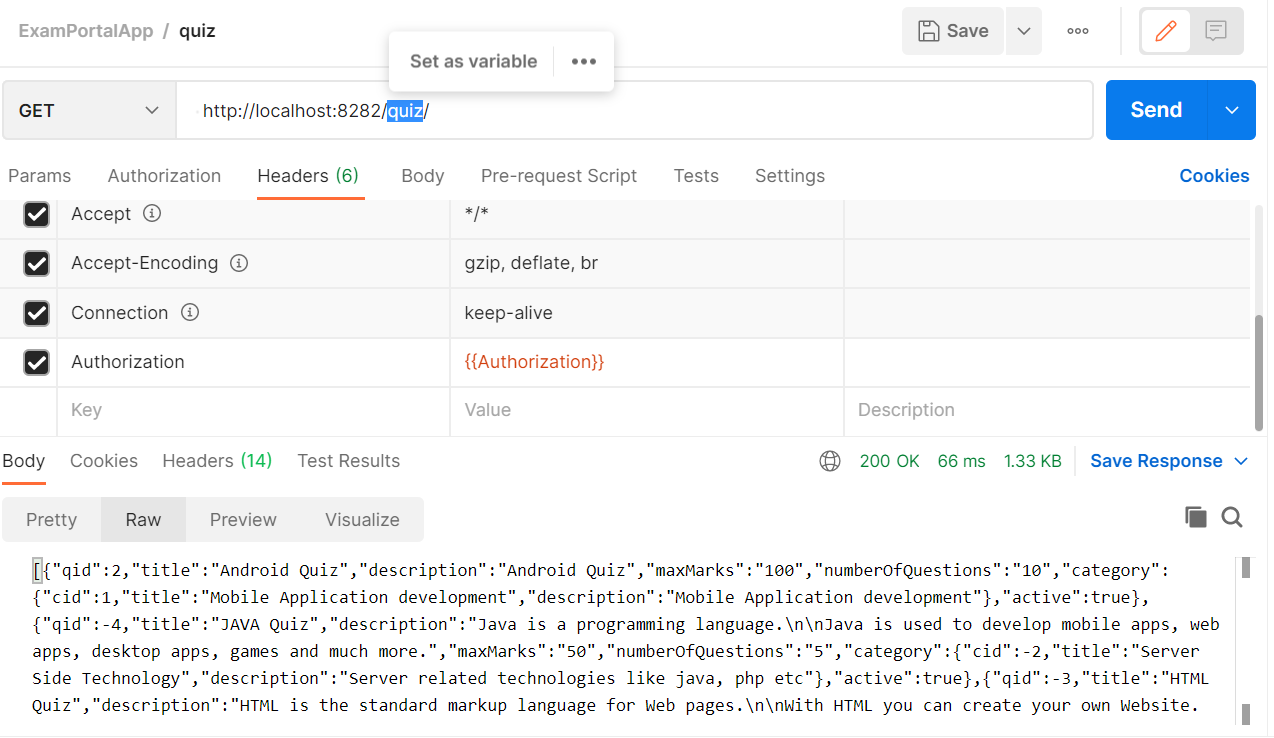
|  |  |
| --- | --- |
| Test Case 2: view all scores | |
| Action | Send GET Request |
| Endpoint URL | http://localhost:8282/user/viewAllScores |
| Request Body | Authorization Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbiIsImV4cCI6MTYyOTY1NzY4MiwiaWF0IjoxNjI5NjIxNjgyfQ.5r\_VjkzexT1u7I-hCwGH1Y-OOLO3CcYRcrBv-7eQ7ic |
| Excepted Response | Returns a list of scores |
| Actual Response | [  {  "id": 1,  "username": "student",  "quizName": "JAVA Quiz",  "marksGot": 30,  "totalMarks": 50,  "attemptedDate": "2021-08-22"  },  {  "id": 2,  "username": "student",  "quizName": "HTML Quiz",  "marksGot": 55,  "totalMarks": 100,  "attemptedDate": "2021-08-22"  }  ] |
| Status | PASS |



|  |  |
| --- | --- |
| Test Case 3: View profile | |
| Action | Send GET Request |
| Endpoint URL | http://localhost:8282/user/admin |
| Request Body |  |
| Excepted Response | Display admin profile information |
| Actual Response | {      "id": 0,      "username": "admin",      "password": "$2a$12$okP1zv5qxKVJqyurIUOKuehpM05plmU9EKTsQRyLYhIc.HUkNBMj2",      "firstName": "admin",      "lastName": **null**,      "email": "admin@examportal.edu",      "phone": **null**,      "enabled": **true**,      "profile": **null**,      "authorities": [          {              "authority": "ADMIN"          }      ],      "accountNonExpired": **true**,      "credentialsNonExpired": **true**,      "accountNonLocked": **true**  } |
| Status | PASS |



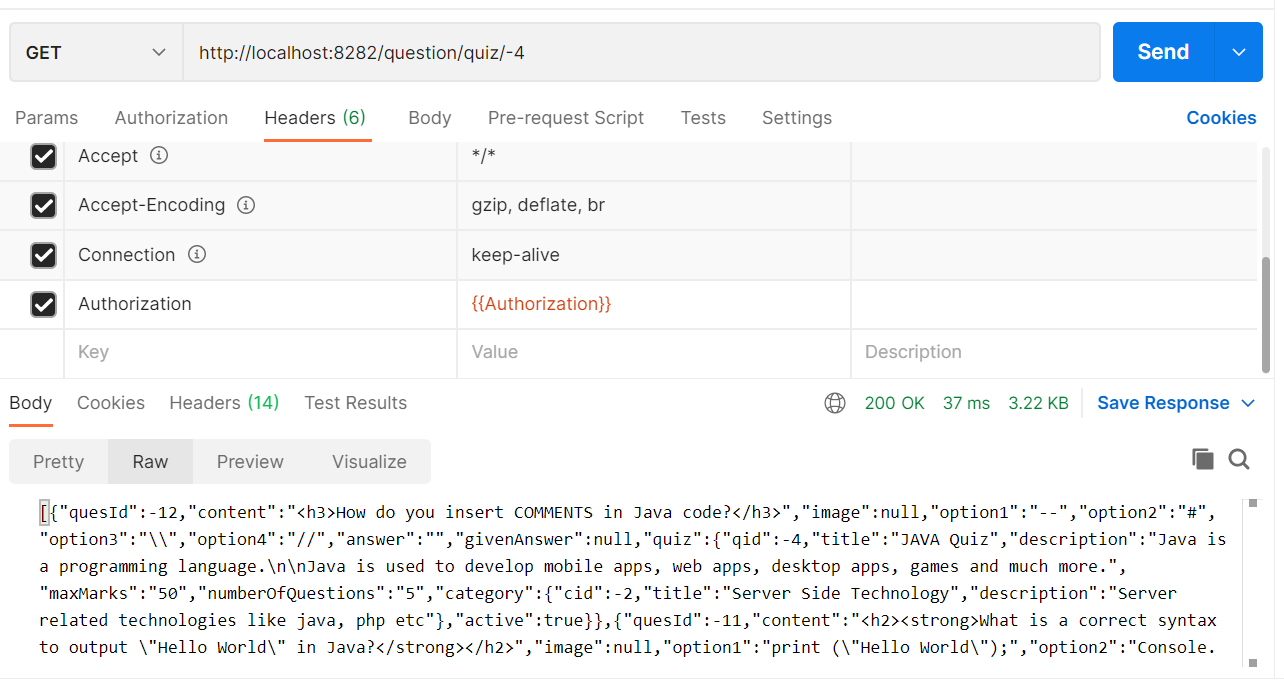
|  |  |
| --- | --- |
| Test Case 4: get all quiz | |
| Action | Send GET Request |
| Endpoint URL | http://localhost:8282/quiz/ |
| Request Body |  |
| Excepted Response | List of all quiz |
| Actual Response | [{"qid":2,"title":"Android Quiz","description":"Android Quiz","maxMarks":"100","numberOfQuestions":"10","category":{"cid":1,"title":"Mobile Application development","description":"Mobile Application development"},"active":true},{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true},{"qid":-3,"title":"HTML Quiz","description":"HTML is the standard markup language for Web pages.\n\nWith HTML you can create your own Website.\n\nHTML is easy to learn - You will enjoy it!","maxMarks":"100","numberOfQuestions":"10","category":{"cid":-1,"title":"Web Technology","description":"Web related technologies quiz like html, css etc"},"active":true}] |
| Status | PASS |



\\

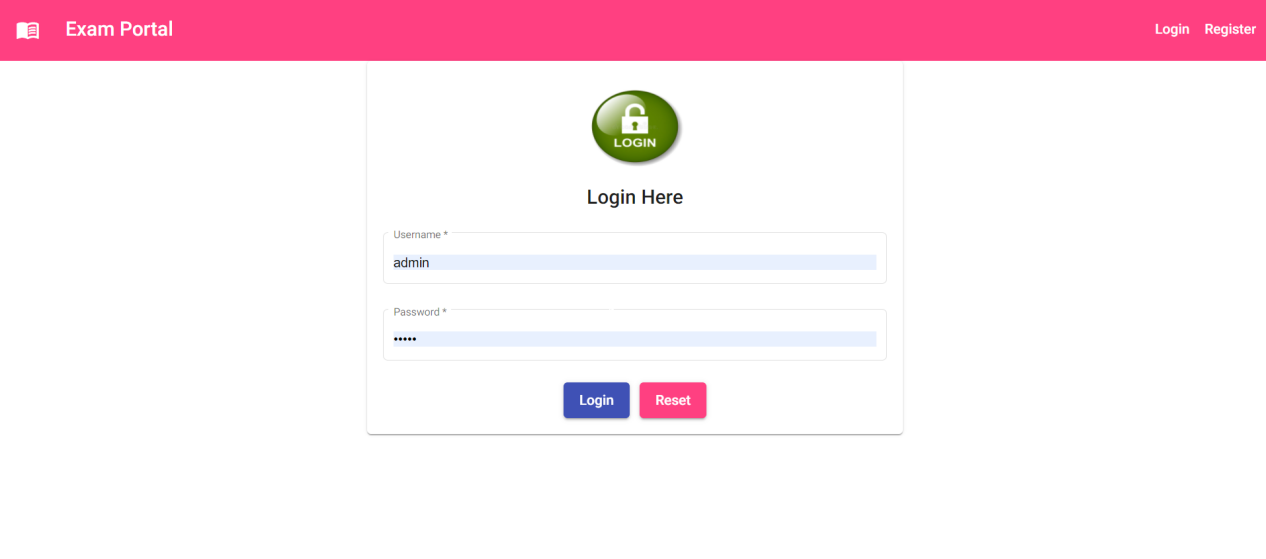
[\\\](\\)

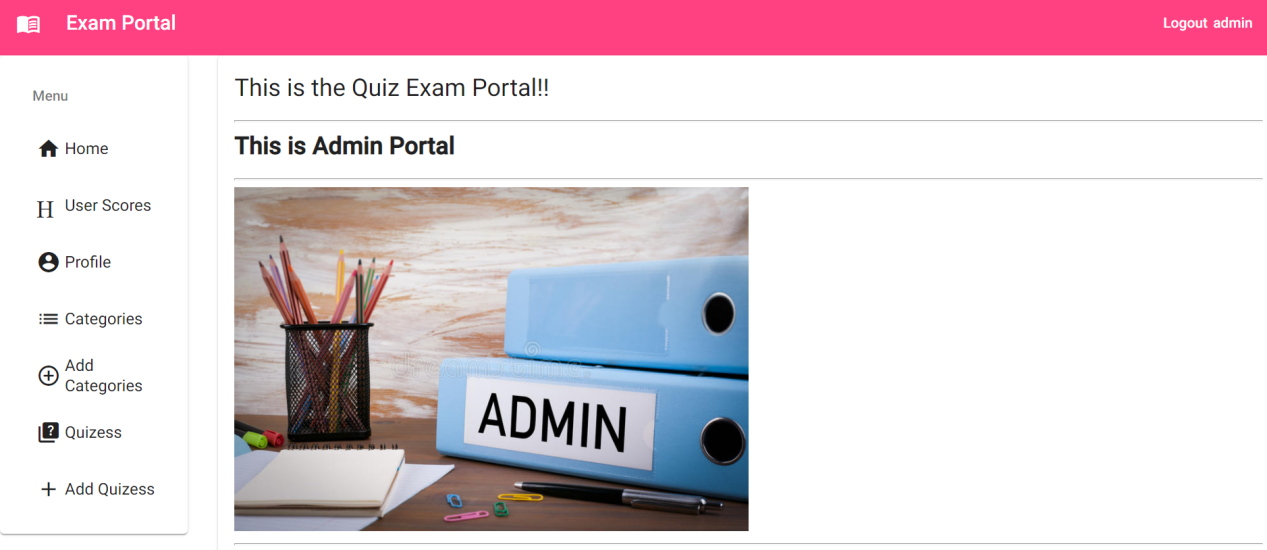
|  |  |
| --- | --- |
| Test Case 5: get all question for a quiz | |
| Action | Send GET Request |
| Endpoint URL | http://localhost:8282/question/quiz/-4 |
| Request Body |  |
| Excepted Response | Get the list of question in json format |
| Actual Response | [{"quesId":-12,"content":"<h3>How do you insert COMMENTS in Java code?</h3>","image":null,"option1":"--","option2":"#","option3":"\\","option4":"//","answer":"","givenAnswer":null,"quiz":{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true}},{"quesId":-11,"content":"<h2><strong>What is a correct syntax to output \"Hello World\" in Java?</strong></h2>","image":null,"option1":"print (\"Hello World\");","option2":"Console.WriteLine(\"Hello World\");","option3":"System.out.println(\"Hello World\");","option4":"echo(\"Hello World\");","answer":"","givenAnswer":null,"quiz":{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true}},{"quesId":-13,"content":"<h3>Which data type is used to create a variable that should store text?</h3>","image":null,"option1":"Str","option2":"string","option3":"String","option4":"varchar","answer":"","givenAnswer":null,"quiz":{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true}},{"quesId":-15,"content":"<h3>Which operator can be used to compare two integer values?</h3>","image":null,"option1":"=","option2":"==","option3":"equals()","option4":"<>","answer":"","givenAnswer":null,"quiz":{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true}},{"quesId":-14,"content":"<h3>Which method can be used to find the length of a string?</h3>","image":null,"option1":"len()","option2":"getLength()","option3":"length()","option4":"size()","answer":"","givenAnswer":null,"quiz":{"qid":-4,"title":"JAVA Quiz","description":"Java is a programming language.\n\nJava is used to develop mobile apps, web apps, desktop apps, games and much more.","maxMarks":"50","numberOfQuestions":"5","category":{"cid":-2,"title":"Server Side Technology","description":"Server related technologies like java, php etc"},"active":true}}] |
| Status | PASS |



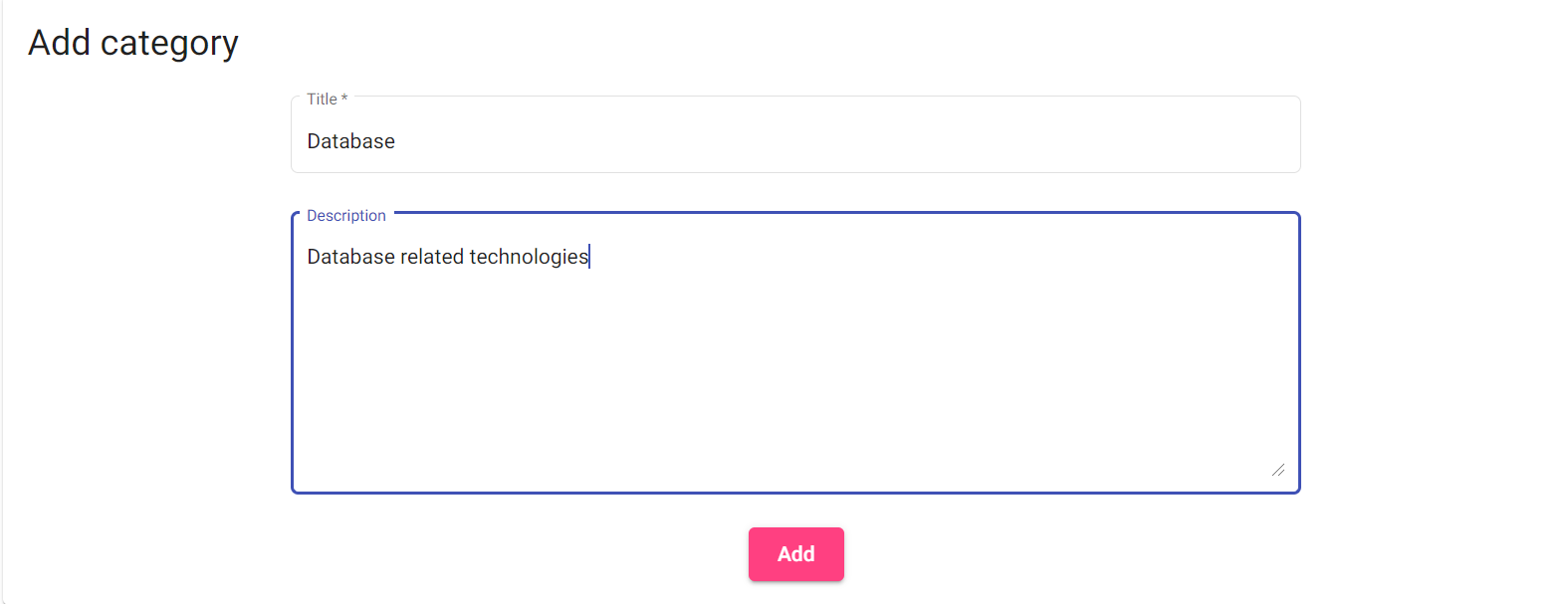
## System Test

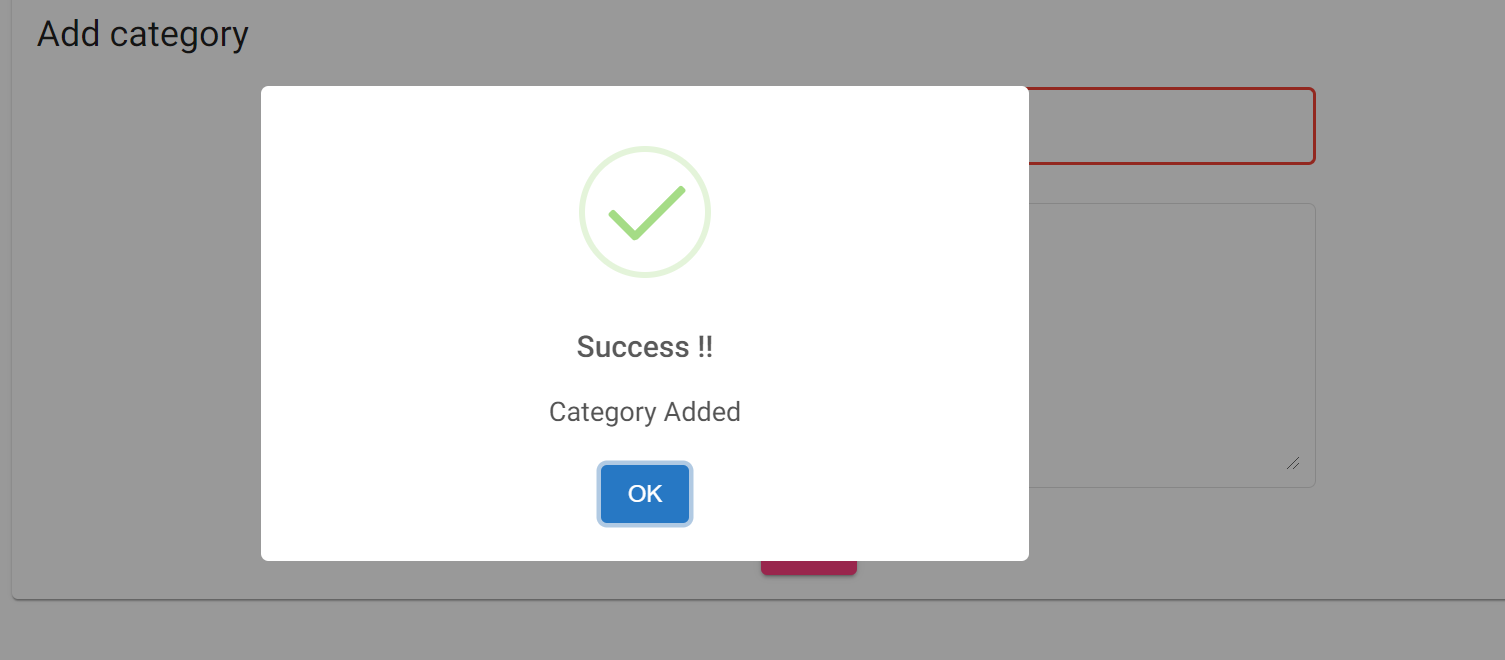
|  |  |
| --- | --- |
| Test case 1 : Login | |
| Action | Open Google chrome browser click on Login link on the nav bar enter username as admin  Password as admin |
| Endpoint URL | http://localhost:4200/login |
| Excepted Response | Login successful |
| Actual Response | Login successful |
| Status | PASS |



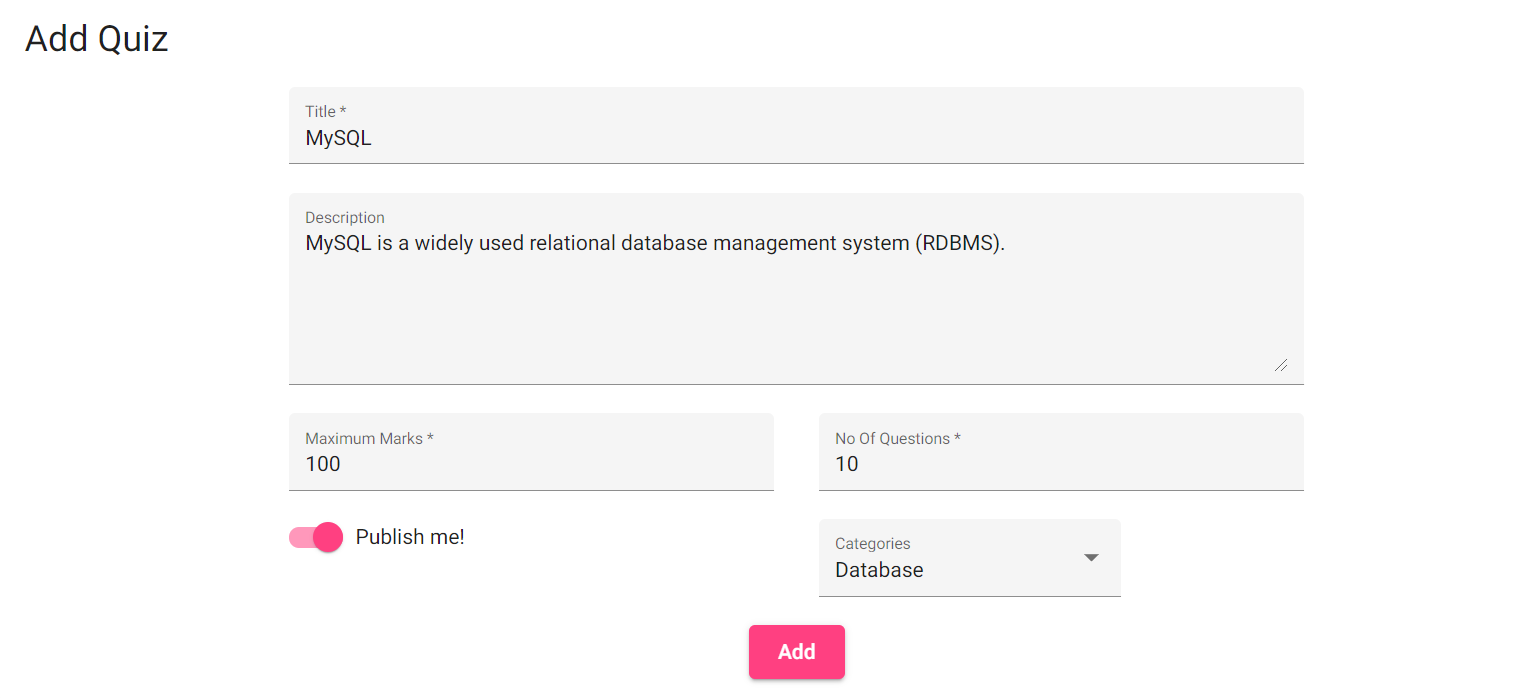


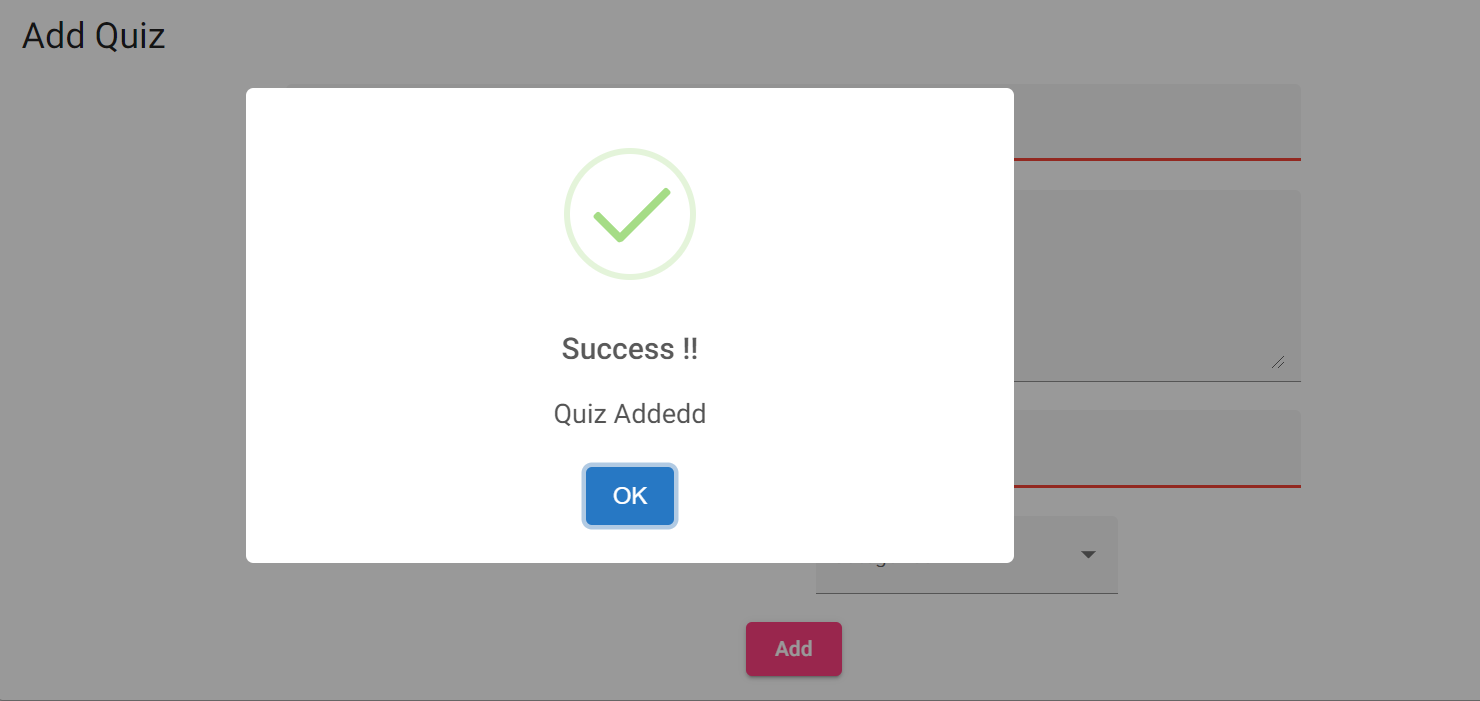
|  |  |
| --- | --- |
| Test case 2 : Add category | |
| Action | Open Google chrome browser  Enter  Title : Database  Description: Database related technologies |
| Endpoint URL | http://localhost:4200/admin/categories |
| Excepted Response | Success category added |
| Actual Response | Success category added |
| Status | PASS |



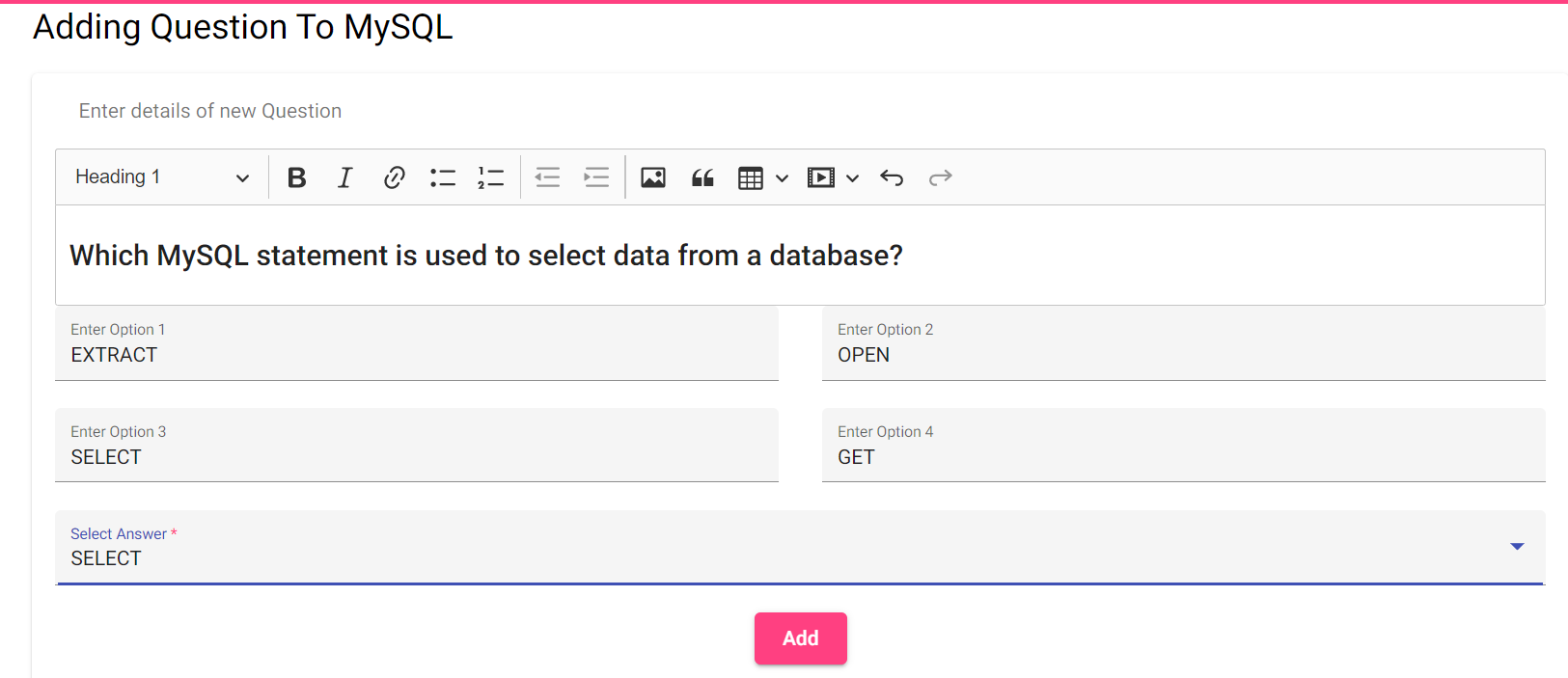


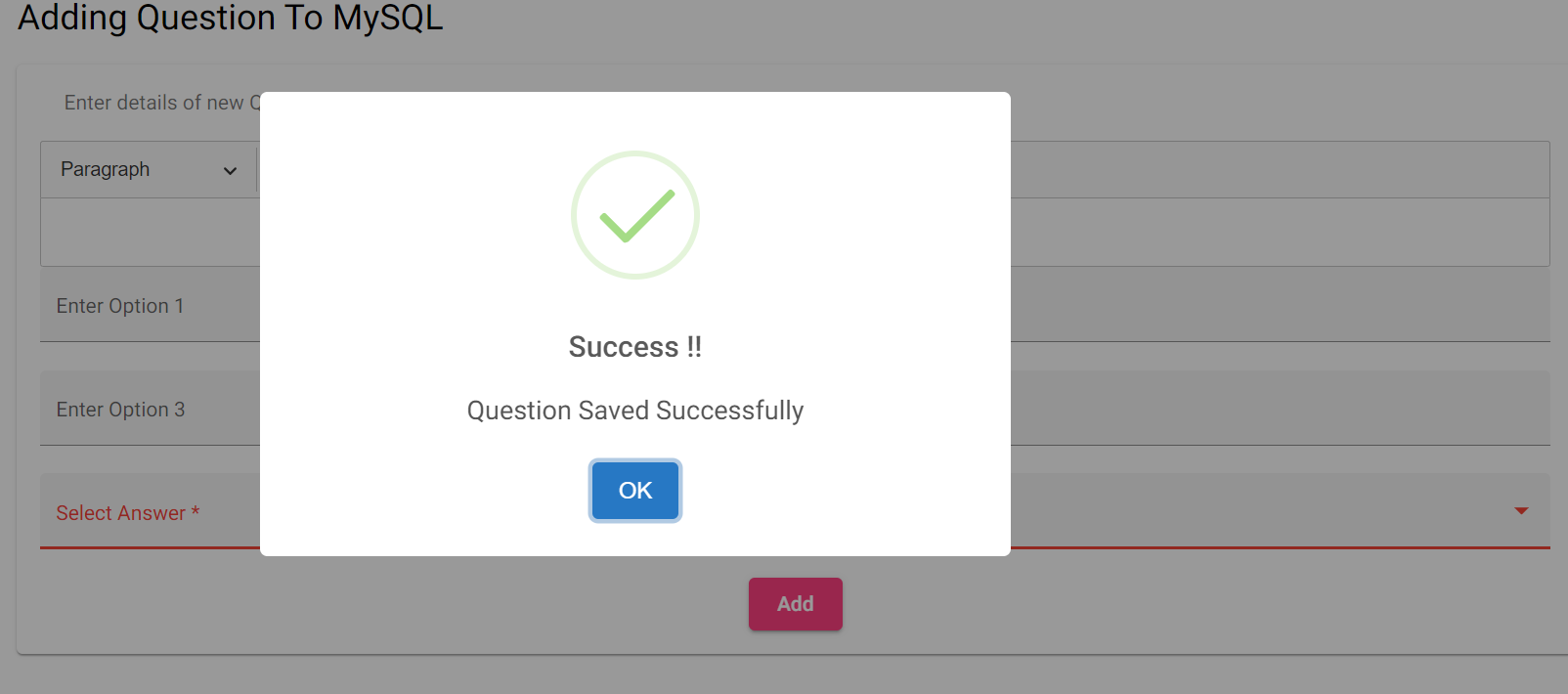
|  |  |
| --- | --- |
| Test case 3 : Add quiz | |
| Action | Enter  Title: MySQL  Description : MySQL is a widely used relational database management system (RDBMS).  Marks: 100  No of question: 10 |
| Endpoint URL | http://localhost:4200/admin/add-quiz |
| Excepted Response | Success quiz added |
| Actual Response | Success quiz added |
| Status | PASS |



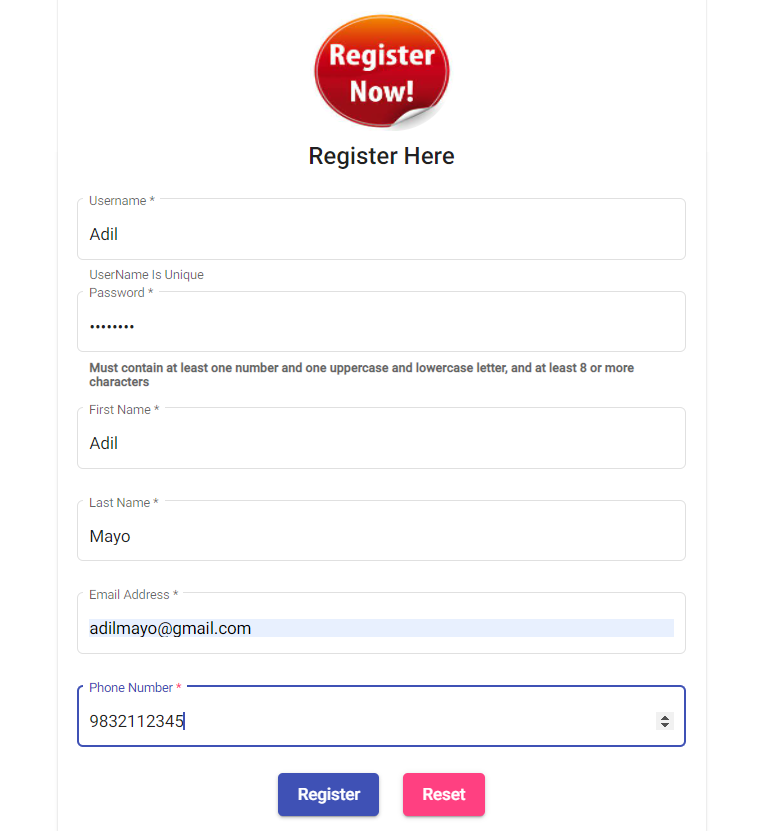


|  |  |
| --- | --- |
| Test case 4 : Add question | |
| Action | Enter a new question Enter 4 options  Enter correct answer |
| Endpoint URL | http://localhost:4200/admin/add-question/5/MySQL |
| Excepted Response | Success question added |
| Actual Response | Success question added |
| Status | PASS |

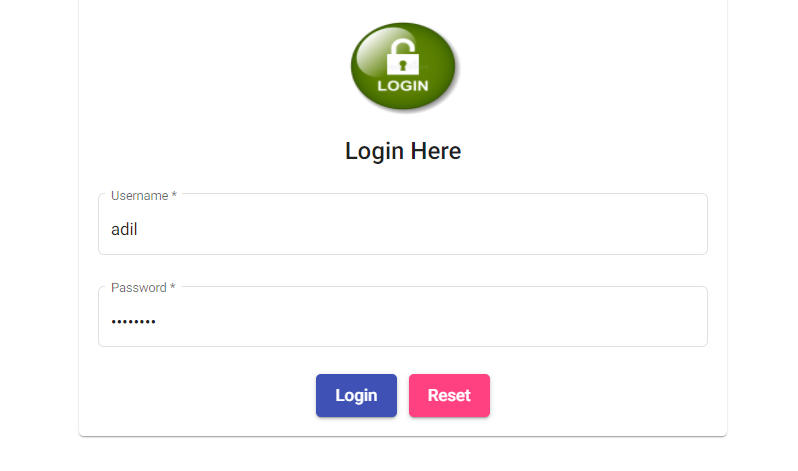


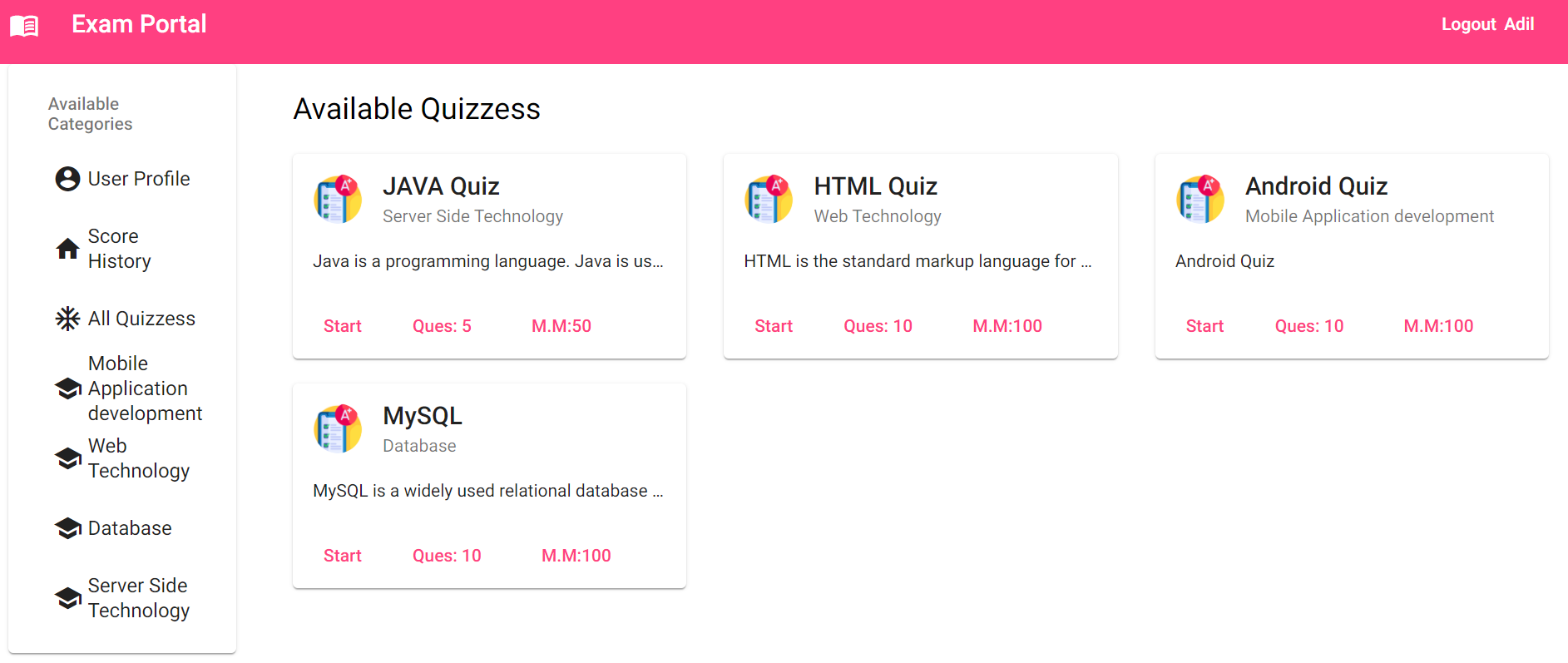


|  |  |
| --- | --- |
| Test case 5 : Sign-up | |
| Action | Enter all required fields **password validation**: Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters  Email validation  Phone number : must of length 10 should starts with 7-9 |
| Endpoint URL | http://localhost:4200/signup |
| Excepted Response | User registered successfully |
| Actual Response | User registered successfully |
| Status | PASS |

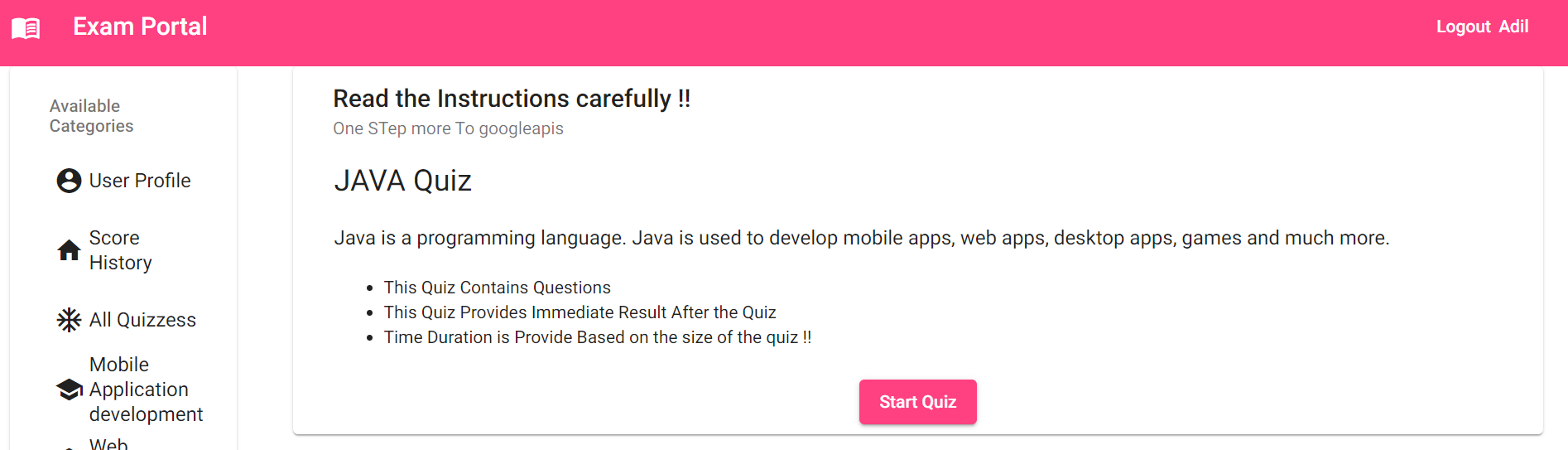


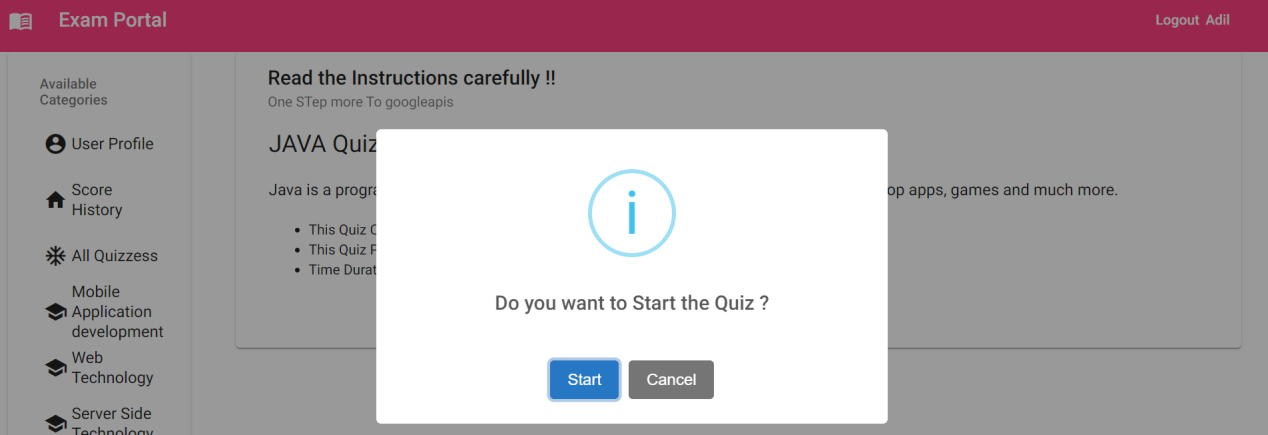
|  |  |
| --- | --- |
| Test case 6 : Student login | |
| Action | Open Google chrome browser click on Login link on the nav bar enter username as adil  Password as Adil@123 |
| Endpoint URL | http://localhost:4200/login |
| Excepted Response | Login successful |
| Actual Response | Login successful |
| Status | PASS |

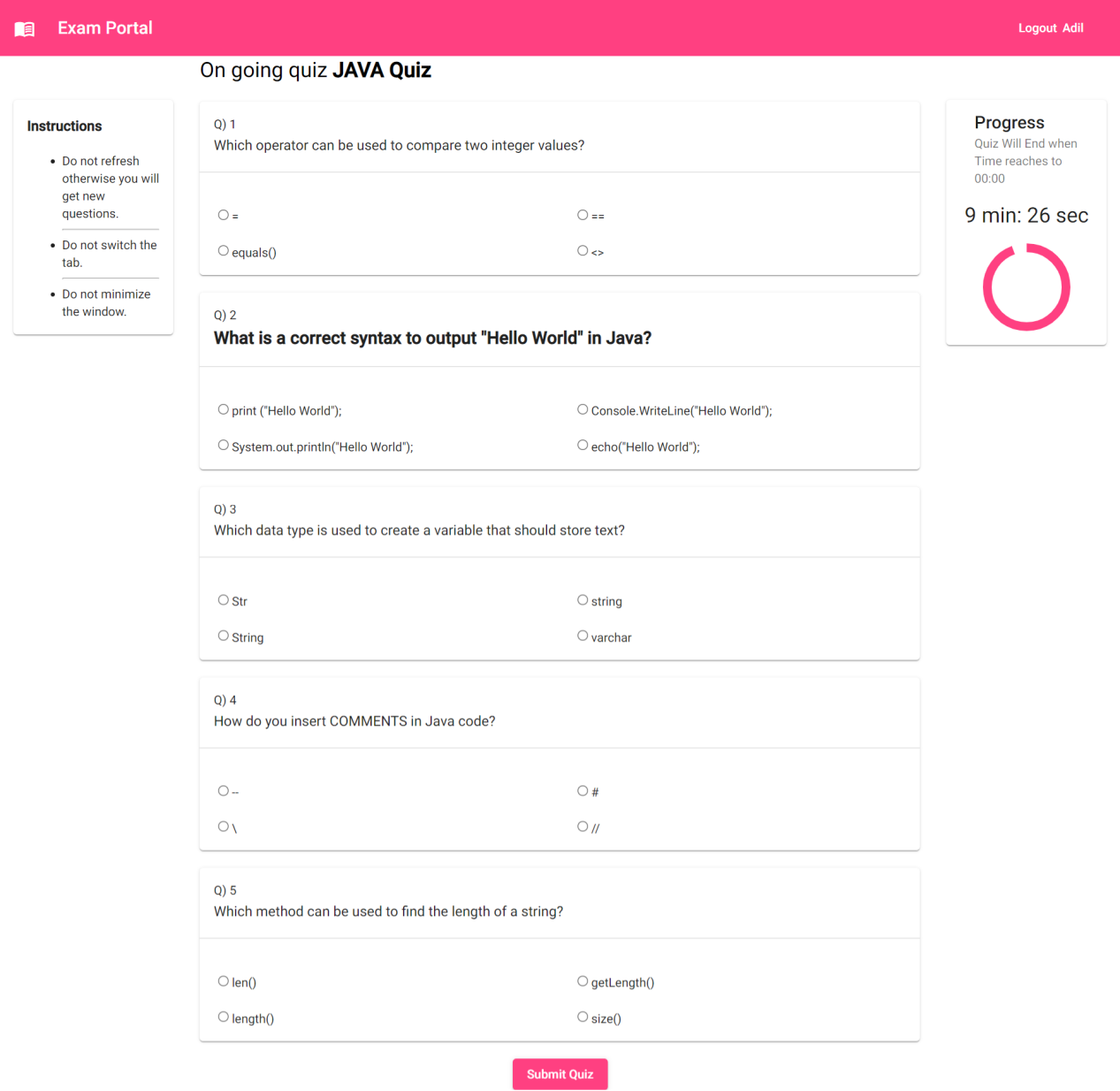




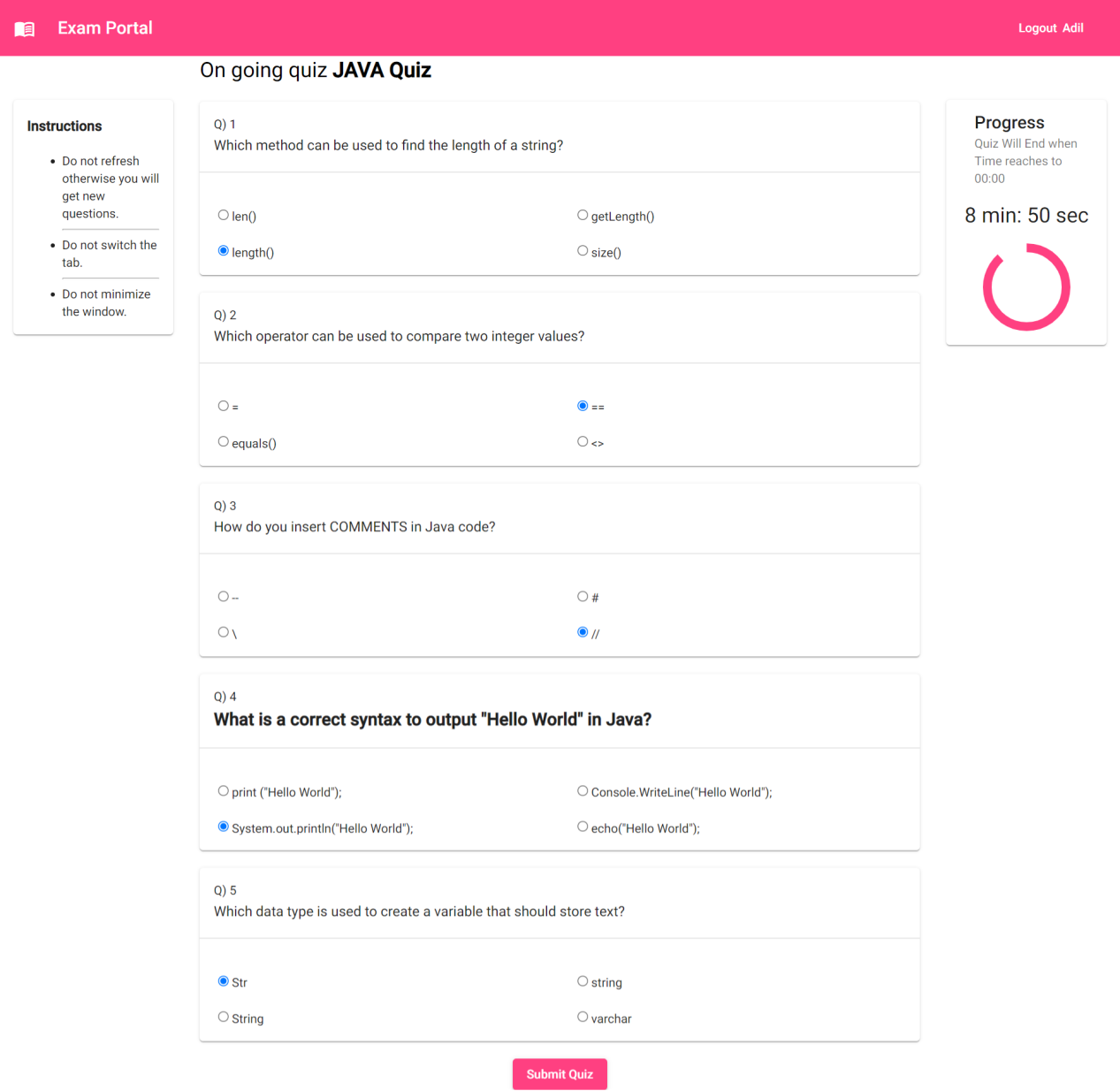
|  |  |
| --- | --- |
| Test case 7 : Attempt quiz | |
| Action | Select Java quiz Click on start button |
| Endpoint URL | http://localhost:4200/user-dashboard/instructions/-4 |
| Excepted Response | Quiz started |
| Actual Response | Quiz started |
| Status | PASS |

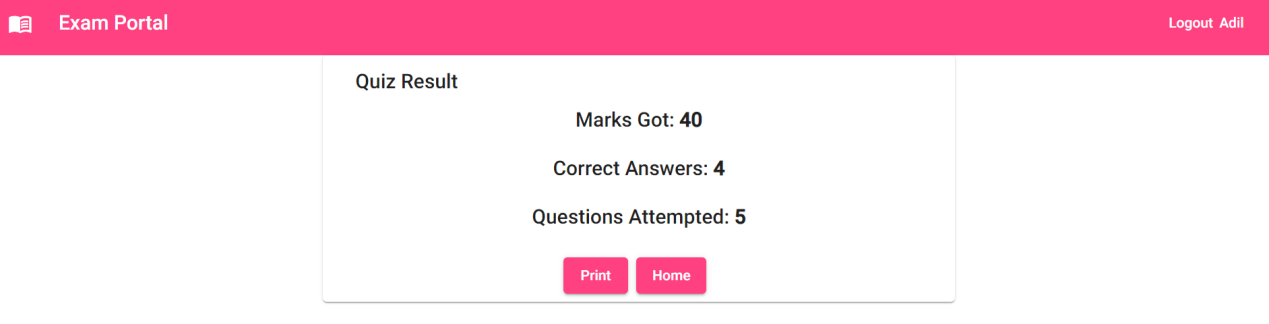




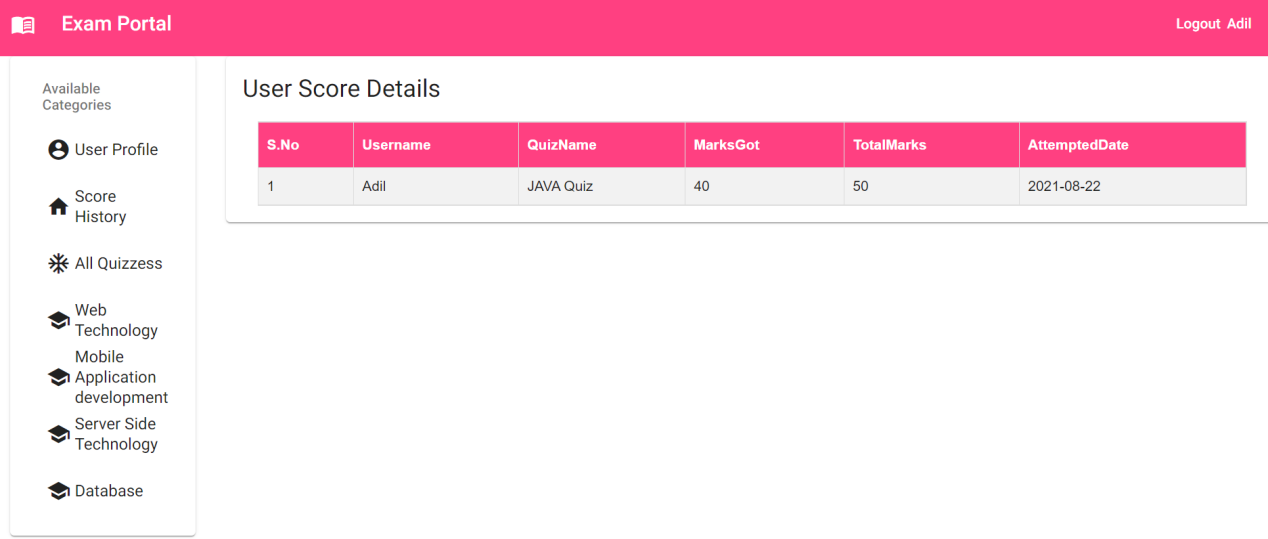


|  |  |
| --- | --- |
| Test case 8 : submit quiz | |
| Action | Enter the answer for the question |
| Endpoint URL | http://localhost:4200/start/-4 |
| Excepted Response | Quiz submitted successfully |
| Actual Response | Quiz submitted successfully |
| Status | PASS |





|  |  |
| --- | --- |
| Test case 9 : View Score | |
| Action | Click on user history |
| Endpoint URL | http://localhost:4200/user-dashboard/score |
| Excepted Response | 200 OK : list of user score |
| Actual Response | 200 OK : list of user score |
| Status | PASS |



# 

# Conclusion

At last I would like to conclude that working on this project was a great learning experience, From the design, development and testing phases. I have learned a lot about the IT industry related technology and best practices followed.I followed SOLID Design principle for the application, I used REST architectural design, learned about different layers of the application like front end layer, business layer and back-end layer followed camel case for coding with Consistent Naming Scheme, Focused on DRY principle. Though I have only tested and deploy on my local machine. In the future scope, I would like to deploy my application to cloud like AWS and observe the performance of the application with real users.  
  
I would like to thank my teacher and friends for supporting and extending help through the project work.

**Future Improvement and Enhancements which can made in this application**

Develop Mobile Apps for IOS And Android  
Session management.

Social Login through Google and Facebook.  
Responsive UI.  
Student monitoring.  
Quiz resume

**References**

[**https://angular.io/tutorial**](https://angular.io/tutorial)

[**https://www.javatpoint.com/spring-boot-tutorial**](https://www.javatpoint.com/spring-boot-tutorial)

[**https://www.baeldung.com/spring-boot**](https://www.baeldung.com/spring-boot)

[**https://jwt.io/introduction**](https://jwt.io/introduction)

[**https://en.wikipedia.org/wiki/Bcrypt**](https://en.wikipedia.org/wiki/Bcrypt)