**Online Practice Test Platform**

**Objective**

* Personalized Math Practice Platform with Adaptive Learning
* This project aims to develop an online learning platform specifically for students in grades 7 through 10 to hone their mathematical skills.

**Key functionalities:**

* Seamless User Access:
* Implement a user-friendly login system with options for both traditional account creation and Google authentication for convenient access.
* Personalized Dashboard:
* Upon successful login, direct users to a personalized dashboard that reflects their progress and learning journey.
* Adaptive Learning with CAT:
* Integrate Computerized Adaptive Testing (CAT) to tailor the learning experience. The platform will assess the student's initial understanding through a starting set of questions. Based on their performance, the difficulty and topic of subsequent questions will dynamically adjust, ensuring a focused and efficient learning experience.
* Practice Question Bank:
* Develop a comprehensive question bank with at least 20 high-quality mathematics questions catering to the curriculum of grades 7 to 10. This bank can be further expanded in the future.
* Benefits:
* Students will benefit from a personalized learning experience that identifies their strengths and weaknesses, allowing them to focus on areas that need improvement.
* Integrate performance tracking.
* The platform offers a convenient and engaging way to practice and improve mathematical skills.
* The use of CAT ensures a challenging yet achievable learning path, promoting a sense of accomplishment and boosting confidence.
* Future Enhancements:
* Expand the question bank to cover a wider range of mathematical topics.
* Analytics to provide detailed feedback to users and identify areas for further improvement.
* Implement gamification elements to make the learning process more engaging and interactive.
* Allow instructors to create custom quizzes and track student progress within a classroom setting.

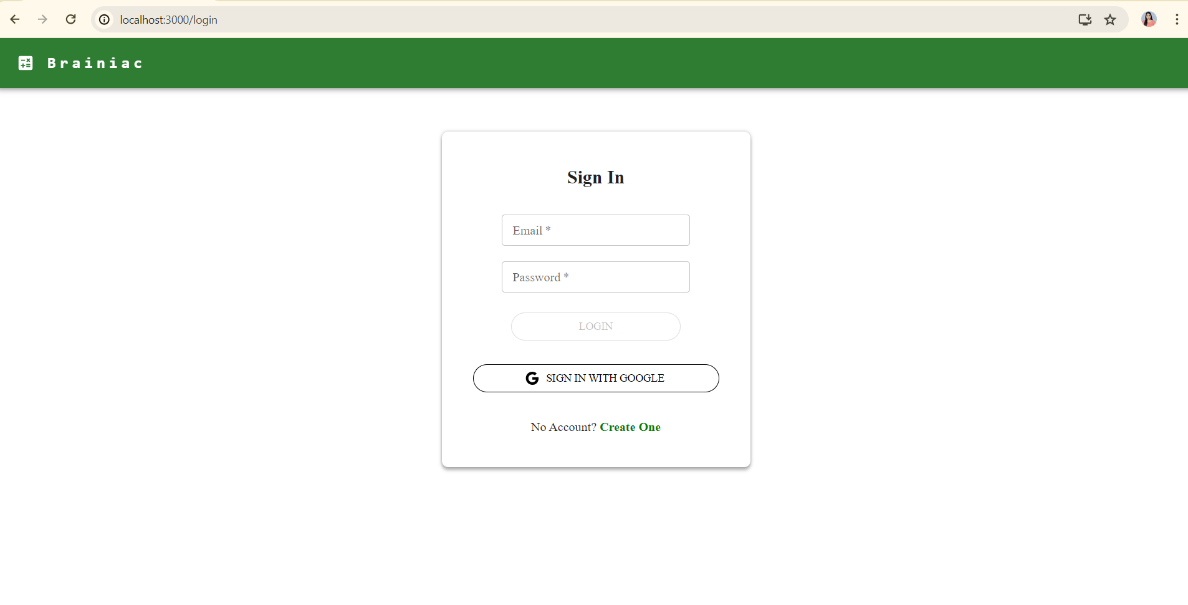
**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Topics | Page No. |
| 1 | User Authentication | **3-6** |
| 2 | Dashboard | **6-9** |
| 3 | Quiz System | **9-11** |
| 4 | Technologies Used | **11-12** |

**User Authentication**

User Authentication Page contain two options for users. Where user can login with Email Id and Password, or user can login with google.

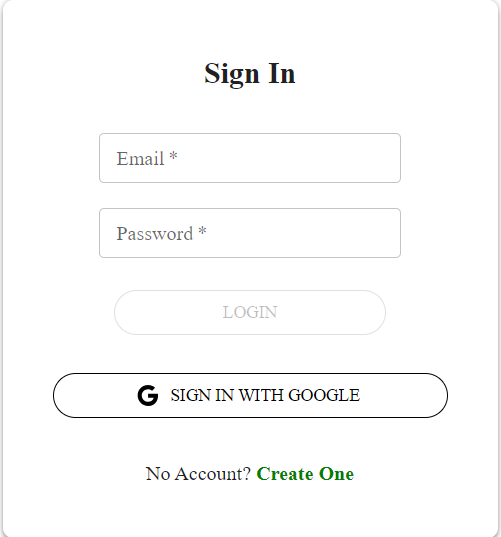
If user doesn’t have any account, user can click on Create One link to create new account.



* 1. User Authentication

1. **Implementation of Login System with Email and Password**

User need to provide valid credential for log in to the system.



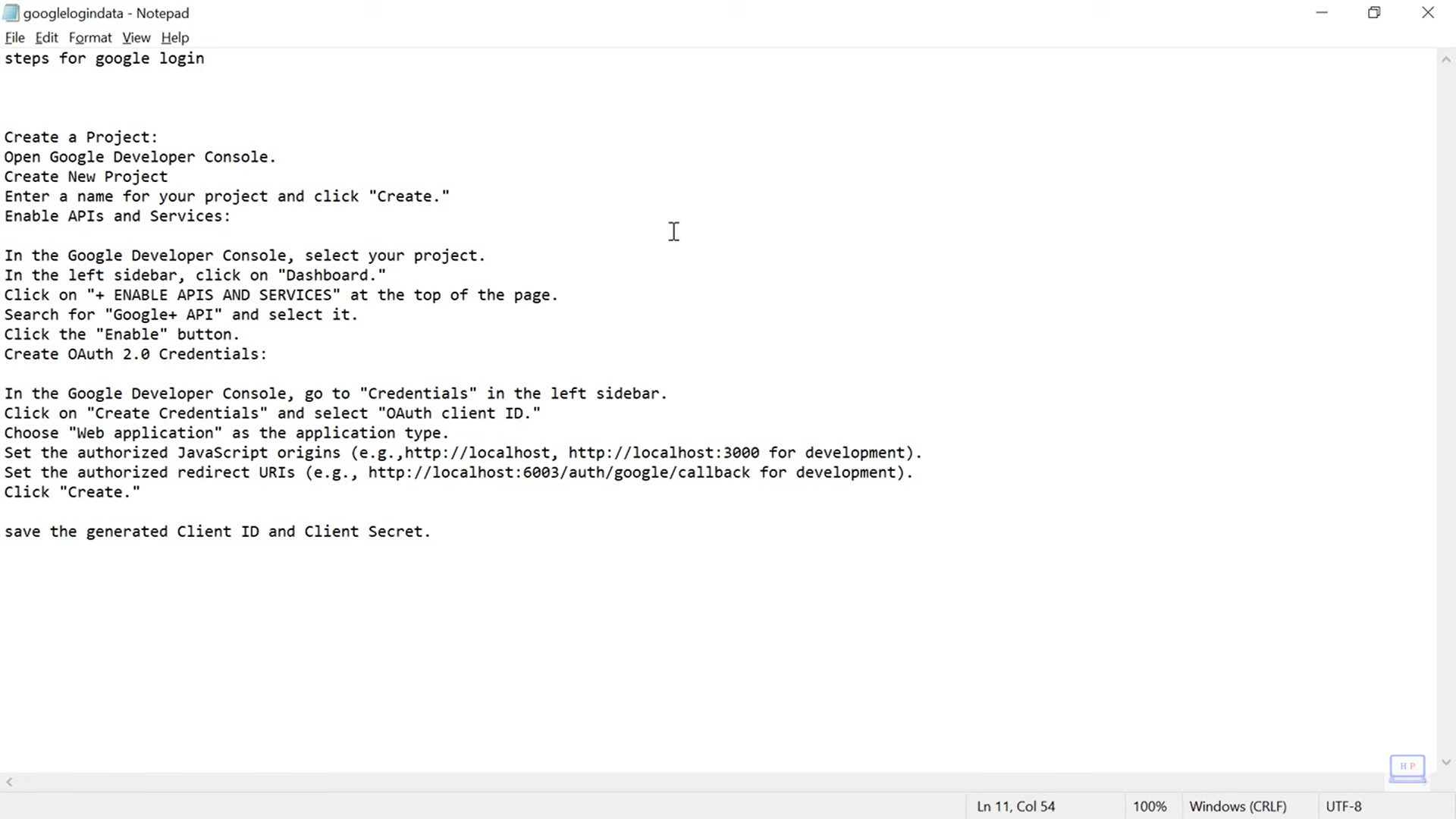
* 1. Log in with email

When user hit Login button, API fetch the user data check if user with this email and password is present in database. If user is present, it will redirect to the Dashboard page.



* 1. Log In error

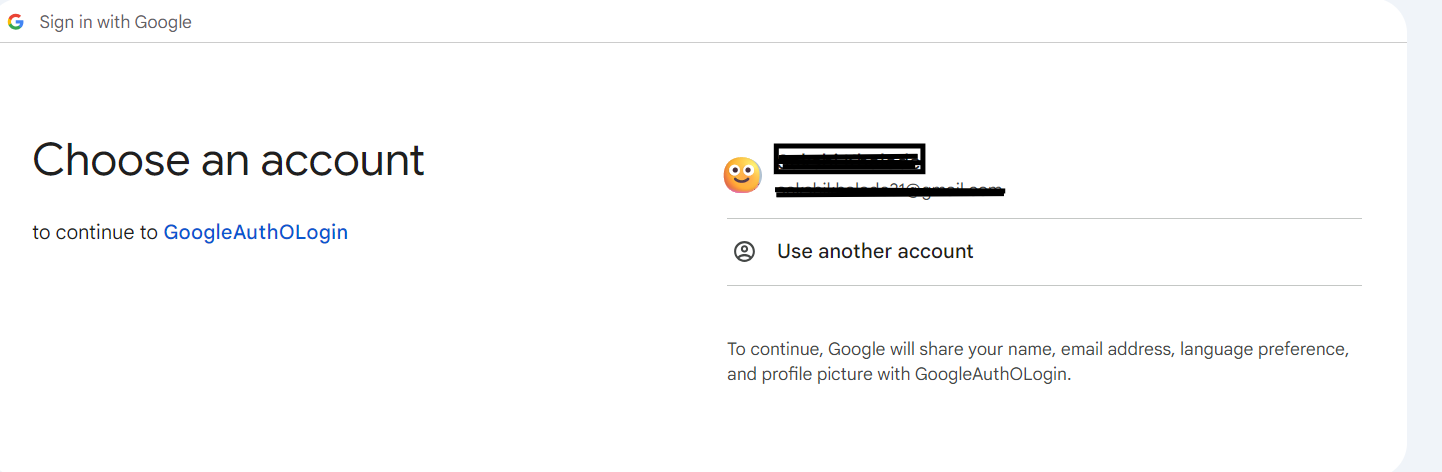
1. **Implementation of Google Authentication**



* 1. Steps to generate Client Id and Secret from google

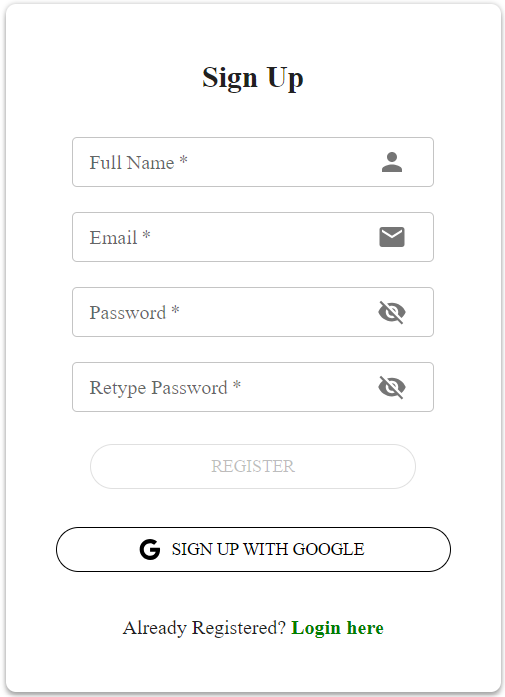
User needs to click on Sign in with Google, if user is already in DB user will be redirected to Dashboard page else user will be created. Same will happen with Sign Up with google.

When user hit the google auth button below page will open and user needs to select their Gmail option to proceed.



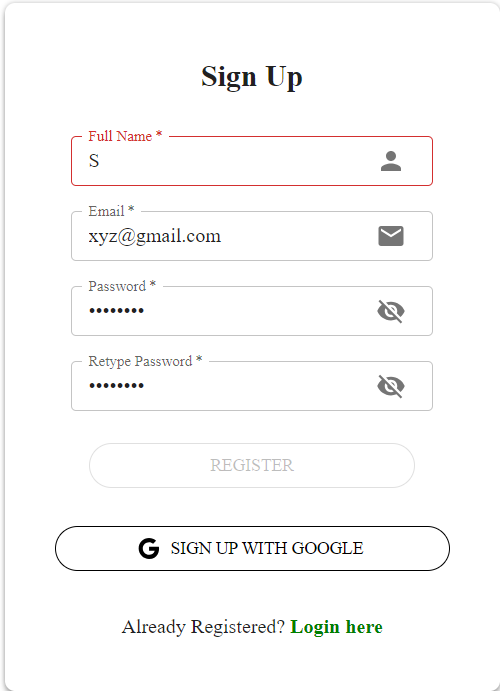
2.2 Google Authentication page

1. **Implementation of Sign-Up page**



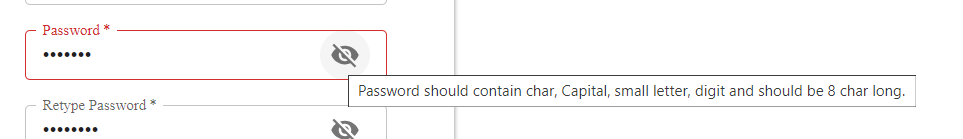
3.1 Sign Up page

On sign Up page user have to enter Name, Email, Password to sign up. As if data is not correct or empty Register button won’t be enabled until and unless correct data is provided.



3.2 Sign Up page with error

In above image as name is not correct as per validation, so input field is red and register button is disabled.



3.3 Sign Up Page with error details

On hovering right icon user can see specific error respected to that field. Once user registered successfully, form will show success message and redirect user to login.

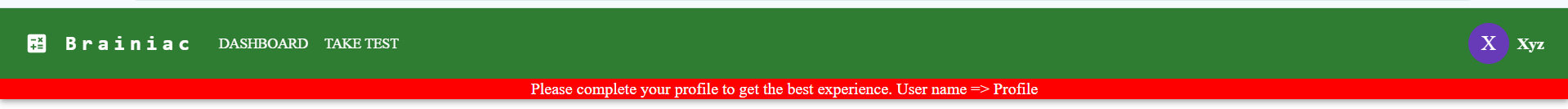
**Dashboard**

****

2.0 Dashboard

After successful login user will able to see above dashboard.

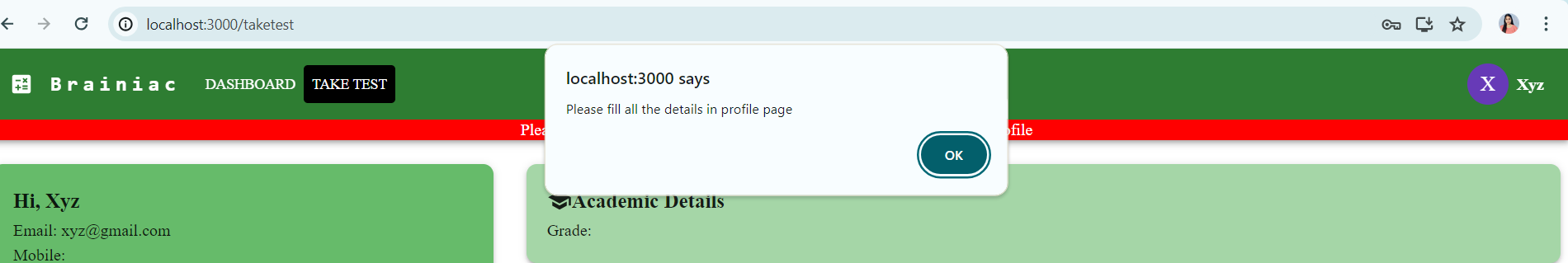
1. **Header**

****

1.1 Header

In header, we can see logo, username and two tabs for Dashboard and Take Test. Below that in red color we can see error, it will come when user details are not completed.

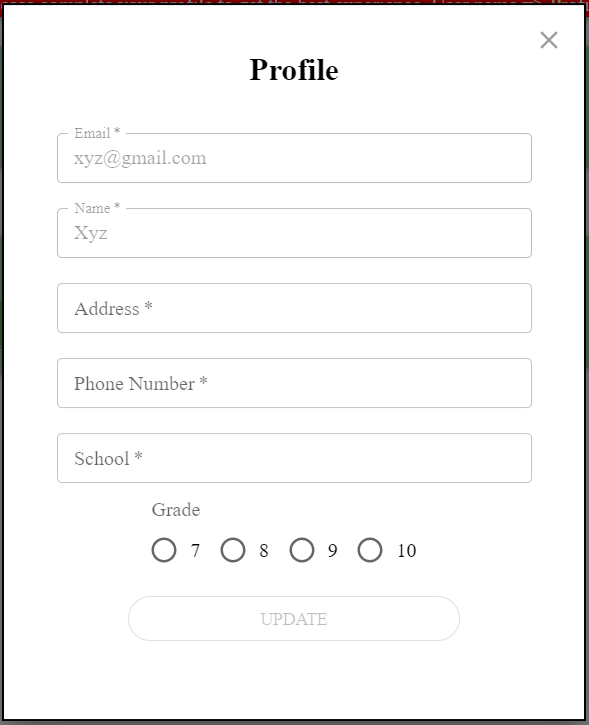
After clicking on Take Test user will redirect to test portal only if user details are fulfilled. Else user will get alert for same.



1.2 Error Take Test

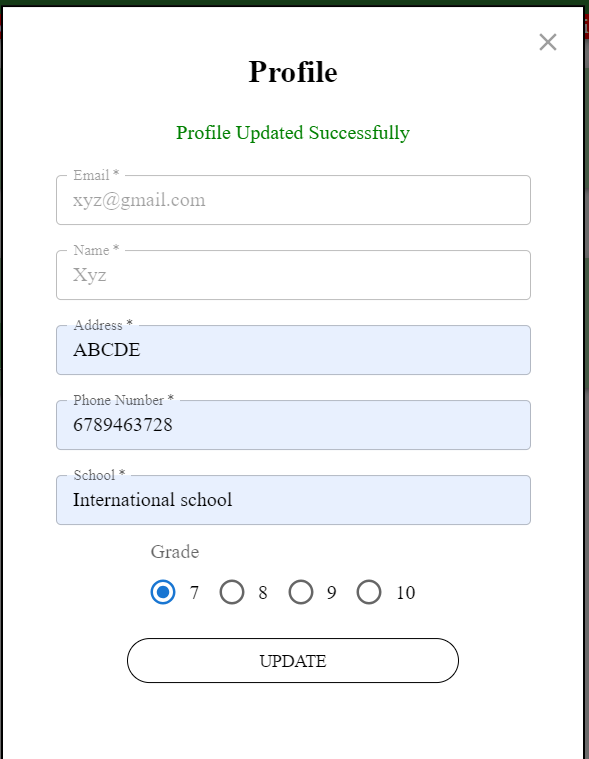
1. **User Details**

To complete user details, click on user name and select profile, user will able to see modal with user details form.



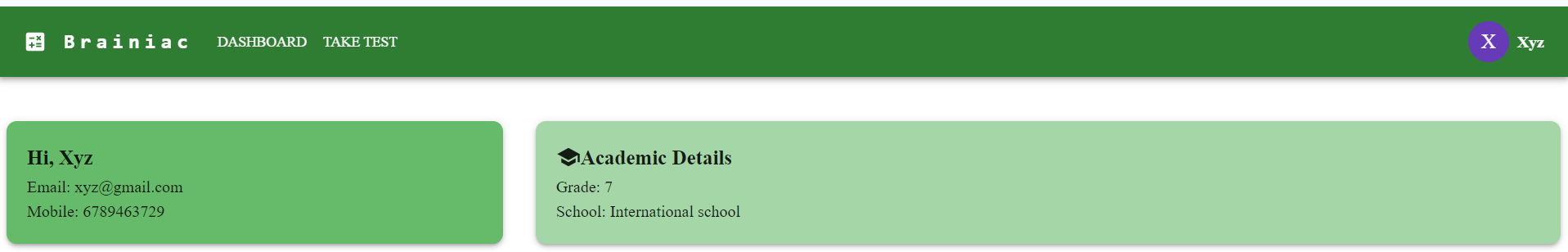
* 1. User Profile

Once user fill required details, details will be saved in backend and message will be print Profile Updated successfully and error will be gone.



* 1. Profile updated

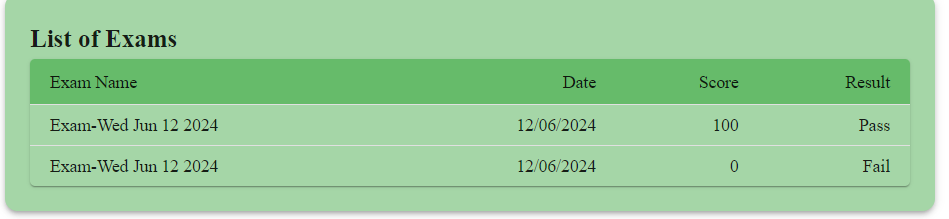
User details on dashboard will be updated as below without error. And user will able to take test.



2.3 User Data updated

1. **List of Exam**

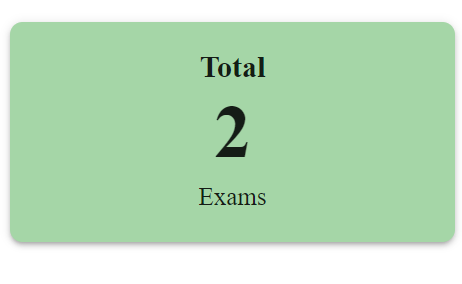
This part of dashboard shows list of exams with exam name(exam-date) with score and result.



* 1. List of Exam

1. **Total Exam**

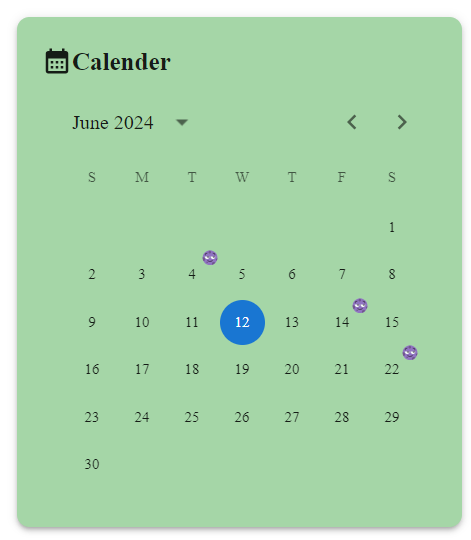
This part of dashboard shows total number of exams you have given.



* 1. Total Exams

1. **Calendar (Incomplete)**

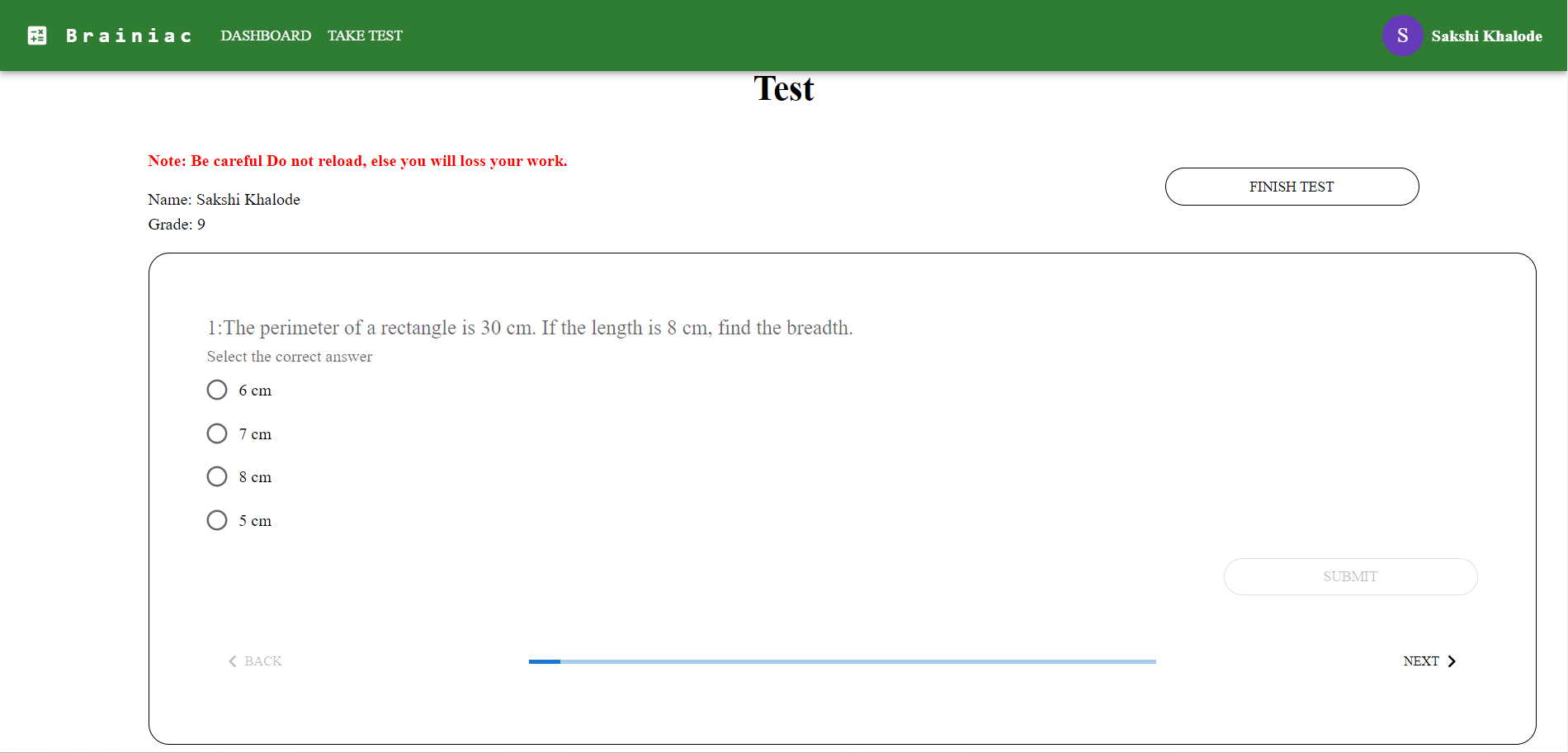
The Idea behind this, user will able to see on what date he/she given exam. It will also manipulate list of exam component.



* 1. Calendar

**Quiz System**

1. **Test**



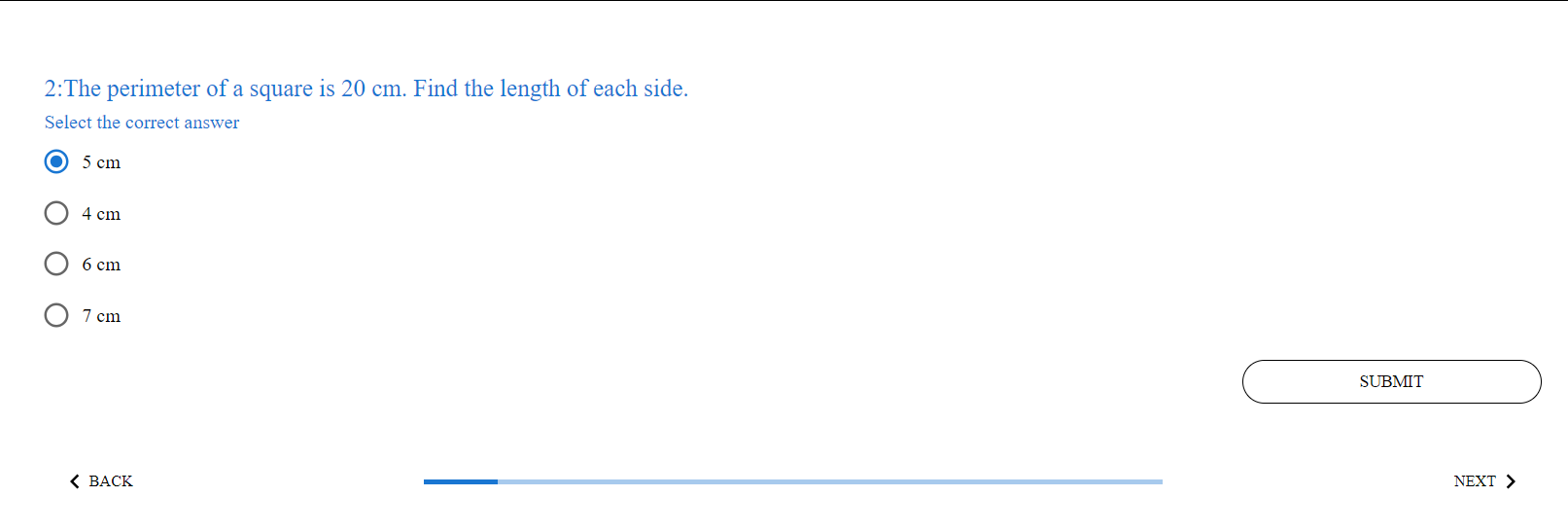
1.1 Test

Test dashboard will show Name of user and grade of user. This test contain question as per user grade. It will have Finish test button as shown, if user wants end exam user can do that.

Question box have, Question, option and submit button it will enable after user select answer. Also, this box contains back and next button to move as user wants.

1. **Submit**

Once select answer, submit button will be enabled and question will move to next if answer has been submitted. There are 20 total questions.

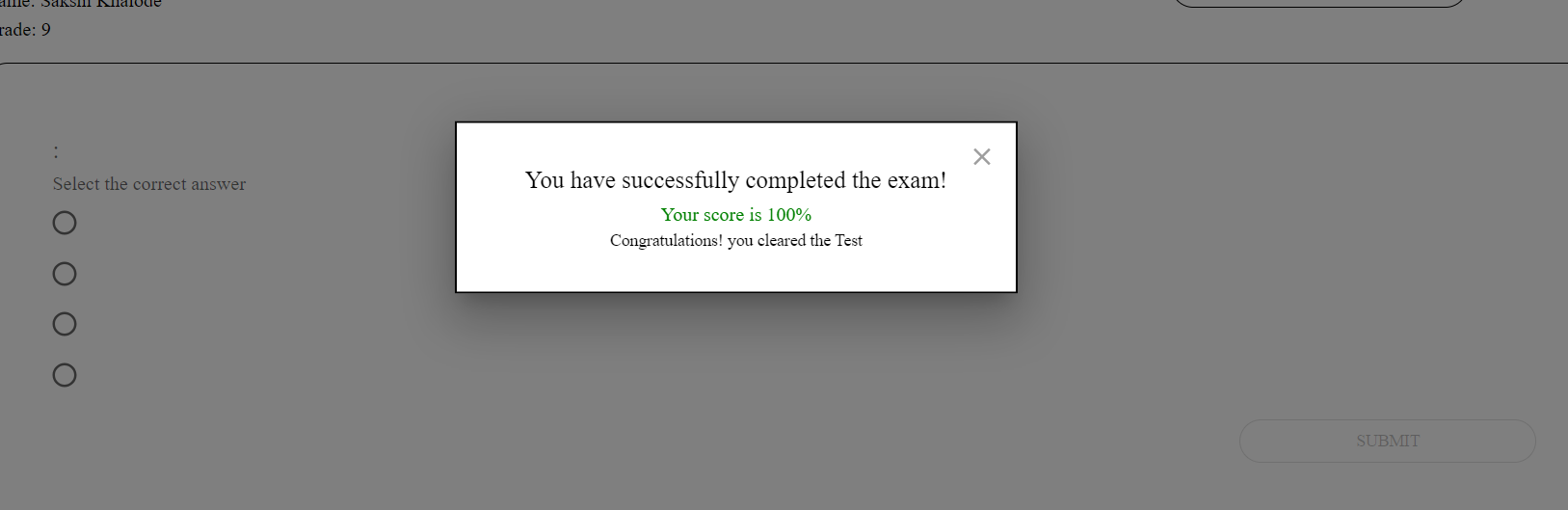


* 1. Question

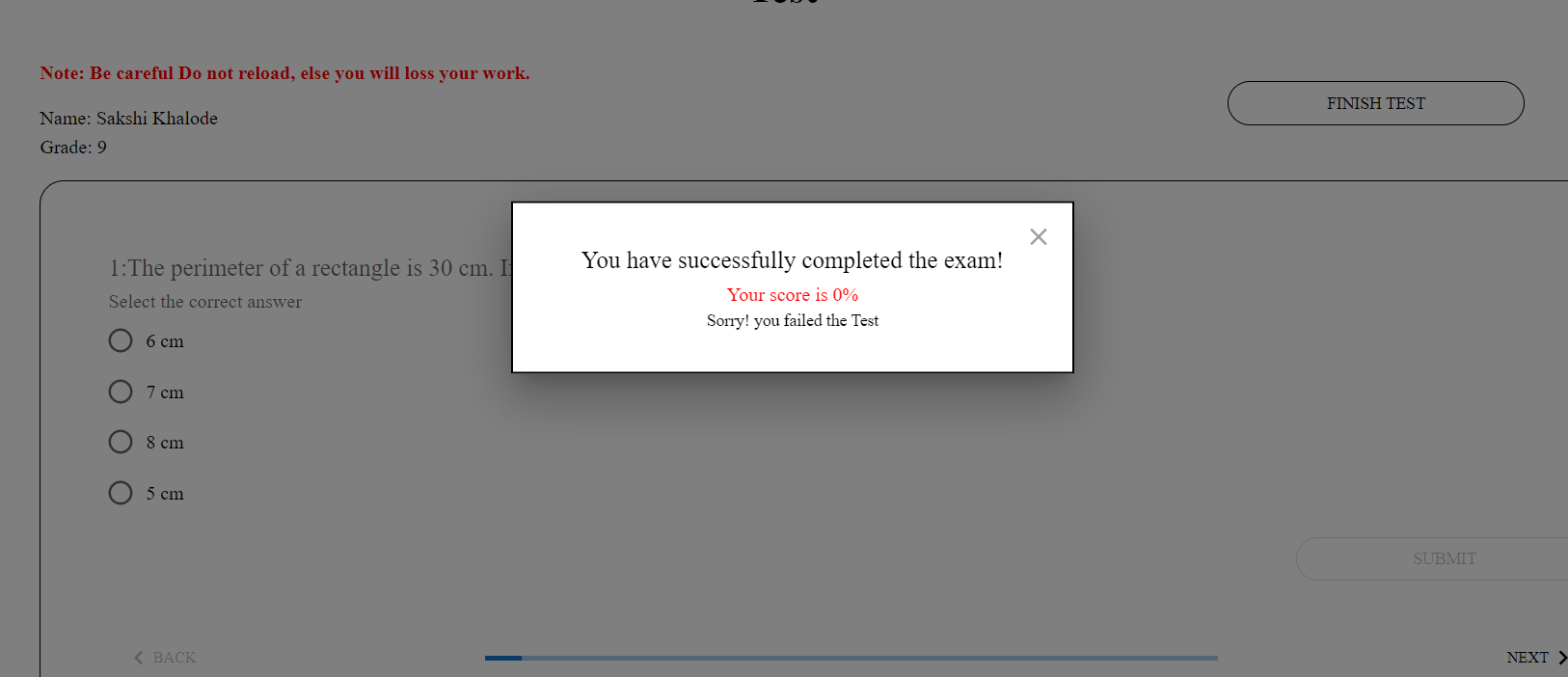
Progressor will increase as question level increases, back button disabled when there is no previous question and next will be disabled when question ends.

1. **Finish**

Once user click on finish button screen will pop up with result.



3.1 Pass



3.2 Fail

**Technologies Used**

1. **React**

**Project complexity:** React is well-suited for complex user interfaces with a lot of interactivities.

**Developer experience:** I am already familiar with JavaScript and web development concepts, React can be a good choice. However, there is a learning curve involved, it was worth considering a more beginner-friendly option.

1. **Redux**

Redux is a popular state management library for JavaScript applications, particularly those built with React. It provides a predictable and centralized way to manage the application's state.

**Centralized state management:** Redux stores the entire application state in a single source of truth, called the store. This eliminates the need to pass data down through component props, a practice known as prop drilling, which can become cumbersome and error-prone in complex applications.

**Improved testability:** Since the state updates are handled through pure functions (reducers), it makes testing your application logic much easier and more reliable.

**Scalability:** As your application grows in complexity, Redux can help you manage the state more effectively and keep your codebase maintainable.

1. **Node/Express**

**JavaScript runtime environment:** Node.js is an open-source, cross-platform runtime environment that executes JavaScript code outside of a web browser.

**Web framework:** Express is a popular web framework built on top of Node.js. It provides a robust set of features for building web applications and APIs.

1. **MongoDB**

**Document-oriented database:** Unlike relational databases that store data in tables with fixed schemas, MongoDB stores data in flexible documents, similar to JSON objects.

**Schema-less (optional schema):** MongoDB allows you to define schemas for your data, but it's not mandatory. This provides flexibility for evolving data models without requiring strict schema changes.

1. **Material UI**

Material UI is a popular open-source React component library that implements Google's Material Design. It provides a wide range of pre-built, customizable UI components that you can use to build attractive and functional user interfaces for your React applications.