



APRIL 2023

## Quiztopia

### PROJECT REPORT

Submitted to  
Mr Ankit Arora  
(Technical Trainer)

Submitted by  
Anveshika Singh (201500134)  
Anurag Sharma(201500126)

in partial fulfillment for the award of the degree of

### BACHELOR OF ENGINEERING

IN

Computer Engineering and Application

GLA University, Mathura

## BONAFIDE CERTIFICATE

Certified that this project report “QUIZTOPIA APP” is the bonafide work of “Anurag Sharma , Anveshika Singh” who carried out the project work under my/our supervision.

### SIGNATURE

Mr. Rohit Agrawal

HEAD OF DEPARTMENT  
(CSE Department)

### SIGNATURE

Mr. Ankit Arora  
(Project Mentor)

Submitted for the project viva-voce examination held on 27 April 2023

INTERNAL EXAMINER

EXTERNAL EXAMINER

# TABLE OF CONTENTS

ABSTRACT.....
GRAPHICAL ABSTRACT.....
ABBREVIATIONS.....
INTRODUCTION.....
Identification of Problem.....
Identification of Tasks.....
Organization of the Report.....
LITERATURE REVIEW/BACKGROUND STUDY.....
Timeline of the reported problem.....
Proposed solutions.....
Bibliometric analysis.....
Review Summary.....
Problem Definition.....
Goals/Objectives.....
DESIGN FLOW/PROCESS.....
Evaluation & Selection of Specifications/Features.....
Design Constraints.....
RESULTS ANALYSIS AND VALIDATION.....
Implementation of solution.....
CONCLUSION AND FUTURE WORK.....
CONCLUSION.....
FUTURE WORK.....
REFERENCES.....
CODE SAMPLES.....
SOME PHOTOGRAPHS FROM OUR WEBSITE.....

## ABSTRACT

This project report details the development of a quiz application that offers users different categories to choose from, each with various difficulty levels. The application provides users with a fun and interactive way to test their knowledge while also providing educational benefits.

The quiz application is designed with a user-friendly interface that makes it easy for users to navigate and select their preferred category and difficulty level. The categories range from sports, history, science, to entertainment, among others, catering to a diverse range of interests. Each category has a set of multiple-choice questions that challenge the user's knowledge, with the number of questions increasing with the difficulty level selected. Upon completion of the quiz, the user receives a score based on their performance, which provides an incentive for users to keep playing and improving their knowledge. The quiz application also features a leaderboard, where users can compare their scores with other players, adding an element of competition to the experience.

Overall, the quiz application provides an engaging and interactive way for users to test their knowledge while also providing educational benefits. The application has the potential to be a valuable tool for individuals who want to enhance their knowledge in various subjects while also providing entertainment.

## ABBREVIATIONS

**Quiztopia** :Our website's name, which can be used as an abbreviation throughout the report.

**Technologies Used:** JSX, CSS, JavaScript, React, MongoDB

**API:** Application Programming Interface, which refers to the methods we use to communicate between the front end and backend of our website.

**CRUD:** Create, Read, Update, and Delete, which are the basic operations we perform on data in our database.

**UI:** User Interface, which refers to the design and layout of our website's frontend.

**DB:** Database, which refers to the storage system we use to store and manage our website's data.

## **INTRODUCTION**

Welcome to quiztopia, the ultimate quiz app for [your target audience]. Test your knowledge, challenge yourself, and have fun with our wide range of quizzes on [your quiz topics]. With [number of questions] to choose from, you can customize your quiz experience by selecting your preferred difficulty level and category. Whether you're a trivia buff or just looking for a fun way to pass the time, our app is perfect for you. Use Quiztopia App today and start quizzing!

With our platform, we hope to make it easier for everyone to achieve this goal.

### **Identification of Tasks**

In order to develop a successful quiz service platform, it is important to identify the tasks that users will need to request and track. These tasks will vary depending on the individual needs of each user, but some common examples include:

1. Provide users with a variety of categories to choose from
2. Offer different levels of difficulty to challenge users
3. Present multiple-choice questions that test the user's knowledge
4. Score the user's performance based on their answers
5. Provide a leaderboard to compare the user's score with others
6. Ensure the accuracy and relevance of the questions presented
7. Create a user-friendly interface that is accessible on different devices
8. Provide an engaging and interactive way to learn and have fun
9. Encourage users to keep playing and improving their knowledge
10. Enhance the user experience through additional features and improvements.

Once a task has been requested, users should be able to track its progress

through the platform and the score and time will be visible on the user's dashboard.

Overall, the task of this project report is to provide a comprehensive overview of the quiz application's design and development, including its features, functionality, challenges, and potential educational benefits.

## Project Scope:

This quiz application has the potential for future improvements and enhancements to provide a better user experience and increase its educational value. Here are some possible future scopes for the project:

1. Integration with other platforms: The quiz application could be integrated with popular social media platforms, such as Facebook and Twitter, to enable users to share their scores and challenge their friends. It could also be integrated with learning management systems used by educational institutions to provide a more structured learning experience.
2. Gamification features: The quiz application could be enhanced with gamification features such as badges, rewards, and achievements, to motivate users to keep playing and learning. This would increase user engagement and make learning more enjoyable.
3. Personalized learning: The quiz application could be personalized for each user based on their interests, learning style, and performance. It could recommend topics or difficulty levels based on the user's previous performance and learning history.
4. User feedback system: The quiz application could have a user feedback system to collect suggestions for new categories, questions, or improvements to existing features. This would help to keep the application up-to-date and relevant.
5. Mobile application: The quiz application could be developed as a mobile application for iOS and Android platform.

Overall, the future scope of the quiz application is to enhance the user experience and increase its educational value through the addition of new features and improvements.

## **Proposed solutions**

The platform will consist of two main components: a customer-facing interface and a service provider-facing interface. The customer-facing interface will allow users to create an account, request for quiz, monitor the status of their requested tasks, and cancel tasks if needed. The interface will be designed to be user-friendly and intuitive.

To ensure the security and privacy of user data, the platform will include robust authentication and authorization systems, as well as encryption of all sensitive data. Regular security audits and vulnerability scans will also be conducted to ensure the platform is secure.

Overall, the proposed solution for the project is to develop a web-based platform that provides a comprehensive and user-friendly interface for customers.

## **Bibliometric analysis**

A bibliometric analysis of the project based on providing with a user-friendly interface for task management can provide valuable insights into the current state of research in this field, as well as identifying potential areas for further research. By conducting a comprehensive search and using bibliometric tools to analyze the data, researchers can gain a deeper understanding of the key concepts and trends in this area of study

## **Review Summary**

The project scope covers the development of a user-friendly interface, database of questions, scoring and leaderboard system, and security measures. The project also has a future scope for enhancements, including adding more categories, improving difficulty levels, gamifying the application, and personalizing the user experience. The project report provides a detailed description of the project's background, methodology, results, and conclusion. It highlights the importance of conducting market research, user testing, and continuous improvement through user feedback. The report also emphasizes the potential benefits of the quiz application as an educational and entertainment tool for users and efficient completion of the quiz test.

In summary, the quiz application project is an innovative and engaging solution to provide users with a fun and interactive way to test their knowledge. The project's goals and objectives are well-defined, and the future scope of the project shows great potential for further development and improvement.

## **Problem Definition**

One of the primary issues is the lack of an interactive and engaging way for users to test their knowledge. Traditional methods of testing knowledge, such as exams or assessments, can be boring and not very motivating for users. This can result in a lack of interest and engagement, leading to a lower level of understanding and retention of information.

## **Goals/Objectives**

Goals:

- To design and develop a web-based quiz application that provides users with an entertaining and interactive way to test their knowledge in various categories.
- To create an application that is user-friendly, responsive, and accessible on different devices.
- To provide a variety of categories and difficulty levels to cater to different interests and skill levels.
- To ensure the accuracy and relevance of the questions presented in the quiz application.
- To motivate users to keep playing and improving their knowledge through a scoring and leader-board system.

Objectives:

- To conduct market research to identify the target audience and their interests.
- To design an intuitive and user-friendly interface that is accessible on different devices.
- To develop a database of questions for the quiz application that are accurate and relevant to the chosen categories
- To incorporate a scoring and leaderboard system that motivates users to keep playing and improving their scores
- To ensure the application's security and reliability by implementing proper testing and debugging processes.

# **DESIGN FLOW/PROCESS**

## **Evaluation & Selection of Specifications/Features**

The evaluation and selection of specific features of the project based on providing knowledge testing service with task management, status tracking, and record-keeping capabilities involve the following steps:

1. Identify User Needs: The first step is to identify the needs of the users, including customers, and admin. This involves understanding their requirements for managing tracking task status, and keeping records of user activity.
2. Brainstorm Features: The next step is to brainstorm potential features that could meet these user needs as user can pick the topic in which he/she is interested. This could include features such as a user-friendly interface, task management tools, status tracking(score), notifications(counter), and reporting capabilities(picking the options).
3. Prioritize Features: Once a list of potential features has been created, the project team should prioritize them based on their importance to users and their feasibility. This involves considering factors such as user experience, technical complexity, and development effort.

## **Design Constraints**

Design constraints are the limitations or restrictions that impact the design and development of a project. The design constraints for the project based on providing with task management, status tracking, and record-keeping capabilities could include the following:

1. Budget: One of the key constraints for the project could be the budget allocated for its development. The project team would need to work within the budget constraints to ensure that the platform is designed and developed within the allocated budget.
2. Timeframe: Another important constraint could be the project timeline. The project team would need to complete the design and development of the platform within the given timeframe, which may require prioritizing certain features over others to ensure timely delivery.

3. Technical Requirements: The platform would need to be designed and developed to meet certain technical requirements, such as compatibility with various operating systems, devices, and browsers. These technical constraints would need to be considered during the design and development process.

4. Security: As the platform would be handling sensitive user information, security would be a critical design constraint. The platform would need to be designed and developed with strong security measures to protect user data from unauthorized access or data breaches.

5. User Experience: The platform would need to be user-friendly and intuitive to use. The project team would need to ensure that the design and development of the platform prioritizes user experience to ensure that users can easily request tasks, track task status, and access their records.

6. Scalability: The platform would need to be designed and developed with scalability in mind to accommodate potential growth in users and tasks. The project team would need to ensure that the platform can handle increased traffic and data storage requirements as the platform grows.

By considering these design constraints, the project team can ensure that the platform is designed and developed to meet the needs of its users, while also operating within the constraints of the project budget, timeline, technical requirements, security, user experience, and scalability.

## **RESULTS ANALYSIS AND VALIDATION**

### **Implementation of solution**

The implementation of the solution for the project based on providing test for the user where users can request tasks, check task status, and delete tasks, and the admin can keep records of every user activity would involve several steps. These steps include:

1. Planning: The project team would need to plan the implementation process by defining the scope of the project, identifying the resources needed, and establishing the timeline and milestones for the project.

2. Development: The development process would involve designing and developing the platform based on the proposed solution. The team would need to

select appropriate technologies and tools to develop the platform and ensure that it meets the design constraints and technical requirements.

3. Testing: The platform would need to undergo various tests to ensure that it is functioning as intended and meets user requirements. This would involve testing for functionality, user experience, security, and performance.

4. Deployment: Once the platform has been developed and tested, it would be deployed to a production environment where it can be accessed by users. The deployment process would involve setting up the necessary infrastructure, configuring the platform, and ensuring that it is accessible to users.

5. Training: The project team would need to train the users and the admin on how to use the platform effectively. This would involve creating user manuals, providing online tutorials, and conducting training sessions.

6. Maintenance and Support: After the platform has been deployed, the project team would need to provide ongoing maintenance and support to ensure that it remains up-to-date, secure, and functioning as intended. This would involve fixing bugs, updating the platform, and providing user support.

By following these implementation steps, the project team can ensure that the platform is successfully developed and implemented, and that it meets the requirements of the users and the admin.

## **CONCLUSION AND FUTURE WORK**

### **CONCLUSION**

In conclusion, the quiz application project aims to develop a web-based quiz application that provides users with an entertaining and interactive way to test their knowledge in various categories. The application will be designed with a focus on user experience and accessibility across different devices.

Furthermore, the future scope of the project includes adding more categories, improving difficulty levels, gamifying the application, and personalizing the user experience. These enhancements will provide a more engaging and immersive experience for users and increase user acquisition and engagement.

Overall, the quiz application project has great potential to provide a valuable educational and entertainment tool to users while also offering opportunities for future development and improvement.

## **FUTURE WORK**

There are several areas of future work that could enhance the functionality and usability of the project based on providing more Quiz questions covering all the hot GK and trending topics with a user-friendly platform where users can request tasks, check the status of skills through marks while the admin can keep a record of every user activity as Database is used.

Some potential future work includes:

1. Integration of additional features: The platform could be enhanced by incorporating new features such as interested topic as API is used to fetch the questions to the interested topic. scheduling of quiz , and payment processing for advance question like Interview questions and others.
2. Mobile application development: Developing a mobile application for the platform could increase the accessibility and convenience for users.
- 3 Expansion to other regions: The project could be expanded to other regions, increasing its potential user base and market reach.
- 4 Machine Learning-based recommendations: Implementation of machine learning algorithms for personalized recommendations could enhance the user experience and increase engagement.

## **REFERENCES**

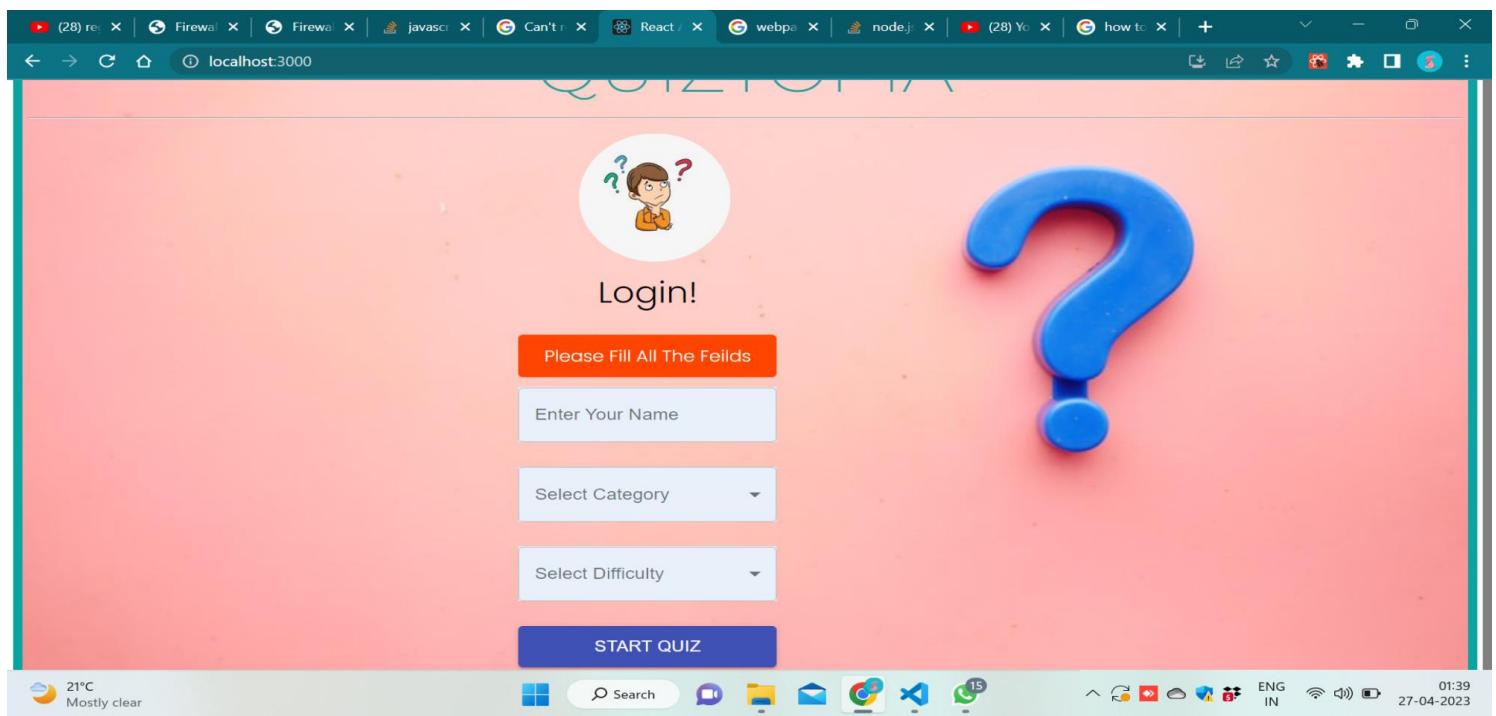
Faculty Guidelines:

Mr Ankit Arora (Technical Trainer in GLA University)

GitHubRepository link:

<https://github.com/ANVESHIIKA17/quiztopia.live>

# Output Sample | Admin



localhost:3000

# QUIZTOPIA



Login!

Enter Your Name

Select Category

Select Difficulty

START QUIZ



21°C  
Mostly clear

Search

File Explorer

Email

Google Chrome

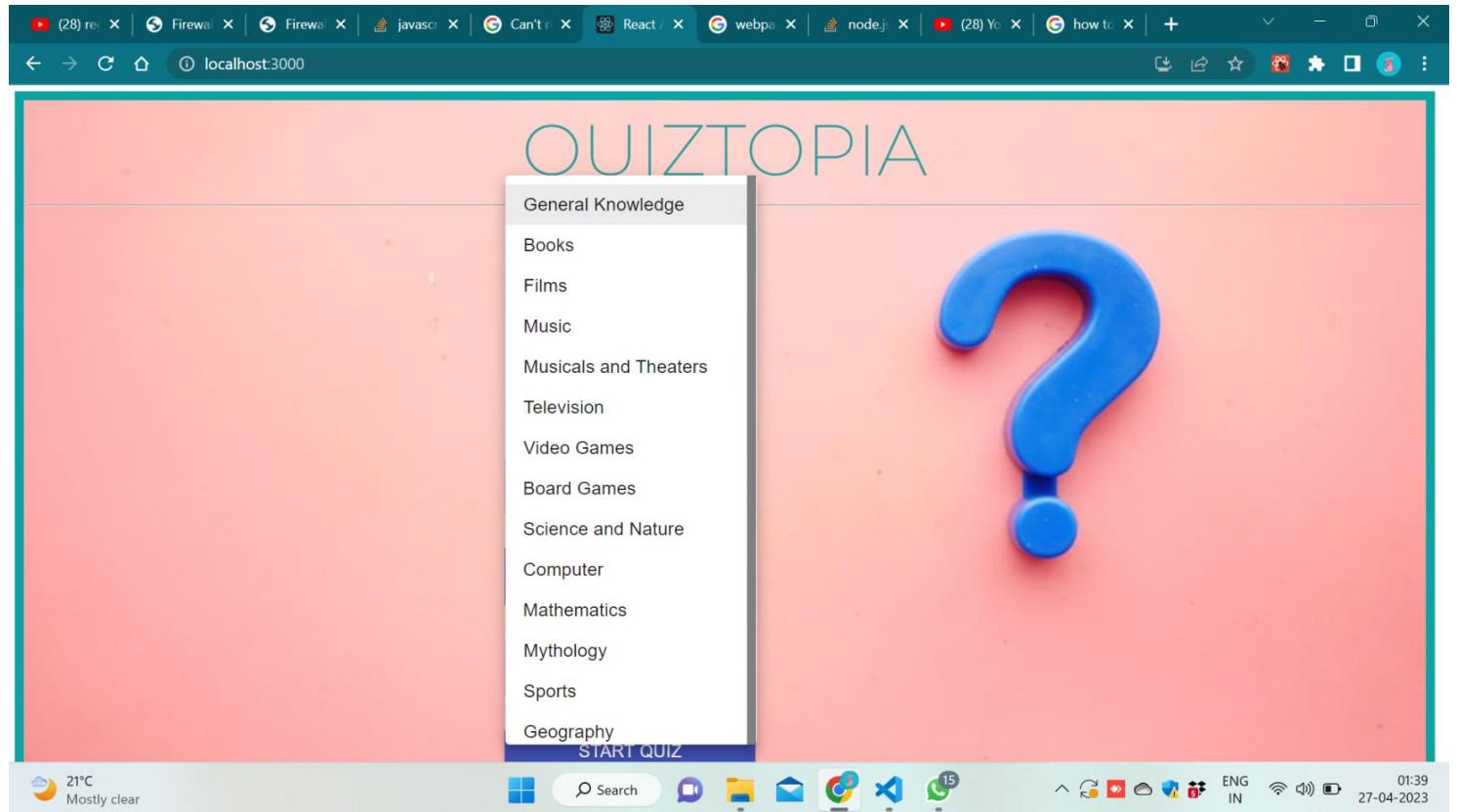
OneDrive

Skype

15 notifications

ENG IN

01:38 27-04-2023



# Quiz Question

The screenshot shows a quiz application titled "QUIZTOPIA" running on a Windows operating system. The application interface includes a header bar with tabs like "React", "webpa", and "node.js", and a status bar at the bottom showing weather, system icons, and the date/time.

**Header:** QUIZTOPIA

**User Information:** Welcome,anveshika

**Category:** SCIENCE: COMPUTERS

**Score:** SCORE : 0

**Question 1:**

In any programming language, what is the most common way to iterate through an array?

**Options:**

- \ While loops
- If Statements
- For loops
- Do-while loops

**Buttons:**

- QUIT (Pink button)
- NEXT QUESTION (Blue button)

**System Status Bar:**

- 21°C Mostly clear
- Windows Start button
- Search bar
- Taskbar icons: File Explorer, Mail, Edge browser, VS Code, GitHub
- System icons: Volume, Battery, Network, etc.
- Language: ENG IN
- Date: 27-04-2023
- Time: 01:39

The final scores with name:-



# The coding section:

A screenshot of the Visual Studio Code interface. The title bar says "Home.js - quiztopiaapp - Visual Studio Code". The left sidebar shows a tree view of the project structure under "OPEN EDITORS". The main editor area displays the code for "Home.js". The code imports React, axios, and various components from material-ui/core and other files. It defines a functional component "Home" that handles states for category, difficulty, and error. It also includes logic for handling form submissions using useState and axios. The bottom status bar shows the file path "PS C:\Users\ANVESHAKA\Desktop\quiztopiaapp>" and the current date and time "04-05-2023 18:40".

```
src > pages > Home > JS Home.js > [?] Home > [?] myHandle
1 import React, { useState } from 'react';
2 import axios from 'axios';
3 import "./Home.css";
4 import { Button, MenuItem, TextField } from '@material-ui/core';
5 import Categories from '../../../../../data/categories';
6 import ErrorMessage from '../../../../../components/ErrorMessage(ErrorMessage)';
7 import { useNavigate } from 'react-router-dom';
8
9
10 const Home = ({name, setName, fetchQuestions}) => {
11   //creating the states
12   const [category, setCategory] = useState("");
13   const [difficulty, setDifficulty] = useState("");
14   const [error, setError] = useState(false);
15
16   const Navigate = useNavigate();
17
18   const myHandle = (e) => {
19     e.preventDefault();
20     axios.post('/api/mydata', { name, category })
21   }
22
23   return (
24     <div>
25       <h1>Quiz Topia</h1>
26       <h2>Welcome to Quiz Topia</h2>
27       <form>
28         <input type="text" value={name} onChange={(e) => setName(e.target.value)} />
29         <input type="button" value="Submit" onClick={myHandle} />
30       </form>
31       <Categories categories={categories} />
32     </div>
33   );
34 }
35
36 export default Home;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\ANVESHAKA\Desktop\quiztopiaapp> \* History restored  
PS C:\Users\ANVESHAKA\Desktop\quiztopiaapp> \* History restored

Ln 19, Col 30 Spaces: 2 UTF-8 CRLF ⚡ JavaScript ⏪ Go Live ✨ Prettier 🔍 ENG IN ⌂ 04-05-2023 18:40 4

```
src > pages > Quiz > JS Quiz.js > ...
1 import { useEffect, useState } from "react";
2 import { CircularProgress } from "@material-ui/core";
3 import "./Quiz.css";
4 import Question from "../../components/Question/Question";
5
6 const Quiz = ({name, score, questions, setQuestions, setScore}) => {
7
8     const [options, setOptions] = useState();
9     const [currQues, setCurrQues] = useState(0);
10
11    useEffect(() =>{
12        console.log(questions);
13
14        setOptions(
15            questions &&
16            handleShuffle([
17                questions[currQues].correct_answer,
18                ...questions[currQues].incorrect_answers,
19                // spread operator will spread the incorrect ans as there are 3 incorrect ans
20            ])
21        );
22        // cuurques as everytime the curr ques changes the options will also get changed
23        ,[questions, currQues]);
24
25        console.log(options);
26
}
// master* 0 △ 0 0 18:40 04-05-2023
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** index.js - quiztopiaapp - Visual Studio Code
- Explorer Sidebar:** Shows the project structure with groups:
  - GROUP 1: server.js (U), index.js (src, M), Home.js (src... U), package.json (U)
  - GROUP 2: QUIZTO... (Quiz, Result, App.css, App.js, App.test.js, index.css), index.js (M), logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json (M)
- Code Editor:** The file index.js is open, showing React code. The code includes imports for React and ReactDOM, and a root render call with an App component.
- Bottom Navigation:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL.
- Terminal:** PS C:\Users\ANVESHIKA\Desktop\quiztopiaapp> [empty]
- System Tray:** Shows icons for power shell, search, folder, file, settings, terminal, battery, signal, and date/time (04-05-2023, 18:41).

File Edit Selection View Go Run Terminal Help

App.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS App.js M X JS Home.js U {} package.json U

OPEN EDITORS GROUP 1 JS server.js ba... U X JS App.js src M JS Home.js src... U {} packagejso... U GROUP 2 QUIZTO... D M

Quiz # Quiz.css U JS Quiz.js U Result # Result.css U JS Result.js U # App.css M JS App.js M JS App.test.js # index.css JS index.js M logo.svg JS reportWebVitals.js JS setupTests.js .gitignore {} package-lock.json M

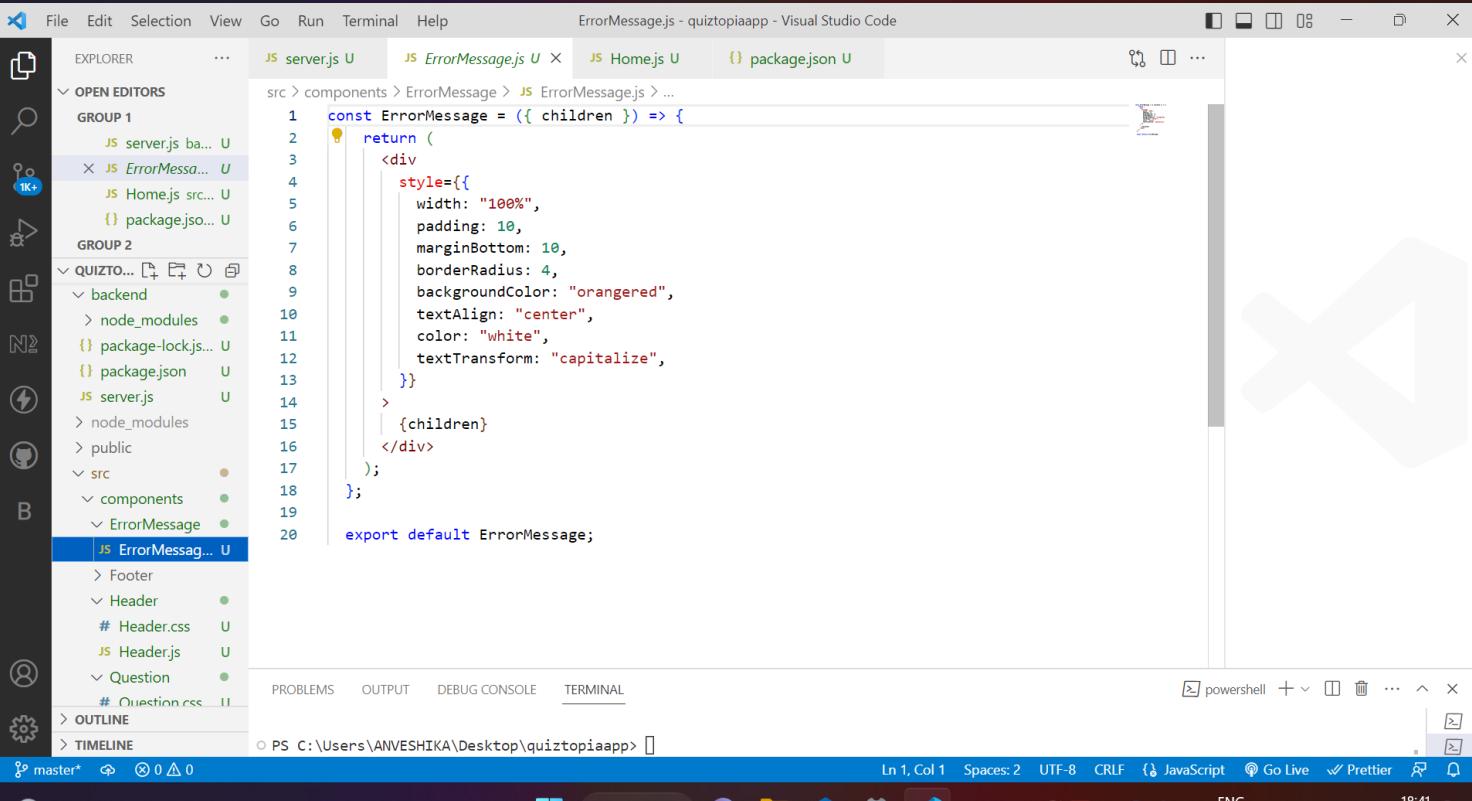
src > JS App.js > ...

```
34 |     setQuestions(data.results);
35 |
36 |
37 | };
38 |
39 | return (
40 |   <BrowserRouter>
41 |     <div className="app" style={{backgroundImage:url('https://images.unsplash.com/photo-16')...}}>
42 |       <Header />
43 |       <Routes>
44 |
45 |         <Route exact path='/' element={<Home name={name} setName={setName} fetchQuestions={fetchQuestions} setScore={setScore} setQuestions={setQuestions}>} />
46 |         <Route exact path='/quiz' element={<Quiz name={name} questions={questions} score={score} setScore={setScore} setQuestions={setQuestions}>} />
47 |       </Routes>
48 |
49 |     </div>
50 |   </BrowserRouter>
51 | )
52 |
53 | </div>
54 |
55 | </div>
56 |
57 | </div>
58 |
59 | </div>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell + -

PS C:\Users\ANVESHIKA\Desktop\quiztopiaapp



File Edit Selection View Go Run Terminal Help ErrorMessage.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS ErrorMessage.js U X JS Home.js U {} package.json U

src > components > ErrorMessage > JS ErrorMessage.js > ...

```
1 const ErrorMessage = ({ children }) => {
2   return (
3     <div
4       style={{
5         width: "100%",
6         padding: 10,
7         margin: 10,
8         borderRadius: 4,
9         backgroundColor: "orangered",
10        textAlign: "center",
11        color: "white",
12        textTransform: "capitalize",
13      }}
14     >
15       {children}
16     </div>
17   );
18 }
19
20 export default ErrorMessage;
```

GROUP 1  
JS server.js ba... U  
X JS ErrorMessage.js U  
JS Home.js src... U  
{} package.json... U

GROUP 2  
QUIZTO... D F O S  
backend  
node\_modules  
{} package-lock.json U  
{} package.json U  
JS server.js U  
node\_modules  
public  
src  
components  
ErrorMessage

B PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + PS C:\Users\ANVESHIIKA\Desktop\quiztopiaapp> Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier ENG IN 18:41 04-05-2023 4

File Explorer Search Taskbar

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** App.js - quiztopiaapp - Visual Studio Code.
- Explorer Sidebar:** Shows the project structure:
  - OPEN EDITORS: server.js (U), App.js (M), Home.js (U), package.json (U).
  - GROUP 1: server.js (U), App.js src (M), Home.js src... (U), package.json... (U).
  - GROUP 2: QUIZTOPIA (D+ E+ U): Quiz (green dot), # Quiz.css (U), JS Quiz.js (U), Result (green dot), # Result.css (U), JS Result.js (U), JS App.css (M).
- Code Editor:** The active file is App.js (M). The code is a functional component named App() that uses useState to manage state for name, questions, and score. It includes an asynchronous fetch function to get questions from an API. The code editor has syntax highlighting and several red squiggly underlines indicating errors or warnings.
- Bottom Status Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL.
- Bottom Taskbar:** powershell, Ln 18, Col 27, Spaces: 2, UTF-8, LF, JavaScript, Go Live, Prettier.
- Bottom Icons:** Cloud, Search, Folder, Mail, Settings, Terminal, and various system icons.
- Bottom Right:** ENG IN, 04-05-2023, 18:41.

File Edit Selection View Go Run Terminal Help

App.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS App.js M X JS Home.js U {} package.json U

OPEN EDITORS

GROUP 1

- JS server.js ba... U
- X JS App.js src M
- JS Home.js src... U
- {} package.json... U

GROUP 2

QUIZTO... [+] [?] [?] [?] [?]

- Quiz ●
- # Quiz.css U
- JS Quiz.js U
- Result ●
- # Result.css U
- JS Result.js U
- # App.css M
- JS App.js M

JS App.test.js

# index.css

JS index.js M

logo.svg

JS reportWebVitals.js

JS setupTests.js

.gitignore

{} package-lock.json M

OUTLINE

TIMELINE

src > JS App.js > ...

```
1 import { BrowserRouter, Route, Routes } from 'react-router-dom';
2 import './App.css';
3 import Header from './components/Header/Header';
4 import Footer from './components/Footer/Footer';
5 import Home from './pages/Home/Home';
6 import Quiz from './pages/Quiz/Quiz';
7 import Result from './pages/Result/Result';
8 import { useState } from 'react';
9 import axios from 'axios';

10
11
12 // const express = require('express');
13 // const app = express();
14 // const mongoose = require('mongoose');

15
16 // mongoose.connect('mongodb://127.0.0.1:27017/quiztopia-users', {
17 //   useNewUrlParser: true,
18 //   useCreateIndex: true,
19 //   useUnifiedTopology: true,
20 //   useFindAndModify: false
21 // })
22 // .then(() => console.log('DB Successfully connected'))
23 // .catch((e) => console.log(e));
24
25 function App() {
26   const [name, setName] = useState("");
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell + ...

Ln 18, Col 27 Spaces: 2 UTF-8 LF { JavaScript Go Live Prettier 18:41 04-05-2023 4

PS C:\Users\ANVESHAKA\Desktop\quiztopiaapp>

```
import { Button } from '@material-ui/core';
import React, { useEffect } from 'react';
import { useNavigate } from 'react-router-dom';
import './Result.css'

const Result = ({name,score}) => {

  const Navigate= useNavigate();

  useEffect(() => {
    if(!name){
      Navigate("/");
    }
  }, [name,Navigate]);

  return (
    <div className='result'>
      <span className='title'{>{name}, Your Final Score Is : {score}</span>
      <Button
        variant="contained"
        color="secondary"
        size="large"
        style={{ alignSelf: "center", marginTop: 20 }}
        href="/>
        Go To HomePage
    </div>
  );
};

export default Result
```

File Edit Selection View Go Run Terminal Help Quiz.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS Quiz.js U JS Home.js U {} package.json U

OPEN EDITORS GROUP 1 JS server.js ba... U X JS Quiz.js src... U JS Home.js src... U {} package.json... U GROUP 2 QUIZTOP... P E U & > public > src > components > data JS categories.js U > pages > Home # Home.css U JS Home.js U > Quiz # Quiz.css U JS Quiz.js U > Result # App.css M JS App.js M JS App.test.js M # index.css > OUTLINE > TIMELINE

src > pages > Quiz > JS Quiz.js > ...

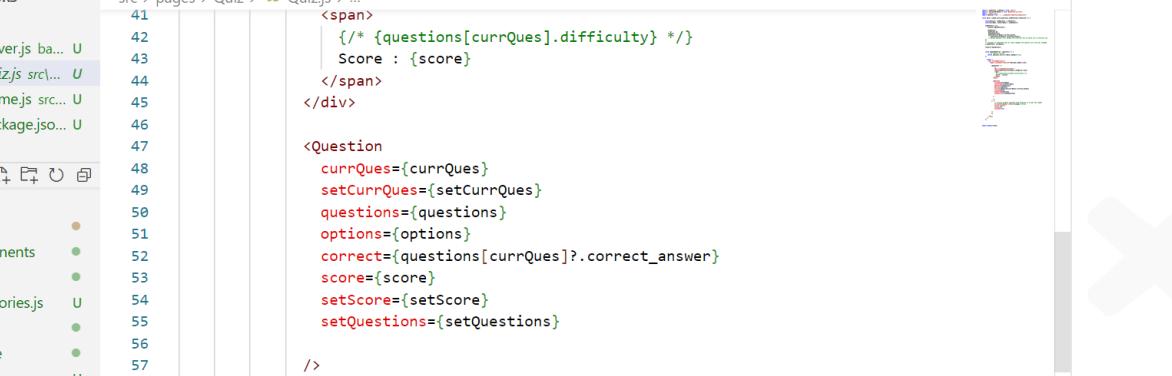
```
54     setScore={setScore}
55     setQuestions={setQuestions}
56
57   
58
59   ): (
60     // circular progree imported from material ui to get the loader
61     <CircularProgress style={{ margin: 100 }}>
62       color='inherit'
63       size={150}
64       thickness={1}
65
66     />
67   )}
68
69   </div>
70 );
71 );
72 );
73 );
74 );
75 );
76 export default Quiz;
77 );
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell + ×

PS C:\Users\ANVESHIKA\Desktop\quiztopiaapp>

master\* 0 △ 0 18:41 04-05-2023 ENG IN



```
src > pages > Quiz > Quiz.js > ...
41   <span>
42     {/* {questions[currQues].difficulty} */}
43     Score : {score}
44   </span>
45 </div>
46
47 <Question
48   currQues={currQues}
49   setCurrQues={setCurrQues}
50   questions={questions}
51   options={options}
52   correct={questions[currQues]?.correct_answer}
53   score={score}
54   setScore={setScore}
55   setQuestions={setQuestions}
56
57   />
58
59   </>
60   ): (
61     // circular progree imported from material ui to get the loader
62     <CircularProgress style={{ margin: 100 }}>
63       color='inherit'
64       size={150}
65       thickness={1}
66   )
```



```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
app.use(cors());

// Connect to MongoDB
mongoose.connect('mongodb://localhost/mydatabase', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(()=>console.log("DB Connected"));

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', function() {

  * History restored

PS C:\Users\ANVESHIIKA\Desktop\quiztopiaapp
* History restored
○ PS C:\Users\ANVESHIIKA\Desktop\quiztopiaapp>
```

File Edit Selection View Go Run Terminal Help Home.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS Home.js U package.json U

OPEN EDITORS GROUP 1 JS server.js ba... U X JS Home.js src... U {} packagejson.o U GROUP 2 QUIZTOPIAAPP backend node\_modules {} package-lock.j... U {} package.json U JS server.js U > node\_modules > public > src > components > ErrorMessage > Footer > Header # Header.css U JS Header.js U > Question # Question.css U JS Question.j... U > OUTLINE > TIMELINE

```
src > pages > Home > JS Home.js > [e] Home > [e] myHandle
  77 style={{ marginBottom: 25, background: "rgb(232, 241, 250)" }
  78   | }
  79   >
  80     <MenuItem key="Easy" value="easy">
  81       Easy
  82     </MenuItem>
  83     <MenuItem key="Medium" value="medium">
  84       Medium
  85     </MenuItem>
  86     <MenuItem key="Hard" value="hard">
  87       Hard
  88     </MenuItem>
  89   </TextField>

  90   <Button variant="contained"
  91     color="primary"
  92     size="large"
  93     onClick={handleSubmit}
  94   >
  95     Start Quiz
  96   </Button>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

\* History restored  
PS C:\Users\ANVESHKA\Desktop\quiztopiaapp  
\* History restored

Ln 19, Col 30 Spaces: 2 UTF-8 CRLF { JavaScript ENG IN 18:40 04-05-2023 4

File Edit Selection View Go Run Terminal Help categories.js - quiztopiaapp - Visual Studio Code

EXPLORER ... JS server.js U JS categories.js U JS Home.js U package.json U

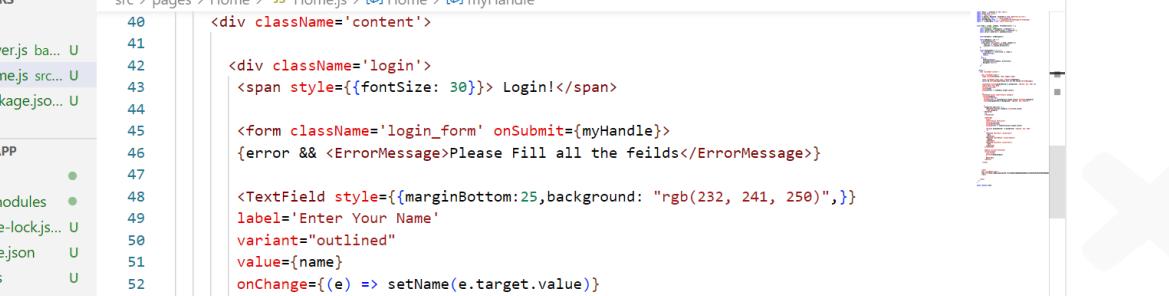
OPEN EDITORS GROUP 1 JS server.js ba... U X JS categories.j... U JS Home.js src... U {} packagejson.o U GROUP 2 QUIZTO... D+ E+ U > backend > node\_modules > public > src > components > data JS categories.js U > pages > Home > Quiz > Recruit > OUTLINE > TIMELINE

```
src > data > JS categories.js > ...
  2   [
  3     { category: "General Knowledge",
  4       value: 9,
  5     },
  6     { category: "Books", value: 10 },
  7     { category: "Films", value: 11 },
  8     { category: "Music", value: 12 },
  9     { category: "Musicals and Theaters", value: 13 },
 10     { category: "Television", value: 14 },
 11     { category: "Video Games", value: 15 },
 12     { category: "Board Games", value: 16 },
 13     { category: "Science and Nature", value: 17 },
 14     { category: "Computer", value: 18 },
 15     { category: "Mathematics", value: 19 },
 16     { category: "Mythology", value: 20 },
 17     { category: "Sports", value: 21 },
 18     { category: "Geography", value: 22 },
 19     { category: "History", value: 23 },
 20     { category: "Politics", value: 24 },
 21     // { category: "Art", value: 25 },
 22     { category: "Celebrities", value: 26 },
 23   ]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

\* History restored  
PS C:\Users\ANVESHKA\Desktop\quiztopiaapp  
\* History restored

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF { JavaScript ENG IN 18:40 04-05-2023 4



```
src > pages > Home > JS Home.js > [x] Home > [x] myHandle
40   <div className='content'>
41     <div className='login'>
42       <span style={{fontSize: 30}}> Login!</span>
43
44     <form className='login_form' onSubmit={myHandle}>
45       {error && <ErrorMessage>Please Fill all the fields</ErrorMessage>}
46
47       <TextField style={{marginBottom:25,background: "rgb(232, 241, 250)"}}>
48         label='Enter Your Name'
49         variant='outlined'
50         value={name}
51         onChange={(e) => setName(e.target.value)}
52
53       />
54       <TextField select label="Select Category"
55         value={category}
56         variant='outlined'
57         onChange={(e) => setCategory(e.target.value)} onSubmit={myHandle}
58         style={{marginBottom:25,background: "rgb(232, 241, 250)"}}
59       >
60     </form>
61   </div>
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it displays the project structure under "OPEN EDITORS". The "src" folder contains "components", "Footer", "Header", "Question", and "Timeline". "components" contains "ErrorMessage". "Header" contains "Header.css" and "Header.js". "Question" contains "Question.css" and "Question.js".
- Code Editor:** The main area shows a code editor with line numbers (77-97) and syntax highlighting for JavaScript and CSS. The code implements a dropdown menu with three options: "Easy", "Medium", and "Hard". Below the menu is a large "Start Quiz" button.
- Terminal:** At the bottom, the terminal shows a PowerShell session with the command "PS C:\Users\ANVESHIIKA\Desktop\quiztopiaapp>". It also displays the message "\* History restored".
- Bottom Bar:** The status bar at the bottom indicates "Ln 19, Col 30", "Spaces: 2", "UTF-8", "CRLF", "JavaScript", "Go Live", "Prettier", and file paths like "C:\Users\ANVESHIIKA\Desktop\quiztopiaapp\src\pages\Home\Home.js".

