

QUIZ MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by :

ROHITH R (2023246038)

**Report submitted to the faculty whose
handling subject**

FULL STACK TECHNOLOGIES- IF3101

SEMESTER - 1

MASTER OF TECHNOLOGY -

INFORMATION TECHNOLOGY

**DEPARTMENT OF INFORMATION SCIENCE AND
TECHNOLOGY**

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

Abstract

The quiz management system implemented using react is a comprehensive web design framework for quiz creation, administration, and participation. This user-friendly platform covers essential functionalities like secure user authentication, streamlined quiz-taking processes, and personalized dashboards displaying toppers' lists and individual marks. Students benefit from clear instructions, seamless access to questions, and an intuitive interface for submitting answers. Notably, the system provides in-depth analytics and reports, including group statistics, empowering educators to make data-driven decisions. By elevating the learning experience through interactive features and a dynamic design, the system not only simplifies administrative tasks but also fosters a more efficient and enriching educational environment. The integration of React ensures a responsive interface, underscoring the system's commitment to user accessibility and engagement

Tools Used:

- ❖ Node.js and npm
- ❖ Vs Code
- ❖ Redux Tool Kit
- ❖ MongoDB
- ❖ Express

Functionalities

i) Manage Questions:

- ❖ Create, update, and delete questions.
- ❖ Specify the correct answer(s) for each question

ii) Quiz Taking:

- ❖ Display a list of available quizzes.
- ❖ Allow users to select and take a quiz.
- ❖ Timer functionality for quizzes.

iii) Scoring and Results:

- ❖ Calculate and display the user's score.
- ❖ Store and display historical quiz results for users

iv) User Profiles:

- ❖ Display user details.
- ❖ Show a history of quizzes taken

Files

i) data.js and questions.js

- ❖ Data.js have a set of questions
- ❖ Questions.js displays the question which is retrieve from fetchquestion.js along with the options for that question

ii) fetchQuestion.js

- ❖ This page calls the set of events like move previous question and move next question with the help of useEffect. UseEffect is mainly used to execute the set of logic after the execution of the functional component.

Quiz Application

1. You will be asked 10 questions one after another.
2. 10 points is awarded for the correct answer.
3. Each question has three to four options. You can choose only one option
4. You can review and changes answers before the quiz finish.
5. The result will be declared at the end of the quiz.

Start Quiz

Quiz Application

Following methods can be used to display in hidden in web browser

☐ `document.write()`

☒ `console.log()`

☐ `window.alert()`

PrevNextSubmit

iii) helper.js and setresult.js

❖ This (helper.js) file calculate the marks for the question. Each question carries 10 mark and finally gives the result whether the user pass or fail for the quiz.

❖ This (setresult.js) file update the result for the user using dispatch. It will take the current state/function, which means the current state of our application and reducer will also take action

iv) result_reducer.js and question_reducer.js

❖ This (result_reducer) file consists function definition for name ,marks, exam result after computation the values are fed to result.js that file displays the content that passed from this file

❖ This (question_reducer) file consists set of events function definition for movenextquestion, prev question

v) result.js and resulttable.js

❖ This file displays the user name, marks, exam result, total questions, user attempts with the help of result_reducer.js

❖ This page displays the same as result.js file in addition of it displays the overall users who are all attend the quiz exam so far with their details. This is the output before the backend connection.

Quiz Application

Username	Rohith
Total Quiz Points:	100
Total Questions:	10
Total Attempts:	7
Total Earn Points:	70
Question Result:	Passed

Restart

Name	Attempt	Earn Points	Result
Rohith	03	50	Passed

vi) app.js

❖ This file routes to main page,quiz page and result page overall js file bundling to sing root file that fed to index.js that renders the app component

vii) Conn.js

❖ To get port no and mongodb cloud db url(including name and password). MongoDB is successfully connected.

viii) Route.js

❖ In express tool we import the router to get the routes API HTTP(post, get, put, delete). Each and every set of methods some set of actions executed, which can insert the user record and delete or truncate the user record. Result page and toppers page is displayed.

Username

vikram

Total Quiz Points :

100

Total Questions :

10

Total Attempts :

5

Total Earn Points :

50

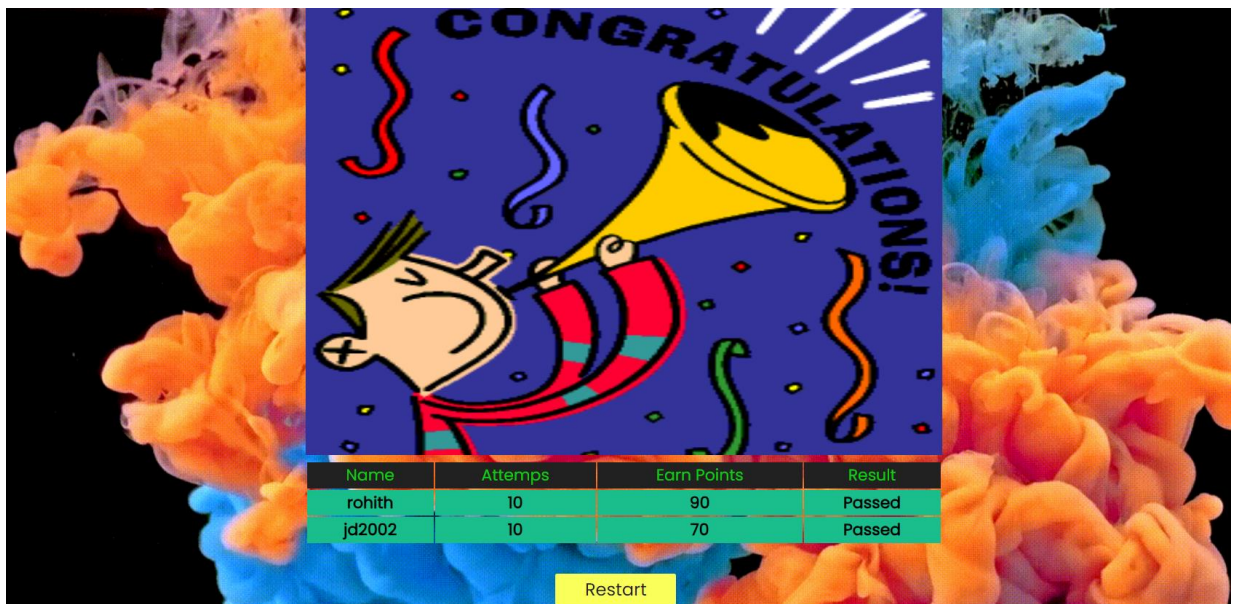
Quiz Result

Failed

Restart

Toppers

Name	Attempts	Earn Points	Result
rohith	10	90	Passed
rathish	1	0	Failed
h	3	10	Failed
jd2002	10	70	Passed
fasi	2	20	Failed
vikram	5	50	Failed



Name	Attempts	Earn Points	Result
rohith	10	90	Passed
jd2002	10	70	Passed

Restart

Conclusion

The quiz management system created with React is a helpful tool for making and taking quizzes. It's easy to use, with secure logins, smooth quiz processes, and personalized dashboards. Students get clear instructions, easy access to questions, and a simple way to submit answers. Teachers can see detailed reports and stats, helping them make smart choices based on data. The system also makes learning more interesting with interactive features and a design that works well on different devices. Using React ensures that the system is responsive and engaging for users. To sum up, this project not only does what it's supposed to for quizzes but also makes learning more enjoyable and effective.