**Online Quiz Portal Using REST APIs**

**UserRegister.java**

package com.simplilearn.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class UserRegister

{

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

private int id;

private String name;

private String email;

private String password;

private String country;

private long phoneno;

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;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public long getPhoneno() {

return phoneno;

}

public void setPhoneno(long phoneno) {

this.phoneno = phoneno;

}

public UserRegister(int id, String name, String email, String password, String country, long phoneno) {

super();

this.id = id;

this.name = name;

this.email = email;

this.password = password;

this.country = country;

this.phoneno = phoneno;

}

public UserRegister() {

// TODO Auto-generated constructor stub

}}

**UserRegisterRepo.java**

package com.simplilearn.demo;

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

import org.springframework.stereotype.Repository;

@Repository

public interface UserRegisterRepo extends JpaRepository<UserRegister, Integer> {

}

**UserRegisterService.java**

package com.simplilearn.demo;

import java.util.List;

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

import org.springframework.stereotype.Service;

@Service

public class UserRegisterService {

@Autowired

private UserRegisterRepo repo;

//add method or CREATE RECORD

public UserRegister addUser(UserRegister u) {

return repo.save(u);

}

//List user Method

public List<UserRegister> getAllUser(){

return repo.findAll();

}

}

**UserRegisterController.java**

package com.simplilearn.demo;

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.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/userregister")

public class UserRegisterController {

@Autowired

private UserRegisterService service;

//create new record

@PostMapping("/")

public ResponseEntity<UserRegister> addUser(@RequestBody UserRegister u){

UserRegister user= service.addUser(u);

if(user!=null)

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

else

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

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<UserRegister> getAllUser(){

return service.getAllUser();

}

}

**User.java**

package com.simplilearn.demo;

public class User

{

private String name;

private String password;

private String role;

public User(String name, String password, String role) {

this.name = name;

this.password = password;

this.role = role;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

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;

}

}

**SpringSecurityConfig.java**

package com.simplilearn.demo;

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.configuration.WebSecurityConfigurerAdapter;

@Configuration

public class SpringSecurityConfig extends WebSecurityConfigurerAdapter {

// Protecting the urls with a role-based access.

@Override

protected void configure(HttpSecurity http) throws Exception {

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

.antMatchers("/").permitAll()

.antMatchers("/protected").hasRole("USER")

.antMatchers("/admin").hasRole("ADMIN");

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.authenticationProvider(new CustomAuthenticationProvider());

}

}

**CustomAuthenticationProvider.java**

package com.simplilearn.demo;

import org.springframework.security.authentication.\*;

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

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

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

public class CustomAuthenticationProvider implements AuthenticationProvider {

List<User> dummyUsers = new ArrayList<>();

public CustomAuthenticationProvider() {

dummyUsers.add(new User("kavi", "123", "ROLE\_USER"));

dummyUsers.add(new User("admin", "pass", "ROLE\_ADMIN"));

}

@Override

public Authentication authenticate(Authentication authentication) throws AuthenticationException {

String name = authentication.getName();

String password = authentication.getCredentials().toString();

//jdk 8 -- stream

Optional<User> authenticatedUser = dummyUsers.stream().filter(

user -> user.getName().equals(name) && user.getPassword().equals(password)

).findFirst();

if(!authenticatedUser.isPresent()){

throw new BadCredentialsException("Some Text");

}

List<GrantedAuthority> authorities = new ArrayList<>();

authorities.add(new SimpleGrantedAuthority(authenticatedUser.get().getRole()));

Authentication auth = new UsernamePasswordAuthenticationToken(name, password, authorities);

return auth;

}

@Override

public boolean supports(Class<?> aClass) {

return aClass.equals(UsernamePasswordAuthenticationToken.class);

}

}

**MainController.java**

package com.simplilearn.demo;

import org.springframework.web.bind.annotation.RequestMapping;

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

@RestController

public class MainController {

@RequestMapping("/")

public String hello(){

return "Hello Welcome to this QUIZ competition";

}

@RequestMapping("/protected")

public String protectedHello(){

return "Hey hi are you ready for the competition";

}

@RequestMapping("/admin")

public String admin(){

return "Hello admin here you can see all the participants details";

}

}

**QuizCompetition.java**

package com.simplilearn.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class QuizCompetition {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private int quizId;

private int questionId;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getQuizId() {

return quizId;

}

public void setQuizId(int quizId) {

this.quizId = quizId;

}

public int getQuestionId() {

return questionId;

}

public void setQuestionId(int questionId) {

this.questionId = questionId;

}

public QuizCompetition(int id, int quizId, int questionId) {

super();

this.id = id;

this.quizId = quizId;

this.questionId = questionId;

}

public QuizCompetition() {

// TODO Auto-generated constructor stub

}

}

**QuizRepo.java**

package com.simplilearn.demo;

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

import org.springframework.stereotype.Repository;

@Repository

public interface QuizRepo extends JpaRepository<QuizCompetition, Integer> {

UserRegister save(UserRegister u);

}

**QuizController.java**

package com.simplilearn.demo;

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.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;

///create new record

@PostMapping("/")

public ResponseEntity<QuizCompetition> addUser(@RequestBody QuizCompetition u){

QuizCompetition user= service.addUser(u);

if(user!=null)

return new ResponseEntity<QuizCompetition>(user,HttpStatus.CREATED);

else

return new ResponseEntity<QuizCompetition>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<QuizCompetition> getAllUser(){

return service.getAllUser();

}}

**QuizService.java**

package com.simplilearn.demo;

import java.util.List;

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

import org.springframework.stereotype.Service;

@Service

public class QuizService {

@Autowired

private QuizRepo repo;

//add method or CREATE RECORD

public QuizCompetition addUser(QuizCompetition u) {

return repo.save(u);

}

//List user Method

public List<QuizCompetition> getAllUser(){

return repo.findAll();

}

}

**Questions.java**

package com.simplilearn.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Questions {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String question;

public String getQuestion() {

return question;

}

public void setQuestion(String question) {

this.question = question;

}

private String OptionA;

private String OptionB;

private String OptionC;

private String OptionD;

private String Answer;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getOptionA() {

return OptionA;

}

public void setOptionA(String optionA) {

OptionA = optionA;

}

public String getOptionB() {

return OptionB;

}

public void setOptionB(String optionB) {

OptionB = optionB;

}

public String getOptionC() {

return OptionC;

}

public void setOptionC(String optionC) {

OptionC = optionC;

}

public String getOptionD() {

return OptionD;

}

public void setOptionD(String optionD) {

OptionD = optionD;

}

public String getAnswer() {

return Answer;

}

public void setAnswer(String answer) {

Answer = answer;

}

public Questions(int id, String question, String optionA, String optionB, String optionC, String optionD,

String answer) {

this.id = id;

this.question = question;

OptionA = optionA;

OptionB = optionB;

OptionC = optionC;

OptionD = optionD;

Answer = answer;

}

public Questions() {

// TODO Auto-generated constructor stub

}

}

**QuestionRepo.java**

package com.simplilearn.demo;

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

import org.springframework.stereotype.Repository;

@Repository

public interface QuestionRepo extends JpaRepository<Questions, Integer> {

}

**QuestionService.java**

package com.simplilearn.demo;

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 Questions addUser(Questions u) {

return repo.save(u);

}

//List user Method

public List<Questions> getAllUser(){

return repo.findAll();

}

//get user by id

public Questions getUserById(int id) {

if(repo.findById(id).isPresent())

return repo.findById(id).get();

else

return null;

}

//update user by id

public Questions updateUser(Questions user, int id) {

if(repo.findById(id).isPresent())

{

Questions old= repo.findById(id).get();

old.setOptionA(user.getOptionA());

old.setOptionB(user.getOptionB());

old.setOptionC(user.getOptionC());

old.setOptionD(user.getOptionD());

old.setAnswer(user.getAnswer());

return repo.save(old);

}

else

{

return null;

}

}

///DELETE USER BY ID

public boolean deleteUser(int id) {

if(repo.findById(id).isPresent())

{

repo.deleteById(id);

return true;

}

else

return false;

}}

QuestionController.java

package com.simplilearn.demo;

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.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;

@RestController

@RequestMapping("/api/questions")

public class QuestionController {

@Autowired

private QuestionService service;

///create new record

@PostMapping("/")

public ResponseEntity<Questions> addUser(@RequestBody Questions u){

Questions user= service.addUser(u);

if(user!=null)

return new ResponseEntity<Questions>(user,HttpStatus.CREATED);

else

return new ResponseEntity<Questions>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<Questions> getAllUser(){

return service.getAllUser();

}

//get user by id

@GetMapping("/{id}")

public ResponseEntity<Questions> getUserById(@PathVariable int id){

Questions user= service.getUserById(id);

if(user!=null)

return new ResponseEntity<Questions>(user,HttpStatus.FOUND);

else

return new ResponseEntity<Questions>(user,HttpStatus.NOT\_FOUND);

}

//update user by id

@PutMapping("/{id}")

public ResponseEntity<Object> updateUser(@RequestBody Questions user,@PathVariable int id){

Questions data= service.updateUser(user, id);

if(data!=null)

return new ResponseEntity<Object>(data,HttpStatus.OK);

else

return new ResponseEntity<Object>("User is Not Available",HttpStatus.NOT\_FOUND);

}

///DELETE USER BY ID

@DeleteMapping("/{id}")

public ResponseEntity<Object> deleteUser(@PathVariable int id ){

if(service.deleteUser(id))

return new ResponseEntity<Object>("User Deleted", HttpStatus.OK);

else

return new ResponseEntity<Object>("No User Found",HttpStatus.NOT\_FOUND);

}}

**UserAnswer.java**

package com.simplilearn.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.ForeignKey;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

//@Data

public class UserAnswer {

//@Table(name = "users", uniqueConstraints = {@UniqueConstraint(columnNames = {"username"})})

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

// @GeneratedValue(strategy = GenerationType.FOREIGNKEY)

private int id;

private int quizid;

private String questionid;

private String useranswer;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getQuizid() {

return quizid;

}

public void setQuizid(int quizid) {

this.quizid = quizid;

}

public String getQuestionid() {

return questionid;

}

public void setQuestionid(String questionid) {

this.questionid = questionid;

}

public String getUseranswer() {

return useranswer;

}

public void setUseranswer(String useranswer) {

this.useranswer = useranswer;

}

public UserAnswer(int id, int quizid, String questionid, String useranswer) {

super();

this.id = id;

this.quizid = quizid;

this.questionid = questionid;

this.useranswer = useranswer;

}

public UserAnswer() {

// TODO Auto-generated constructor stub

}

}

**UserAnswerRepo.java**

package com.simplilearn.demo;

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

import org.springframework.stereotype.Repository;

@Repository

public interface UserAnswerRepo extends JpaRepository<UserAnswer, Integer> {

}

**UserAnswerService.java**

package com.simplilearn.demo;

import java.util.List;

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

import org.springframework.stereotype.Service;

@Service

public class UserAnswerService {

@Autowired

private UserAnswerRepo repo;

//add method or CREATE RECORD

public UserAnswer addUser(UserAnswer u) {

return repo.save(u);

}

//List user Method

public List<UserAnswer> getAllUser(){

return repo.findAll();

}

}

**UserAnswerController.java**

package com.simplilearn.demo;

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.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/useranswer")

public class UserAnswerController {

@Autowired

private UserAnswerService service;

//create new record

@PostMapping("/")

public ResponseEntity<UserAnswer> addUser(@RequestBody UserAnswer u){

UserAnswer user= service.addUser(u);

if(user!=null)

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

else

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

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<UserAnswer> getAllUser(){

return service.getAllUser();

}}

**UserQuizStoreResponse.java**

package com.simplilearn.demo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class UserQuizStoreResponse {

@Id

@Column(name="id")

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String response;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getResponse() {

return response;

}

public void setResponse(String response) {

this.response = response;

}

public UserQuizStoreResponse(int id, String response) {

super();

this.id = id;

this.response = response;

}

public UserQuizStoreResponse() {

// TODO Auto-generated constructor stub

}

}

**UserQuizStoreResponseRepo.java**

package com.simplilearn.demo;

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

import org.springframework.stereotype.Repository;

@Repository

public interface UserQuizStoreResponseRepo extends JpaRepository<UserQuizStoreResponse, Integer> {

}

**UserQuizStoreResponseService.java**

package com.simplilearn.demo;

import java.util.List;

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

import org.springframework.stereotype.Service;

@Service

public class UserQuizStoreResponseService {

@Autowired

private UserQuizStoreResponseRepo repo;

//add method or CREATE RECORD

public UserQuizStoreResponse addUser(UserQuizStoreResponse u) {

return repo.save(u);

}

//List user Method

public List<UserQuizStoreResponse> getAllUser(){

return repo.findAll();

}}

**UserQuizStoreResponseController.java**

package com.simplilearn.demo;

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.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/userquizstoreresponse")

public class UserQuizStoreResponseController {

@Autowired

private UserQuizStoreResponseService service;

///create new record

@PostMapping("/")

public ResponseEntity<UserQuizStoreResponse> addUser(@RequestBody UserQuizStoreResponse u){

UserQuizStoreResponse user= service.addUser(u);

if(user!=null)

return new ResponseEntity<UserQuizStoreResponse>(user,HttpStatus.CREATED);

else

return new ResponseEntity<UserQuizStoreResponse>(user, HttpStatus.INTERNAL\_SERVER\_ERROR);

}

//List of Users Or RETRIVE DATA

@GetMapping("/")

public List<UserQuizStoreResponse> getAllUser(){

return service.getAllUser();

}

}

**Mysql Query**

CREATE DATABASE OnlineQuiz;

use OnlineQuiz;

CREATE TABLE Questions (

id INT AUTO\_INCREMENT PRIMARY KEY,

question VARCHAR(255),

OptionA VARCHAR(255),

OptionB VARCHAR(255),

OptionC VARCHAR(255),

OptionD VARCHAR(255),

Answer VARCHAR(255)

);

CREATE TABLE User (

name VARCHAR(255),

password VARCHAR(255),

role VARCHAR(255)

);

CREATE TABLE UserAnswer (

id INT AUTO\_INCREMENT PRIMARY KEY,

quizid INT,

questionid VARCHAR(255),

useranswer VARCHAR(255)

);

CREATE TABLE UserRegister (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255),

password VARCHAR(255),

country VARCHAR(255),

phoneno BIGINT

);