**OnlineQuiz**

**Index:**

**Git URL:**  [**https://github.com/RajaRajeswari12/P4OnlineQuizProj.git**](%20https://github.com/RajaRajeswari12/P4OnlineQuizProj.git%20)

Angular Components: QuizClient

1. Header Component
2. get-quiz Component
3. index-page Component
4. login Component
5. review-result Component
6. view-quiz Component

Angular Service: QuizClient

1. loginRouting.service.ts
2. quiz-service.service.ts

Angular Models:

1. User.ts
2. QuizQuestion.ts
3. quizAnswer.ts
4. quiz.ts

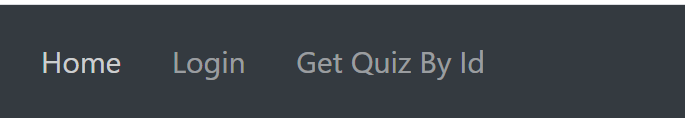
Server Side:

1. **Servlets [com.onlineQuiz.OnlineQuizRest.controller]**
   1. QuizController.java
2. **Repositories [com.onlineQuiz.OnlineQuizRest.repository]**
   1. AnswerRepository.java
   2. QuestionRepository.java
   3. QuizRepository.java
   4. UserRepository.java
3. **Models[com.onlineQuiz.OnlineQuizRest.models]**
   1. Answer.java
   2. Question.java
   3. Quiz.java
   4. User.java

Angular Components: QuizClient

1. **Header Component:**

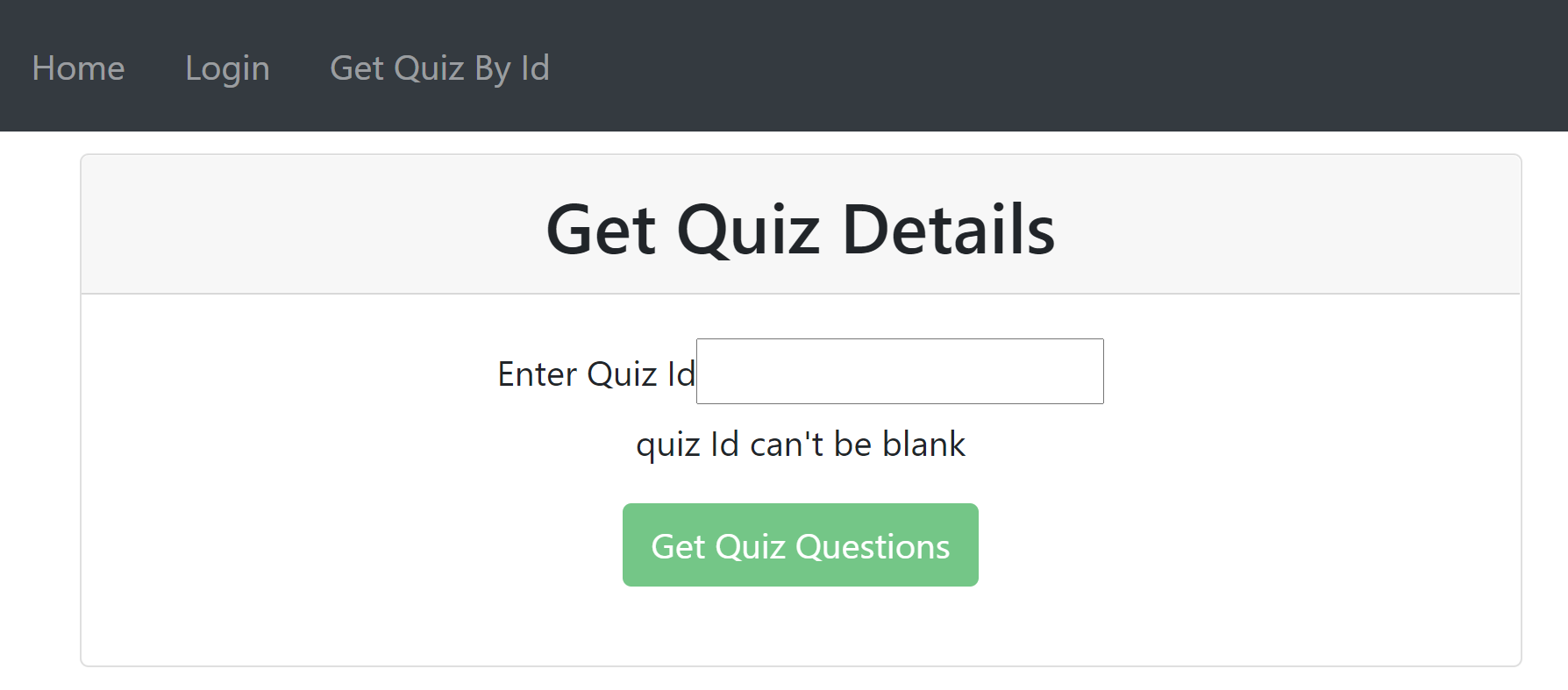
Header html file present in this component contains navigation links like Home, Login and GetQuiz



1. **get-quiz Component:**

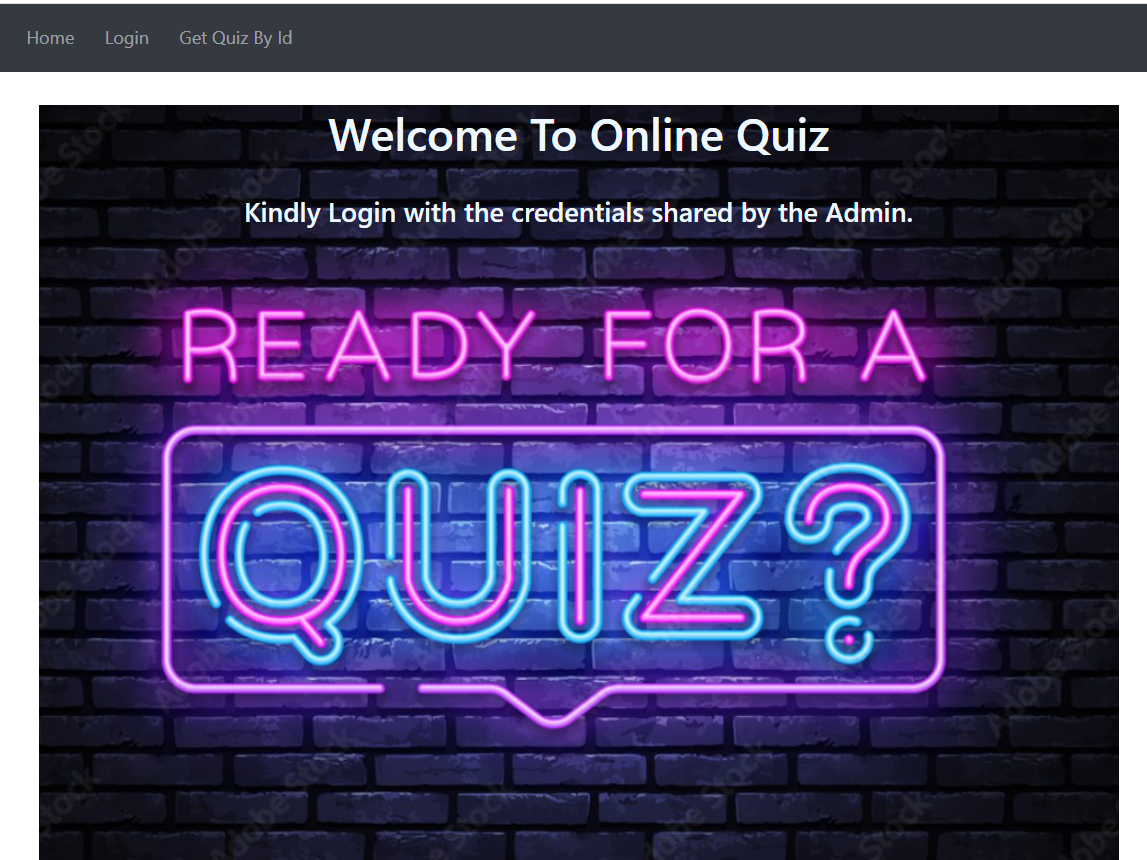
This component contains the html and ts file to handle the get Quiz Page. In the getQui page the validations like quizId cannot be blank, Invalid quizId, ets are handled.

On entering the correct quiz id, It navigates to the viewQuiz Page.

****

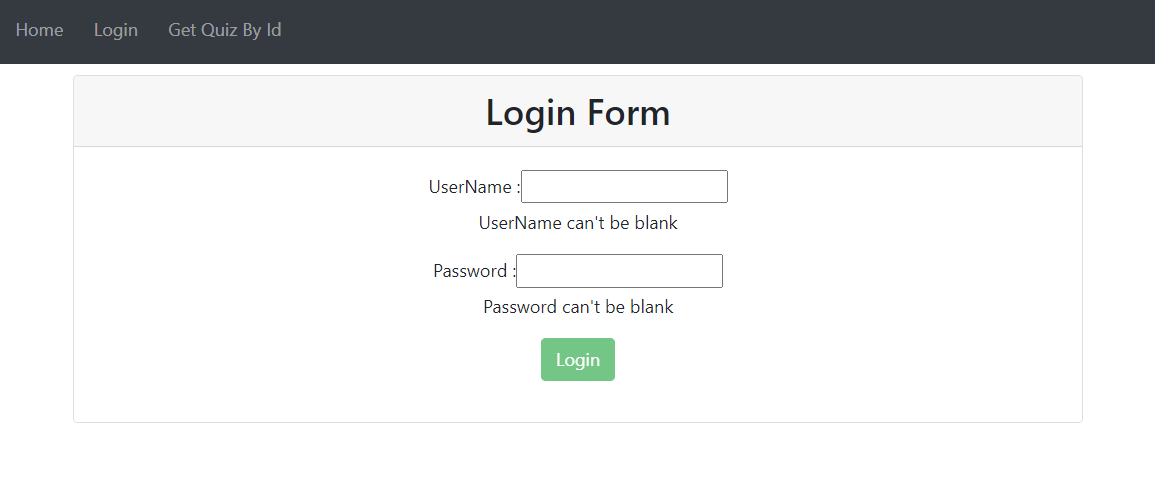
1. **index-page Component**

This component is used to display the index page.

****

1. **login Component**

This component handles the login part. The User has to login with the credentials shared. Validations like Username can’t be blank, password can’t be blank and invalid login credentials are handled. On successful login the user enters the GetQuiz page.

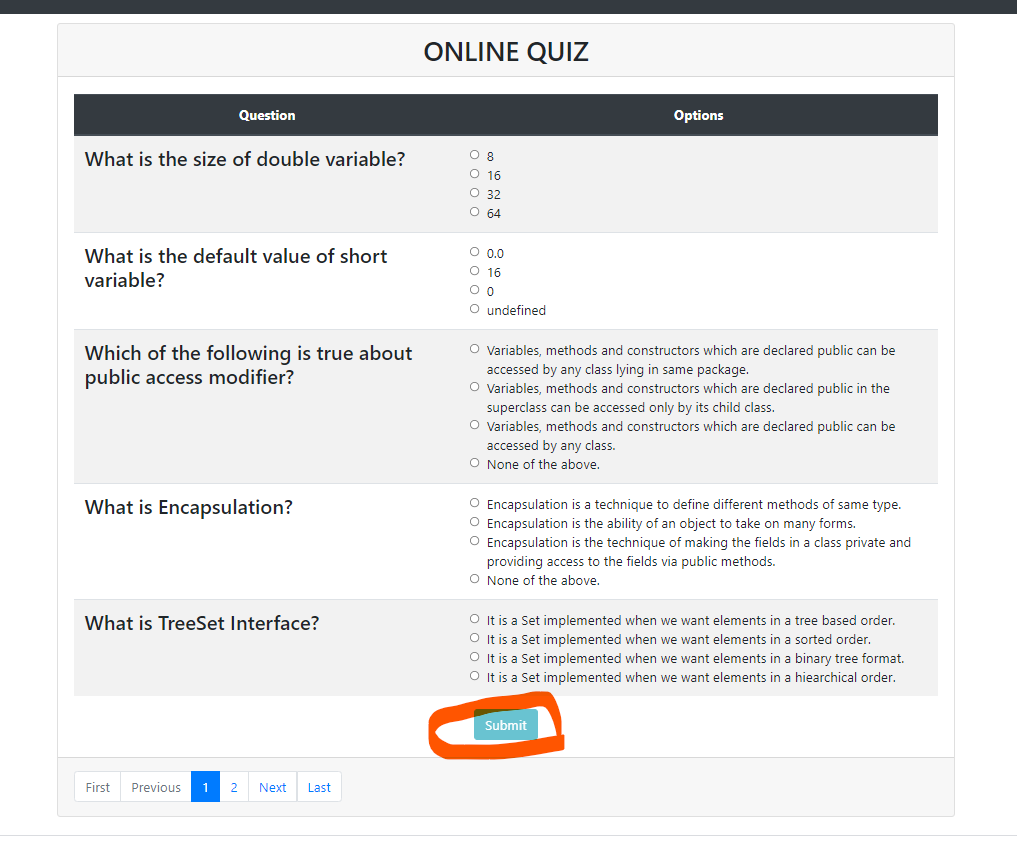


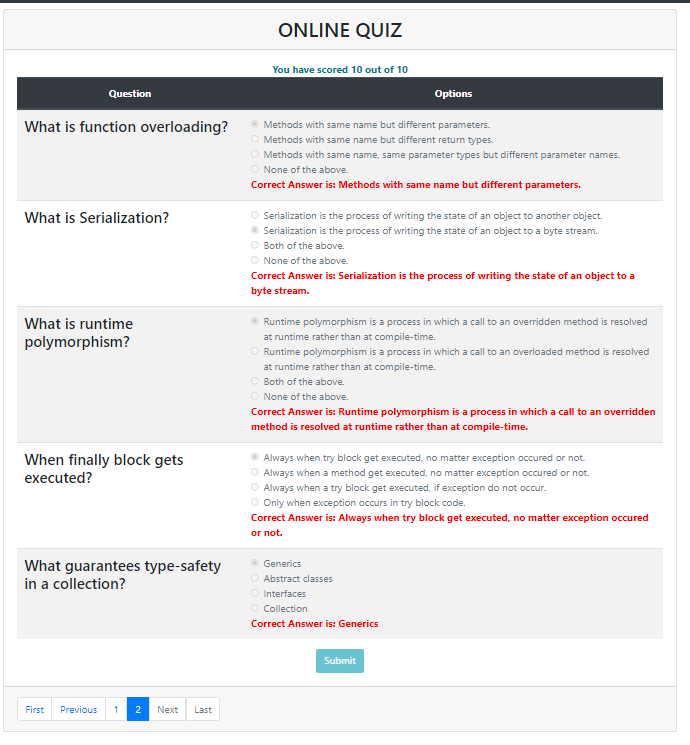
1. **review-result Component**

This component just loads the quiz answer and matches the correct answer with the questions.

1. **view-quiz Component**

User lands in this page on entering the shared quizId correctly. Here validations like submit button will be disabled state till the user answers all the question are handled. On Clicking the submit button, the results are displayed on the spot with the correct answer at the bottom of every question.





Angular Service: QuizClient

1. **loginRouting.service.ts**

This service is used to route the links and it also helps to work the application as SinglePageApplication.

1. **quiz-service.service.ts**

This service is used to contact the rest point and get the required data for each page through HttpClient.

Angular Models:

1. User.ts
2. QuizQuestion.ts
3. quizAnswer.ts
4. quiz.ts

Server Side:

1. **Servlets [com.onlineQuiz.OnlineQuizRest.controller]**
   1. QuizController.java

The Rest Points for retrieving the quiz,Retrieving the quizAnswer , Retrieving the user login details are handled by getMapping. Post Mapping for loading the qui questions and answer were also handled here

1. **Repositories [com.onlineQuiz.OnlineQuizRest.repository]**
   1. AnswerRepository.java
   2. QuestionRepository.java
   3. QuizRepository.java
   4. UserRepository.java
2. **Models[com.onlineQuiz.OnlineQuizRest.models]**
   1. Answer.java
   2. Question.java
   3. Quiz.java
   4. User.java

Data To Be Populated

**create database onlinequiz;**

**use onlinequiz;**

**select \* from quiz;**

**select \* from question;**

**select \* from answer;**

**insert into quiz(quiz\_id,quiz\_name) values(1,'Java');**

**insert into user(userid,user\_name,password) values(1,'Default','12345678');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (1,1,"What is the size of double variable?",'8 bit','16 bit','32 bit','64 bit');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (2,1,"What is the default value of short variable?",'0.0','16','0','undefined');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (3,1," Which of the following is true about public access modifier?",**

**'Variables, methods and constructors which are declared public can be accessed by any class lying in same package.',**

**'Variables, methods and constructors which are declared public in the superclass can be accessed only by its child class.',**

**'Variables, methods and constructors which are declared public can be accessed by any class.',**

**'None of the above.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (4,1," What is Encapsulation?",**

**'Encapsulation is a technique to define different methods of same type.',**

**'Encapsulation is the ability of an object to take on many forms.',**

**'Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods.',**

**'None of the above.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (5,1," What is TreeSet Interface?",**

**'It is a Set implemented when we want elements in a tree based order.',**

**'It is a Set implemented when we want elements in a sorted order.',**

**'It is a Set implemented when we want elements in a binary tree format.',**

**'It is a Set implemented when we want elements in a hiearchical order.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (6,1,"What is function overloading?",**

**'Methods with same name but different parameters.',**

**'Methods with same name but different return types.',**

**'Methods with same name, same parameter types but different parameter names.',**

**'None of the above.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (7,1,"What is Serialization?",**

**'Serialization is the process of writing the state of an object to another object.',**

**'Serialization is the process of writing the state of an object to a byte stream.',**

**'Both of the above.',**

**'None of the above.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (8,1," What is runtime polymorphism?",**

**'Runtime polymorphism is a process in which a call to an overridden method is resolved at runtime rather than at compile-time.',**

**'Runtime polymorphism is a process in which a call to an overloaded method is resolved at runtime rather than at compile-time.',**

**'Both of the above.',**

**'None of the above.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (9,1," When finally block gets executed?",**

**'Always when try block get executed, no matter exception occured or not.',**

**'Always when a method get executed, no matter exception occured or not.',**

**'Always when a try block get executed, if exception do not occur.',**

**'Only when exception occurs in try block code.');**

**insert into question(quest\_id,quiz\_id,quiz\_question,answer\_option1,answer\_option2,answer\_option3,answer\_option4)**

**values (10,1,"What guarantees type-safety in a collection?",**

**'Generics',**

**'Abstract classes',**

**'Interfaces',**

**'Collection');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(1,1,'64 bit');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(2,2,'0.0');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(3,3,'Variables, methods and constructors which are declared public can be accessed by any class.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(4,4,'Encapsulation is the technique of making the fields in a class private and providing access to the fields via public methods.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(5,5,'It is a Set implemented when we want elements in a sorted order.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(6,6,'Methods with same name but different parameters.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(7,7,'Serialization is the process of writing the state of an object to a byte stream.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(8,8,'Runtime polymorphism is a process in which a call to an overridden method is resolved at runtime rather than at compile-time.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(9,9,'Always when try block get executed, no matter exception occured or not.');**

**insert into answer(ans\_id,quest\_id,quiz\_answer)**

**values(10,10,'Generics');**