**SourceCodes**

**Question.java**

**package** com.abc.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table

**public** **class** Question {

@Id

@Column

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** Long questionId;

@Column

**private** String questionText;

@Column

**private** String questionAnswer;

**public** Long getQuestionId() {

**return** questionId;

}

**public** **void** setQuestionId(Long questionId) {

**this**.questionId = questionId;

}

**public** String getQuestionText() {

**return** questionText;

}

**public** **void** setQuestionText(String questionText) {

**this**.questionText = questionText;

}

**public** String getQuestionAnswer() {

**return** questionAnswer;

}

**public** **void** setQuestionAnswer(String questionAnswer) {

**this**.questionAnswer = questionAnswer;

}

**public** Question() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Question(Long questionId, String questionText, String questionAnswer) {

**super**();

**this**.questionId = questionId;

**this**.questionText = questionText;

**this**.questionAnswer = questionAnswer;

}

@Override

**public** String toString() {

**return** "Question [questionId=" + questionId + ", questionText=" + questionText + ", questionAnswer="

+ questionAnswer + "]";

}

}

**Quiz.java:**

**package** com.abc.model;

**import** java.util.List;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.ManyToMany;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.OneToMany;

**import** javax.persistence.OneToOne;

@Entity

**public** **class** Quiz {

@Id

**private** **long** quizId;

@Column

**private** String quizName;

@OneToMany()

//@JoinColumn(name="questionlist" )

**private** List<Question> questionList;

**public** **long** getQuizId() {

**return** quizId;

}

**public** **void** setQuizId(**long** quizId) {

**this**.quizId = quizId;

}

**public** String getQuizName() {

**return** quizName;

}

**public** **void** setQuizName(String quizName) {

**this**.quizName = quizName;

}

**public** List<Question> getQuestionList() {

**return** questionList;

}

**public** **void** setQuestionList(List<Question> questionList) {

**this**.questionList = questionList;

}

**public** Quiz(**long** quizId, String quizName, List<Question> questionList) {

**super**();

**this**.quizId = quizId;

**this**.quizName = quizName;

**this**.questionList = questionList;

}

**public** Quiz() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Quiz [quizId=" + quizId + ", quizName=" + quizName + ", questionList=" + questionList + "]";

}

}

**CheckAnswer.java:**

**package** com.abc.model;

**public** **class** CheckAnswer {

**private** **int** qusId;

**private** String qusText;

**private** String yourAnswer;

**private** String correctAnswer;

**private** Boolean correct;

**public** **int** getQusId() {

**return** qusId;

}

**public** **void** setQusId(**int** qusId) {

**this**.qusId = qusId;

}

**public** String getQusText() {

**return** qusText;

}

**public** **void** setQusText(String qusText) {

**this**.qusText = qusText;

}

**public** String getYourAnswer() {

**return** yourAnswer;

}

**public** **void** setYourAnswer(String yourAnswer) {

**this**.yourAnswer = yourAnswer;

}

**public** String getCorrectAnswer() {

**return** correctAnswer;

}

**public** **void** setCorrectAnswer(String correctAnswer) {

**this**.correctAnswer = correctAnswer;

}

**public** **boolean** isCorrect() {

**return** correct;

}

**public** **void** setCorrect(**boolean** correct) {

**this**.correct = correct;

}

**public** CheckAnswer() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** CheckAnswer(**int** qusId, String qusText, String yourAnswer, String correctAnswer, **boolean** correct) {

**super**();

**this**.qusId = qusId;

**this**.qusText = qusText;

**this**.yourAnswer = yourAnswer;

**this**.correctAnswer = correctAnswer;

**this**.correct = correct;

}

}

**PAnswer.java**

**package** com.abc.model;

**public** **class** PAnswer {

**private** **int** qlid;

**private** String ans;

**public** **int** getQlid() {

**return** qlid;

}

**public** **void** setQlid(**int** qlid) {

**this**.qlid = qlid;

}

**public** String getAns() {

**return** ans;

}

**public** **void** setAns(String ans) {

**this**.ans = ans;

}

**public** PAnswer(**int** qlid, String ans) {

**super**();

**this**.qlid = qlid;

**this**.ans = ans;

}

**public** PAnswer() {

**super**();

// **TODO** Auto-generated constructor stub

}

}

**ParticipnatDetails.java**

**package** com.abc.model;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

@Entity

**public** **class** ParticipantDetails {

@Id

**private** **int** pId;

**private** String pName;

**private** String pEmail;

**private** String password;

**public** ParticipantDetails() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** **int** getpId() {

**return** pId;

}

**public** **void** setpId(**int** pId) {

**this**.pId = pId;

}

**public** String getpName() {

**return** pName;

}

**public** **void** setpName(String pName) {

**this**.pName = pName;

}

**public** String getpEmail() {

**return** pEmail;

}

**public** **void** setpEmail(String pEmail) {

**this**.pEmail = pEmail;

}

**public** ParticipantDetails(**int** pId, String pName, String pEmail, String password) {

**super**();

**this**.pId = pId;

**this**.pName = pName;

**this**.pEmail = pEmail;

**this**.password = password;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** ParticipantDetails(**int** pId, String pName, String pEmail) {

**super**();

**this**.pId = pId;

**this**.pName = pName;

**this**.pEmail = pEmail;

}

}

**QuestionToParticipant:**

**package** com.abc.model;

**public** **class** QuestionToParticipant {

**private** Long id;

**private** String qusText;

**public** Long getId() {

**return** id;

}

**public** String getQusText() {

**return** qusText;

}

**public** **void** setQusText(String qusText) {

**this**.qusText = qusText;

}

**public** QuestionToParticipant() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** QuestionToParticipant(Long id, String qusText) {

**super**();

**this**.id = id;

**this**.qusText = qusText;

}

**public** **void** setId(Long id) {

**this**.id = id;

}

}

**QuizToParticipant:**

**package** com.abc.model;

**public** **class** QuizToParticipant {

**private** **int** id;

**private** String name;

**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** QuizToParticipant() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** QuizToParticipant(**int** id, String name) {

**super**();

**this**.id = id;

**this**.name = name;

}

**Statistics.java:**

**package** com.abc.model;

**public** **class** Statistics {

**private** Long quizNumber;

**private** Long questionNumber;

**private** Long participantNumber;

**public** Long getQuizNumber() {

**return** quizNumber;

}

**public** **void** setQuizNumber(Long quizNumber) {

**this**.quizNumber = quizNumber;

}

**public** Long getQuestionNumber() {

**return** questionNumber;

}

**public** **void** setQuestionNumber(Long questionNumber) {

**this**.questionNumber = questionNumber;

}

**public** Long getParticipantNumber() {

**return** participantNumber;

}

**public** **void** setParticipantNumber(Long participantNumber) {

**this**.participantNumber = participantNumber;

}

**public** Statistics() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** Statistics(Long quizNumber, Long questionNumber, Long participantNumber) {

**super**();

**this**.quizNumber = quizNumber;

**this**.questionNumber = questionNumber;

**this**.participantNumber = participantNumber;

}

@Override

**public** String toString() {

**return** "Statistics [quizNumber=" + quizNumber + ", questionNumber=" + questionNumber + ", participantNumber="

+ participantNumber + "]";

}

}

**QuestionRepository.java:**

**package** com.abc.repository;

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

**import** com.abc.model.Question;

**public** **interface** QuestinonRepository **extends** JpaRepository<Question, Long> {

}

QuizRepository.java:

**package** com.abc.repository;

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

**import** com.abc.model.Quiz;

**public** **interface** QuizRepository **extends** JpaRepository<Quiz,Long>{

}

ParticipantDetailsRepo.java

**package** com.abc.repository;

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

**import** com.abc.model.ParticipantDetails;

**public** **interface** ParticipantDetailsRepo **extends** JpaRepository<ParticipantDetails,Integer>{

}

**QuiztestApplication:**

**package** com.abc;

**import** org.springframework.boot.SpringApplication;

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

**import** org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication

@EnableJpaRepositories

**public** **class** QuiztestApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(QuiztestApplication.**class**, args);

}

}

**AuthConfig.java**

**package** com.abc;

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

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

**import** org.springframework.security.config.annotation.web.builders.HttpSecurity;

**import** org.springframework.security.config.annotation.web.builders.WebSecurity;

**import** org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

**import** org.springframework.security.config.annotation.web.configuration.~~WebSecurityConfigurerAdapter~~;

**import** org.springframework.security.core.userdetails.UserDetailsService;

**import** org.springframework.security.crypto.password.~~NoOpPasswordEncoder~~;

**import** org.springframework.security.crypto.password.PasswordEncoder;

@Configuration

@EnableWebSecurity

**public** **class** AuthConfig **extends** ~~WebSecurityConfigurerAdapter~~{

@Autowired

UserDetailsService userDetailsService;

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.httpBasic().and().authorizeRequests()

.antMatchers("/participant/register").permitAll()

.antMatchers("/participant").hasRole("participant")

.antMatchers("/admin").hasRole("admin").anyRequest().authenticated()

.and().formLogin().permitAll().and().logout().invalidateHttpSession(**true**).permitAll().clearAuthentication(**true**);

http

.csrf().disable();

}

@Override

**protected** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception {

auth.userDetailsService(userDetailsService);

}

@Bean

**public** PasswordEncoder passwordEncoder() {

**return** ~~NoOpPasswordEncoder~~.~~getInstance~~();

}

}

**User.java**

**package** com.abc;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

@Entity

**public** **class** User {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

**private** **int** id;

**private** String username;

**private** String password;

**private** String role;

//private boolean active;

**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;

}

**public** String getRole() {

**return** role;

}

**public** **void** setRole(String role) {

**this**.role = role;

}

// public boolean isActive() {

// return active;

// }

// public void setActive(boolean active) {

// this.active = active;

// }

**public** User(**int** id, String username, String password, String role, **boolean** active) {

**super**();

**this**.id = id;

**this**.username = username;

**this**.password = password;

**this**.role = role;

//this.active = active;

}

**public** User() {

**super**();

// **TODO** Auto-generated constructor stub

}}

**UserRepository.java:**

**package** com.abc;

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

**public** **interface** UserRepository **extends** JpaRepository<User,Integer>{

User findByUsername(String username);

}

**MyUserDetailsService.java**

**package** com.abc;

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

**import** org.springframework.security.core.userdetails.UserDetails;

**import** org.springframework.security.core.userdetails.UserDetailsService;

**import** org.springframework.security.core.userdetails.UsernameNotFoundException;

**import** org.springframework.stereotype.Service;

@Service

**public** **class** MyUserDetailsService **implements** UserDetailsService {

@Autowired

UserRepository uRepo;

@Override

**public** UserDetails loadUserByUsername(String username) **throws** UsernameNotFoundException {

User user=uRepo.findByUsername(username);

**if**(user==**null**) {

**throw** **new** UsernameNotFoundException("not found");

}

**return** **new** MyUserDetailsImp(user);

}

}

**MyUserDetailsImp:**

**package** com.abc;

**import** java.util.Collection;

**import** java.util.Collections;

**import** org.springframework.security.core.GrantedAuthority;

**import** org.springframework.security.core.authority.SimpleGrantedAuthority;

**import** org.springframework.security.core.userdetails.UserDetails;

**public** **class** MyUserDetailsImp **implements** UserDetails {

**private** User user;

**public** MyUserDetailsImp(User user){

**super**();

**this**.user=user;

}

@Override

**public** Collection<? **extends** GrantedAuthority> getAuthorities() {

// **TODO** Auto-generated method stub

**return** Collections.*singleton*(**new** SimpleGrantedAuthority("USER"));

}

@Override

**public** String getPassword() {

// **TODO** Auto-generated method stub

**return** user.getPassword();

}

@Override

**public** String getUsername() {

// **TODO** Auto-generated method stub

**return** user.getUsername();

}

@Override

**public** **boolean** isAccountNonExpired() {

// **TODO** Auto-generated method stub

**return** **true**;

}

@Override

**public** **boolean** isAccountNonLocked() {

// **TODO** Auto-generated method stub

**return** **true**;

}

@Override

**public** **boolean** isCredentialsNonExpired() {

// **TODO** Auto-generated method stub

**return** **true**;

}

@Override

**public** **boolean** isEnabled() {

// **TODO** Auto-generated method stub

**return** **true**;

}

}

**CreationController.java:**

**package** com.abc.controller;

**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.RestController;

**import** com.abc.User;

**import** com.abc.UserRepository;

**import** com.abc.model.ParticipantDetails;

**import** com.abc.repository.ParticipantDetailsRepo;

@RestController

**public** **class** CreationController {

@Autowired

ParticipantDetailsRepo ptRepo;

@Autowired

UserRepository urepo;

@GetMapping("/admin/home")

**public** String adminhome() {

**return** "<h1>Welcome Admin</h1>";

}

@PostMapping("/participant/register")

**public** ResponseEntity<String> registerParticpant(@RequestBody ParticipantDetails pd) {

**try** {

ptRepo.save(pd);

String uname=pd.getpName();

String pass=pd.getPassword();

User user=**new** User();

user.setUsername(uname);

user.setPassword(pass);

user.setRole("participant");

urepo.save(user);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>("Failed",HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

**return** **new** ResponseEntity<>("success",HttpStatus.***CREATED***);

}

@PostMapping("/admin/createadmin")

**public** ResponseEntity<String>createAdmin(@RequestBody User user) {

**try** {

urepo.save(user);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>("Failed",HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

**return** **new** ResponseEntity<>("success",HttpStatus.***CREATED***);

}

@PutMapping("/admin/updateadmin/{id}")

**public** ResponseEntity<String>updateAdmin(@PathVariable ("id") Integer id,@RequestBody User user) {

**try** {

User u=urepo.~~getById~~(id);

u.setUsername(user.getUsername());

u.setPassword(user.getPassword());

u.setRole(user.getRole());

urepo.save(u);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>("Failed",HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

**return** **new** ResponseEntity<>("success",HttpStatus.***CREATED***);

}

}

**QuestionController.java:**

**package** com.abc.controller;

**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;

**import** com.abc.model.Question;

**import** com.abc.repository.QuestinonRepository;

@RestController

**public** **class** QuestionController {

@Autowired

QuestinonRepository qrepo;

@RequestMapping("/")

**public** String home() {

**return** "<h1>Home</h1>";

}

@PostMapping("/admin/question")

**public** ResponseEntity<Question> createQuestion(@RequestBody Question qus) {

**try** {

qrepo.save(qus);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null** ,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

**return** **new** ResponseEntity<>(qus ,HttpStatus.***CREATED***);

}

@GetMapping("/question/{questionId}")

**public** ResponseEntity<Question> findQuestionById(@PathVariable ("questionId") Long questionId ){

**try** {

Question qus=qrepo.~~getById~~( questionId);

**return** **new** ResponseEntity<>(qus,HttpStatus.***FOUND***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@GetMapping("/admin/question")

**public** ResponseEntity<List<Question>> allQuestion(@RequestBody Question qus) {

**try** {

List<Question> qlist=qrepo.findAll();

**return** **new** ResponseEntity<>(qlist,HttpStatus.***FOUND***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@PutMapping("/question/{questionId}")

**public** ResponseEntity<Question> updateQuestionById(@PathVariable ("questionId") Long questionId ,@RequestBody Question newQuestion){

**try** {

Question qus=qrepo.~~getById~~( questionId);

qus.setQuestionAnswer(newQuestion.getQuestionAnswer());

qus.setQuestionText(newQuestion.getQuestionText());

qrepo.save(qus);

**return** **new** ResponseEntity<>(qus,HttpStatus.***OK***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

}

**QuizController.java:**

**package** com.abc.controller;

**import** java.util.ArrayList;

**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.RestController;

**import** com.abc.model.ParticipantDetails;

**import** com.abc.model.Question;

**import** com.abc.model.QuestionToParticipant;

**import** com.abc.model.Quiz;

**import** com.abc.model.Statistics;

**import** com.abc.repository.ParticipantDetailsRepo;

**import** com.abc.repository.QuestinonRepository;

**import** com.abc.repository.QuizRepository;

@RestController

**public** **class** QuizController {

@Autowired

QuizRepository qzRepo;

@Autowired

ParticipantDetailsRepo ptRepo;

@Autowired

QuestinonRepository qusRepo;

@GetMapping("/admin/quizhome")

**public** String quizHome() {

**return** "<h1>Quiz Home</h1>";

}

@PostMapping("/admin/quiz")

**public** ResponseEntity<Quiz> createQuiz(@RequestBody Quiz qz){

**try** {

qzRepo.save(qz);

**return** **new** ResponseEntity<>(qz,HttpStatus.***CREATED***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@GetMapping("/admin/quiz")

**public** ResponseEntity<List<Quiz>> allQuiz(){

**try** {

List <Quiz> qzList=qzRepo.findAll();

**return** **new** ResponseEntity<>(qzList,HttpStatus.***OK***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@PutMapping("/admin/quiz/{id}")

**public** ResponseEntity<Quiz> update(@PathVariable ("id") Long id,@RequestBody Quiz qz){

**try** {

Quiz q=qzRepo.~~getById~~(id);

q.setQuestionList(qz.getQuestionList());

q.setQuizName(q.getQuizName());

**return** **new** ResponseEntity<>(q,HttpStatus.***ACCEPTED***);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null** ,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@GetMapping("admin/quiz/{id}")

**public** ResponseEntity<Quiz> quizbyid(@PathVariable("id") Long id){

**try** {

Quiz qz=qzRepo.~~getById~~(id);

**return** **new** ResponseEntity<>(qz,HttpStatus.***FOUND***);

}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***NOT\_FOUND***);

}

}

@GetMapping("/participant/questionByQuizId/{id}")

**public** ResponseEntity<List<QuestionToParticipant>> questionbyquiz(@PathVariable("id") Long id) {

**try** {

Quiz qz=qzRepo.~~getById~~(id);

List<Question> ql=qz.getQuestionList();

List<QuestionToParticipant> lqtp=**new** ArrayList<>();

**for**(Question q:ql) {

QuestionToParticipant qtp=**new** QuestionToParticipant();

qtp.setId(q.getQuestionId());

qtp.setQusText(q.getQuestionText());

lqtp.add(qtp);

}

**return** **new** ResponseEntity<>(lqtp,HttpStatus.***OK***);

}**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@GetMapping("/admin/stat")

**public** ResponseEntity<Statistics> stat() {

**try** {

Statistics st=**new** Statistics();

st.setQuizNumber(qzRepo.count());

st.setParticipantNumber(ptRepo.count());

st.setQuestionNumber(qusRepo.count());

**return** **new** ResponseEntity<>(st,HttpStatus.***OK***) ;}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***OK***);

}

}

}

**ParticipantController.java:**

**package** com.abc.controller;

**import** java.util.ArrayList;

**import** java.util.List;

//import org.aspectj.weaver.patterns.TypePatternQuestions.Question;

**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.RequestParam;

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

**import** com.abc.model.CheckAnswer;

**import** com.abc.model.PAnswer;

**import** com.abc.model.Question;

**import** com.abc.model.Quiz;

**import** com.abc.model.QuizToParticipant;

**import** com.abc.repository.QuestinonRepository;

**import** com.abc.repository.QuizRepository;

@RestController

**public** **class** ParticipantController {

@Autowired

QuizRepository qzRepo;

@Autowired

QuestinonRepository qusRepo;

@PostMapping("/participation/{quizId}")

**public** ResponseEntity<List<CheckAnswer>> participation(@PathVariable("quizId") Long quizId, @RequestBody ArrayList<PAnswer> pa) {

**try** {

List<CheckAnswer> cAList=**new** ArrayList<>();

**for**(PAnswer p:pa) {

CheckAnswer cA=**new** CheckAnswer();

Question q1=qusRepo.~~getById~~((**long**) p.getQlid());

cA.setQusId(p.getQlid());

cA.setQusText(q1.getQuestionText());

cA.setCorrectAnswer(q1.getQuestionAnswer());

cA.setYourAnswer(p.getAns());

String an1=p.getAns();

String ans2=q1.getQuestionAnswer();

**if**(an1.equals(ans2)) {

cA.setCorrect(**true**);

}

**else** {

cA.setCorrect(**false**);

}

cAList.add(cA);

}

**return** **new** ResponseEntity<>(cAList,HttpStatus.***OK***);}**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***OK***);

}

}

@GetMapping("/participant/getAllQuiz")

**public** ResponseEntity<List<QuizToParticipant>> getAllQuiz() {

**try** {

List<QuizToParticipant> qp=**new** ArrayList<>();

List<Quiz> ql=qzRepo.findAll();

**for**(Quiz qz:ql) {

QuizToParticipant pp=**new** QuizToParticipant();

pp.setId((**int**) qz.getQuizId());

pp.setName(qz.getQuizName());

qp.add(pp);

}

**return** **new** ResponseEntity<>(qp,HttpStatus.***OK***);}

**catch**(Exception e) {

**return** **new** ResponseEntity<>(**null**,HttpStatus.***INTERNAL\_SERVER\_ERROR***);

}

}

@GetMapping("/standing")

**public** **void** standing(){

}

}