Phase 3 Project

Online Quiz Portal Using REST API's

Create a new database "quiz" in mysql.

Create the following tables in quiz database.

1) CREATE TABLE `login_session` (

```
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `access_token` varchar(100) NOT NULL,
  `last_access` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2)CREATE TABLE `question` (

```
  `id` int(11) NOT NULL,
  `question` varchar(1000) NOT NULL,
  `option_a` varchar(45) NOT NULL,
  `option_b` varchar(45) NOT NULL,
  `option_c` varchar(45) NOT NULL,
  `option_d` varchar(45) NOT NULL,
  `right_option` varchar(1) NOT NULL,
  `deleted` varchar(1) NOT NULL DEFAULT 'N',
  `created_on` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

3)CREATE TABLE `quiz` (

```
  `id` int(11) NOT NULL,
  `title` varchar(45) NOT NULL,
  `category` varchar(45) NOT NULL,
  `deleted` varchar(1) NOT NULL DEFAULT 'N',
  `created_on` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4)CREATE TABLE `quiz_question` (

```
  `id` int(11) NOT NULL,
  `quiz_id` int(11) NOT NULL,
  `question_id` int(11) NOT NULL,
  `deleted` varchar(1) NOT NULL DEFAULT 'N',
  `created_on` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
          5)CREATE TABLE `user` (
  `id` int(11) NOT NULL,
  `first_name` varchar(45) NOT NULL,
  `last_name` varchar(45) NOT NULL,
  `email_id` varchar(255) NOT NULL,
  `password` varchar(45) NOT NULL,
  `mobile_number` varchar(10) NOT NULL,
  `is_admin` varchar(1) NOT NULL DEFAULT 'N',
  `created_on` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
          6)CREATE TABLE `user_quiz_ques_ans` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `quiz_question_id` int(11) NOT NULL,
  `selected_option` char(1) NOT NULL,
  `created_on` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Add the constraints to tables as per the requirements.

Coding part :

- Now in eclipse create 3 spring boot projects for different rest api’s.

1) Auth service

2) Admin service

3) Quiz service

1) Auth service :

- This project is mainly for authentication and login.
- Add required dependencies in pom.xml file
- Create entity classes and data transfer object classes.
- We use jdbc template, write necessary query in repoistory
- In controller class we have to process the requests like create user , login , logout , change password.
- We don’t want to create jsp files for view since we required to handle back-end process.

2) Admin service :

- This project is mainly for admin features.
- Add required dependencies in pom.xml file
- Create entity classes and data transfer object classes.
- We use jdbc template, write necessary query in repoistory
- We need different controller classes for admin features.
- In user controller class we have to process the requests like get profile, update profile.
- In question controller class we have to process the requests like add question, question update, delete question,get all questions.
- In quiz controller class we have to process the requests like create quiz, update quiz, display all quiz, delete quiz.
- In quiz question controller class we have to process the requests like question added to quiz, get question by its id, delete quiz question.

- We don’t want to create jsp files for view since we required to handle back-end process.

3) Quiz service :

- This project is mainly for user features.
- Add required dependencies in pom.xml file
- Create entity classes and data transfer object classes.
- We use jdbc template, write necessary query in repoistory
- In controller class we have to process the requests like explore questions, answering questions,to get next questions and leader board.
- We don’t want to create jsp files for view since we required to handle back-end process.

Postman :

1) We will use postman for checking the requests.
2) Give the url with specified port number givien in application.properties.
3) As per the mapping, the requests will send along with header and body if required.
4) For each requests the mapping will be different and output will displayed in body after the status code is successful.
5) For example after creating the user, we have login.
6) Upon successful login, access token will be generated and we will use it for authorization for up coming requests.

Git link : https://github.com/Priyakumar11/Simplilearn-Projects/tree/main/Phase%203%20Project