***Front End With AngularJS and Angular Material***

* ***Download Node JS -***

[Download | Node.js (nodejs.org)](https://nodejs.org/en/download/)

* ***Command to install angular cli -***

npm install -g @angular/cli

* ***Command to install angular material -***

npm install -g @angular/material

* ***Create Angular***
* ***App -***

ng new examfront

* ***Command to run Angular App -***

ng serve

* ***Components -***

***1. components -*** Folder for Reusable components

***a. Navbar Component -***

***Command -*** ng g c Navbar

***b. Footer Component -***

***Command -*** ng g c Footer

***2. pages -*** Component folder which we are going to access using URL

***a. admin -*** Folder for admin panel components.

* ***add-category -*** Component for adding new category.

***Command -*** ng g c add-category

* ***add-question -*** Component for adding new Questions under different quizzes.

***Command -*** ng g c add-question

* ***add-quiz -*** Component for adding new quiz under different categories.

***Command -*** ng g c add-quiz

* ***dashboard -*** Component for admin dashboard page.

***Command -*** ng g c dashboard

* ***sidebar -*** Component for creating admin panel sidebar.

***Command -*** ng g c sidebar

* ***update-question -*** Component for updating question if any thing got wrong

***Command -*** ng g c update-question

* ***update-quiz -*** Component for updating quiz if any thing got wrong

***Command -*** ng g c update-quiz

* ***view-categories -*** Component for displaying all the categories available in the database.

***Command -*** ng g c view-categories

* ***view-quiz-questions -*** Component for displaying all the questions available in the database corresponding to specific quizzes.

***Command -*** ng g c view-quiz-questions

* ***view-quizzes -*** Component for displaying all the quizzes available in the database.

***Command -*** ng g c view-quizzes

* ***welcome -*** This is the admin landing page after login.

***Command -*** ng g c welcome

***b. Home -*** This is the main landing component of our site.

***Command -*** ng g c Home

***c. login -*** This is the component for the login page for admin and user.

***Command -*** ng g c login

***d. signup -*** This is the component for user registration page

***Command -*** ng g c signup

***e. user -*** Folder for user panel components.

* ***instruction -*** component for display quiz specific instructions.

***Command -*** ng g c instruction

* ***load-quiz -*** Component for displaying all the quizzes for users.

***Command -*** ng g c load-quiz

* ***sidebar -*** Component for creating user panel sidebar.

***Command -*** ng g c sidebar

* ***start -*** component for displaying all the questions for specific quiz and show them result after submission.

***Command -*** ng g c start

* ***dashboard -*** component for display sidebar and using router-outlet

***Command -*** ng g c dashboard

***3. services -***

***a. admin.guard.ts -*** Typescript file for authentication of admin role.

***b. auth.interceptor.ts -*** Typescript file for user authentication.

***c. category.service.ts -*** Typescript file for managing all category rest api.

***d. helper.ts -*** In this typescript file we have defined base url.

***e. login.service.ts -*** Typescript file for login functions.

***f. question.service.ts -*** Typescript file for managing all question rest api.

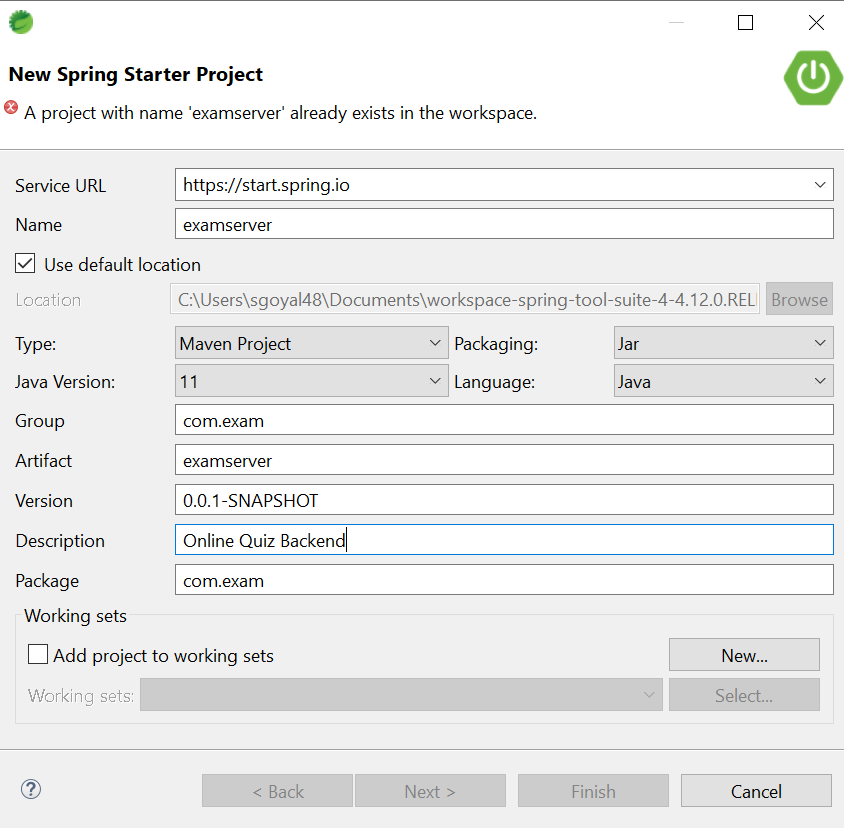
***g. quiz.service.ts -*** Typescript file for managing all quizzes rest api.

***h. user.service.ts -*** Typescript file for user add rest api.

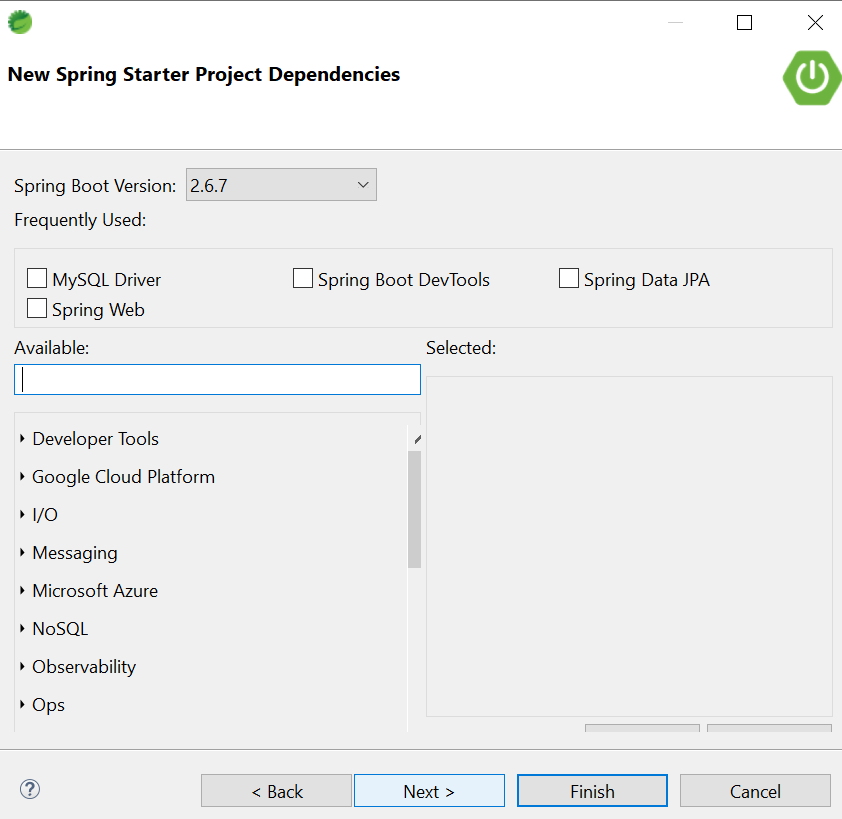
* ***app-routing.module.ts -*** In this typescript file all the routing of the project takes place.
* ***app.module.ts -*** In this typescript fill import of all the user defined components and angular material components takes place.

***Back End With Spring Boot***

* File -> New -> Spring Starter Project ->



* -> New Spring Starter Project Dependency



* -> Select Spring Data JPA, Spring Boot DevTools, Spring Web and Oracle Driver Dependency -> Finish
* Project is ready

1. ***src/main/java***

***a. com.exam (Package)***

* ***ExamserverApplication.java (Class file)-*** It is the .main file for spring boot app.

***b. com.exam.config (Package)***

* ***JwtAuthenticationEntryPoint.java (Class file) -***
* ***JwtAuthenticationFilter.java (Class file) -***
* ***JwtUtils.java (Class fie) -***
* ***MySecurityConfig.java (Class file) -***

***c. com.exam.controller (Package) -***

* ***AuthenticationController.java (Class file) -*** In this class file we define API's with GET, PUT, POST, DELETE, PATCH with client for authentication.
* ***CategoryController.java (Class file) -*** In this class file we define API's with GET, PUT, POST, DELETE, PATCH with client for category.
* ***QuestionController.java (Class file) -*** In this class file we define API's with GET, PUT, POST, DELETE, PATCH with client for question.
* ***QuizController.java (Class file) -*** In this class file we define API's with GET, PUT, POST, DELETE, PATCH with client for quiz.
* ***UserController.java (Class file) -*** In this class file we define API's with GET, PUT, POST, DELETE, PATCH with client for user.

***d. com.exam.helper (Package)***

* ***UserFoundException.java (Class file) -*** It will throw exception if user already found
* ***UserNotFoundException.java (Class file) -*** It will throw exception if user is not found

***e. com.exam.model (Package)***

* ***Authority.java (Class file) -*** This class file contains all the database related content like variable, getter and setter methods.
* ***JwtRequest.java (Class file) -*** This class file contains getter and setter methods for username and password.
* ***JwtResponse.java (Class file) -*** This class file is a response of JwtRequest.java file it will send a token for username and password.
* ***Role.java (Class file) -*** This class file contains getter and setter methods for user roleId and roleName.
* ***User.java (Class file) -*** This class file contains user related setter and getter methods.

***f. com.exam.model.exam (Package)***

* ***Category.java (Class file) -*** This class file contains category related methods.
* ***Question.java (Class file) -*** This class file contains questions related to getter and setter methods such as getQuestion and setQuestion etc.
* ***Quiz.java (Class file) -*** This class file contains quiz related methods.

***g. com.exam.repo (Package)***

* ***CategoryRepository.java (Class file) -***
* ***QuestionRepository.java (Class file) -***
* ***QuizRepository.java (Class file) -***
* ***RoleRepository.java (Class file) -***
* ***UserRepository.java (Class file) -***

***h. com.exam.service (Package)***

* ***CategoryService.java (Interface) -*** Contains methods such as addCategory(), updateCategory() etc.
* ***QuestionService.java (Interface) -*** Contains methods such as addQuestion(), updateQuestion() etc.
* ***QuizService.java (Interface) -*** Contains methods such as addQuiz(), updateQuiz() etc.
* ***UserService.java (Interface) -*** Contains methods such as addUser(), updateUser() etc.

***i. com.exam.service.impl (Package)***

* ***CategoryServiceImpl.java (class) -*** This is the implementation class of the methods defined in categoryService.java.
* ***QuestionServiceImpl.java (Class) -*** This is the implementation class of the methods defined in QuestionService.java.
* ***QuizServiceImpl.java (Class) -*** This is the implementation class of the methods defined in QuizService.java.
* ***UserServiceImpl.java (Class) -*** This is the implementation class of the methods defined in UserService.java.

***2. src/main/resources***

***a. application.properties -*** This file contains Database configuration, JPA Configuration.

***3. pom.xml -*** This file contains all the dependencies.

* Right click on examserver(Project) -> maven -> update project.
* Right click on examserver(Project) -> run as -> Spring Boot App

***Database: MySQL***

* ***Download MySQL -***

[***https://www.mysql.com/downloads/***](https://www.mysql.com/downloads/)

* ***Command for create database***

create database database\_name;

* ***Command for change database***

use database\_name;

* ***Command for display table data***

Select \* from table\_name;