**Online Quiz Portal**

**Source Code**

**Admin.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Admin {

@Id

private String username;

private String password;

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Admin [username=" + username + ", password=" + password + "]";

}

public Admin(String username, String password) {

super();

this.username = username;

this.password = password;

}

public Admin() {

super();

// TODO Auto-generated constructor stub

}

}

**AdminController.java**

package com.example.demo;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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.ResponseBody;

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

@RestController

public class AdminController {

@Autowired

AdminRepo erepo;

@Autowired

quizRepo qrepo;

@Autowired

QuestionRepo Quesrepo;

@Autowired

ScoreRepo srepo;

@Autowired

StandingRepo sarepo;

@Autowired

UserRepo urepo;

@ResponseBody

@RequestMapping("/admin/validate")

public String validate(@RequestBody Admin a)

{

String username=a.getUsername();

String password=a.getPassword();

String val="";

if((erepo.findByName(username)!=null)&&(erepo.findByPassword(password)!=null))

{

if(erepo.findByName(username).equals(erepo.findByPassword(password)))

{

System.out.println(erepo.findByName(username));

System.out.println(erepo.findByPassword(password));

val= "Successfully Logged In \n /changepassword \n /insertquestion"

+ "\n /allquizes \n /getbyquizid/{id} \n"

+ " /getbyquestionid/{id}";

}

}

else

{

val= "Invalid Username or password Please Try again!";

}

return val;

}

private Logger log=LoggerFactory.getLogger(this.getClass());

@PutMapping("/changepassword")

public String changepassword(@RequestBody Admin a)

{

erepo.save(a);

return "Updated Successfully \n To see the quizes TYPE /allquizes \n To insert the questions TYPE /insertquestion "

+ "\n To get all Quizes TYPE /allquizes \n To get quiz by quiz id TYPE /getbyquizid/{id} \n "

+ "To get question by question id TYPE /getbyquestionid/{id} "

+ "To see the scores of all the users TYPE /score\n"

+ "To see the position of all the users TYPE /position \n"

+ "To see the all the user details who attended the quiz TYPE /users";

}

@RequestMapping("/allquizes")

public List<Quiz> show()

{

return qrepo.findAll();

}

@PostMapping("/insertquestion")

public Question insert(@RequestBody Question q)

{

log.info("{}",q);

return Quesrepo.save(q);

}

@PostMapping("/insertquiz")

public Quiz insert(@RequestBody Quiz q)

{

log.info("{}",q);

return qrepo.save(q);

}

@GetMapping("/getbyquizid/{id}")

public Quiz quiz(@PathVariable("id") int id)

{

Quiz q= qrepo.findById(id).orElse(null);

return q;

}

@GetMapping("/getbyquestionid/{id}")

public Question question(@PathVariable("id") int id)

{

Question q= Quesrepo.findById(id).orElse(null);

return q;

}

@RequestMapping("/score")

public List<Standings> find()

{

List<Standings> s1=sarepo.findAll();

return s1;

}

@RequestMapping("/position")

public List<Standings> position()

{

return sarepo.findByPos();

}

@RequestMapping("/users")

public List<User> user()

{

return urepo.findAll();

}

}

**AdminRepo.java**

package com.example.demo;

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

import org.springframework.data.repository.CrudRepository;

public interface AdminRepo extends CrudRepository<Admin,String> {

@Query("Select a from Admin a where a.username=?1")

public Admin findByName(String username);

@Query("Select a from Admin a where a.password=?1")

public Admin findByPassword(String password);

}

**Question.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

@Entity

public class Question {

@Id

private int qid;

private String qname;

private String qoption1;

private String qoption2;

private String qoption3;

private String qoption4;

private String answer;

@ManyToOne()

private Quiz quizid;

public int getQid() {return qid;}

public void setQid(int qid) {this.qid = qid;}

public String getQname() {return qname;}

public void setQname(String qname) {this.qname = qname;}

public String getQoption1() {return qoption1;}

public void setQoption1(String qoption1) {this.qoption1 = qoption1;}

public String getQoption2() {return qoption2;}

public void setQoption2(String qoption2) {this.qoption2 = qoption2;}

public String getQoption3() {return qoption3;}

public void setQoption3(String qoption3) {this.qoption3 = qoption3;}

public String getQoption4() {return qoption4;}

public void setQoption4(String qoption4) {this.qoption4 = qoption4;}

public String getAnswer() {return answer;}

public void setAnswer(String answer) {this.answer = answer;}

//hide this two for display

public Quiz getQuizid() {return quizid;}

public void setQuizid(Quiz quizid) {this.quizid = quizid;}

@Override

public String toString() {return "Question [qid=" + qid + ", qname=" + qname+"]";}

public Question(int qid, String qname, String qoption1, String qoption2, String qoption3, String qoption4,

String answer, Quiz quizid) {

super();

this.qid = qid;

this.qname = qname;

this.qoption1 = qoption1;

this.qoption2 = qoption2;

this.qoption3 = qoption3;

this.qoption4 = qoption4;

this.answer = answer;

this.quizid = quizid;

}

public Question() {super();}

}

**QuestionRepo.java**

package com.example.demo;

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

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

public interface QuestionRepo extends JpaRepository<Question,Integer> {

@Query("Select a from Question a where a.answer=?1")

public Question findByAnswer(String answer);

}

**quizRepo.java**

package com.example.demo;

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

public interface quizRepo extends JpaRepository<Quiz,Integer>{

}

**Score.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Score {

@Id

@GeneratedValue

private int scoreid;

private int userid;

private String username;

private int qid;

private int correct;

private int wrong;

public Score() {

super();

// TODO Auto-generated constructor stub

}

public Score(int scoreid, int userid, String username, int qid, int correct, int wrong) {

super();

this.scoreid = scoreid;

this.userid = userid;

this.username = username;

this.qid = qid;

this.correct = correct;

this.wrong = wrong;

}

@Override

public String toString() {

return "Score [scoreid=" + scoreid + ", userid=" + userid + ", username=" + username + ", qid=" + qid

+ ", correct=" + correct + ", wrong=" + wrong + "]";

}

public int getScoreid() {

return scoreid;

}

public void setScoreid(int scoreid) {

this.scoreid = scoreid;

}

public int getUserid() {

return userid;

}

public void setUserid(int userid) {

this.userid = userid;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public int getQid() {

return qid;

}

public void setQid(int qid) {

this.qid = qid;

}

public int getCorrect() {

return correct;

}

public void setCorrect(int correct) {

this.correct = correct;

}

public int getWrong() {

return wrong;

}

public void setWrong(int wrong) {

this.wrong = wrong;

}

}

**Quiz.java**

package com.example.demo;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.OneToMany;

@Entity

public class Quiz {

@Id

private int quizid;

private String quizname;

@OneToMany(mappedBy="quizid")

private List<Question> Question=new ArrayList<>();

public Quiz() {

super();

}

public int getQuizid() {

return quizid;

}

public void setQuizid(int quizid) {

this.quizid = quizid;

}

public String getQuizname() {

return quizname;

}

public void setQuizname(String quizname) {

this.quizname = quizname;

}

// public List<Question> getQuestion() { //hide this for insert

// return Question;

// }

public void addQuestion(Question question) {

Question.add(question);

}

public void removeQuestion(Question question) {

Question.remove(question);

}

@Override

public String toString() {

return "Quiz [quizid=" + quizid + ", quizname=" + quizname + ", Question=" + Question + "]";

}

public Quiz(int quizid, String quizname, List<com.example.demo.Question> question) {

super();

this.quizid = quizid;

this.quizname = quizname;

Question = question;

}

}

**ScoreRepo.java**

package com.example.demo;

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

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

public interface ScoreRepo extends JpaRepository<Score,Integer>{

@Query("Select count(\*) from Score a where a.correct=1 and a.userid=?1")

public int findByCorrect(int id);

}

**Standings.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Standings {

@Id

private int userid;

private int score;

public int getUserid() {

return userid;

}

public void setUserid(int userid) {

this.userid = userid;

}

public int getScore() {

return score;

}

public void setScore(int score) {

this.score = score;

}

@Override

public String toString() {

return "Standings [userid=" + userid + ", score=" + score + "]";

}

public Standings() {

super();

// TODO Auto-generated constructor stub

}

public Standings(int userid, int score) {

super();

this.userid = userid;

this.score = score;

}

}

**StandingRepo.java**

package com.example.demo;

import java.util.List;

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

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

public interface StandingRepo extends JpaRepository<Standings,Integer>{

@Query("select s from Standings s order by s.score desc")

public List<Standings> findByPos();

}

**User.java**

package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class User {

@Id

private int userid;

private String username;

private String password;

public int getUserid() {

return userid;

}

public void setUserid(int userid) {

this.userid = userid;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public User(int userid, String username, String password) {

super();

this.userid = userid;

this.username = username;

this.password = password;

}

public User() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "User [userid=" + userid + ", username=" + username + ", password=" + password + "]";

}

}

**UserRepo.java**

package com.example.demo;

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

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

public interface UserRepo extends JpaRepository<User,Integer> {

@Query("Select a from User a where a.username=?1")

public User findByName(String username);

@Query("Select a from User a where a.password=?1")

public User findByPassword(String password);

}

**UserController.java**

package com.example.demo;

import java.util.List;

import java.util.Optional;

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

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

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.ResponseBody;

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

@RestController

public class UserController {

@Autowired

UserRepo urepo;

@Autowired

quizRepo qrepo;

@Autowired

QuestionRepo quesrepo;

@Autowired

ScoreRepo srepo;

@Autowired

StandingRepo sarepo;

@ResponseBody

@RequestMapping("/login")

public String validate(@RequestBody User u)

{

int userid=u.getUserid();

String username=u.getUsername();

String password=u.getPassword();

String val="";

if((urepo.findById(userid)!=null)&&(urepo.findByName(username)!=null)&&(urepo.findByPassword(password)!=null))

{

if(urepo.findByName(username).equals(urepo.findByPassword(password)))

{

val= "Warning already attempted !!!

}

}

else

{

urepo.save(u);

val= "Welcome to quiz portal \n To see all the quizes go to /allquizes\n"

+ "To see all questions in the particular quiz id TYPE /quiz/{id}\n"

+ "To answer the questions TYPE /quiz/answer\n"

+ "To see the score TYPE /score/{id}\n"

+ "To see the Standing/position TYPE /position\n";

}

return val;

}

@RequestMapping("/allquizes")

public List<Quiz> allquizes()

{

return qrepo.findAll();

}

@RequestMapping("/quiz/{id}")

public Optional<Quiz> allquestionbyid(@PathVariable int id)

{

return qrepo.findById(id);

}

@RequestMapping("/quiz/answer")

public String answer(@RequestParam(name="userid") int userid,

@RequestParam(name="username") String username,

@RequestParam(name="answer") String ans,

@RequestParam(name="qid") int questionid)

{

String val;

Question q=quesrepo.findById(questionid).orElse(null);

System.out.println(q);

Question q1=quesrepo.findByAnswer(ans);

System.out.println(q1);

Score s=new Score();

if(q.equals(q1))

{

val="correct answer";

s.setCorrect(1);

s.setWrong(0);

s.setUserid(userid);

s.setUsername(username);

s.setQid(questionid);

srepo.save(s);

}

else

{

s.setCorrect(0);

s.setWrong(1);

s.setUserid(userid);

s.setUsername(username);

s.setQid(questionid);

srepo.save(s);

val="wrong answer";

}

return val;

}

@RequestMapping("score/{id}")

public Standings find(@PathVariable("id") int uid)

{

Standings s=new Standings();

int score= srepo.findByCorrect(uid);

s.setUserid(uid);

s.setScore(score);

sarepo.save(s);

System.out.println(s);

Standings s1=sarepo.findById(uid).orElse(null);

return s1;

}

@RequestMapping("/position")

public List<Standings> position()

{

return sarepo.findByPos();

}

}

**application.properties-user**

server.port=8083

#jpa-hibernate

spring.jpa.hibernate.ddl-auto=update

spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.show-sql=true

#datasource

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

spring.datasource.username=root

spring.datasource.password=123456

**application.properties-admin**

server.port=8088

#jpa-hibernate

spring.jpa.hibernate.ddl-auto=update

spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect

spring.jpa.show-sql=true

#datasource

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

spring.datasource.username=root

spring.datasource.password=123456