Quiz.java

```java
package com.online;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;


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

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Id
    @Column(name="SerialId")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name="QuizId")
    private String quizname;

    //private int questionId;
    //@JsonIgnore
    @ManyToMany(fetch = FetchType.EAGER)
    private List<QuestionEntity>listques = new ArrayList<>();

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

    public String getQuizname() {
        return quizname;
    }
    public void setQuizname(String quizname) {
        this.quizname = quizname;
    }
    public List<QuestionEntity> getListques() {
        return listques;
    }
    public void setListques(List<QuestionEntity> listques) {
        this.listques = listques;
```

```
    }


}
```

QuizRepo.java

```
package com.online;

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

@Repository
public interface QuizRepo extends JpaRepository<Quiz, Integer>
{

}
```

Response.java

```
package com.online;

public class Response {

    private int id;
    private String resp;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getResp() {
        return resp;
    }
    public void setResp(String resp) {
        this.resp = resp;
    }



}
```

QuizService.java

```
package com.online;

import java.util.List;

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

@Service
public class QuizService

{

    @Autowired
    private QuizRepo quizRepo;

    //add or CREATE method
    public Quiz addquizques(Quiz q)
    {
        System.out.println(q);
        return quizRepo.save(q);
    }


    //List all Quiz
    public List<Quiz> getAllQuizQues(){
        return quizRepo.findAll();
    }

    // Get Quiz by id of Quiz
    public Quiz getQuizById(int id) {
        if (quizRepo.findById(id).isPresent())
            return quizRepo.findById(id).get();
        else
            return null;

    }
}
```

QuestionRepo.java

```
package com.online;

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

@Repository
public interface QuestionRepo extends JpaRepository<QuestionEntity, Integer>
{

}
```

UserResponse.java

```
package com.online;

import java.util.List;

public class UserResponse {

    private List<Response> res;
    private int id;

    public List<Response> getRes() {
        return res;
    }
    public void setRes(List<Response> res) {
        this.res = res;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }


}
```

QuestionEntity.java

```
package com.online;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;

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

import com.fasterxml.jackson.annotation.JsonIgnore;




@Entity(name="questions")
public class QuestionEntity implements Serializable {


    private static final long serialVersionUID = 1L;



    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name="QId")
    private int id;
    private String question;
    private String optionA;
    private String optionB;
    private String optionC;
    private String optionD;
    private String answer;

    @JsonIgnore
    @ManyToMany( mappedBy = "listques")
    private List<Quiz>quizlists =new ArrayList<>();

    public List<Quiz> getQuizlists() {
        return quizlists;
    }
    public void setQuizlists(List<Quiz> quizlists) {
        this.quizlists = quizlists;
    }


    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getQuestion() {
```

QuestionEntity.java

```
    return question;
}
public void setQuestion(String question) {
    this.question = question;
}
public String getOptionA() {
    return optionA;
}
public void setOptionA(String optionA) {
    this.optionA = optionA;
}
public String getOptionB() {
    return optionB;
}
public void setOptionB(String optionB) {
    this.optionB = optionB;
}
public String getOptionC() {
    return optionC;
}
public void setOptionC(String optionC) {
    this.optionC = optionC;
}
public String getOptionD() {
    return optionD;
}
public void setOptionD(String optionD) {
    this.optionD = optionD;
}
public String getAnswer() {
    return answer;
}
public void setAnswer(String answer) {
    this.answer = answer;
}
/*public QuestionEntity(int id, String question, String optionA, String optionB, String optionC, String opt
        String answer) {
    super();
    this.id = id;
    this.question = question;
    this.optionA = optionA;
    this.optionB = optionB;
    this.optionC = optionC;
    this.optionD = optionD;
    this.answer = answer;
}
public QuestionEntity() {
    super();
    // TODO Auto-generated constructor stub
}-*/



}
```

QuizController.java

```
package com.online;

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

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

    @Autowired
    private QuizService service;

    @PostMapping("/")
    public ResponseEntity<String> addquizques(@RequestBody Quiz q)
    {
        Quiz user= service.addquizques(q);

        if(user!=null)
            return new ResponseEntity<String>("Quiz is Created",HttpStatus.CREATED);
        else
            return new ResponseEntity<String>("Quiz was not Created", HttpStatus.INTERNAL_SERVER_EI
    }

    @GetMapping("/")
    public  List<Quiz> getAllQuizQues(){
        return service.getAllQuizQues();
    }

    @RequestMapping("/{id}")
    public ResponseEntity<Quiz> getQuizById(@PathVariable int id){
          Quiz user= service.getQuizById(id);

          if(user!=null)
              return new ResponseEntity<Quiz>(user,HttpStatus.FOUND);
          else
              return new  ResponseEntity<Quiz>(user,HttpStatus.NOT_FOUND);
    }
}
```

UserController.java

```
package com.online;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
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;

@RestController
@RequestMapping("/api/user")
public class UserController {

    @Autowired
    private UserInputService service;

    @PostMapping("/submit")
    public ResponseEntity<String> addQues(@RequestBody UserResponse submittedResponse) {

        try {
            int score = service.calculateScore(submittedResponse);

            return new ResponseEntity<String>("Score for this Quiz = "+String.valueOf(score), HttpStatus.AC
        } catch (Exception e) {
            return new ResponseEntity<String>("Wrong Input : Question Id Mismatch", HttpStatus.BAD_REQ


        }


    }

}
```

QuestionService.java

```java
package com.online;

import java.util.List;

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

@Service
public class QuestionService {

    @Autowired
    private QuestionRepo repo;


    //add method or CREATE RECORD
    public QuestionEntity addQues(QuestionEntity u) {
        return repo.save(u);
    }


    //List user Method
    public List<QuestionEntity> getAllQues(){
        return repo.findAll();
    }

    //get ques by id
        public QuestionEntity getQuesById(int id) {
            if(repo.findById(id).isPresent())
                return repo.findById(id).get();
            else
                return null;
        }


    //update question by id
        public QuestionEntity updateById(QuestionEntity ques, int id) {

            if(repo.findById(id).isPresent())
            {
                QuestionEntity old=repo.findById(id).get();
                old.setQuestion(ques.getQuestion());
                old.setOptionA(ques.getOptionA());
                old.setOptionB(ques.getOptionB());
                old.setOptionC(ques.getOptionC());
                old.setOptionD(ques.getOptionD());
                old.setAnswer(ques.getAnswer());

                return repo.save(old);

            }
            else
                return null;

        }
```

QuestionService.java

```
}
```

UserInputService.java

```java
package com.online;

import java.util.List;
import java.util.Set;
import java.util.stream.Collectors;

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



@Service
public class UserInputService {

    @Autowired
    private QuestionRepo repo;
    @Autowired
    private QuizRepo quizRepo;

    @SuppressWarnings("deprecation")
    public int calculateScore(UserResponse submittedResponse) throws Exception {

        List<Response> submittedAns = submittedResponse.getRes();
        int score = 0;
        int quizId = submittedResponse.getId();
        Quiz currentQuiz = quizRepo.getById(quizId);
        List<QuestionEntity> validQuestionList = currentQuiz.getListques();

        Set<Integer> validQuestionIdList = validQuestionList.stream().map(ele -> ele.getId())
                .collect(Collectors.toSet());


       // System.out.println(validQuestionIdList);
        for (Response currentQuestion : submittedAns) {

            //System.out.println(validQuestionIdList.contains(currentQuestion.getId()));

            if (validQuestionIdList.contains(currentQuestion.getId())) {
                String correctAns = repo.getById(currentQuestion.getId()).getAnswer();

                if (correctAns.equalsIgnoreCase(currentQuestion.getResp())) {
                    score++;
                }
            } else {
                throw new Exception("Question Id mismatch for this Quiz");
            }
        }
        return score;
    }

}
```

QuestionController.java

```
package com.online;

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.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;



@RestController
@RequestMapping("/api/questions")
public class QuestionController {


    @Autowired
    private QuestionService service;

    //create new question
    @PostMapping("/")
    public ResponseEntity<QuestionEntity> addQues(@RequestBody QuestionEntity u){
        QuestionEntity user= service.addQues(u);
        if(user!=null)
            return new ResponseEntity<QuestionEntity>(user,HttpStatus.CREATED);
        else
            return new ResponseEntity<QuestionEntity>(user, HttpStatus.INTERNAL_SERVER_ERROR);
    }

    //List of all Questions
    @GetMapping("/")
    public  List<QuestionEntity> getAllQues(){
        return service.getAllQues();
    }

    //Get a question by id
    @RequestMapping("/{id}")
    public ResponseEntity<QuestionEntity> getQuesById(@PathVariable int id){
          QuestionEntity user= service.getQuesById(id);

          if(user!=null)
              return new ResponseEntity<QuestionEntity>(user,HttpStatus.FOUND);
          else
              return new  ResponseEntity<QuestionEntity>(user,HttpStatus.NOT_FOUND);
    }

    //Update Question by id
    @PutMapping("/{id}")
    public ResponseEntity<Object> updateById(@RequestBody QuestionEntity ques,@PathVariable int id
    {
```

QuestionController.java

```
        QuestionEntity data= service.updateById(ques, id);
        if(data!=null)
            return new ResponseEntity<Object>(data,HttpStatus.OK);
        else
            return new ResponseEntity<Object>("User is Not Available",HttpStatus.NOT_FOUND);
    }

}
```

OnlineQuizApplication.java

```
package com.online;

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

@SpringBootApplication

public class OnlineQuizApplication {

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

}
```

OnlineQuizApplicationTests.java

```
package com.quiz;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class OnlineQuizApplicationTests {

    @Test
    void contextLoads() {
    }

}
```

application.properties

```
spring.datasource.username=root
spring.datasource.password=qwertyuiop
spring.datasource.url=jdbc:mysql://localhost:3306/quiz
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.jpa.show-sql=true
server.port=8082
```