

EXPERIMENT 4b

NAME: Shivam Pawar

UID: 2019230068

NAME: Vishal Salvi

UID: 2019230069

NAME: Shreyas Patel

UID: 2018130043

CLASS: TE COMPS

BATCH: C

DATE:

Aim: Understanding and use of Nodejs, Angularjs, Reactjs etc.

Theory:

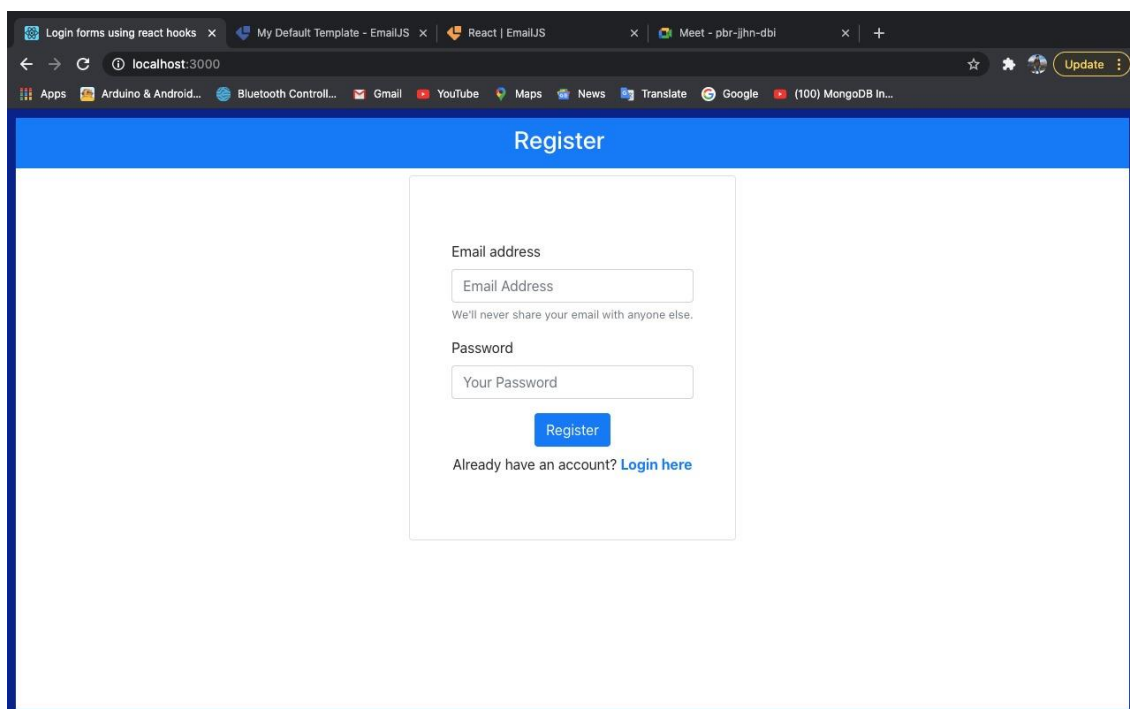
Prob Definition:

This application is developed for educational purpose, allowing the users to prepare the multiple-choice questions for different examinations. The main goal of the application is to enable users to practice for MCQ tests. Our application display score of user how many questions are right from test at the end of the result and also display at the time of submission of answer that selected answer is correct or not. After end of the test if user want to mail the score which is secured in test then user can click on Send mail button and user get email that how much score user get.

Also, login and registration are there to keep track of user to know which user score how many marks in respective quiz.

Screenshots:

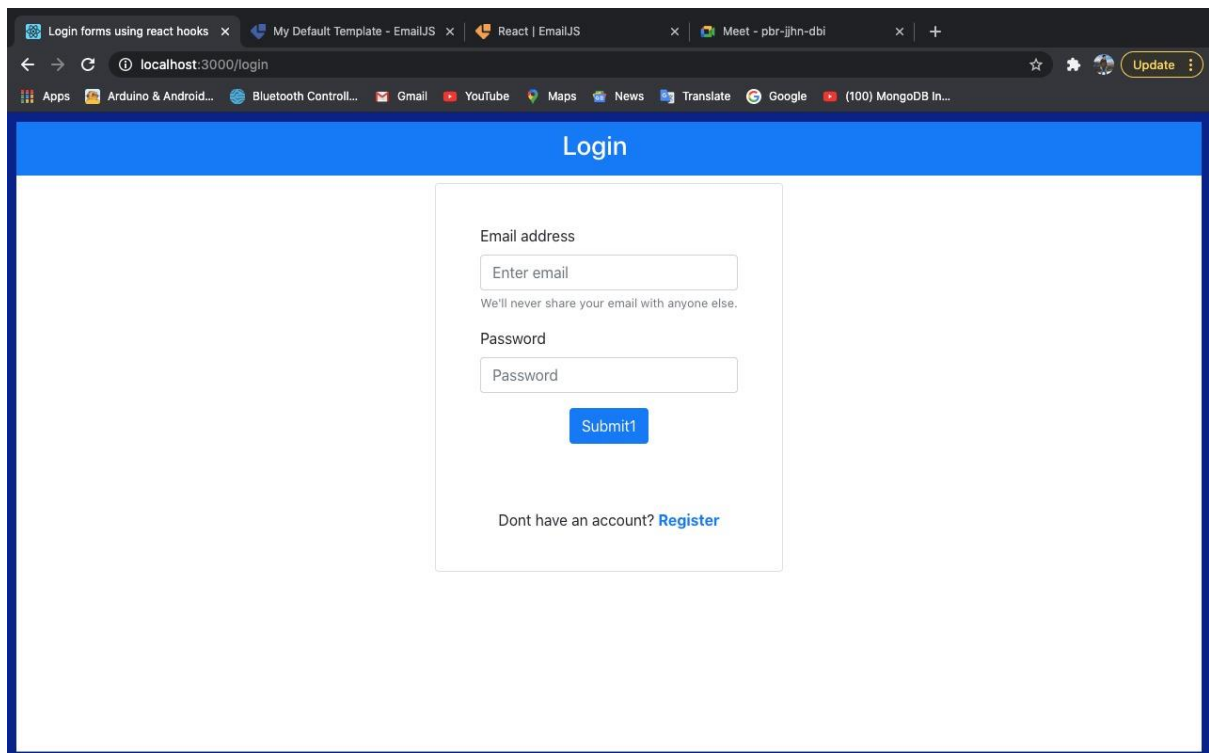
This is Registration page if user is not register in system the user need to first register and then user can log in the application.



The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'React | EmailJS'. The address bar shows 'localhost:3000'. The browser's toolbar includes various icons and an 'Update' button. The main content area displays a registration form with a blue header bar labeled 'Register'. The form contains two input fields: 'Email address' with a placeholder 'Email Address' and 'Password' with a placeholder 'Your Password'. Below the password field is a blue 'Register' button. At the bottom of the form, there is a link that says 'Already have an account? [Login here](#)'. A small text line above the password field states 'We'll never share your email with anyone else.'

After completion of registration user log into the application .

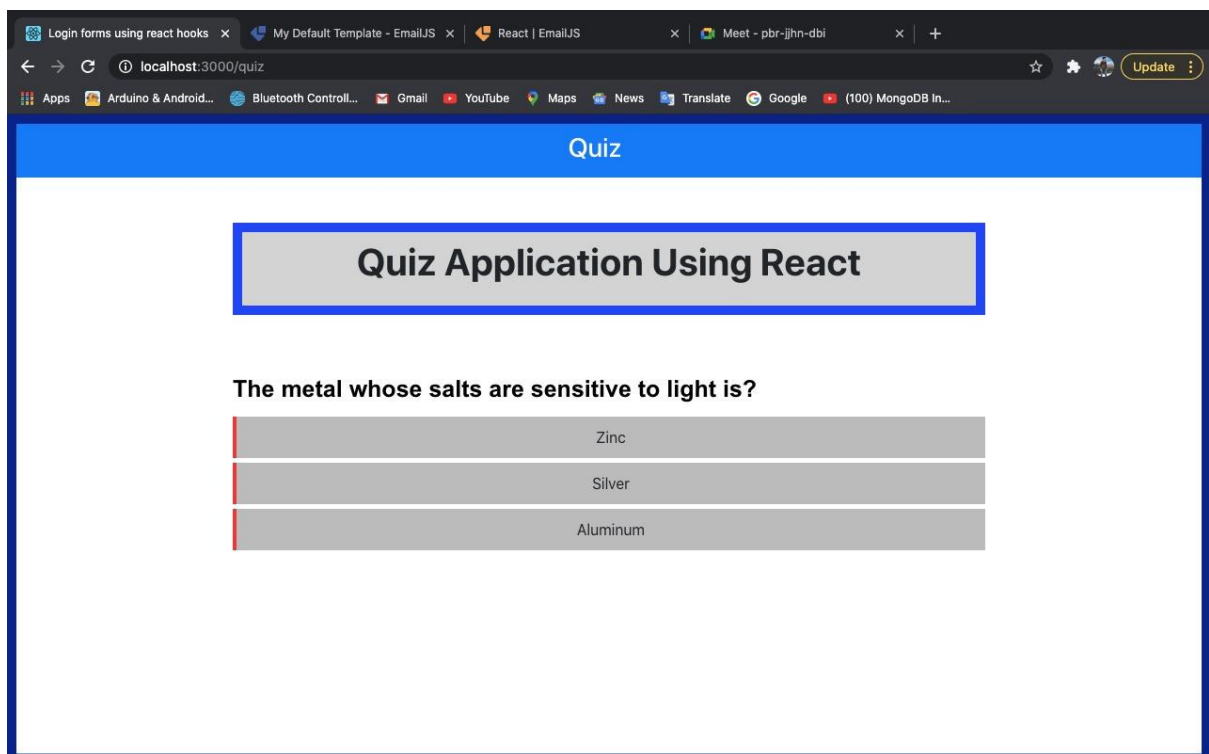
This is login page.



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The page has a blue header with the word 'Login' in white. Below the header, there is a white form box containing the following elements:

- Email address**: A text input field with the placeholder 'Enter email'.
- We'll never share your email with anyone else.
- Password**: A text input field with the placeholder 'Password'.
- Submit1**: A blue button with white text.
- Don't have an account? Register**: A link in blue text.

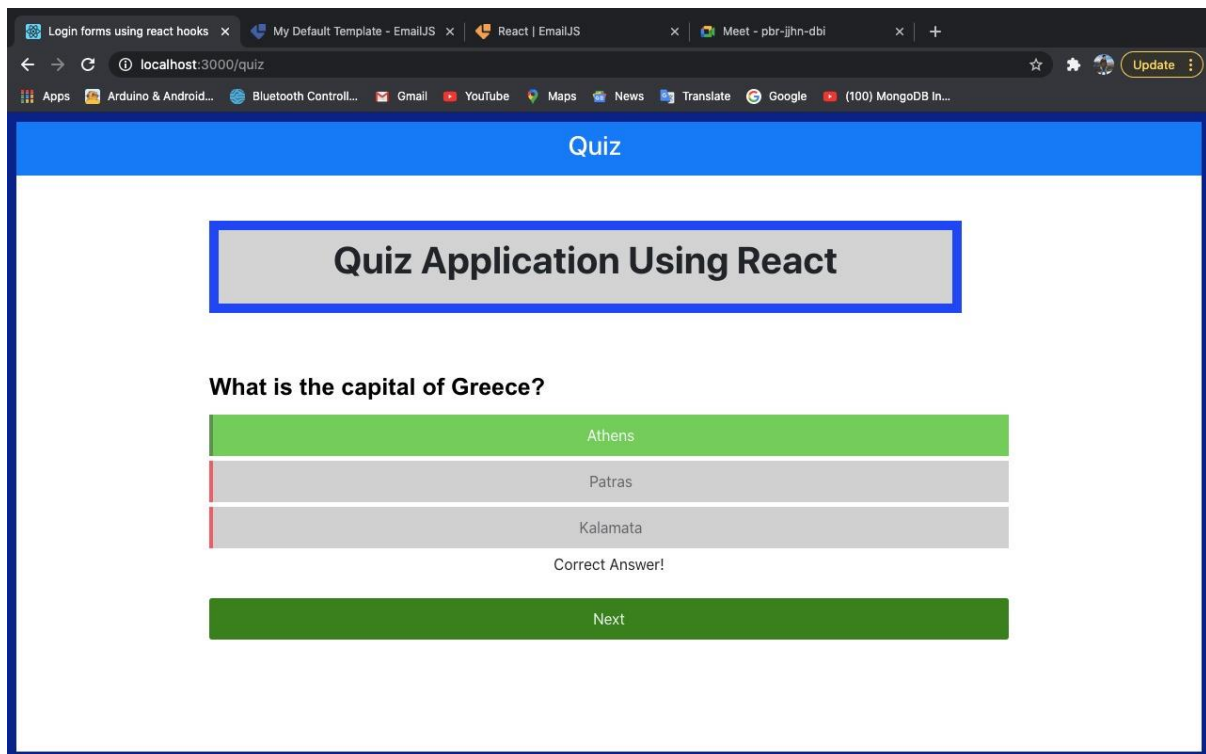
After login user can able to give quiz



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/quiz'. The page has a blue header with the word 'Quiz' in white. Below the header, there is a white form box containing the following elements:

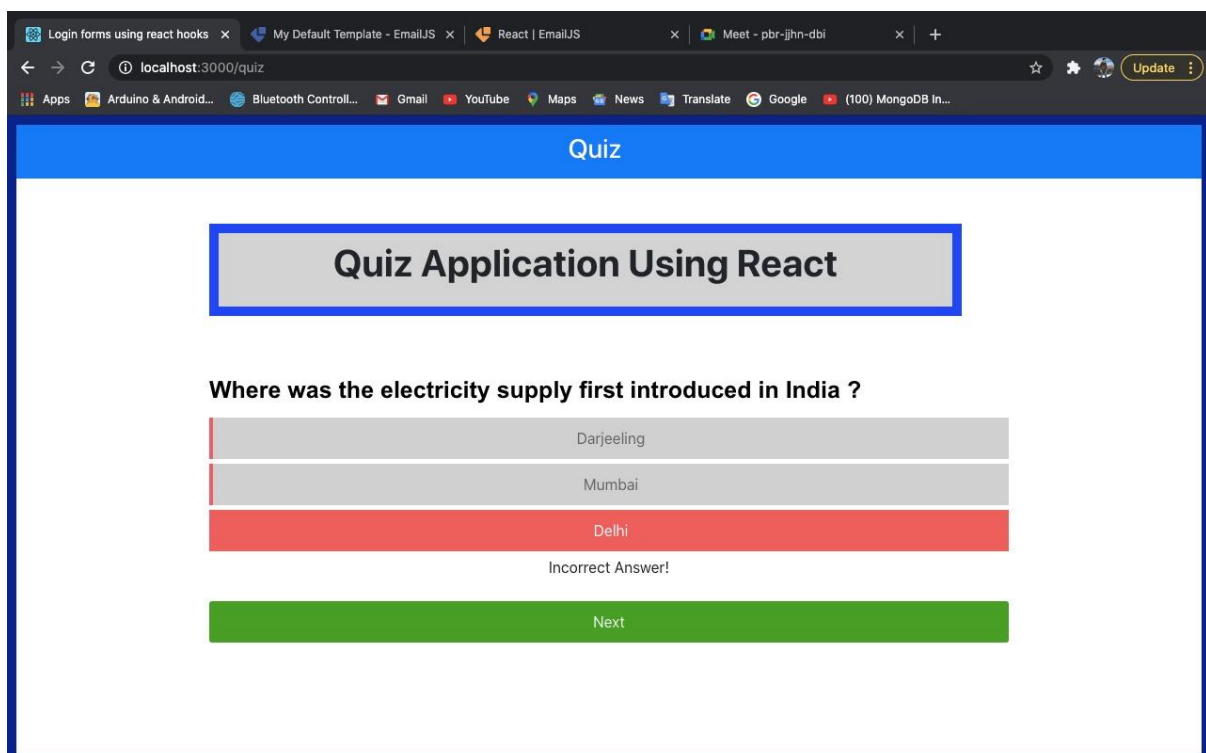
- Quiz Application Using React**: A title in bold black text, enclosed in a blue-bordered box.
- The metal whose salts are sensitive to light is?**: A question in bold black text.
- Options**: Three radio button options in a list:
 - ☐ Zinc
 - ☐ Silver
 - ☐ Aluminum

Whenever user select answer if that answer is correct then other options turn red and Correct answer text display on screen.

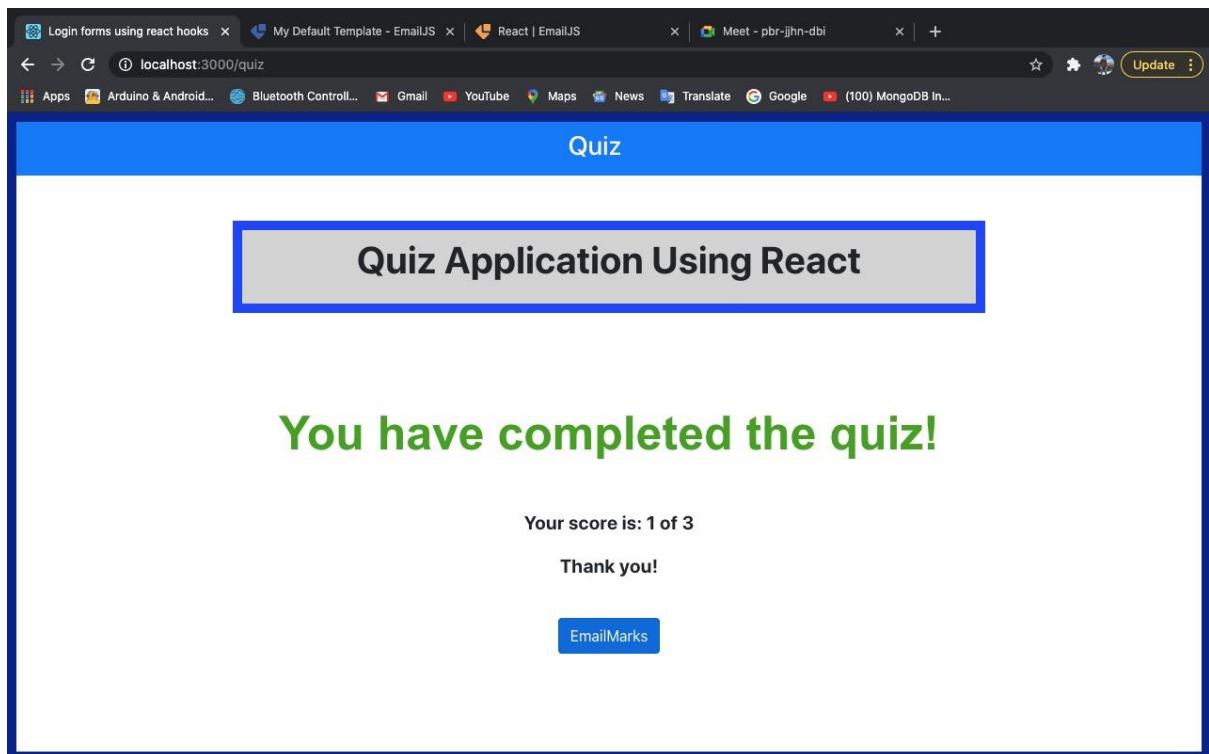


Incorrect Answer

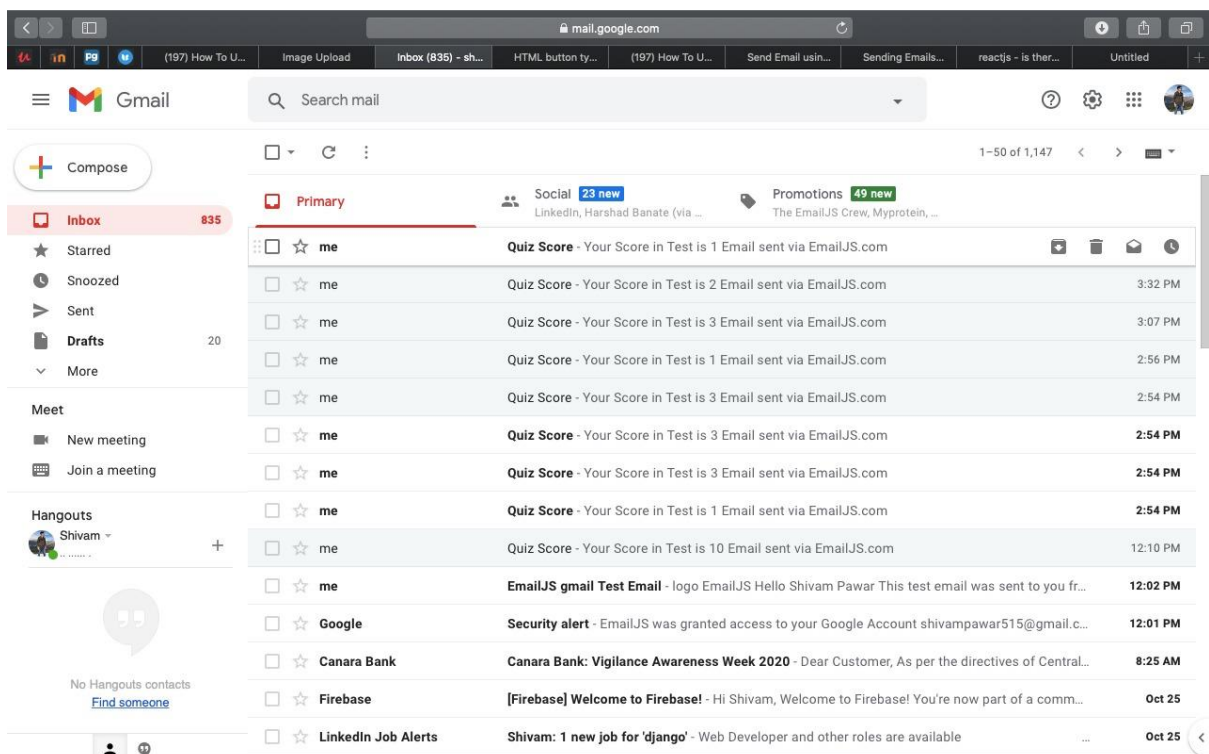
After click on next user get next question.



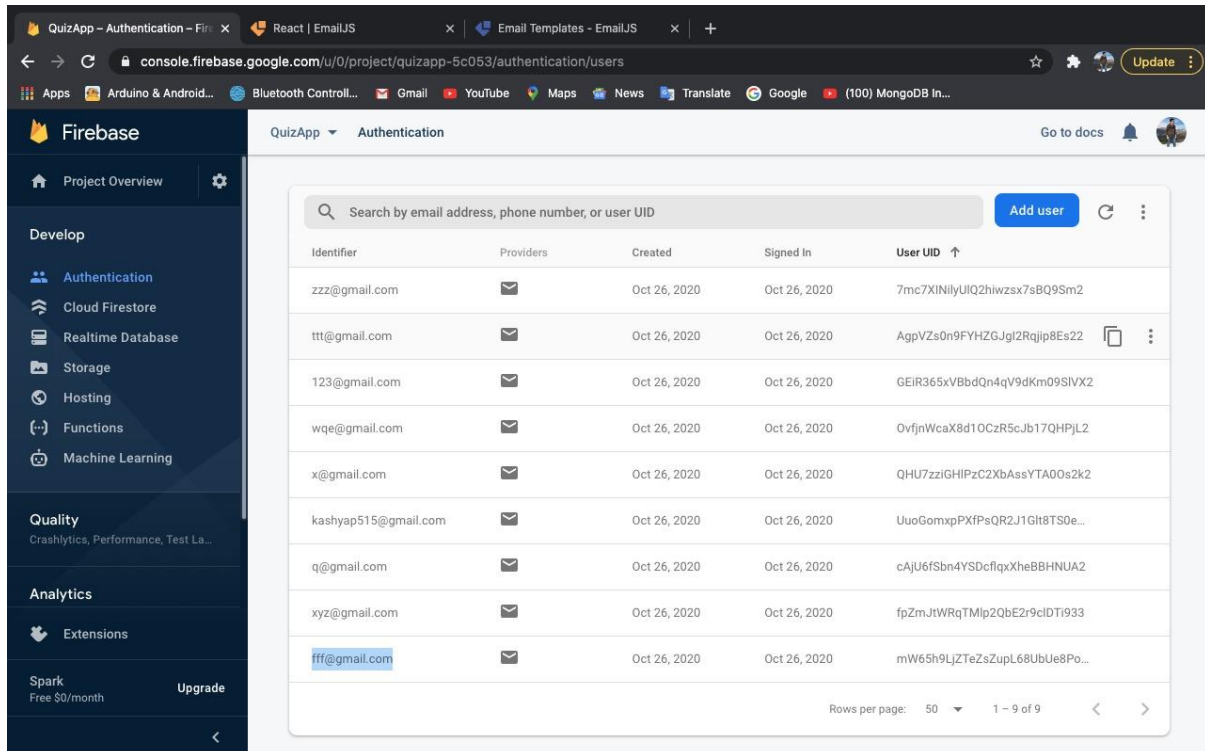
After completion of quiz user get overall score on screen.



After click on Email marks user get marks on Email.



This is our Firebase database whatever users are login our application that are store in database here.



The screenshot shows the Firebase Authentication console for a project named 'QuizApp'. The left sidebar contains navigation links for Project Overview, Develop (Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Quality (Crashlytics, Performance, Test Lab), Analytics, Extensions, and Spark (Free \$0/month, Upgrade). The main content area displays a table of users with columns: Identifier, Providers, Created, Signed In, and User UID. The table lists 10 users, all created on Oct 26, 2020, and signed in on the same date. The last user, 'fff@gmail.com', is highlighted. At the bottom right, it shows 'Rows per page: 50' and '1 - 9 of 9'.

Identifier	Providers	Created	Signed In	User UID
zzz@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	7mc7XINilyUIQ2hiwzsx7sBQ9Sm2
ttt@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	AgpVZs0n9FYHZGJgl2Rqip8Es22
123@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	GEIR365xVBbdQn4qV9dKm09SIVX2
wqe@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	OvfjnWcaX8d1OCzR5cJb17QHPJL2
x@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	QHU7zziGHIPzC2XbAssYTA00s2K2
kashyap515@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	UuoGomxpPXfsQR2J1Gh8TS0e...
q@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	cAjU6fSbn4YSdcflqxXheBBHNUA2
xyz@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	fpZmJtWRqTMlp2QbE2r9cIDTi933
fff@gmail.com	📧	Oct 26, 2020	Oct 26, 2020	mW65h9LjZTeZsZupL68UbUe8Po...

Code Snippet/Main Functions:

Firebase Database:

```
import firebase from 'firebase';

import auth from 'firebase'

var firebaseConfig = {
  apiKey: "AIzaSyDa_de6WxToTra5pmOvdwZ-c4iLXg_fCK4",
  authDomain: "quizapp-5c053.firebaseio.com",
  databaseURL: "https://quizapp-5c053.firebaseio.com",
  projectId: "quizapp-5c053",
  storageBucket: "quizapp-5c053.appspot.com",
  messagingSenderId: "7971630829",
  appId: "1:7971630829:web:1a004e4616e01cfb1ea6e4"
};

// Initialize Firebase

const fire = firebase.initializeApp(firebaseConfig);
```

```
export default fire;
```

Email Function:

```
sendEmail = (e) => {  
    const { score } = this.state;  
  
    this.setState({  
        score : score  
    });  
  
    let templateParams = {  
        sc : score,  
    }  
  
    emailjs.send('gmail', 'template_hcw9ewr', templateParams, 'user_kRPH7x1MvufSw094GmZvi')  
        .then((result) => {  
            console.log(result.text);  
        }, (error) => {  
            console.log(error.text);  
        });  
  
}
```

Registration:

```
registrationform(e) {  
    e.preventDefault();  
    fire.auth().createUserWithEmailAndPassword(this.state.email,this.state.password).catch((error) =>  
    { console.log(error);  
    });  
}  
  
handleChange(e){  
  
    this.setState({ [e.target.name]:e.target.value });
```

```
}
```

```
redirectToLogin = () => {  
  const { history } = this.props;  
  
  if(history) history.push('/login');  
}
```

Login:

```
loginform(e) {  
  e.preventDefault();  
  
  fire.auth().signInWithEmailAndPassword(this.state.email,this.state.password).catch((error) =>  
{ console.log(error);  
  });  
}
```

```
handleChange(e){  
  this.setState({ [e.target.name]:e.target.value });  
}
```

```
redirectToQuiz = () => {  
  const { history } = this.props;  
  
  if(history) history.push('/quiz');  
}
```

Conclusion:

Thus, we implement web application by using ReactJS Framework also learn the Email Verification and database connectivity by using Firebase database .