

SOURCE-CODE

Com.bean Folder:-

- Admin.java code

```
package com.bean;  
import javax.persistence.Entity;  
import javax.persistence.Id;
```

```
@Entity
```

```
public class Admin {
```

```
    @Id
```

```
    private String emailid;
```

```
    private String password;
```

```
    /**
```

```
     * @return the emailid
```

```
     */
```

```
    public String getEmailid() {
```

```
        return emailid;
```

```
    }
```

```
    /**
```

```
     * @param emailid the emailid to set
```

```
     */
```

```
    public void setEmailid(String emailid) {
```

```
        this.emailid = emailid;
```

```
    }
```

```
    /**
```

```
     * @return the password
```

```
     */
```

```
    public String getPassword() {
```

```
        return password;
```

```
    }
```

```
    /**
```

```

        * @param password the password to set
        */
        public void setPassword(String password) {
            this.password = password;
        }
        @Override
        public String toString() {
            return "Admin [emailid=" + emailid + ", password=" + password
+ "];"
        }
    }

```

- **Participant.java code:-**

```
package com.bean;
```

```
import javax.persistence.Entity;
```

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

```
@Entity
```

```
public class Participant {
```

```
    @Id
```

```
    private String emailid;
```

```
    private int pid;
```

```
    private String name;
```

```
    private int age;
```

```
private String password;

/**
 * @return the emailid
 */
public String getEmailid() {
    return emailid;
}

/**
 * @param emailid the emailid to set
 */
public void setEmailid(String emailid) {
    this.emailid = emailid;
}

/**
 * @return the pid
 */
public int getPid() {
    return pid;
}

/**
 * @param pid the pid to set
 */
```

```
public void setPid(int pid) {  
    this.pid = pid;  
}  
  
/**  
 * @return the name  
 */  
public String getName() {  
    return name;  
}  
  
/**  
 * @param name the name to set  
 */  
public void setName(String name) {  
    this.name = name;  
}  
  
/**  
 * @return the age  
 */  
public int getAge() {  
    return age;  
}  
  
/**
```

```

    * @param age the age to set
    */
    public void setAge(int age) {
        this.age = age;
    }

    /**
     * @return the password
     */
    public String getPassword() {
        return password;
    }

    /**
     * @param password the password to set
     */
    public void setPassword(String password) {
        this.password = password;
    }

    @Override
    public String toString() {
        return "Participant [emailid=" + emailid + ", pid=" + pid + ", name=" +
name + ", age=" + age + ", password="
        + password + "];"
    }

```

- **Question.java Code**

```
package com.bean;
```

```
import javax.persistence.Entity;
```

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

```
@Entity
```

```
public class Question {
```

```
    @Id
```

```
    private int qid;
```

```
    private String question;
```

```
    private String optionA;
```

```
    private String optionB;
```

```
    private String optionC;
```

```
    private String optionD;
```

```
    private String correctanswer;
```

```
    public int getQid() {
```

```
        return qid;
```

```
    }
```

```
    public void setQid(int qid) {
```

```
        this.qid = qid;
```

```
    }
```

```
    public String getQuestion() {
```

```
        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 getCorrectanswer() {
        return correctanswer;
    }
    public void setCorrectanswer(String correctanswer) {
        this.correctanswer = correctanswer;
    }
    @Override
    public String toString() {
        return "Question [qid=" + qid + ", question=" + question + ",
optionA=" + optionA + ", optionB=" + optionB
        + ", optionC=" + optionC + ", optionD=" + optionD
+ ", correctanswer=" + correctanswer + "]\n";
    }

```

```
}}
```

- **Quiz.java**

```
package com.bean;
```

```
import javax.persistence.Entity;
```

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

```
@Entity
```

```
public class Quiz {
```

```
    @Id
```

```
    private int qzid;
```

```
    private String subject;
```

```
    private int qid;
```

```
    public int getQzid() {
```

```
        return qzid;
```

```
    }
```

```
    public void setQzid(int qzid) {
```

```
        this.qzid = qzid;
```

```
    }
```

```
    public String getSubject() {
```

```
        return subject;
```



```

    }

    public void setSubject(String subject) {

        this.subject = subject;

    }

    public int getQid() {

        return qid;

    }

    public void setQid(int qid) {

        this.qid = qid;

    }

    @Override

    public String toString() {

        return "Quiz [qzid=" + qzid + ", subject=" + subject + ", qid=" + qid +

    "];

    }}

```

- **Scores.java Code-**

```

package com.bean;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Scores {

```

@Id

```
private String emailid;

private String name;

private String subject;

private int score;

/**
 * @return the emailid
 */
public String getEmailid() {
    return emailid;
}

/**
 * @param emailid the emailid to set
 */
public void setEmailid(String emailid) {
    this.emailid = emailid;
}

/**
 * @return the name
 */
public String getName() {
    return name;
}
```

```

}

/**
 * @param name the name to set
 */
public void setName(String name) {
    this.name = name;
}

/**
 * @return the subject
 */
public String getSubject() {
    return subject;
}

/**
 * @param subject the subject to set
 */
public void setSubject(String subject) {
    this.subject = subject;
}

/**
 * @return the score
 */

```

```

public int getScore() {
    return score;
}

/**
 * @param score the score to set
 */
public void setScore(int score) {
    this.score = score;
}

@Override
public String toString() {
    return "Score [emailid=" + emailid + ", name=" + name + ", subject="
+ subject + ", score=" + score + "];"
}
}

```

- **UserQuiz.java Code**

```

package com.bean;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity
public class Userquiz {

    @Id

```

```

private int uqid;
private String emailid;
private int qzid;
private int qid;
private String usercorrectans;
/**
 * @return the uqid
 */
public int getUqid() {
    return uqid;
}
/**
 * @param uqid the uqid to set
 */
public void setUqid(int uqid) {
    this.uqid = uqid;
}
/**
 * @return the emailid
 */
public String getEmailid() {
    return emailid;
}
/**
 * @param emailid the emailid to set
 */
public void setEmailid(String emailid) {
    this.emailid = emailid;
}
/**
 * @return the qzid
 */
public int getQzid() {

```

```

        return qzid;
    }
    /**
     * @param qzid the qzid to set
     */
    public void setQzid(int qzid) {
        this.qzid = qzid;
    }
    /**
     * @return the qid
     */
    public int getQid() {
        return qid;
    }
    /**
     * @param qid the qid to set
     */
    public void setQid(int qid) {
        this.qid = qid;
    }
    /**
     * @return the usercorrectans
     */
    public String getUsercorrectans() {
        return usercorrectans;
    }
    /**
     * @param usercorrectans the usercorrectans to set
     */
    public void setUsercorrectans(String usercorrectans) {
        this.usercorrectans = usercorrectans;
    }
}
@Override

```

```

        public String toString() {
            return "Userquiz [uqid=" + uqid + ", emailid=" + emailid + ",
qzid=" + qzid + ", qid=" + qid
                + ", usercorrectans=" + usercorrectans + "];"
        }
    }

```

Com.controller folder:-

- **AdminController.java code-**

```

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

```

```

import com.bean.Admin;
import com.bean.Question;
import com.bean.Quiz;
import com.bean.Scores;
import com.service.AdminService;

```

```

@RestController

```

```

@RequestMapping("admin")
public class AdminController {

    @Autowired
    AdminService adminService;

    @PostMapping(value = "signIn",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String checkAdmin(@RequestBody Admin ad) {
        return adminService.CheckAdmin(ad);
    }

    @PatchMapping(value = "updatePassword",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String updatePassword(@RequestBody Admin ad) {
        return adminService.updatePassword(ad);
    }

    @PostMapping(value = "addQuestion",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String addQuestion(@RequestBody Question qn) {
        return adminService.addQuestion(qn);
    }

    @GetMapping(value = "questionbyID/{id}",produces =
MediaType.APPLICATION_JSON_VALUE)
    public Question findQuestionById(@PathVariable("id") int qid) {
        return adminService.findQuestionById(qid);
    }

    @PostMapping(value = "addQuiz",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String addQuiz(@RequestBody Quiz qz) {

```



```

        return adminService.addQuiz(qz);
    }

    @GetMapping(value = "Ranking",produces =
    MediaType.APPLICATION_JSON_VALUE)
    public List<Scores> getstandingPosition() {
        return adminService.getstandingPosition();
    }
}

```

- **ParticipantController.java Code-**

```

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

import com.bean.Participant;
import com.bean.Question;
import com.bean.Quiz;
import com.bean.Scores;
import com.bean.Userquiz;
import com.service.ParticipantService;

```

```

@RestController
@RequestMapping("participant")
public class ParticipantController {

    @Autowired
    ParticipantService participantService;

    @PostMapping(value = "userRegister",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String userRegister(@RequestBody Participant pt) {
        return participantService.userRegister(pt);
    }

    @PostMapping(value = "signIn")
    public String checkUser(@RequestBody Participant pt) {
        return participantService.checkUser(pt);
    }

    @PostMapping(value = "takeQuiz",consumes =
MediaType.APPLICATION_JSON_VALUE)
    public String takeQuiz(@RequestBody Userquiz uq) {
        return participantService.takeQuiz(uq);
    }

    @GetMapping(value = "getResult/{emailid}",produces =
MediaType.APPLICATION_JSON_VALUE)
    public Scores getResult(@PathVariable("emailid") String emailid) {
        return participantService.getResult(emailid);
    }

    @GetMapping(value = "getAllQuiz",produces =
MediaType.APPLICATION_JSON_VALUE)
    public List<Quiz> getAllEmployeeInfoDB() {

```

```
        return participantService.getAllQuiz();  
    }  
}
```

Com.Service Folder:-

- **AdminService Code-**

```
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.Scores;  
import com.dao.AdminDao;
```

```
@Service
```

```
public class AdminService {
```

```
    @Autowired
```

```
    AdminDao adminDao;
```

```
    public String CheckAdmin(Admin ad) {  
        if(adminDao.checkAdmin(ad)>0) {  
            return "Admin Login Successful";  
        }  
        else {  
            return "Failure! Provide Correct Credentials!";  
        }  
    }  
}
```

```
    }  
}
```

```
public String updatePassword(Admin ad) {  
    Admin a = adminDao.getAdminDetails(ad.getEmailid());  
    if(a==null) {  
        return "Admin details not present, cannot be updated";  
    }  
    else if(adminDao.updatePassword(ad)>0) {  
        return "Password Updated";  
    }else {  
        return "Password Not Updated";  
    }  
}
```

```
public String addQuestion(Question qn) {  
    if(adminDao.addQuestion(qn)>0) {  
        return "Question Added Successfully";  
    }  
    else {  
        return "Question didnt added";  
    }  
}
```

```
public String addQuiz(Quiz qz) {  
    if(adminDao.addQuiz(qz)>0) {  
        return "Quiz Created Successfully";  
    }  
    else {  
        return "Question didnt created";  
    }  
}
```

```

    public Question findQuestionById(int qid) {
        return adminDao.getQestionDetails(qid);
    }

    public List<Scores> getstandingPosition(){
        return adminDao.getstandingPosition();
    }
}

```

- **ParticipantService.java Code**

```

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.Participant;
import com.bean.Question;
import com.bean.Quiz;
import com.bean.Scores;
import com.bean.Userquiz;
import com.dao.ParticipantDao;

```

```

@Service

```

```

public class ParticipantService {

```

```

    @Autowired

```

```

    ParticipantDao participantDao;

```

```
public String userRegister(Participant pt) {  
    if(participantDao.userRegister(pt)>0) {  
        return "User Registration Successful";  
    }  
    else {  
        return "User Registration Failure";  
    }  
}
```

```
public String checkUser(Participant pt) {  
    if(participantDao.checkUser(pt)>0) {  
        return "Participant Login Successful";  
    }  
    else {  
        return "Failure! Provide Correct Credentials!";  
    }  
}
```

```
public String takeQuiz(Userquiz uq) {  
    if(participantDao.takeQuiz(uq)>0) {  
        return "Quiz Attempted Successfully";  
    }  
    else {  
        return "Try Again use a unique user id!";  
    }  
}
```

```
public Scores getResult(String emailid) {  
    return participantDao.getResult(emailid);  
}
```

```
public List<Quiz> getAllQuiz(){  
    return participantDao.getAllQuiz();}}
```

com.Dao Folder

- **AdminDao.java Code:-**

```
package com.dao;
```

```
import java.util.List;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.EntityManagerFactory;
```

```
import javax.persistence.EntityTransaction;
```

```
import javax.persistence.Query;
```

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

```
import org.springframework.stereotype.Repository;
```

```
import com.bean.Admin;
```

```
import com.bean.Question;
```

```
import com.bean.Quiz;
```

```
import com.bean.Scores;
```

```
@Repository
```

```
public class AdminDao {
```

```
    @Autowired
```

```
    EntityManagerFactory emf;
```

```
    public Admin getAdminDetails(String email) {
```

```
        EntityManager manager = emf.createEntityManager();
```

```
        Admin ad = manager.find(Admin.class, email);
```

```
        return ad;
```

```
}
```

```
public Question getQestionDetails(int qid) {  
    EntityManager manager = emf.createEntityManager();  
    Question qn = manager.find(Question.class, qid);  
    return qn;  
}
```

```
public int checkAdmin(Admin ad) {  
  
    EntityManager manager = emf.createEntityManager();  
    Admin a = manager.find(Admin.class, ad.getEmailid());  
    if(a==null) {  
        return 0;  
    }  
    else {  
        String pass=ad.getPassword();  
        if(pass.equals(a.getPassword())) {  
            return 1;  
        }  
        return 0;  
    }  
  
}
```

```
public int updatePassword(Admin ad) {  
    EntityManager manager = emf.createEntityManager();  
    EntityTransaction tran = manager.getTransaction();  
    Admin e = manager.find(Admin.class, ad.getEmailid());  
    if(e==null) {  
        return 0;  
    }  
    else {
```



```

        e.setPassword(ad.getPassword());
        tran.begin();
        manager.merge(e);
        tran.commit();
        return 1;
    }
}

public int addQuestion(Question qn) {
    try {
        EntityManager manager = emf.createEntityManager();
        EntityTransaction tran = manager.getTransaction();
        tran.begin();
        manager.persist(qn);
        tran.commit();
        return 1;

    } catch (Exception e) {
        // TODO: handle exception
        System.out.println(e);
        return 0;
    }
}

public int addQuiz(Quiz qz) {
    try {
        EntityManager manager = emf.createEntityManager();
        EntityTransaction tran = manager.getTransaction();
        tran.begin();
        manager.persist(qz);
        tran.commit();
        return 1;
    }
}

```

```

        } catch (Exception e) {
            // TODO: handle exception
            System.out.println(e);
            return 0;
        }
    }

    public List<Scores> getstandingPosition(){
        EntityManager manager = emf.createEntityManager();
        Query qry = manager.createQuery("select
s.emailid,s.subject,s.score FROM Scores s ORDER BY s.score DESC");//JPQL
        List<Scores> list = qry.getResultList();
        return list;
    }
}

```

- **ParticipantDao.java Code:-**

```
package com.dao;
```

```
import java.util.List;
```

```
import javax.persistence.EntityManager;
```

```
import javax.persistence.EntityManagerFactory;
```

```
import javax.persistence.EntityTransaction;
```

```
import javax.persistence.Query;
```

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

```
import org.springframework.stereotype.Repository;
```

```
import com.bean.Admin;
```

```
import com.bean.Participant;
```

```
import com.bean.Question;
```

```
import com.bean.Quiz;  
import com.bean.Scores;  
import com.bean.Userquiz;
```

```
@Repository
```

```
public class ParticipantDao {
```

```
    @Autowired
```

```
    EntityManagerFactory emf;
```

```
    public int userRegister(Participant pt) {
```

```
        try {
```

```
            EntityManager manager = emf.createEntityManager();
```

```
            EntityTransaction tran = manager.getTransaction();
```

```
            tran.begin();
```

```
            manager.persist(pt);
```

```
            tran.commit();
```

```
            return 1;
```

```
        } catch (Exception e) {
```

```
            // TODO: handle exception
```

```
            System.out.println(e);
```

```
            return 0;
```

```
        }
```

```
    }
```

```
    public int checkUser(Participant pt) {
```

```
        EntityManager manager = emf.createEntityManager();
```

```
        Participant p = manager.find(Participant.class, pt.getEmailid());
```

```
        if(p==null) {
```

```
            return 0;
```

```
        }
```

```
        else {
```

```

        String pass=pt.getPassword();
        if(pass.equals(p.getPassword())) {
            return 1;
        }
        return 0;
    }

}

public int takeQuiz(Userquiz uq) {
    try {
        EntityManager manager = emf.createEntityManager();
        EntityTransaction tran = manager.getTransaction();
        tran.begin();
        manager.persist(uq);
        Scores s = manager.find(Scores.class,uq.getEmailid());
        Question q = manager.find(Question.class, uq.getQid());
        Quiz qz = manager.find(Quiz.class, uq.getQzid());
        String res = uq.getUsercorrectans();
        if(res.equals(q.getCorrectanswer())) {
            s.setEmailid(uq.getEmailid());
            s.setSubject(qz.getSubject());
            s.setScore(s.getScore()+1);
        }
        tran.commit();
        return 1;

    } catch (Exception e) {
        // TODO: handle exception
        System.out.println(e);
        return 0;
    }
}

```

```
    }  
}
```

```
public Scores getResult(String emailid) {  
    EntityManager manager = emf.createEntityManager();  
    Scores s = manager.find(Scores.class, emailid);  
    return s;  
}
```

```
public List<Quiz> getAllQuiz(){  
    EntityManager manager = emf.createEntityManager();  
    Query qry = manager.createQuery("select q from Quiz  
q");//JPQL  
    List<Quiz> list = qry.getResultList();  
    return list;  
}}
```