please go through this code if the screenshots are not visible--->(The data getting exceeded than 10MB thats becoming really trouble. i tried to add every sceenshot. please go through them.)

1.Here I used microservices joining.

for 1st application/Project (Quiz)---->

```
package com.project.quiz;

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

@SpringBootApplication
public class QuizApplication {

	public static void main(String[] args) {
		SpringApplication.run(QuizApplication.class, args);
	}

}
```

```
package com.project.quiz.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="admin")
public class Admin {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int adminid;
	private String adminname;
	private String password;

	public int getAdminid() {
```

```
		return adminid;
	}

	public void setAdminid(int adminid) {
		this.adminid = adminid;
	}

	public String getAdminname() {
		return adminname;
	}

	public void setAdminname(String adminname) {
		this.adminname = adminname;
	}

	public String getPassword() {
		return password;
	}

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

	@Override
	public String toString() {
		return "Admin [adminid=" + adminid + ", adminname=" + adminname +
", password=" + password + "]";
	}

}
```

```
package com.project.quiz.entity;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="questions")
```

```
public class Questions {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int quesid;

	private String ques;

	@OneToMany(mappedBy = "ques", cascade = { CascadeType.ALL })
	private Set<Answer> answers = new HashSet<>();

	public int getId() {
		return quesid;
	}

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

	public String getQues() {
		return ques;
	}

	public void setQues(String ques) {
		this.ques = ques;
	}

	@Override
	public String toString() {
		return "Questions [id=" + quesid + ", ques=" + ques + ", answers="
+ answers + "]";
	}

}
```

```
package com.project.quiz.entity;



import javax.persistence.Entity;
import javax.persistence.FetchType;
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 com.fasterxml.jackson.annotation.JsonBackReference;

@Entity
@Table(name="answer")
public class Answer {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int id;
	private String ans;

	@JsonBackReference
	@ManyToOne(fetch = FetchType.EAGER)
	@JoinColumn(name = "ques_id", referencedColumnName = "quesid")
	private Questions ques;

	public Answer() {
		super();

	}

	public int getId() {
		return id;
	}

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

	public String getAns() {
		return ans;
	}

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

	public Questions getQues() {
		return ques;
	}

	public void setQues(Questions ques) {
		this.ques = ques;
	}

	@Override
	public String toString() {
		return "Answer [id=" + id + ", ans=" + ans + ", ques=" + ques +
```

```
"]";
	}

}-------------------------------------------------------------

package com.project.quiz.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="quiz")
public class Quiz {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int quizid;
	private String quizname;

	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;
	}

	@Override
	public String toString() {
		return "Quiz [quizid=" + quizid + ", quizname=" + quizname + "]";
	}

}
_____________________________________________
package com.project.quiz.Repository;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;
```

```
import com.project.quiz.entity.Admin;

public interface AdminRepository extends CrudRepository<Admin, Integer> {

        Optional<Admin> findByAdminnameAndPassword(String name, String password);

}
```

```
package com.project.quiz.Repository;

import org.springframework.data.repository.CrudRepository;

import com.project.quiz.entity.Questions;

public interface QuestionRepository extends CrudRepository<Questions, Integer> {

}
```

```
package com.project.quiz.Repository;

import org.springframework.data.repository.CrudRepository;

import com.project.quiz.entity.Quiz;

public interface QuizRepository extends CrudRepository<Quiz, Integer> {

}
```

```
package com.project.quiz.exceptions;

public class AdminNotFoundException extends RuntimeException {

        public AdminNotFoundException(int id) {
        super("Admin with id " + id + " not found.");
        }

        public AdminNotFoundException(String username, String password) {
                super("Admin with Username: " + username + " password " + password
+ " not found.");
        }

}
```

```
package com.project.quiz.exceptions;
```

```
public class QuestionNotFoundException extends RuntimeException {

	public QuestionNotFoundException(int id) {
		super("Question with id " + id + " not found.");
	}

}
```

```
package com.project.quiz.exceptions;

public class QuizNotFoundException extends RuntimeException {

	public QuizNotFoundException(int id) {
		super("Quiz with id " + id + " not found.");
	}


}
```

```
package com.project.quiz.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
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.quiz.Repository.AdminRepository;
import com.project.quiz.entity.Admin;
import com.project.quiz.exceptions.AdminNotFoundException;

@RestController
@RequestMapping("/admin")
public class AdminController {

	@Autowired
	AdminRepository adminrepo;

	@GetMapping
	public Iterable<Admin> getAdmin() {
		return adminrepo.findAll();
	}
```

```
	@GetMapping("/{id}")
	public Admin getAdmin(@PathVariable("id") Integer id) {
		Optional<Admin> opt = adminrepo.findById(id);
		if (opt.isEmpty()) {
			throw new AdminNotFoundException(id);
		}
		return opt.get();
	}

	@PostMapping
	public Admin create(@RequestBody Admin admin) {
		return adminrepo.save(admin);
	}

	@PutMapping
	public Admin update(@RequestBody Admin admin) {
		return adminrepo.save(admin);
	}

	@DeleteMapping("/{id}")
	public void delete(@PathVariable("id") Integer id) {
		adminrepo.deleteById(id);
	}
}
```

```
package com.project.quiz.controller;

import java.util.Optional;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;

import com.project.quiz.Repository.AdminRepository;
import com.project.quiz.entity.Admin;
import com.project.quiz.exceptions.AdminNotFoundException;

@Controller
@RequestMapping("/verifyadmin")
public class AdminVerifyController {

	@Autowired
	private AdminRepository repo;
```

```
	@RequestMapping("/showLogin")
	public String showLoginPage() {
		return "login";
	}

	@RequestMapping(value = "/loginverify", method = RequestMethod.POST)
	@ResponseBody
	public String ValidateAdmin(HttpServletRequest request) {

		String username = request.getParameter("adminname");
		String password = request.getParameter("password");
		Optional<Admin> optproduct =
repo.findByAdminnameAndPassword(username, password);
		if (optproduct.isEmpty()) {
			throw new AdminNotFoundException(username, password);
		}

		return "Login Successful";
	}

}
```

```
package com.project.quiz.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.project.quiz.exceptions.AdminNotFoundException;

@ControllerAdvice
public class AdminExceptionController {

	@ExceptionHandler(value = AdminNotFoundException.class)
	public ResponseEntity<Object> handleException(AdminNotFoundException ex) {
		return new ResponseEntity<Object>(ex.getMessage(),
HttpStatus.NOT_FOUND);
	}

}
```

```
package com.project.quiz.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
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.quiz.Repository.QuestionRepository;
import com.project.quiz.entity.Questions;
import com.project.quiz.exceptions.QuestionNotFoundException;

@RestController
@RequestMapping("/question")
public class QuestionController {

	@Autowired
	QuestionRepository quesrepo;

	@GetMapping
	public Iterable<Questions> getQuiz() {
		return quesrepo.findAll();
	}

	@GetMapping("/{id}")
	public Questions getQues(@PathVariable("id") Integer id) {
		Optional<Questions> opt = quesrepo.findById(id);
		if (opt.isEmpty()) {
			throw new QuestionNotFoundException(id);
		}
		return opt.get();
	}

	@PostMapping
	public Questions create(@RequestBody Questions ques) {
		return quesrepo.save(ques);
	}

	@PutMapping
	public Questions update(@RequestBody Questions ques) {
		return quesrepo.save(ques);
	}

	@DeleteMapping("/{id}")
	public void delete(@PathVariable("id") Integer id) {
		quesrepo.deleteById(id);
	}

}
```

```
package com.project.quiz.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.project.quiz.exceptions.QuestionNotFoundException;

@ControllerAdvice
public class QuestionExceptionController {

	@ExceptionHandler(value = QuestionNotFoundException.class)
	public ResponseEntity<Object> handleException(QuestionNotFoundException ex)
{
		return new ResponseEntity<Object>(ex.getMessage(),
HttpStatus.NOT_FOUND);
	}

}_____

package com.project.quiz.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
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.quiz.Repository.QuizRepository;
import com.project.quiz.entity.Quiz;
import com.project.quiz.exceptions.QuizNotFoundException;

@RestController
@RequestMapping("/quiz")
public class QuizController {

	@Autowired
	QuizRepository quizrepo;

	@GetMapping
	public Iterable<Quiz> getQuiz() {
		return quizrepo.findAll();
	}
```

```
	@GetMapping("/{id}")
	public Quiz getQuiz(@PathVariable("id") Integer id) {
		Optional<Quiz> opt = quizrepo.findById(id);
		if (opt.isEmpty()) {
			throw new QuizNotFoundException(id);
		}
		return opt.get();
	}

	@PostMapping
	public Quiz create(@RequestBody Quiz quiz) {
		return quizrepo.save(quiz);
	}

	@PutMapping
	public Quiz update(@RequestBody Quiz quiz) {
		return quizrepo.save(quiz);
	}

	@DeleteMapping("/{id}")
	public void delete(@PathVariable("id") Integer id) {
		quizrepo.deleteById(id);
	}
}
```

```
package com.project.quiz.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.project.quiz.exceptions.QuizNotFoundException;

@ControllerAdvice
public class QuizExceptionController {

	@ExceptionHandler(value = QuizNotFoundException.class)
	public ResponseEntity<Object> handleException(QuizNotFoundException ex) {
		return new ResponseEntity<Object>(ex.getMessage(),
HttpStatus.NOT_FOUND);
	}

}
```

```
package com.project.quiz.controller;

import java.util.Optional;
```

```
import org.springframework.beans.factory.annotation.Autowired;
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.quiz.Repository.AnswerRepository;
import com.project.quiz.entity.Answer;

@RestController
@RequestMapping("/answer")
public class AnswerController {

	@Autowired
	AnswerRepository ansrepo;

	@GetMapping
	public Iterable<Answer> getQuiz() {
		return ansrepo.findAll();
	}

	@GetMapping("/{id}")
	public String getQues(@PathVariable("id") Integer id) {
		Optional<Answer> opt = ansrepo.findById(id);
		if (opt.isEmpty()) {
			return "Answer not found";
		}
		return opt.get().toString();
	}

	@PostMapping
	public Answer create(@RequestBody Answer ans) {
		return ansrepo.save(ans);
	}

	@PutMapping
	public Answer update(@RequestBody Answer ans) {
		return ansrepo.save(ans);
	}

	@DeleteMapping("/{id}")
	public void delete(@PathVariable("id") Integer id) {
		ansrepo.deleteById(id);
	}

}
```

```
_____________________________________________________
server.port=8082
spring.datasource.url=jdbc:mysql://localhost:3306/myquizportal
spring.datasource.username=root
spring.datasource.password=Veda@1202189
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.hibernate.ddl-auto=update
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
logging.level.org.hibernate=INFO
logging.level.org.hibernate.SQL=INFO
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
logging.level.org.springframework=INFO
#logging.level.org.apache=ERROR
#logging.level.org.springframework.web=TRACE


_____________________________________________
<!DOCTYPE html>
<html>
<head>
<meta charset="US-ASCII">
<title>Admin Login Page</title>
<h1>Please login to continue</h1>
</head>
<body>

        <form action="loginverify" method="post">

                Username: <input type="text" name="adminname"> <br>
                Password: <input type="password" name="password"> <br> <br>
                <input type="submit" value="Login">
        </form>
</body>
</html>
_____________________________________________
<?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.example</groupId>
        <artifactId>Quiz</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
<name>Quiz</name>
<description>Demo project for Spring Boot</description>
<properties>
        <java.version>17</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.apache.tomcat.embed</groupId>
                <artifactId>tomcat-embed-jasper</artifactId>
                <version>9.0.44</version>
                <scope>provided</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-tomcat</artifactId>
                <version>2.4.4</version>
                <scope>provided</scope>
        </dependency>
        <dependency>
                <groupId>javax.servlet</groupId>
                <artifactId>jstl</artifactId>
                <version>1.2</version>
        </dependency>
        <dependency>
                <groupId>taglibs</groupId>
                <artifactId>standard</artifactId>
                <version>1.1.2</version>
        </dependency>

        <dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <scope>runtime</scope>
        </dependency>
        <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-devtools</artifactId>
                <scope>runtime</scope>
                <optional>true</optional>
        </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>
```

---

2. Second Project **********(QuizUser)---->

```
package com.project.quizuser;

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

@SpringBootApplication
public class QuizUserApplication {

        public static void main(String[] args) {
                SpringApplication.run(QuizUserApplication.class, args);
        }

}
```

```
package com.project.quizuser.entity;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="user")
public class User {

        @Id
```

```
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int id;
	private String name;
	private String email;
	private String password;

	public int getId() {
		return id;
	}

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

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	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 "User [id=" + id + ", name=" + name + ", email=" + email +
", password=" + password + "]";
	}

}
```

```
package com.project.quizuser.entity;

import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class QuizPage {

	@Id
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int pageid;
	private int quizid;
	private int quesid;

	public int getPageid() {
		return pageid;
	}

	public void setPageid(int pageid) {
		this.pageid = pageid;
	}

	public int getQuizid() {
		return quizid;
	}

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

	public int getQuesid() {
		return quesid;
	}

	public void setQuesid(int quesid) {
		this.quesid = quesid;
	}

	@Override
	public String toString() {
		return "QuizPage [pageid=" + pageid + ", quizid=" + quizid + ",
quesid=" + quesid + "]";
	}

}

----_____________________________

package com.project.quizuser.entity;

import javax.persistence.Entity;
```

```
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="ScoreCompares")
public class ScoreCompare {

	@Id
	private String uname;
	private int scorecard;

	public String getUname() {
		return uname;
	}

	public void setUname(String uname) {
		this.uname = uname;
	}

	public int getScorecard() {
		return scorecard;
	}

	public void setScorecard(int scorecard) {
		this.scorecard = scorecard;
	}

	@Override
	public String toString() {
		return "ScoreCompare [uname=" + uname + ", scorecard=" + scorecard
+ "]";
	}

}
```

```
package com.project.quizuser.Repo;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;

import com.project.quizuser.entity.User;

public interface UserRepository extends CrudRepository<User, Integer> {

	Optional<User> findByNameAndPassword(String name, String password);

}
```

```
_____________________________________________
package com.project.quizuser.Repo;

import org.springframework.data.repository.CrudRepository;

import com.project.quizuser.entity.QuizPage;

public interface QuizPageRepository extends CrudRepository<QuizPage, Integer> {

}
_____________________________________________
package com.project.quizuser.Repo;

import org.springframework.data.repository.CrudRepository;

import com.project.quizuser.entity.ScoreCompare;

public interface ScoreCompareRepository extends CrudRepository<ScoreCompare,
String> {

}
_____________________________________________
package com.project.quizuser.exceptions;

public class UserNotFoundException extends RuntimeException {

        public UserNotFoundException(int id) {
                super("User with id " + id + " not found.");
        }

        public UserNotFoundException(String username, String password) {
                super("Admin with Username: " + username + " password " + password
+ " not found.");
        }

}
_____________________________________________________________________________________

package com.project.quizuser.dto;

public class Answer {

        private int id;
        private String ans;
        private Questions ques;

        public int getId() {
                return id;
        }
```

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

	public String getAns() {
		return ans;
	}

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

	public Questions getQues() {
		return ques;
	}

	public void setQues(Questions ques) {
		this.ques = ques;
	}

	@Override
	public String toString() {
		return "Answer [id=" + id + ", ans=" + ans + ", ques=" + ques +
"]";
	}

}
```

```
package com.project.quizuser.dto;

public class Quiz {

	private int quizid;
	private String quizname;

	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;
```

```
	}

	@Override
	public String toString() {
		return "Quiz [quizid=" + quizid + ", quizname=" + quizname + "]";
	}

}
```

```
package com.project.quizuser.dto;

import java.util.HashSet;
import java.util.Set;

public class Questions {

	private int quesid;
	private String ques;

	private Set<Answer> answers = new HashSet<>();



	public Set<Answer> getAnswers() {
		return answers;
	}

	public void setAnswers(Set<Answer> answers) {
		this.answers = answers;
	}

	public int getQuesid() {
		return quesid;
	}

	public void setQuesid(int quesid) {
		this.quesid = quesid;
	}

	public String getQues() {
		return ques;
	}

	public void setQues(String ques) {
		this.ques = ques;
	}

	@Override
	public String toString() {
```

```
		return "Questions [quesid=" + quesid + ", ques=" + ques + ",
answers=" + answers + "]";
	}

}
```

```
package com.project.quizuser.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
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.quizuser.Repo.UserRepository;
import com.project.quizuser.entity.User;
import com.project.quizuser.exceptions.UserNotFoundException;

@RestController
@RequestMapping("/user")
public class UserContoller {

	@Autowired
	UserRepository userrepo;

	@GetMapping
	public Iterable<User> getUser() {
		return userrepo.findAll();
	}

	@GetMapping("/{id}")
	public User getUser(@PathVariable("id") Integer id) {
		Optional<User> opt = userrepo.findById(id);
		if (opt.isEmpty()) {
			throw new UserNotFoundException(id);
		}
		return opt.get();
	}

	@PostMapping
	public User create(@RequestBody User user) {
		return userrepo.save(user);
	}
```

```
	@PutMapping
	public User update(@RequestBody User user) {
		return userrepo.save(user);
	}

	@DeleteMapping("/{id}")
	public void delete(@PathVariable("id") Integer id) {
		userrepo.deleteById(id);
	}
}
```

```
package com.project.quizuser.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.client.RestTemplate;

import com.project.quizuser.Repo.QuizPageRepository;
import com.project.quizuser.Repo.ScoreCompareRepository;
import com.project.quizuser.dto.Answer;
import com.project.quizuser.dto.Questions;
import com.project.quizuser.dto.Quiz;
import com.project.quizuser.entity.QuizPage;

@Controller
@RequestMapping("/quiz")
public class UserQuizController {
	@Autowired
	private QuizPageRepository repo;

	@Autowired
	private ScoreCompareRepository srepo;

	private RestTemplate restTemplate = new RestTemplate();

	@RequestMapping(value = "/showpage", method = RequestMethod.GET)
	public String ShowCreateQuizPage(ModelMap model) {
		model.addAttribute("QuizPages", repo.findAll());
		List<Quiz> quiz = (List<Quiz>)
restTemplate.getForObject("http://localhost:8082/quiz", List.class);
		model.addAttribute("quiz", quiz);
		List<Questions> ques = (List<Questions>)
restTemplate.getForObject("http://localhost:8082/question",
```

```
                                List.class);
                model.addAttribute("ques", ques);
                return "showAddQuizPage";
        }

        @RequestMapping(value = "/addtoquiz", method = RequestMethod.POST)
        public String addToquiz(@ModelAttribute("shoppingCart") QuizPage qp,
ModelMap model) {
                repo.save(qp);
                model.addAttribute("QuizPages", repo.findAll());
                List<Quiz> quiz = (List<Quiz>)
restTemplate.getForObject("http://localhost:8082/quiz", List.class);
                model.addAttribute("quiz", quiz);
                List<Questions> ques = (List<Questions>)
restTemplate.getForObject("http://localhost:8082/question",
                                List.class);
                model.addAttribute("ques", ques);
                return "showAddQuizPage";
        }

        @RequestMapping("/showQuizPage")
        public String showQuizPage(ModelMap model) {
                List<Quiz> quiz = (List<Quiz>)
restTemplate.getForObject("http://localhost:8082/quiz", List.class);
                model.addAttribute("quiz", quiz);
                return "showQuizStartPage";
        }

        @RequestMapping("/{quizname}")
        public String showTestPage(@PathVariable("quizname") String quizname,
ModelMap model) {
                List<Questions> ques = (List<Questions>)
restTemplate.getForObject("http://localhost:8082/question",
                                List.class);
                model.addAttribute("ques", ques);
                List<Quiz> quiz = (List<Quiz>)
restTemplate.getForObject("http://localhost:8082/quiz", List.class);
                model.addAttribute("quiz", quiz);
                List<Answer> answer = (List<Answer>)
restTemplate.getForObject("http://localhost:8082/answer", List.class);
                model.addAttribute("answer", answer);
                return "javaQuiz";
        }
//compare
        @RequestMapping("/cppQuiz")
        public String showTestPage1(@PathVariable("quizname") String quizname,
ModelMap model) {
                List<Questions> ques = (List<Questions>)
restTemplate.getForObject("http://localhost:8082/question",
                                List.class);
```

```
                model.addAttribute("ques", ques);
                List<Quiz> quiz = (List<Quiz>)
restTemplate.getForObject("http://localhost:8082/quiz", List.class);
                model.addAttribute("quiz", quiz);
                List<Answer> answer = (List<Answer>)
restTemplate.getForObject("http://localhost:8082/answer", List.class);
                model.addAttribute("answer", answer);
                return "cppQuiz";
        }

        @RequestMapping("/result")

        public String viewResult1(ModelMap model) {
                model.addAttribute("scores", srepo.findAll());
                return "final";
        }

}
```

```
package com.project.quizuser.controller;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.project.quizuser.exceptions.UserNotFoundException;

@ControllerAdvice
public class UserExceptionController {

        @ExceptionHandler(value = UserNotFoundException.class)
        public ResponseEntity<Object> handleException(UserNotFoundException ex) {
                return new ResponseEntity<Object>(ex.getMessage(),
HttpStatus.NOT_FOUND);
        }

}
```

```
package com.project.quizuser.controller;

import java.util.Optional;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.ResponseBody;

import com.project.quizuser.Repo.UserRepository;
import com.project.quizuser.entity.User;
import com.project.quizuser.exceptions.UserNotFoundException;

@Controller
@RequestMapping("/verifyuser")
public class UserVerifyController {

	@Autowired
	UserRepository repo;

	@RequestMapping("/userLogin")
	public String showLoginPage() {
		return "UserLogin";
	}

	@RequestMapping(value = "/userloginverify", method = RequestMethod.POST)
	@ResponseBody
	public String ValidateAdmin(HttpServletRequest request) {

		String username = request.getParameter("name");
		String password = request.getParameter("password");
		Optional<User> optproduct = repo.findByNameAndPassword(username,
password);
		if (optproduct.isEmpty()) {
			throw new UserNotFoundException(username, password);
		}

		return "Login Successful";
	}

}
```

```
package com.project.quizuser.controller;

import org.springframework.beans.factory.annotation.Autowired;
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.project.quizuser.Repo.ScoreCompareRepository;
import com.project.quizuser.entity.ScoreCompare;

@RestController
@RequestMapping("/scorecard")
public class ScoreCompareController {
```

```
	@Autowired
	ScoreCompareRepository repo;

	@GetMapping
	public Iterable<ScoreCompare> getUser() {
		return repo.findAll();
	}

	@PostMapping
	public ScoreCompare create(@RequestBody ScoreCompare sc) {
		return repo.save(sc);
	}
}
```

```
server.port=8083
spring.datasource.url=jdbc:mysql://localhost:3306/myquizportal
spring.datasource.username=root
spring.datasource.password=Veda@1202189
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.hibernate.ddl-auto=update
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
logging.level.org.hibernate=INFO
logging.level.org.hibernate.SQL=INFO
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
logging.level.org.springframework=INFO
#logging.level.org.apache=ERROR
#logging.level.org.springframework.web=TRACE
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="US-ASCII">
<title>User Login Page</title>
<h1>Please login to continue</h1>
</head>
<body>

	<form action="userloginverify" method="post">

		Username: <input type="text" name="name"> <br> Password:
		<input type="password" name="password"> <br> <br> <input
			type="submit" value="Login">
	</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Add Quiz</title>
</head>
<body>
        <h1>Add Quiz</h1>
        <form action="addtoquiz" method="post">
                <table>

                        <tr>
                                <td>QuizID</td>
                                <td><select name="quizid">
                                                <c:forEach items="${quiz}"
var="quiz">

<option>${quiz.quizid}</option>
                                                </c:forEach>
                                </select></td>
                        </tr>
                        <tr>
                                <td>QuesID</td>
                                <td><select name="quesid">
                                                <c:forEach items="${ques}"
var="ques">
                                                        <option>${ques.id}</option>
                                                </c:forEach>
                                </select></td>
                        <tr>
                                <td></td>
                                <td><input type="submit" value="Add" /></td>
                        </tr>
                </table>
        </form>
        <br />
        <br />

</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Quiz Portal</title>
</head>
<body>

	<form action="testpage">
		<table>
			<tr>
				<td>Search for the quiz:</td>
				<td><select name="quiz">
						<c:forEach items="${quiz}"
var="quiz">

<option>${quiz.quizname}</option>
						</c:forEach>
				</select></td>

			</tr>
			<tr>
				<td></td>
				<td><input type="submit" value="start" /></td>
			</tr>
		</table>
	</form>
	<br />
	<br />

</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Java Quiz</title>
</head>
<body>
	<form action="result">
		<c:forEach items="${quiz}" var="quiz" begin="1" end="1">
			<c:forEach items="${ques}" var="ques" begin="0" end="0">

				<h2>
					<c:out value="${ques.ques}" />
				</h2>
```

```
                                <br />
                                <c:forEach var="answer" items="${answer}" begin="0"
end="1">
                                        <li><input type="radio"
name="question_${ques.id}"
                                                value="${answer.id}" /> <c:out
value="${answer.ans}" /></li>

                                </c:forEach>

                        </c:forEach>
                        <c:forEach items="${ques}" var="ques" begin="1" end="1">

                                <h2>
                                        <c:out value="${ques.ques}" />
                                </h2>
                                <br />
                                <c:forEach var="answer" items="${answer}" begin="2"
end="3">
                                        <li><input type="radio"
name="question_${ques.id}"
                                                value="${answer.id}" /> <c:out
value="${answer.ans}" /></li>

                                </c:forEach>

                        </c:forEach>
                </c:forEach>
                <input type="submit" value="submit" />
        </form>


</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Java Quiz</title>
</head>
<body>
        <form action="result">
                <c:forEach items="${quiz}" var="quiz" begin="2" end="2">
                        <c:forEach items="${ques}" var="ques" begin="2" end="2">
```

```
                                <h2>
                                        <c:out value="${ques.ques}" />
                                </h2>
                                <br />
                                <c:forEach var="answer" items="${answer}" begin="4"
end="5">
                                        <li><input type="radio"
name="question_${ques.id}"
                                                value="${answer.id}" /> <c:out
value="${answer.ans}" /></li>

                                </c:forEach>

                        </c:forEach>
                        <c:forEach items="${ques}" var="ques" begin="3" end="3">

                                <h2>
                                        <c:out value="${ques.ques}" />
                                </h2>
                                <br />
                                <c:forEach var="answer" items="${answer}" begin="6"
end="7">
                                        <li><input type="radio"
name="question_${ques.id}"
                                                value="${answer.id}" /> <c:out
value="${answer.ans}" /></li>

                                </c:forEach>

                        </c:forEach>
                </c:forEach>
                <input type="submit" value="submit" />
        </form>


</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1" isELIgnored="false" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h2>Your Final Result is 1</h2>
<<c:forEach items="${scores}" var="score">
```

```
                        <li>${score.uname} | ${score.scorecard} </li>
                </c:forEach>
</body>
</html>
```

_____**********_____

please go through this code if the screenshots are not visible