Online Quiz Portal Using REST APIs :- code

Admin class:

```
package com.bean;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import org.springframework.stereotype.Component; @Component
@Entity @Table(name="admin") public class Admin {
	@Id
	private int id;
	private String username; private String password;
	@Override
	public String toString() {
		return "Admin [id=" + id + ", username=" + username + ",
password=" + password + "]";
	}
	public int getId() {
		return id;
		}
	public void setId(int id) {
		this.id = id;
	}
	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;
	}
}
```

Question class:

```
package com.bean;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;

import javax.persistence.Id;
import javax.persistence.Table;
import org.springframework.stereotype.Component;

@Component
@Entity
@Table(name = "question")
public class Question {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int qid;
    private String quest;
    private String opt1;
    private String opt2;
    private String opt3;
    private String opt4;
    private int ans;

    public int getQid() {
        return qid;
    }

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

    public String getQuest() {
        return quest;
    }

    public void setQuest(String quest) {
        this.quest = quest;
    }

    public String getOpt1() {
```

```
    return opt1;
}

public void setOpt1(String opt1) {
    this.opt1 = opt1;
}

public String getOpt2() {
    return opt2;
}

public void setOpt2(String opt2) {
    this.opt2 = opt2;

}

public String getOpt3() {
    return opt3;
}

public void setOpt3(String opt3) {
    this.opt3 = opt3;
}

public String getOpt4() {
    return opt4;
}

public void setOpt4(String opt4) {
    this.opt4 = opt4;
}

public int getAns() {
    return ans;
}

public void setAns(int ans) {
    this.ans = ans;
}

@Override
public String toString() {
    return "Question [qid=" + qid + ", quest=" + quest + ",
  opt1=" + opt1 + ", opt2=" + opt2 + ", opt3=" + opt3
            + ", opt4=" + opt4 + ", ans=" + ans + "]";
```

```
    }
}
```

Quiz class:

```
package com.bean;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import org.springframework.stereotype.Component;

@Component
@Entity
@Table(name = "quiz")
public class Quiz {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int quid;
    private String title;
    private int quizno;
    private String subject;
    @ManyToOne
    @JoinColumn(referencedColumnName = "qid")
    private Question qid;

    public int getQuizno() {
        return quizno;
    }

    public void setQuizno(int quizno) {
        this.quizno = quizno;
    }

    public int getQuid() {
        return quid;
    }
```

```
    public void setQuid(int quid) {
        this.quid = quid;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getSubject() {
        return subject;
    }

    public void setSubject(String subject) {
        this.subject = subject;
    }

    public Question getQid() {
        return qid;
    }

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

    @Override
    public String toString() {
        return "Quiz [quid=" + quid + ", title=" + title + ", quizno=" + quizno + ",
subject=" + subject + ", qid="
                + qid + "]";
    }
}
```

Result Class:

```
package com.bean;

public class Result implements Comparable<Result> {
    private String email;
    private Integer marks;

    public Result() {
    }

    public Result(String email2, int mark) {

    }

    public String getEmail() {
        return email;
    }

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

    public Integer getMarks() {
        return marks;
    }

    public void setMarks(Integer marks) {
        this.marks = marks;
    }

    @Override
    public String toString() {
        return "Result [email=" + email + ", marks=" + marks + "]";
    }

    @Override
    public int compareTo(Result r) {
        // TODO Auto-generated method stub
        int comparemarks = r.getMarks();
        return comparemarks - this.marks;
    }
}
```

Statistics class:

```
package com.bean;

import java.util.List;
import org.springframework.stereotype.Component;

@Component
public class Statistics {
    private int users;
    private List<Object> quiz;
    private int questions;

    @Override
    public String toString() {
        return "Statistics [users=" + users + ", quiz=" + quiz + ", questions=" +
questions + "]";
    }

    public int getUsers() {
        return users;
    }

    public void setUsers(int users) {
        this.users = users;
    }

    public List<Object> getQuiz() {
        return quiz;
    }

    public void setQuiz(List<Object> quiz) {
        this.quiz = quiz;
    }

    public int getQuestions() {
        return questions;
    }

    public void setQuestions(int questions) {
        this.questions = questions;
    }
}
```

Test class

```
package com.bean;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import org.springframework.stereotype.Component;

@Component
@Entity
@Table(name = "test")
public class Test {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int tid;
    @ManyToOne
    @JoinColumn(referencedColumnName = "uid")
    private User userid;
    @ManyToOne
    @JoinColumn(referencedColumnName = "quid")
    private Quiz quizid;
    @ManyToOne
    @JoinColumn(referencedColumnName = "qid")
    private Question questionid;
    private int testans;

    public int getTid() {
        return tid;
    }

    public void setTid(int tid) {
        this.tid = tid;
    }

    public User getUserid() {
        return userid;
    }

    public void setUserid(User userid) {
```

```
        this.userid = userid;
    }

    public Quiz getQuizid() {
        return quizid;
    }

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

    public Question getQuestionid() {
        return questionid;
    }

    public void setQuestionid(Question questionid) {
        this.questionid = questionid;
    }

    public int getTestans() {
        return testans;
    }

    public void setTestans(int testans) {
        this.testans = testans;
    }

@Override
public String toString() {
return "Test [tid=" + tid + ", userid=" + userid + ", quid=" + quizid + ",
questionid=" + questionid + ", testans=" + testans + "]";
}
```

User class

```
package com.bean;

import java.io.Externalizable;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectOutput;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;
import org.springframework.stereotype.Component;

@Component
@Entity
@Table(name = "user")
public class User implements Externalizable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int uid;
    private String emailid;
    private String password;
    private long phno;

    public int getUid() {
        return uid;
    }

    public void setUid(int uid) {
        this.uid = uid;
    }

    public String getEmailid() {
        return emailid;
    }

    public void setEmailid(String emailid) {
        this.emailid = emailid;
    }

    public String getPassword() {
        return password;
    }

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

    public long getPhno() {
        return phno;
    }
```

```
    public void setPhno(long phno) {
        this.phno = phno;
    }

    @Override
    public void writeExternal(ObjectOutput out) throws IOException {
        // TODO Auto-generated method stub
    }

    @Override
    public void readExternal(ObjectInput in) throws IOException,
ClassNotFoundException {
    }
}
```

Maincontroller:

```
package com.controller;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.bean.Admin;
import com.bean.Question;
import com.bean.Quiz;
import com.bean.Result;
import com.bean.Statistics;
import com.bean.Test;
import com.bean.User;
import com.service.AdminSer;
import com.service.UserSer;

@RestController
@RequestMapping("mcq")
public class MainController {
    @Autowired
    UserSer us;
    @Autowired
```

```
    AdminSer as;

    // http://localhost:8080/mcq/userLogin
    @PostMapping(value = "userLogin", consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String userLogin(@RequestBody User u) {
        return us.userLogin(u.getEmailid(), u.getPassword());
    }

    // http://localhost:8080/mcq/userRegister
@PostMapping(value="userRegister", consumes =
MediaType.APPLICATION_JSON_VALUE) public String
userRegiter(@RequestBody User u)
{
      return us.userRegister(u);
}

    // http://localhost:8080/mcq/adminLogin
@PostMapping(value="adminLogin", consumes =
MediaType.APPLICATION_JSON_VALUE) public String
adminLogin(@RequestBody Admin u)
{
      return as.adminLogin(u);
}

    // http://localhost:8080/mcq/adminupdate
@PostMapping(value="adminupdate", consumes =
MediaType.APPLICATION_JSON_VALUE) public String
adminUpdate(@RequestBody Admin u)
{
      return as.adminupdate(u);
}

    // http://localhost:8080/mcq/addQuestions
@PostMapping(value="addQuestions", consumes =
MediaType.APPLICATION_JSON_VALUE) public String
addQuestion(@RequestBody Question q)
{
      return as.addQuestion(q);
}

    // http://localhost:8080/mcq/addQuiz
@PostMapping(value="addQuiz", consumes =
MediaType.APPLICATION_JSON_VALUE) public String
addQuiz(@RequestBody Quiz q)
```

```
{
	return as.addQuiz(q);
}

   // http://localhost:8080/mcq/viewAllQuiz
@GetMapping(value="viewAllQuiz", produces=
MediaType.APPLICATION_JSON_VALUE) public List<Quiz> viewAllQuiz()
{
	return as.viewAllQuiz();
}

   // http://localhost:8080/mcq/quizinfo
   @GetMapping(value = "quizinfo", produces =
MediaType.APPLICATION_JSON_VALUE)
   public Statistics quizinfo() {
	 return as.quizInfo();
   }

@GetMapping(value="viewQuiz", produces=
MediaType.APPLICATION_JSON_VALUE) public List<Object> viewQuiz()
{
	return us.viewAllQuiz();
}

   // http://localhost:8080/mcq/takeTest
@PostMapping(value="takeTest", consumes =
MediaType.APPLICATION_JSON_VALUE)
	public String takeTest(@RequestBody Test t) {
	return us.takeTest(t);
}

   // http://localhost:8080/mcq/getAllTest
@GetMapping(value="getAllTest", produces=
MediaType.APPLICATION_JSON_VALUE)
	public List<Test> getAllTest() {
	return us.getTestList();
}

   @GetMapping(value = "getresult", produces =
MediaType.APPLICATION_JSON_VALUE)public List<Result> getresult() {
	 return us.result();
   }


   public List<Result> getAdminResult() {
```

```
        return us.result();
    }
}
```

OnlineQuizPortalRestApplication:

```
package com.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication; import
org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
@SpringBootApplication(scanBasePackages = "com")
@EntityScan("com.bean")
@EnableJpaRepositories("com.repository")
public class OnlineQuizPortalRestApplication {
public static void main(String[] args{
      SpringApplication.run(OnlineQuizPortalRestApplication.class, args);
      }
}
```

AdminRepo:

```
package com.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.bean.Admin;
public interface AdminRepo extends JpaRepository<Admin, Integer> {
}
```

Questionrepo:

```
package com.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import com.bean.Question;
@Repository
public interface Questionrepo extends JpaRepository<Question, Integer> {
 }
```

Quizrepo:

```
package com.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;
import com.bean.Quiz;
@Repository
public interface Quizrepo extends JpaRepository<Quiz, Integer>{
	@Query("select q.title,count(distinct q.quizno) from Quiz as q group by
q.quizno")
	public List<Object> listOfQuiz();
}
```

Testrepo:

```
package com.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import com.bean.Test;
public interface Testrepo extends JpaRepository<Test, Integer>{
	@Query("Select t from Test as t group by t.userid")
	List<Test> getIndividual();
}
```

Userrepo:

```
package com.repository;
import org.springframework.data.jpa.repository.JpaRepository; import
org.springframework.stereotype.Repository;
import com.bean.User;
@Repository
public interface Userrepo extends JpaRepository<User, Integer>{
	public User findByEmailid(String emailid);
 }
```

*** Service ***

Admin:

```
package com.service;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.bean.Admin;
import com.bean.Question;
import com.bean.Quiz;
import com.bean.Result;
import com.bean.Statistics;
import com.bean.User;
import com.repository.AdminRepo;
import com.repository.Questionrepo;
import com.repository.Quizrepo;
import com.repository.Userrepo;

@Service
public class AdminSer {
    @Autowired
    Questionrepo qr;
    @Autowired
    Quizrepo qur;
    @Autowired
    Userrepo ur;
    @Autowired
    Statistics stat;
    @Autowired
    AdminRepo adr;

public String adminLogin(Admin u) {
Admin ad= adr.findById(1).get();
if(u.getUsername().equals(ad.getUsername())&&u.getPassword().equals(ad.g
etPassword())) {
return "Welcome admin";
} else {
} }return"invalid Credentials";

public String adminupdate(Admin a) {
}

    Admin ad = adr.findById(1)
```

```
        .get();ad.setUsername(a.getUsername());ad.setPassword(a.getUsern
ame());adr.saveAndFlush(ad);return"Updated";

public String addQuestion(Question q) {
if(q!=null) {
qr.save(q);
return "question added";
} else {
} }return"failed to add";

public String addQuiz(Quiz q) {
if(q!=null) {
qur.save(q);
return "quiz added";
} else {
} }return"failed to add";

public List<Quiz> viewAllQuiz() {
return qur.findAll();
}

    public Statistics quizInfo() {
        stat.setUsers(ur.findAll().size());
        stat.setQuestions(qr.findAll().size());
        stat.setQuiz(qur.listOfQuiz());
        return stat;

    }
}
```

User :

```
package com.service;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.bean.Result;
import com.bean.Test;
import com.bean.User;
import com.repository.Quizrepo;
```

```
@Service
public class UserSer {
    List<Result> finalList = new ArrayList<>();
    @Autowired
    Userrepo ur;
    @Autowired
    Quizrepo qr;
    @Autowired
    Testrepo tr;
    @Autowired

    User u;
    @Autowired
    Test t;
    Result r = new Result();

public String userLogin(String email,String password) {
u=ur.findByEmailid(email); if(u!=null)
{
if(u.getEmailid().equals(email)&&u.getPassword().equals(password)) {
return "login sucessfull";
}
else{
    return "invalid credentials";
}

else{
    return "User not found";
}
}

    public String userRegister(User u) {
        if (ur.findByEmailid(u.getEmailid()) == null) {
            ur.save(u);
            return "registered";
        } else {
            return "User already exists";
        }
    }

    public List<Object> viewAllQuiz() {
        return qr.listOfQuiz();
    }

    public String takeTest(Test t) {
```

```
        if (t != null) {
            tr.save(t);
            return "submitted";
        } else {
            return "submission failed";
        }
    }

    public List<Test> getTestList() {
        return tr.findAll();
    }

    public List<Result> result() {
        String email = "";
        int mark = 0;
        List<Test> obj = tr.findAll();
        List<User> u = ur.findAll();
        for (User user : u) {
            mark = 0;
            email = user.getEmailid();
            System.out.println(user.getEmailid());
            for (Test ob : obj) {
                if (user.getUid() == ob.getUserid().getUid()) {
                    if (ob.getTestans() == ob.getQuestionid().getAns()) {
                        mark++;
                    }
                    System.out.println("inside" + mark);
                }
            }
            System.out.println("outside" + mark);
            finalList.add(new Result(email, mark));
        }
        System.out.println("final :" + mark);
        Collections.sort(finalList);
        return finalList;
    }
}
```

Application.properties:

```
Spring.application.name=Online_Quiz
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/onlineexam1
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql: true
spring.jpa.properties.hibernate.format_sql=true
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type=TRACE
```

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>2.6.11</version>
		<relativePath /> <!-- lookup parent from repository -->
	</parent>
	<groupId>com</groupId>
	<artifactId>OnlineQuizPortal-REST</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>OnlineQuizPortal-REST</name>
	<description>Demo project for Spring Boot with rest</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>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </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>
```