**Source code for Online Quiz Portal:**

**MyQuizApplication.java:**

**package** com.project;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.annotation.ComponentScan;

@SpringBootApplication

@ComponentScan("com.project")

**public** **class** MyQuizApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(MyQuizApplication.**class**, args);

}

}

**AdminController.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.Admin;

import com.project.service.AdminService;

@RestController

@RequestMapping("/quizapi")

public class AdminController {

@Autowired

private AdminService adminService;

@PostMapping("/login")

public String adminLogin(@RequestBody Admin ad)

{

if(adminService.verifyAdmin(ad.getEmail(), ad.getPassword()))

{

return "Welcome!";

}

else

{

return "Invalid Credentials.";

}

}

@PostMapping("/changePassword/{id}&{newPassword}")

public String changePassword(@PathVariable int id, @PathVariable String newPassword)

{

adminService.changePassword(id, newPassword);

return "Password Changed Successfully";

}

}

**Createquizcontroller.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.Quiz;

import com.project.service.CreateQuizService;

@RestController

@RequestMapping("/quizapi/createquiz")

public class CreateQuizController {

@Autowired

private CreateQuizService service;

///create new record

@PostMapping("/addingnewquiz")

public ResponseEntity<Quiz> addUser(@RequestBody Quiz u){

Quiz user= service.addUser(u);

if(user!=null)

return new ResponseEntity<Quiz>(user,HttpStatus.CREATED);

else

return new ResponseEntity<Quiz>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/gettingallquizzes")

public List<Quiz> getAllUser(){

return service.getAllUser();

}

//get user by id

@GetMapping("/{id}")

public ResponseEntity<Quiz> getUserById(@PathVariable int id){

Quiz user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<Quiz>(user,HttpStatus.FOUND);

else

return new ResponseEntity<Quiz>(user,HttpStatus.NOT\_FOUND);

}

}

**Questioncontroller.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.Questions;

import com.project.service.QuestionService;

@RestController

@RequestMapping("/quizapi/questions")

public class QuestionController {

@Autowired

private QuestionService service;

///create new record

@PostMapping("/addnewquestions")

public ResponseEntity<Questions> addUser(@RequestBody Questions u){

Questions user= service.addUser(u);

if(user!=null)

return new ResponseEntity<Questions>(user,HttpStatus.CREATED);

else

return new ResponseEntity<Questions>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<Questions> getAllUser(){

return service.getAllUser();

}

//attend quiz id

@GetMapping("/{id}")

public ResponseEntity<Questions> getUserById(@PathVariable int id){

Questions user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<Questions>(user,HttpStatus.FOUND);

else

return new ResponseEntity<Questions>(user,HttpStatus.NOT\_FOUND);

}

//update user by id

@PutMapping("/update/{id}")

public ResponseEntity<Object> updateUser(@RequestBody Questions user,@PathVariable int id){

Questions data= service.updateUser(user, id);

if(data!=null)

return new ResponseEntity<Object>(data,HttpStatus.OK);

else

return new ResponseEntity<Object>("User is Not Available",HttpStatus.NOT\_FOUND);

}

///DELETE USER BY ID

@DeleteMapping("/delete/{id}")

public ResponseEntity<Object> deleteUser(@PathVariable int id ){

if(service.deleteUser(id))

return new ResponseEntity<Object>("Question Deleted !!", HttpStatus.OK);

else

return new ResponseEntity<Object>("Question Not Found.",HttpStatus.NOT\_FOUND);

}

}

**Quizcontroller.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.QuizCompetition;

import com.project.service.QuizService;

@RestController

@RequestMapping("/quizapi/quiz")

public class QuizController {

@Autowired

private QuizService service;

///create new record

@PostMapping("/addquiz")

public ResponseEntity<QuizCompetition> addUser(@RequestBody QuizCompetition u){

QuizCompetition user= service.addUser(u);

if(user!=null)

return new ResponseEntity<QuizCompetition>(user,HttpStatus.CREATED);

else

return new ResponseEntity<QuizCompetition>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/getallquizzes")

public List<QuizCompetition> getAllUser(){

return service.getAllUser();

}

}

**Scoreboardcontroller.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.Scoreboard;

import com.project.service.ScoreboardService;

@RestController

@RequestMapping("/quizapi/scoreboard")

public class ScoreboardController {

@Autowired

private ScoreboardService service;

//List of Users

@GetMapping("/")

public List<Scoreboard> getAllUser(){

return service.getAllUser();

}

//get user score by id

@GetMapping("/{id}")

public ResponseEntity<Scoreboard> getUserById(@PathVariable int id){

Scoreboard user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<Scoreboard >(user,HttpStatus.FOUND);

else

return new ResponseEntity<Scoreboard >(user,HttpStatus.NOT\_FOUND);

}

}

**Usercontroller.java:**

package com.project.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.project.entity.Questions;

import com.project.entity.User;

import com.project.service.UserService;

@RestController

@RequestMapping("/quizapi/user")

public class UserController {

@Autowired

private UserService service;

//login

@PostMapping("/user1")

public String adminLogin(@RequestBody User u)

{

if(service.verifyUser(u.getEmail(), u.getPassword()))

{

return "Welcome User!";

}

else

{

return "Invalid Credentials.";

}

}

//add User

@PostMapping("/addUser")

public ResponseEntity<User> addUser(@RequestBody User u){

User user= service.addUser(u);

if(user!=null)

return new ResponseEntity<User>(user,HttpStatus.CREATED);

else

return new ResponseEntity<User>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<User> getAllUser(){

return service.getAllUser();

}

//get user by id

@GetMapping("/{id}")

public ResponseEntity<User> getUserById(@PathVariable int id){

User user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<User>(user,HttpStatus.FOUND);

else

return new ResponseEntity<User>(user,HttpStatus.NOT\_FOUND);

}

}

**Admin.java:**

package com.project.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String email;

private String password;

public Admin(String email, String password) {

super();

this.email = email;

this.password = password;

}

public Admin() {}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Admin [id=" + id + ", email=" + email + ", password=" + password + "]";

}

}

**Questions.java:**

package com.project.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Questions {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String question;

public String getQuestion() {

return question;

}

public void setQuestion(String question) {

this.question = question;

}

private String OptionA;

private String OptionB;

private String OptionC;

private String OptionD;

private String Answer;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getOptionA() {

return OptionA;

}

public void setOptionA(String optionA) {

OptionA = optionA;

}

public String getOptionB() {

return OptionB;

}

public void setOptionB(String optionB) {

OptionB = optionB;

}

public String getOptionC() {

return OptionC;

}

public void setOptionC(String optionC) {

OptionC = optionC;

}

public String getOptionD() {

return OptionD;

}

public void setOptionD(String optionD) {

OptionD = optionD;

}

public String getAnswer() {

return Answer;

}

public void setAnswer(String answer) {

Answer = answer;

}

public Questions(int id, String question, String optionA, String optionB, String optionC, String optionD,

String answer) {

this.id = id;

this.question = question;

OptionA = optionA;

OptionB = optionB;

OptionC = optionC;

OptionD = optionD;

Answer = answer;

}

public Questions() {

// TODO Auto-generated constructor stub

}

}

**Quiz.java:**

package com.project.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Quiz {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String quizCode;

private String question1;

private String question2;

private String question3;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getQuizCode() {

return quizCode;

}

public void setQuizCode(String quizCode) {

this.quizCode = quizCode;

}

public String getQuestion1() {

return question1;

}

public void setQuestion1(String question1) {

this.question1 = question1;

}

public String getQuestion2() {

return question2;

}

public void setQuestion2(String question2) {

this.question2 = question2;

}

public String getQuestion3() {

return question3;

}

public void setQuestion3(String question3) {

this.question3 = question3;

}

public Quiz(int id, String quizCode, String question1, String question2, String question3) {

super();

this.id = id;

this.quizCode = quizCode;

this.question1 = question1;

this.question2 = question2;

this.question3 = question3;

}

public Quiz() {

// TODO Auto-generated constructor stub

}

}

**Quizcompetition.java:**

package com.project.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class QuizCompetition {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private int quizId;

private int questionId;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getQuizId() {

return quizId;

}

public void setQuizId(int quizId) {

this.quizId = quizId;

}

public int getQuestionId() {

return questionId;

}

public void setQuestionId(int questionId) {

this.questionId = questionId;

}

public QuizCompetition(int id, int quizId, int questionId) {

super();

this.id = id;

this.quizId = quizId;

this.questionId = questionId;

}

public QuizCompetition() {

// TODO Auto-generated constructor stub

}

}

**Scoreboard.java:**

package com.project.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Scoreboard {

@Id

@Column(name="position")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int position;

private String user;

private String score;

public int getPosition() {

return position;

}

public void setPosition(int position) {

this.position = position;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

public String getScore() {

return score;

}

public void setScore(String score) {

this.score = score;

}

public Scoreboard(int position, String user, String score) {

super();

this.position = position;

this.user = user;

this.score = score;

}

public Scoreboard() {

}

}

**User.java:**

package com.project.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String email;

private String password;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public User(int id, String email, String password) {

super();

this.id = id;

this.email = email;

this.password = password;

}

public User() {

}

}

**AdminRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.Admin;

@Repository

public interface AdminRepository extends JpaRepository<Admin, Integer>{

}

**CreatequizRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.Quiz;

@Repository

public interface CreateQuizRepo extends JpaRepository<Quiz, Integer> {

}

**QuestionRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.Questions;

@Repository

public interface QuestionRepo extends JpaRepository<Questions, Integer> {

}

**QuizRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.QuizCompetition;

@Repository

public interface QuizRepo extends JpaRepository<QuizCompetition, Integer> {

}

**ScoreboardRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.Scoreboard;

@Repository

public interface ScoreboardRepo extends JpaRepository<Scoreboard, Integer> {

}

**UserRepository.java:**

package com.project.repo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.project.entity.User;

@Repository

public interface UserRepo extends JpaRepository<User, Integer> {

}

**AdminService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.Admin;

import com.project.repo.AdminRepository;

@Service

public class AdminService {

@Autowired

private AdminRepository adminRepository;

public Boolean verifyAdmin(String email, String password)

{

List<Admin> admins = adminRepository.findAll();

for(Admin admin : admins)

{

if(admin.getEmail().equals(email) && admin.getPassword().equals(password))

{

return true;

}

}

return false;

}

public void changePassword(int id, String newPassword)

{

Admin admin = adminRepository.findById(id).get();

admin.setPassword(newPassword);

adminRepository.save(admin);

System.out.println("Password Changed Successfully");

}

}

**CreateQuizService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.Quiz;

import com.project.repo.CreateQuizRepo;

@Service

public class CreateQuizService {

@Autowired

private CreateQuizRepo repo;

//add method or CREATE RECORD

public Quiz addUser(Quiz u) {

return repo.save(u);

}

//List user Method

public List<Quiz> getAllUser(){

return repo.findAll();

}

//get user by id

public Quiz getUserById(int id) {

if(repo.findById(id).isPresent())

return repo.findById(id).get();

else

return null;

}

}

**QuestionService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.Questions;

import com.project.repo.QuestionRepo;

@Service

public class QuestionService {

@Autowired

private QuestionRepo repo;

//add method or CREATE RECORD

public Questions addUser(Questions u) {

return repo.save(u);

}

//List user Method

public List<Questions> getAllUser(){

return repo.findAll();

}

//get user by id

public Questions getUserById(int id) {

if(repo.findById(id).isPresent())

return repo.findById(id).get();

else

return null;

}

//update user by id

public Questions updateUser(Questions user, int id) {

if(repo.findById(id).isPresent())

{

Questions old= repo.findById(id).get();

old.setOptionA(user.getOptionA());

old.setOptionB(user.getOptionB());

old.setOptionC(user.getOptionC());

old.setOptionD(user.getOptionD());

old.setAnswer(user.getAnswer());

return repo.save(old);

}

else

{

return null;

}

}

///DELETE USER BY ID

public boolean deleteUser(int id) {

if(repo.findById(id).isPresent())

{

repo.deleteById(id);

return true;

}

else

return false;

}

}

**QuizService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.QuizCompetition;

import com.project.repo.QuizRepo;

@Service

public class QuizService {

@Autowired

private QuizRepo repo;

//add method or CREATE RECORD

public QuizCompetition addUser(QuizCompetition u) {

return repo.save(u);

}

//List user Method

public List<QuizCompetition> getAllUser(){

return repo.findAll();

}

}

**ScoreboardService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.Scoreboard;

import com.project.repo.ScoreboardRepo;

@Service

public class ScoreboardService {

@Autowired

private ScoreboardRepo repo;

//List user Method

public List<Scoreboard> getAllUser(){

return repo.findAll();

}

//get user by id

public Scoreboard getUserById(int id) {

if(repo.findById(id).isPresent())

return repo.findById(id).get();

else

return null;

}

}

**UserService.java:**

package com.project.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.project.entity.Questions;

import com.project.entity.User;

import com.project.repo.UserRepo;

@Service

public class UserService {

@Autowired

private UserRepo userRepository;

public Boolean verifyUser(String email, String password)

{

List<User> users = userRepository.findAll();

for(User user : users)

{

if(user.getEmail().equals(email) && user.getPassword().equals(password))

{

return true;

}

}

return false;

}

//add User

public User addUser(User u) {

return userRepository.save(u);

}

//List user Method

public List<User> getAllUser(){

return userRepository.findAll();

}

//get user by id

public User getUserById(int id) {

if(userRepository.findById(id).isPresent())

return userRepository.findById(id).get();

else

return null;

}

}

Pom.xml:

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.project</groupId>

<artifactId>MyQuiz</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>MyQuiz</name>

<description>Ecommerce - Sporty Shoes</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Application.properties:**

spring.datasource.username=root

spring.datasource.password=Mohan@4635

spring.datasource.url=jdbc:mysql://localhost:3306/onlinequiz

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.show-sql=true

server.port=8088