**Online Quiz Application**

**Developer Details:**

N Prasad

BTECH (Department of Computer science and Engineering)

BS Abdur Rahman Crescent Institute of science and technology.

**Contact**: [prasadnithi15@gmail.com](mailto:prasadnithi15@gmail.com)

**Source code:**

Git repository link: (frontend)🡪 <https://github.com/prasad-ops/project3>

Git repository link: (backend)🡪 <https://github.com/prasad-ops/project3backend>

**PROBLEM STATEMENT:**

The Online Test Application system creates an application that enables users to provide online tests, review them, and display the results. This system contains three main modules: Quiz, Review, and Result. The quiz section of the online test application accepts the questions in JSON format. The JSON file can be easily shared from the server in the pre-defined format. The application renders the test at client side.

The “Review and display result” section allows users to declare the results immediately. You can simply call another JSON with the answers in it and evaluate and display the results immediately.

**REQUIREMENTS:**

* Eclipse/IntelliJ: An IDE to code for the application
* Java 8: A programming language to develop the prototype
* Git: To connect and push files from the local system to GitHub
* GitHub: To store the application code and track its versions
* Scrum: An efficient agile framework to deliver the product incrementally
* Specification document: Any open-source document or Google Docs
* Spring boot with hibernate.
* Visual studio code with angular installed.
* Browser (chrome).
* H2 database enabled.

**Git Repository links:**

Git repository link: (frontend)🡪 <https://github.com/prasad-ops/project3>

Git repository link: (backend)🡪 <https://github.com/prasad-ops/project3backend>

**ALGORITHM:**

**Backend:**

* Start the program.
* Create a spring boot project with spring web, h2 database, jpa repository.
* An entity class Quiz should be made with annotations to link it with the QUIZ table
* A repository class should then map the entity class to the CrudRepository interface
* Create a REST controller class to create the REST endpoint. It should take in parameters using the POST protocol
* Data received in the REST controller will be then saved into the database.
* Run the backend in the port 1100,
* Stop the program

**Frontend:**

* Start the program.
* Create an angular project with routing enabled.
* Create three components (take quiz, review answer, result).
* Develop an interface by using components.html file.
* Map the functions for those components by using typescript file.
* Provide the url in the router for routing process.
* Serve the program by using ng s - -o
* Stop the program.

**HOW TO EXECUTE:**

**Backend:**

STEP 1:

Download the backend code from the git repository and run the code as **Spring application** default port used is 1100. For changing the port

Server.port=\*\*\*\*

This code is written in application.properties

After executing the code the table is created in h2 database.

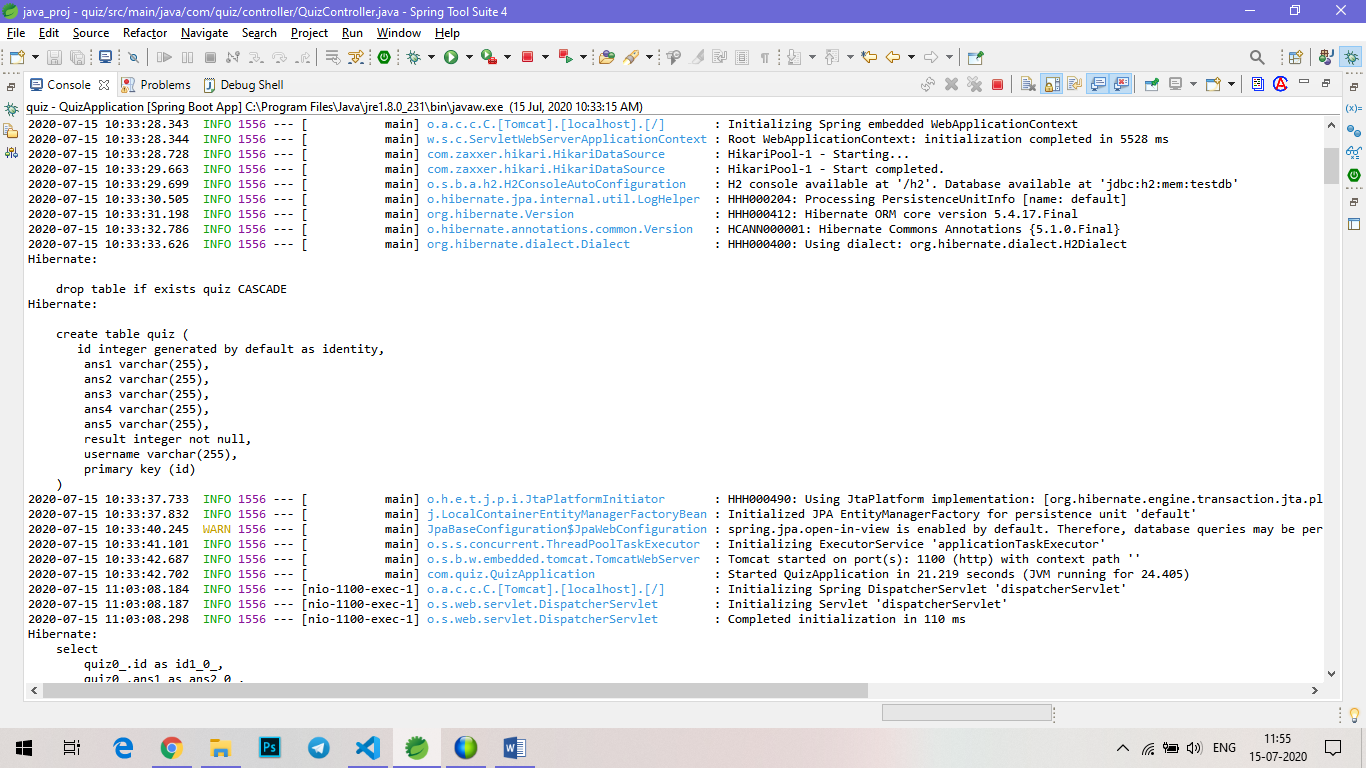


Fig 1. Table creation query

STEP 2:

In-order to view the fields in the database.

Open this URL in the browser - l<http://localhost:1100/h2>

Set JDBC URL as - jdbc:h2:mem:testdb

Then click connect.

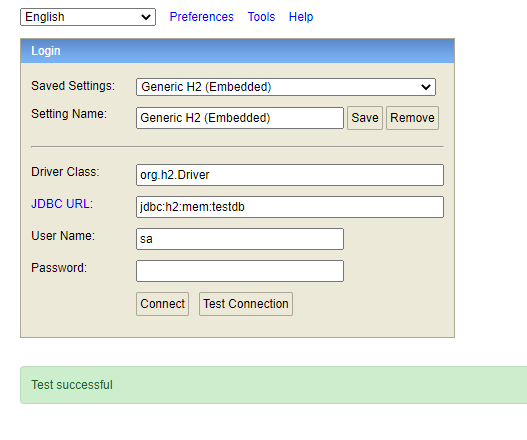


Fig 2. H2 database connection.

STEP 3:

The crud operation can be performed using postman tool.

POST operation – it will create the fields in the form of Jason data and store it into the h2 database.

PUT operation – it will update the data based on primary key (id),

GET operation – it will fetch the data from the database.

DELETE operation – it will delete the data based on primary key (id).

Since h2 database is an embedded database. It will drop the table once the connection is closed. For permanent storage of data, use MSQL database. The same program with change in configuration is required in application.properties.

**Frontend:**

STEP 1:

Download the frontend code from the git repository and open visual studio code.

Open new terminal

Run the code using angular command -- ng s - -o

It will serve the program in the default port 4200.

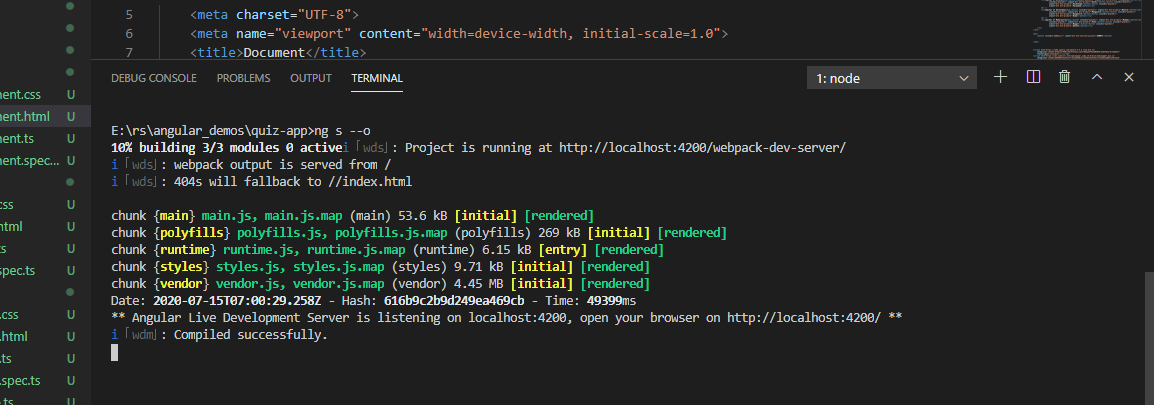


Fig 3. Serve the angular program.

STEP 2:

The output page will be open in the default browser or open chrome and type the url: <http://localhost:4200>

It will launch the app.

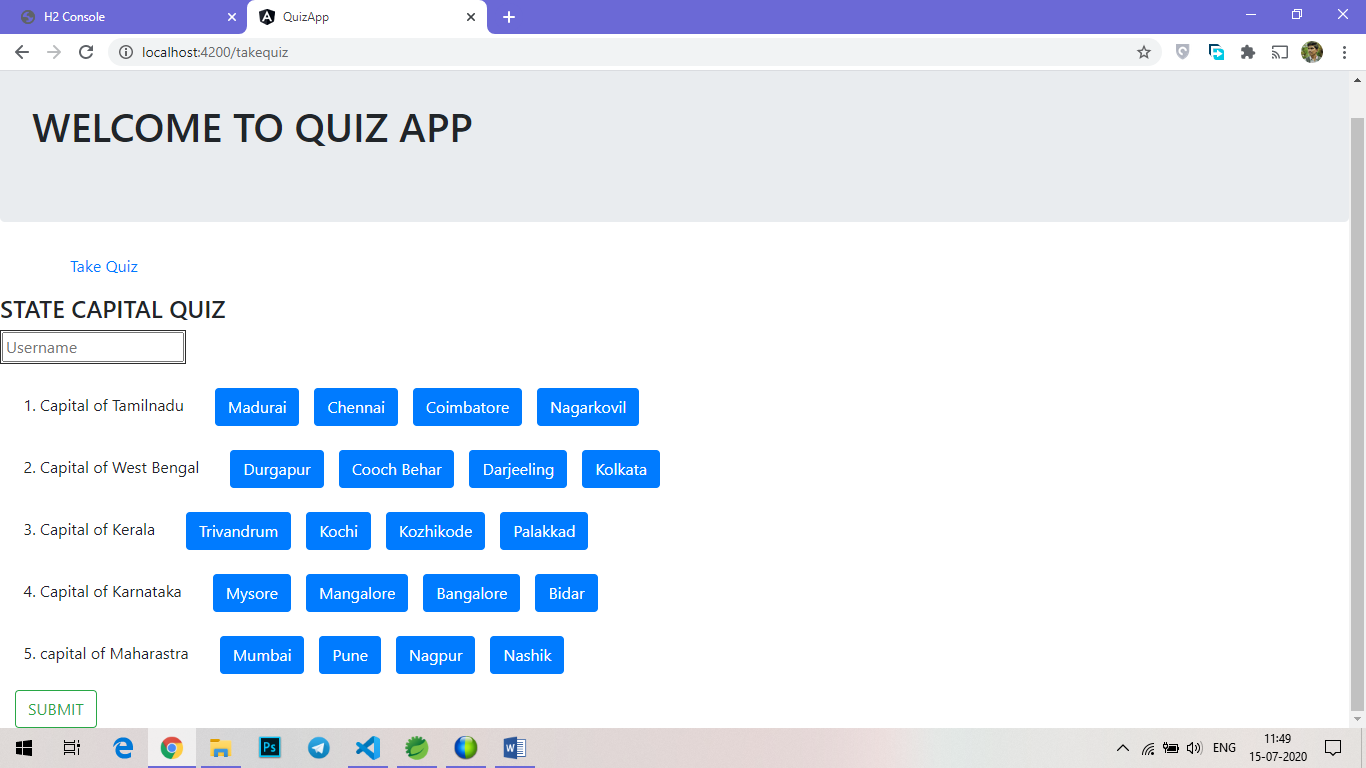


Fig 4. Quiz app(take quiz page)

STEP 3:

After launching the app, enter the username (username must be email id of the participant).

Answer the quiz by clicking the button which has the answer in it. (buttons are used instead of radio button or checkbox, only for design purpose).

After attending all the questions, click on submit for getting result.

It will take to result page. The result page contains the result of all the participant which is displayed in the table format.

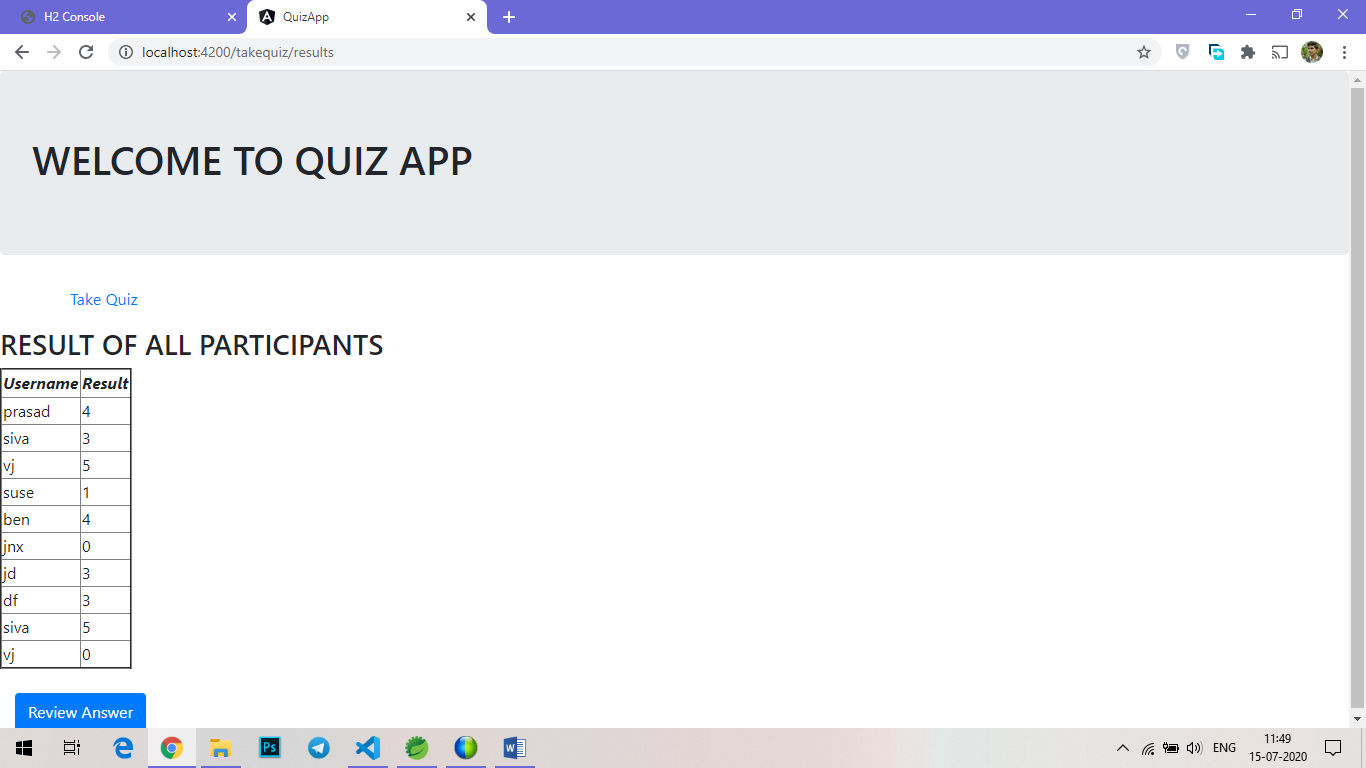


Fig 5. Result page

STEP 4:

Check the backend, that the datas were inserted in the table or not.

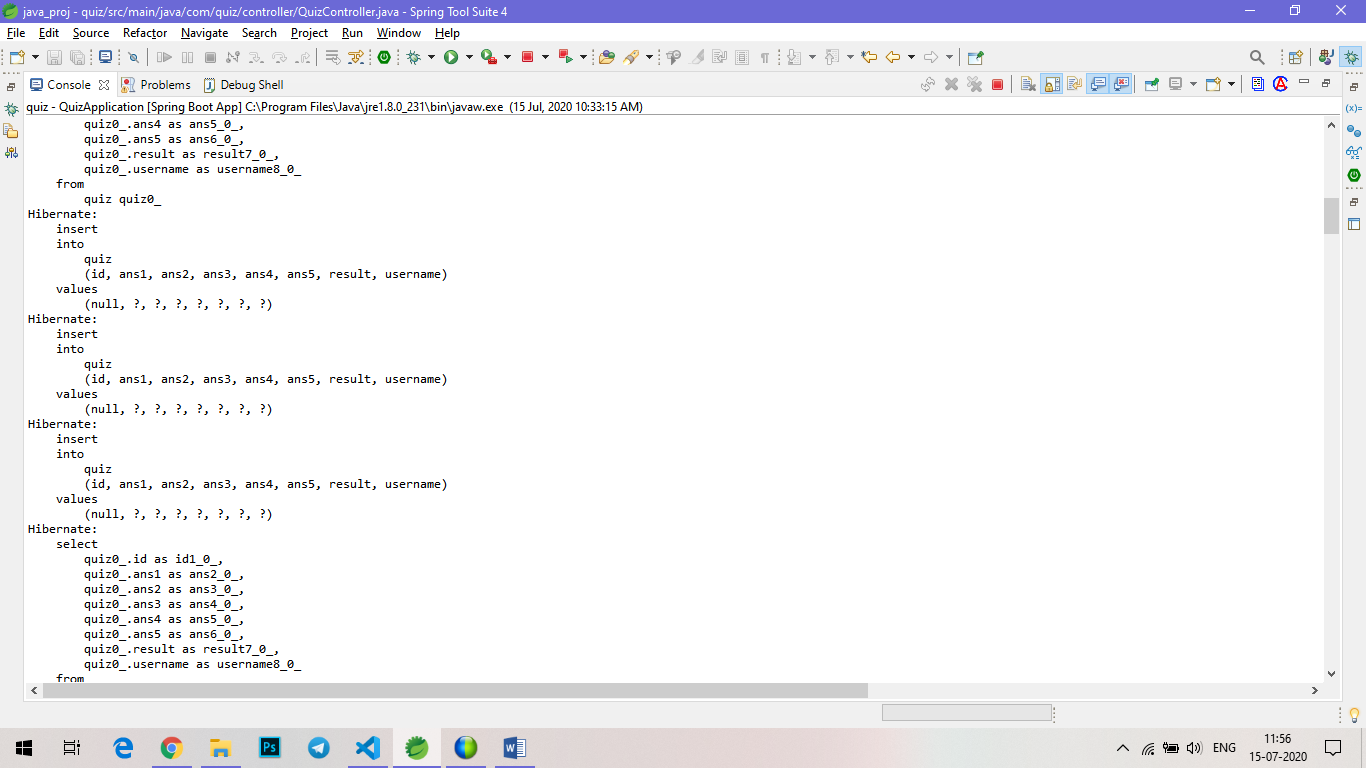


Fig 6. Insert query in spring console.

STEP 5:

By clicking on the Review Answer button, the correct answer for the quiz is displayed.

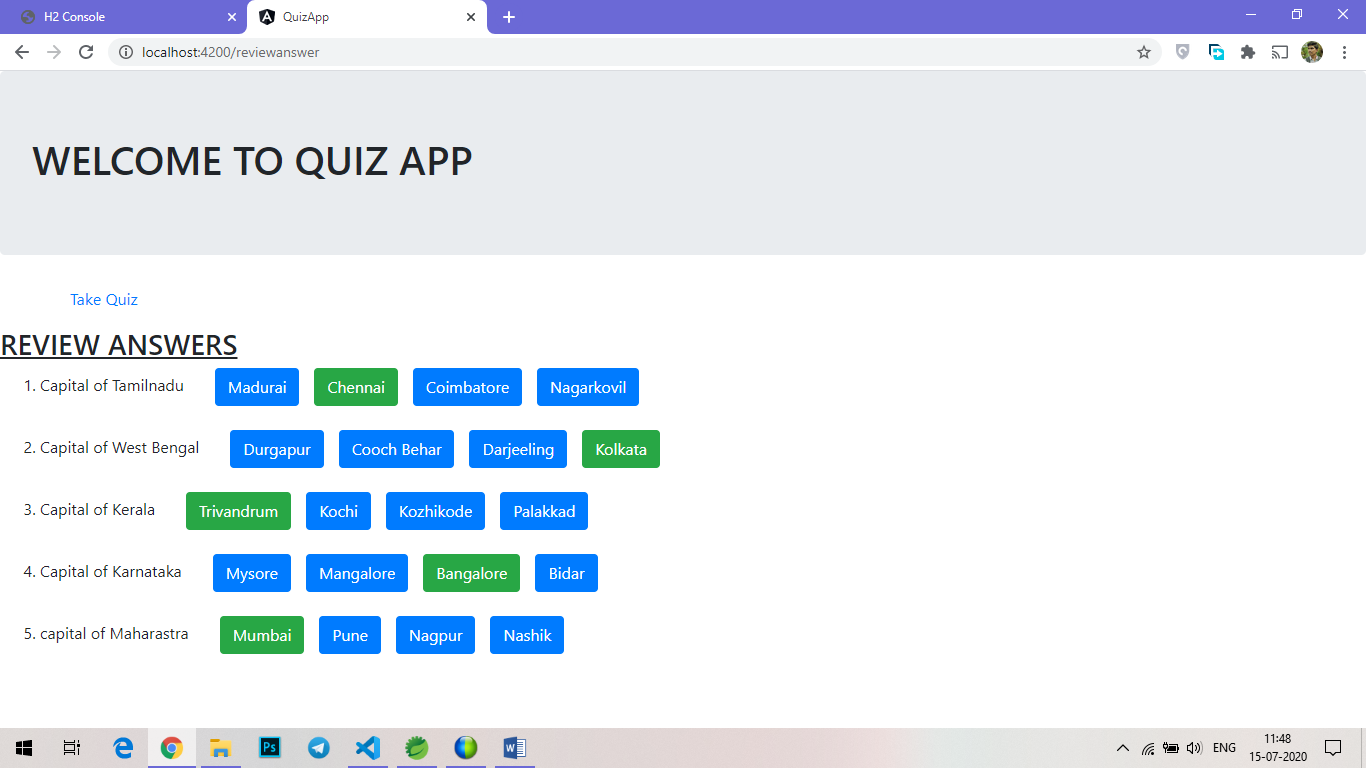


Fig 7. Review Answer page.

STEP 6:

After completing the quiz, you can refer the database. It will display all the user details, answers, and their results.

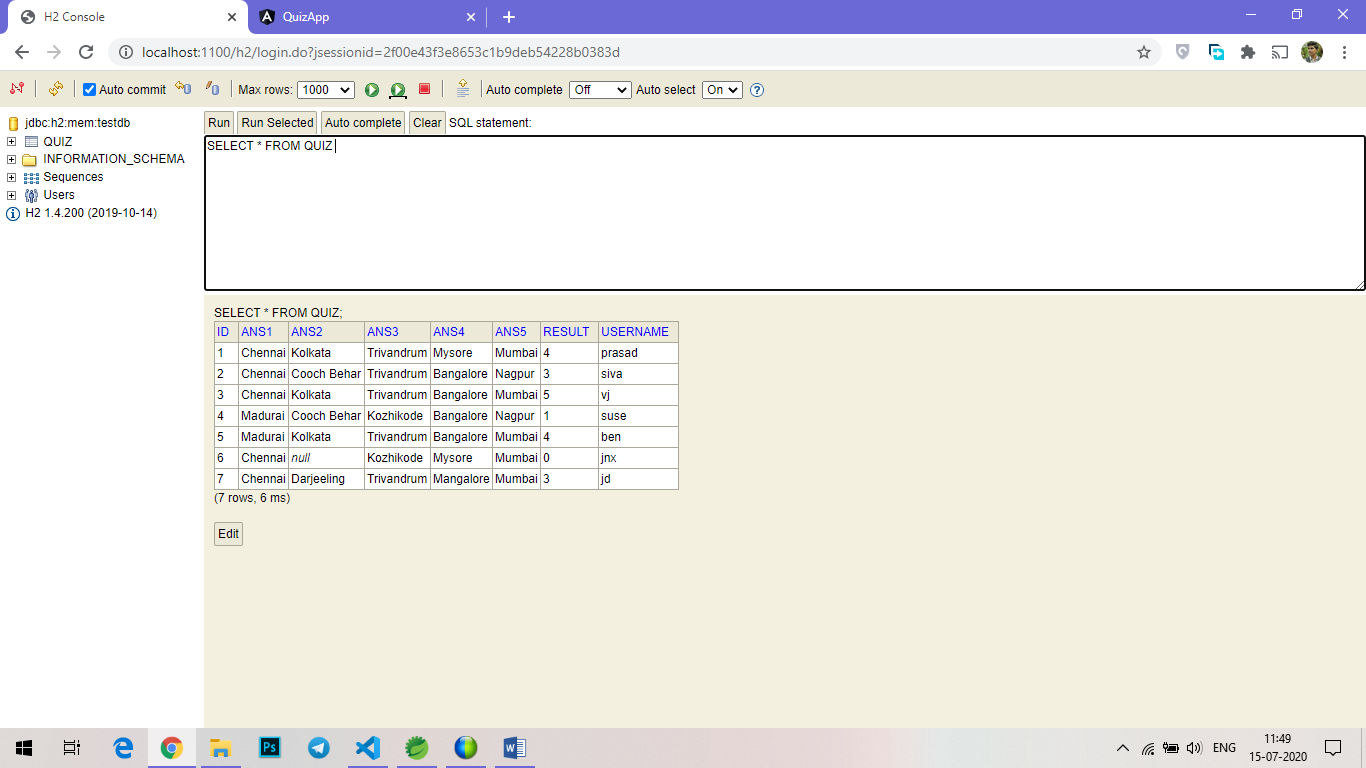


Fig 8. H2 database (QUIZ table)

**CONCLUSION:**

Thus, the mentioned task in the problem statement are executed. The source code is pushed into the git repository. The git link for downloading the source code is given in document. This system contains three main modules: Quiz, Review, and Result. The quiz section of the online test application accepts the questions in JSON format. The JSON file can be easily shared from the server in the pre-defined format. The application renders the test at client side. The “Review and display result” section allows users to declare the results immediately. You can simply call another JSON with the answers in it and evaluate and display the results immediately.