

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

**DEPARTMENT OF INFORMATION SCIENCE AND
ENGINEERING**



“QUIZ APPLICATION”

Submitted to -

AKARSHA D P, SWETHA R, AKSHATHA ULLAL

Submitted by-

SUNAYANA M-1NT19IS165

SUPREETHA G S-1NT19IS166

SWETHA A-1NT19IS172

CERTIFICATE

It is hereby certified that the work which is being presented in the B.E Minor
Project Report entitled "QUIZ APPLICATION" in partial fulfilment of the
requirements for the award of the degree of Bachelor of Engineering and
submitted in the Department of Information Science & Engineering of Nitte
Meenakshi Institute of Technology.

SUNAYANA M

SUPREETHA G S

SWETHA A

1NT19IS165

1NT19IS166

1NT19IS172

This is to certify that the above statement made by the candidate is correct to the
best of my knowledge.

Ms. Akarsha D P

Prof. (Dr.) Mohan S.G

Asst. Prof. ISE

HOD, ISE

Mrs. Swetha R

(Signature of Examiner)

Asst. Prof. ISE

ABSTRACT

ONLINE QUIZ forms the lifeline of the Educational Institutes to the functioning of the Examination. It is very essential for an Institute to handle the Examinations and their results. It is very useful for an Institute to test its students continuously for their mutual development. This system is helpful for conducting (M.C) Multiple Choice Examinations which can be conducted regularly as well as for surprise tests and provides immediate results saving the precious time of faculties to check the papers and prepare mark sheets.

"Quiz Application" is a collection of a number of different types of questions. There will be a limited number of questions and for each correct answer the user will get a credit score. Users can see answers as well as high scores of the quiz. There are many quiz applications available currently on the internet. But there are few Which provide better understanding between users and the application like, providing proper answers. To develop a user friendly quiz application which will contain: Numbers of quiz, Answers to every question and to improve the knowledge level of users. To develop an application which will contain solutions to the above problems. By this application the user will come to know about his/her level and can learn additional knowledge. Also by this application a user can expand his/her knowledge among the world.

TABLE OF CONTENTS

Content	Pg. No.
Certificate	ii
Abstract	iii
Table of Contents	iv
List of Figures	v
Introduction	6
Literature Review	6
Objectives	8
Problem Statement	9
Methodology	9
Algorithm	11
Testing	12
Results and Snapshots	14
Conclusion	16
References	16

LIST OF CONTENTS

Figure Name	Pg. no.
Fig. 1 - MERN Stack	9
Fig. 2 - Use Case	10
Fig. 3 - Entity Diagram	11
Fig. 4 - Testing Snapshot 1	12
Fig. 5 - Testing Snapshot 2	12
Fig. 6 - Testing Snapshot 3	13
Fig. 7 - Testing Snapshot 4	13
Fig. 8 - Result Snapshot 1	14
Fig. 9 - Result Snapshot 2	14
Fig. 10 - Result Snapshot 3	15
Fig. 11 - Result Snapshot 4	15

INTRODUCTION

The ONLINE QUIZ is a web application for taking online tests in an efficient manner and not wasting time for checking the paper. The main objective of ONLINE QUIZ is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves a lot of time but also gives fast results. For students they give questions according to their convenience and time and there is no need to use extra things like paper, pen etc. Online quizzes are a popular form of entertainment for web surfers. Online quiz is set up to actually test knowledge or identify a person's attribute. Some companies use online quizzes as an efficient way of testing a potential hire's knowledge without that candidate needing to travel. This can be used in educational institutions as well as in the corporate world. This Can be used anywhere any time as it is a web based application where user's Location doesn't matter. There is no restriction that examiner has to be present when the candidate takes the test.

LITERATURE REVIEW

Today Developers around the world are making efforts to enhance user experience of using applications as well as to enhance the developer's workflow of designing applications to deliver projects and rollout change requests under strict timeline. Stacks can be used to build web applications in the shortest span of time. The stacks used in web development are basically the response of software engineers to current demands. They have essentially adopted pre-existing frameworks (including JavaScript) to make their lives easier.

While there are many, MEAN and MERN are just two of the popular stacks that have evolved out of JavaScript. Both stacks are made up of open source components and offer an end-to-end framework for building comprehensive web apps that enable browsers to connect with databases. The common theme between the two is JavaScript

and this is also the key benefit of using either stack. One can basically avoid any syntax errors or any confusion by just coding in one programming language, JavaScript. Another advantage of building web projects with MERN is the fact that one can benefit from its enhanced flexibility.

Basic Components of Mern includes:

1. MongoDB
2. Express.JS
3. React
4. Node.JS

1. **MongoDB:** - MongoDB is a cross-platform document-oriented NoSQL database used for high volume data storage that provides high performance, high availability and easy scalability. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time. The document model maps to the objects in the application code, making data easy to work with. The data model available within MongoDB allows users to represent hierarchical relationships, to store arrays, and other more complex structures more easily. MongoDB works on concept of collections and documents. Each database contains collections which in turn contains documents. Each document can have a varying number of fields. The size and content of each document can also be different from each other.

2. **Express.JS:** - Express is a minimal and flexible Node.js. web application framework that provides a robust feature set for developing web and mobile applications. This facilitates the rapid development of node-based web applications. You can configure the middleware to respond to HTTP requests. Defines a routing table used to perform various actions based on HTTP methods and URLs. Allows dynamic rendering of HTML pages based on passing arguments to the template.

3. **React:** - React (ReactJS) is a free open source front-end JavaScript library for building user interfaces based on UI components. React JS is a JavaScript library used in web development to create interactive elements on websites. ReactJS uses downward data flow. This is because it ensures that the smallest of changes that happen in child structure do not affect the parents. When an object is changed by a developer it needs to be made sure that only the particular objects are updated and that just its state is modified. The data is bound together with the object and this structure sees.

4. **Node.JS:** - Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single- page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world. Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009. Node.js applications are written in JavaScript and can be

run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

The Welcome(user profile) will appear and the user has to choose any of two options 'Start Quiz', 'View Score'. After choosing the start quiz, multiple choice questions appear automatically one by one with a specific time limit. User has to choose any of four existing options and then the user has to hit the submit button and each right answer a score of 10 marks is awarded and for each wrong answer or if he/she decides to skip the question a negative 5 marking is done. At the end of the game it will show the score of the user and record the user's name to the score page. If the user fails to answer the questions on time, then the user can either go back to home or retake the quiz again.

OBJECTIVES

- The main objective of “Quiz Application ” is to facilitate a user-friendly environment for all users and reduce the manual effort.
- In the past, quizzes were conducted manually but with further resolution of the technology we are able to generate the score and pose the queries automatically.
- The functional requirements include to create users that participate in the quiz, automatic score and report generation.

PROBLEM STATEMENT

“Our aim is to develop a application for the users in which a user can attempt a quiz and view the high scores of quizz”

METHODOLOGY

Quiz application is built using ReactJs in frontend and ExpressJs , Nodejs and MongoDB in backend.

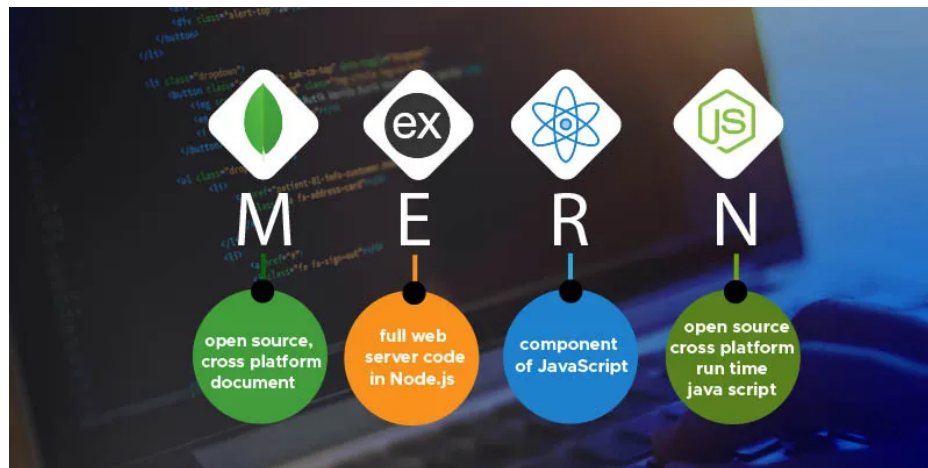


Fig. 1

1. The project is built using MERN Stack i.e; Mongo Db, Express, React and Node JS.
2. The application has a user who can take up a quiz and view his scores.
3. The user can first login to the quiz and answer all the questions.
4. For every right answer he is awarded with 10 marks and for every wrong answer a negative 5 marks is cut.
5. After the quiz, we can join the score page by entering our name.
6. The scores will get arranged in the descending order.
7. First an index page gets opened and from the index page we can either navigate to the quiz page or score page.
8. We can go to the score page and view the scores of all the players who attended the quiz.
9. From the quiz page we can attend the quiz and answer the questions for the quiz a timer of 5 minutes is provided.
10. We are required to answer the quiz in 5 minutes, if we answer the quiz in time we are directed to the score page.
11. We can register ourselves and enter into the score page.
12. If we cannot answer the questions in 5 minutes, we will be directed to a page where it asks us to retake the quiz or return to home since the time for the quiz has expired.
13. After 20 records on the score page , the lower scores will get deleted automatically.
14. For every player, a player's identity is created.
15. For every player's score, the high score is updated in descending order.

16. From the MongoDB atlas we can view the players who registered, we can create more players, edit their names and scores (update), read the scores and delete the player and the scores.
17. Also, new questions can be added, deleted and updated along with the options.
18. Timers can also be managed.
19. The CRUD operations such as addition,update,deletion of new players and questions can be performed and these changes can be read by our project.
20. Web APIs such as .get , .post are used to retrieve and update records. And promises are also used by making use of then catch blocks.

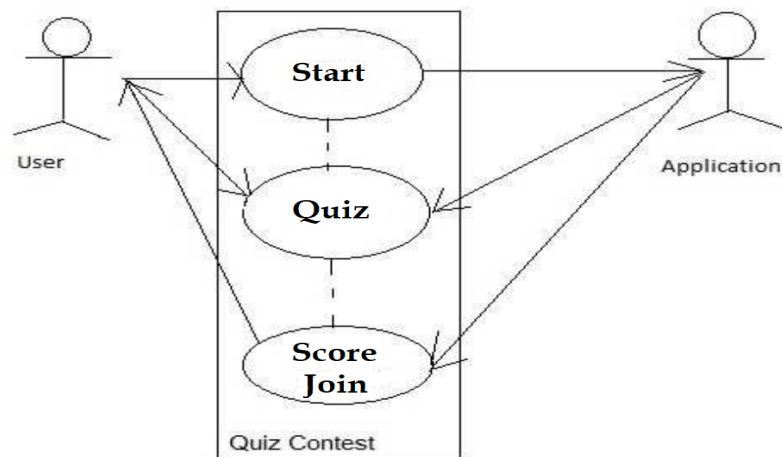


Fig. 2



Fig. 3

ALGORITHM

- Enter into the client directory in the terminal using the command “cd client”.
- Run the client in the terminal using the command “npm start”.
- Enter into the root directory using the command - “cd ..”.
- Run the root in the terminal using the command “npm start”.
- Execution is done using the command - “npm run dev”.
- The quiz application will run in the browser.
- The user can start the quiz.
- Once the user starts the quiz the timer starts.
- For each correct answer his scores will be increased by 10 points and for each wrong answer his scores will be reduced by 5 points.
- When the user finishes the quiz, his scores are displayed. He can join or register.
- The scores of the joined users will be seen in the decreasing order.

GITLINK: <https://github.com/Supreetha-GS/Quiz-Application.git>

TESTING

```

PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master> cd client
PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master\client> npm install
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

removed 1 package, and audited 1418 packages in 4s

169 packages are looking for funding
  run `npm fund` for details

10 vulnerabilities (8 high, 2 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master\client> 

```

Fig. 4

```

PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master\client> cd ..
PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master> npm install
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

up to date, audited 29 packages in 557ms

5 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\suppi\OneDrive\Desktop\QuizAppMain-master> 

```

Fig. 5

```

> concurrently "npm run client" "npm run server"

[0] npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
[0] WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
[1]
[1] > quizappmain@1.0.0 server
[1] > cd Backend && npm run dev
[1]
[0] > quizappmain@1.0.0 client
[0] > cd client && npm run start
[0]
[1] npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
[0] npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
[1]
[1] > backend@1.0.0 dev
[1] > nodemon index
[1]
[0] > client@0.1.0 start
[0] > react-scripts start
[0]
[1] [nodemon] 2.0.15
[1] [nodemon] to restart at any time, enter `rs`
[1] [nodemon] watching path(s): *.*
[1] [nodemon] watching extensions: js,mjs,json
[1] [nodemon] starting `node index index.js`
[1] Server Started
[1] connection established
[0] (node:7632) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' is deprecated. Use the 'setupMiddlewares' option.
[0] (Use `node --trace-deprecation ...` to show where the warning was created)
[0] (node:7632) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' is deprecated. Use the 'setupMiddlewares' option.
[0] Starting the development server...
[0]
[0] Compiled successfully!

```

Fig. 6

```

[0]
[0] You can now view client in the browser.
[0]
[0] Local:      http://localhost:3000
[0] On Your Network: http://172.20.10.4:3000
[0]
[0] Note that the development build is not optimized.
[0] To create a production build, use npm run build.
[0]
[0] assets by path static/ 2.08 MiB
[0]   assets by path static/js/*.js 2.02 MiB
[0]     asset static/js/bundle.js 2.01 MiB [emitted] (name: main) 1 related asset
[0]     asset static/js/node_modules_web-vitals_dist_web-vitals_js.chunk.js 6.92 KiB [emitted] 1 related asset
[0]   assets by path static/media/ 62.5 KiB
[0]     asset static/media/icon.e776eeb5ec820b07ffd0.png 60.9 KiB [emitted] [immutable] [from: src/components/Navbar/icon.png] (auxiliary name: main)
[0]     asset static/media/background.095f476a8f688579620d.svg 1.66 KiB [emitted] [immutable] [from: src/components/addRecord/images/background.svg] (auxiliary name: main)
[0] asset index.html 1.78 KiB [emitted]
[0] asset asset-manifest.json 622 bytes [emitted]
[0] cached modules 1.77 MiB (javascript) 123 KiB (asset) 31.4 KiB (runtime) [cached] 224 modules
[0] webpack 5.70.0 compiled successfully in 3804 ms

```

Fig. 7

RESULT AND SNAPSHOTS

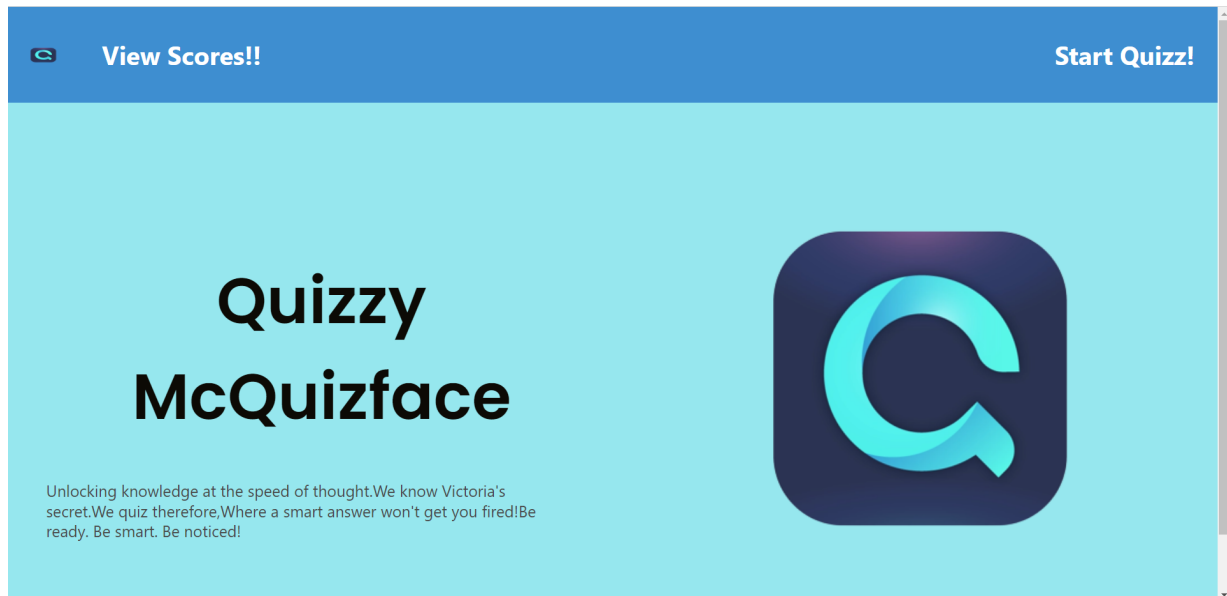


Fig. 8

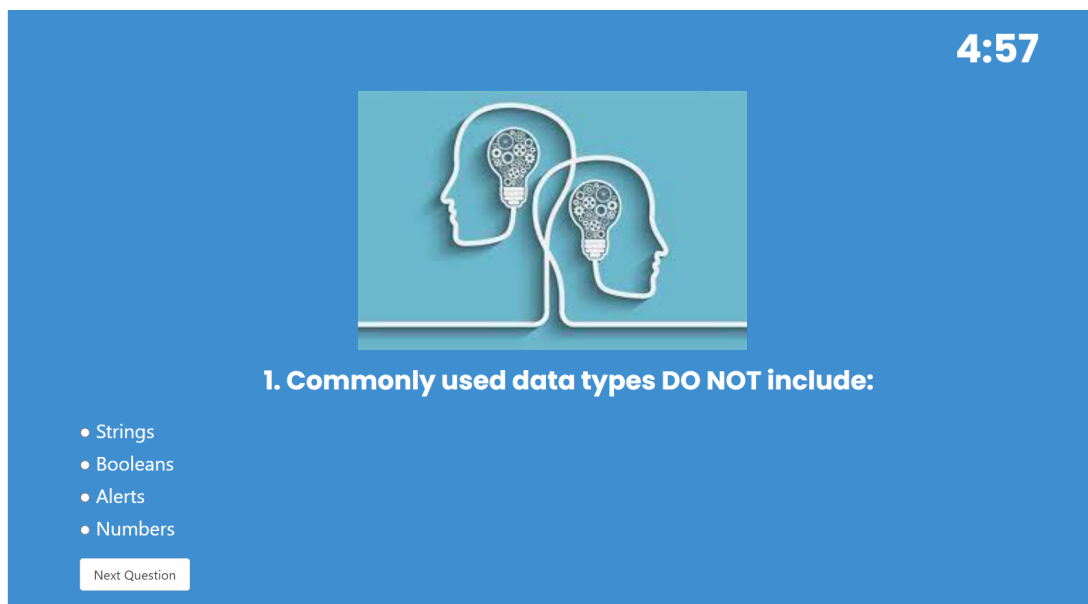


Fig. 9

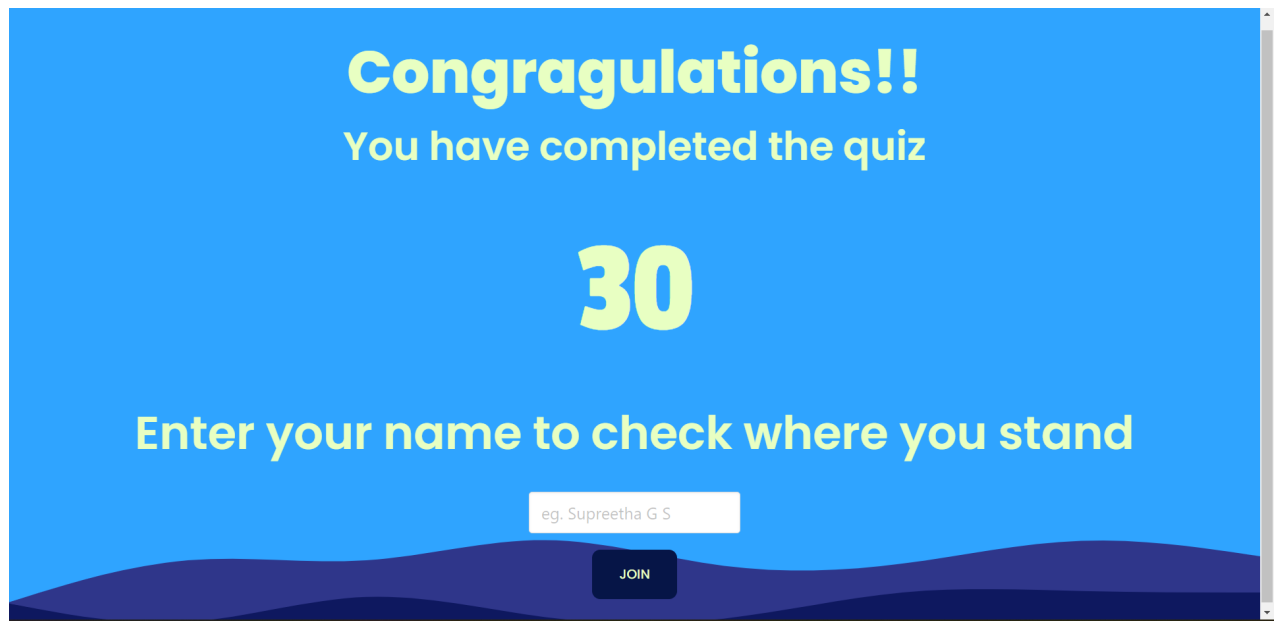


Fig. 10

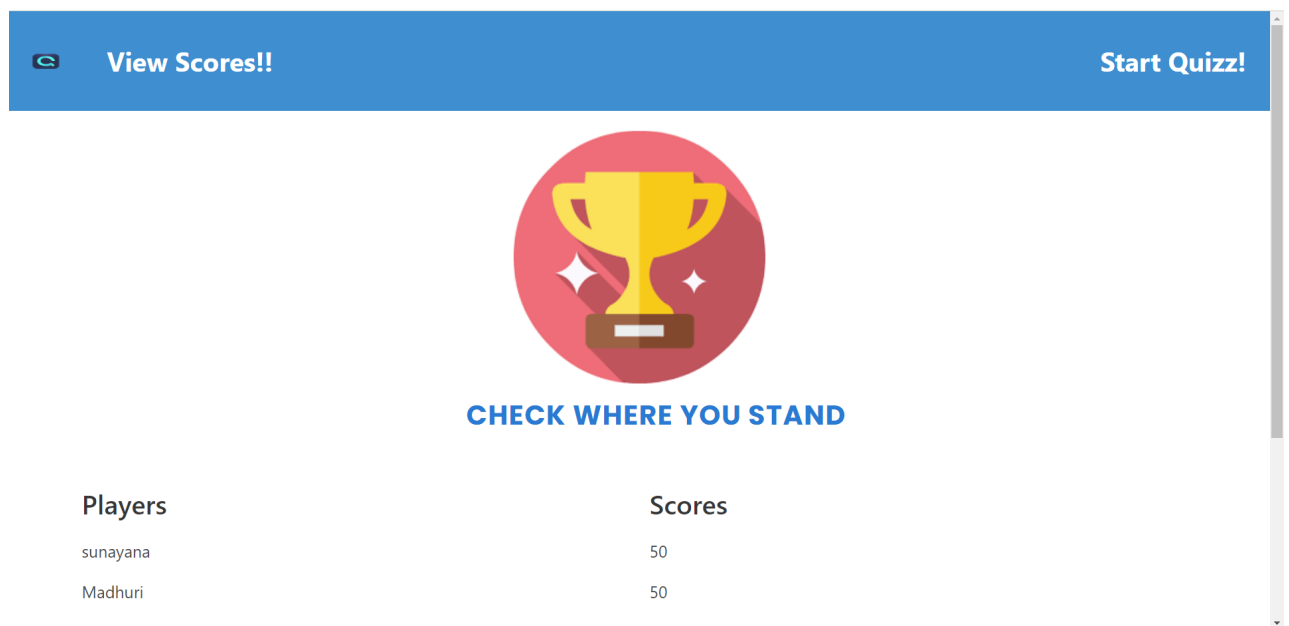


Fig. 11

CONCLUSION

- Due to increased manual work, the available system becomes more time consuming.
- So in the given work, an attempt has been made to implement fully automatic quiz application
- This online quiz application provides a facility to play quiz anywhere and anytime.
- It saves time since the user does not need to wait for the result.
- Administrator has a privilege to put as many questions in any category given in the application.
- Users can register and give the test with his/her name, and can see the results .
- Users can see the high scores as well.

REFERENCES

- www.stackoverflow.com
- www.w3schools.com
- www.wikipedia.org