

Abstract

This is a Simple Chat Application Project in HTML, CSS, JS and for data base we have used firebase. In this the users can create their id with the help of email and then they can chat with each other

Key concept used:

1. Basics of HTML, CSS, Figma(to design the UI of the page),
Font Awesome(to use icons).
2. Prior Knowledge of Bootstrap, JavaScript.
3. How to set up the firebase and the functionalites of the
firebase.

OBJECTIVE

The objective of this project is to develop a user-friendly and responsive web-based chat application using a combination of HTML, CSS, JavaScript, and the Bootstrap framework. This chat application aims to provide a seamless and real-time messaging experience for users, allowing them to:

1. Communicate in real-time: Enable users to exchange text messages with one another instantly, promoting real-time communication.
2. User-friendly interface: Design an intuitive and user-friendly interface that is easy to navigate, making it accessible to users of all skill levels.
3. Responsive design: Ensure the application is responsive, adapting to various screen sizes and devices, including desktops, tablets, and mobile phones.
4. Multiple users: Allow multiple users to join and participate in chat conversations, facilitating group discussions and private messaging.

INTRODUCTION

In today's fast-paced digital world, communication has become more essential than ever. The need for efficient and user-friendly chat applications is evident, and this project aims to provide just that. Welcome to our Simple Chat Application, a web-based platform built using a combination of HTML, CSS, JavaScript, and Bootstrap.

Our Simple Chat Application is designed to provide a straightforward yet efficient means of communication, making it accessible to users of all levels of technical expertise. Whether you want to chat with friends, collaborate with colleagues, or simply stay connected with loved ones, our application offers an easy-to-use interface that ensures a smooth and enjoyable chatting experience.

Why This Project Matters:

The Simple Chat Application project not only showcases the power of web technologies like HTML, CSS, JavaScript, and Bootstrap but also addresses the ever-growing need for efficient communication tools. In an era where remote work and online collaboration have become the norm, a reliable and user-friendly chat application is essential for staying connected and productive. This project serves as a starting point for those interested in web development, providing valuable insights into building interactive and responsive web applications.

METHODOLOGY

1. Design and Wireframing:

Create wireframes and mockups of the application's user interface (UI) using design tools or paper sketches.

Develop a responsive design layout that adapts to various screen sizes, ensuring a consistent user experience.

2. Frontend Development:

Implement the frontend of the chat application using HTML for structure, CSS for styling, and JavaScript for interactivity.

Utilize the Bootstrap framework to enhance the UI design, responsiveness, and overall aesthetics.

Incorporate real-time messaging functionality using JavaScript libraries or frameworks (e.g., Socket.io) to enable instant communication.

3. User Authentication:

Develop a user authentication system to secure the application, allowing users to register, log in, and manage their profiles. Implement encryption and best security practices to protect user data.

4. Scalability and Future Development:

Plan for scalability by designing the application to accommodate a growing user base and additional features in the future.

Explore options for integrating additional functionalities like file sharing, multimedia support, or advanced chat features.

By following this methodology, we aim to systematically and efficiently develop a simple chat application that meets user requirements while maintaining high standards of quality, security, and user experience.

CODE

INDEX:-

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Login | onestop.ai | Simple Chat App</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzA
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-ic5ziedLmask
    <script>try{Typekit.load({async:true});}catch(e){}</script>
    <link href="style.css" rel="stylesheet">
  </head>
  <body>
    <main>
      <div class="container">
        <div class="row justify-content-center">
          <div class="card" style="width: 700px;">
            <div class="card-header">Login</div>
            <div class="card-body">
              <form>
                <div class="form-group row mt-2">
                  <label for="email" class="col-md-4 text-md-right">Email Address</label>
                  <div class="col-md-8">
                    <input type="text" id="email" class="form-control" name="email" autofocus placeholder="Enter your E
                  <span class="text-danger" id="emailErr"></span>
                </div> /.col-md-8
              </div> /.form-group.row.mt-2
              <div class="form-group row mt-2">
                <label for="password" class="col-md-4 text-md-right">Password</label>
                <div class="col-md-8">
                  <input type="password" id="password" class="form-control" name="password" autofocus placeholder="En
                </div> /.col-md-8
              </div> /.form-group.row.mt-2
            </div> /.card-body
          </div> /.card
        </div> /.row
      </div> /.container
    </main>
  </body>
</html>
```

The screenshot shows a code editor with a dark theme. The top navigation bar includes File, Edit, Selection, View, Go, Run, and a search bar labeled "chat application". On the left, there's a vertical toolbar with icons for file operations like Open, Save, Find, and Settings.

The main area displays the following HTML code:

```
<label for="password" class="col-md-4 text-md-right">Password</label>
<div class="col-md-8">
    <input type="password" id="password" class="form-control" name="password" autofocus placeholder="Enter your password here" />
    <span class="text-danger" id="passwordErr"></span>
</div>/.col-md-8
</div>/.form-group.row.mt-2
<div class="form-group row mt-2">
    <div class="col-md-4"></div>
    <div class="col-md-8">
        <div class="checkbox">
            <label>
                <input type="checkbox" name="remember"> Remember Me
            </label>
        </div>/.checkbox
    </div>/.col-md-8
</div>/.form-group.row.mt-2
<div class="form-group row mt-2">
    <div class="col-md-4"></div>
    <div class="col-md-8">
        <button type="button" id="authenticateUser" class="btn btn-primary">
            Login
        </button>/#authenticateUser.btn.btn-primary
        <p class="text-black mt-2 mb-2">I do not have an Account <a href="#signup.html" style="text-decoration: underline;">Sign Up</a>
    </div>/.col-md-8
</div>/.form-group.row.mt-2
</form>
</div>/.card-body
</div>/.card
```

The code is part of a card structure, specifically the card-body section. It includes a password input field, a remember me checkbox, and a primary button labeled "Login". There is also a link to "Sign Up". The code uses Bootstrap classes like col-md-4, col-md-8, and form-group, along with custom CSS classes like text-md-right, text-danger, and btn-primary.

```
53     </form>
54   </div>/.card-body
55 </div>/.card
56 </div>/.row.justify-content-center
57 </div>/.container
58 </main>
59 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
60 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-+ZT5ssoF4ZtElWVl51yTr0puNpduQvH95jY3oQwXl5EPCZJ+q7hRyGiwDg=="></script>
61 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmf</script>
62 <script src="validation.js"></script>
63 <script type="module">
64   import { initializeApp } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-app.js";
65   import { getAuth,signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-auth.js";
66   const firebaseConfig = [
67     apiKey: "AIzaSyBg9XJWm4L0n-ZT1SUqXOWS-Se6wp-sFhg",
68     authDomain: "chatapp-e602f.firebaseio.com",
69     databaseURL: "https://chatapp-e602f-default-rtdb.firebaseio.com",
70     projectId: "chatapp-e602f",
71     storageBucket: "chatapp-e602f.appspot.com",
72     messagingSenderId: "150115543267",
73     appId: "1:150115543267:web:0322faf04cfbeccee57dd6",
74     measurementId: "G-Z5K276QYET"
75   ];
76   const app = initializeApp(firebaseConfig);
77   function authenticateFromFirebase(){
78     if(validateLoginInputCredentials()){
79       const auth = getAuth(app);
80       signInWithEmailAndPassword(auth,email,password)
81       .then((userCredentials)=>{
82         const user = userCredentials.user;
83         localStorage.setItem('loggedin_user_id',user.uid)
84         localStorage.setItem('loggedin_user_email',user.email)
85         setTimeout(location.href="start-chat.html",2000)
86       }).catch((error)=>{
87         const errorCode = error.code;
88         const errorMessage = error.message;
89       })
90     }
91   }
92   $("#authenticateUser").on('click',function(){
93     authenticateFromFirebase();
94   })
95 </script>
96 </body>
97 </html>
```

```
76   const app = initializeApp(firebaseConfig);
77   function authenticateFromFirebase(){
78     if(validateLoginInputCredentials()){
79       const auth = getAuth(app);
80       signInWithEmailAndPassword(auth,email,password)
81       .then((userCredentials)=>{
82         const user = userCredentials.user;
83         localStorage.setItem('loggedin_user_id',user.uid)
84         localStorage.setItem('loggedin_user_email',user.email)
85         setTimeout(location.href="start-chat.html",2000)
86       }).catch((error)=>{
87         const errorCode = error.code;
88         const errorMessage = error.message;
89       })
90     }
91   }
92   $("#authenticateUser").on('click',function(){
93     authenticateFromFirebase();
94   })
95 </script>
96 </body>
97 </html>
```

SIGN UP:-

A screenshot of a code editor showing the initial state of a sign-up form. The file is `signup.html`. The code includes meta tags, a title, and a main container with a card for sign-up. The card header says "Sign Up". The card body contains a form with fields for "User name" and "Email Address". The "User name" field has an autofocus attribute and a placeholder "Enter your name". The "Email Address" field also has an autofocus attribute and a placeholder "Enter your email address". There are validation spans for both fields.

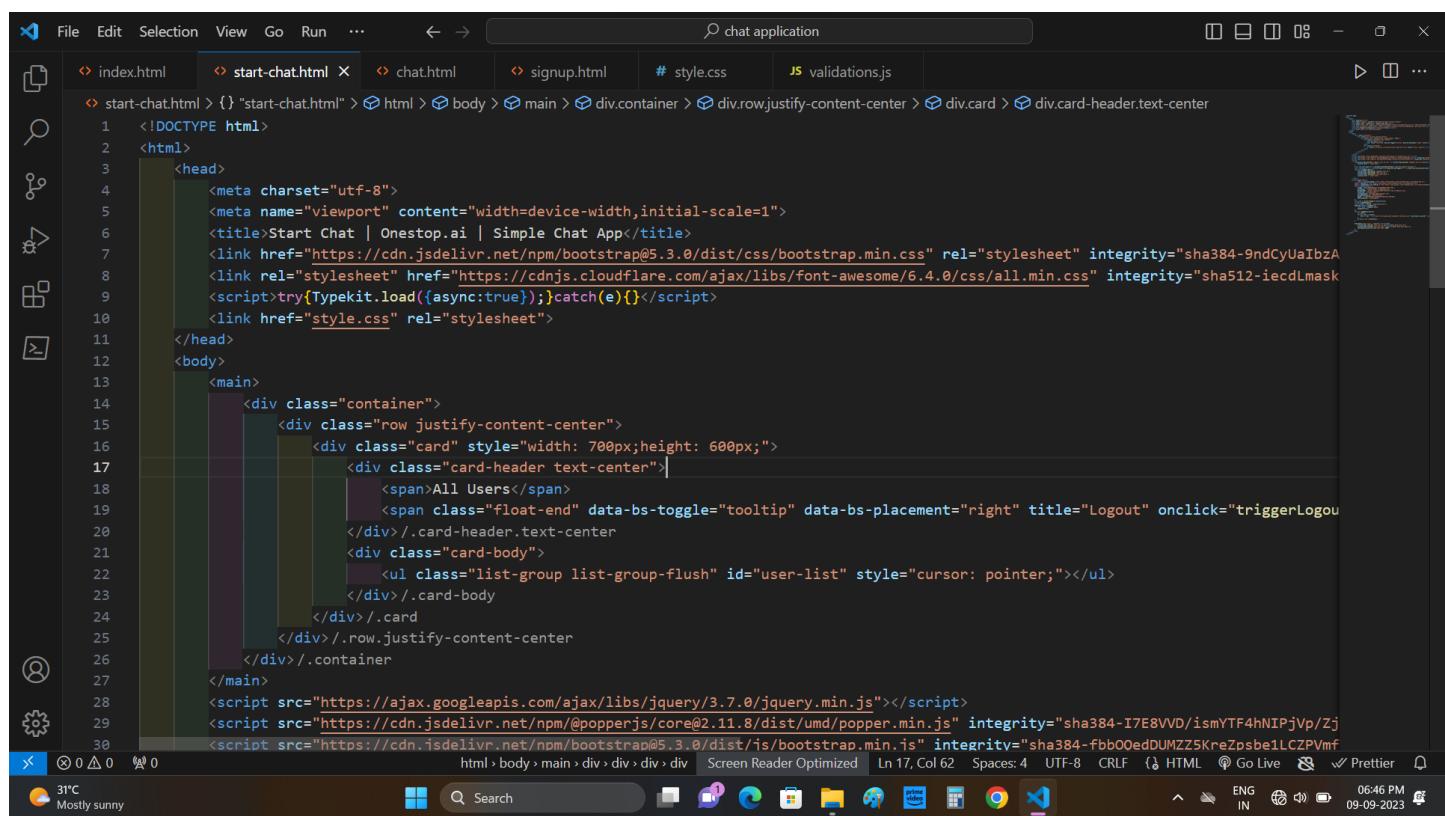
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Signup | onestop.ai | Simple Chat App</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzA" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-iecdLmask" />
    <script>try{Typekit.load({async:true});}catch(e){}</script>
    <link href="style.css" rel="stylesheet">
</head>
<body>
    <main>
        <div class="container">
            <div class="row justify-content-center">
                <div class="card" style="width: 700px;">
                    <div class="card-header">Sign Up</div>
                    <div class="card-body">
                        <form>
                            <div class="form-group row mt-2">
                                <label for="name" class="col-md-4 text-md-right">User name </label>
                                <div class="col-md-8">
                                    <input type="text" id="name" class="form-control" name="name" autofocus placeholder="Enter your name" />
                                    <span class="text-danger" id="nameErr"></span>
                                </div>
                            </div>
                            <div class="form-group row mt-2">
                                <label for="email" class="col-md-4 text-md-right">Email Address</label>
                                <div class="col-md-8">
                                    <input type="text" id="email" class="form-control" name="email" autofocus placeholder="Enter your email address" />
                                    <span class="text-danger" id="emailErr"></span>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </main>
    </body>

```

A screenshot of a code editor showing the completed sign-up form. The file is `signup.html`. The code now includes fields for "Email Address", "Password", "Confirm Password", and a "Save User" button. The "Email Address" and "Password" fields have autofocus attributes and placeholders. Validation spans are present for each. The "Save User" button is labeled "Signup". A link to the index page is provided for users who already have an account.

```
<label for="email" class="col-md-4 text-md-right">Email Address</label>
<div class="col-md-8">
    <input type="text" id="email" class="form-control" name="email" autofocus placeholder="Enter your email address" />
    <span class="text-danger" id="emailErr"></span>
</div>
</div>
<div class="form-group row mt-2">
    <label for="password" class="col-md-4 text-md-right">Password</label>
    <div class="col-md-8">
        <input type="password" id="password" class="form-control" name="password" autofocus placeholder="Enter your password" />
        <span class="text-danger" id="passwordErr"></span>
    </div>
</div>
<div class="form-group row mt-2">
    <label for="confirm_password" class="col-md-4 text-md-right">Confirm Password</label>
    <div class="col-md-8">
        <input type="password" id="confirm_password" class="form-control" name="confirm_password" autofocus placeholder="Enter your password again" />
        <span class="text-danger" id="confirmPasswordErr"></span>
    </div>
</div>
<div class="form-group row mt-2">
    <div class="col-md-4"></div>
    <div class="col-md-8">
        <button type="button" id="saveUser" class="btn btn-primary">
            Signup
        </button>
        <a href="index.html" style="text-decoration: none; color: inherit;">I already have an Account </a>
    </div>
</div>
```

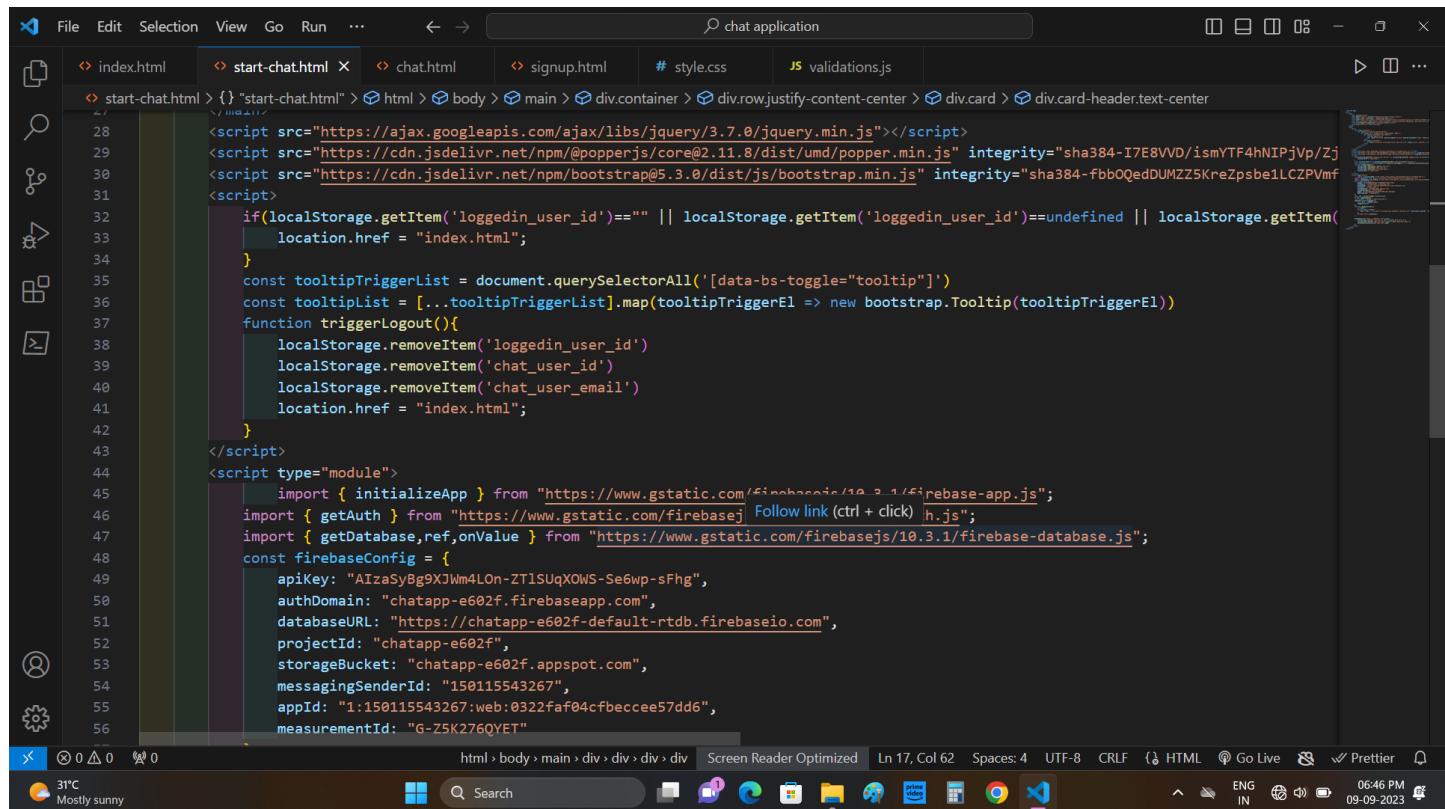

START-CHAT:-



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Start Chat | Onestop.ai | Simple Chat App</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUabzA" data-bbox="118 168 891 181"/>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-ieCDLmask" data-bbox="118 183 891 196"/>
    <script>try{Typekit.load({async:true});}catch(e){}</script>
    <link href="style.css" rel="stylesheet">
  </head>
  <body>
    <main>
      <div class="container">
        <div class="row justify-content-center">
          <div class="card" style="width: 700px; height: 600px;">
            <div class="card-header text-center">
              <span>All Users</span>
              <span class="float-end" data-bs-toggle="tooltip" data-bs-placement="right" title="Logout" onclick="triggerLogout()></span>
            </div>/.card-header.text-center
            <div class="card-body">
              <ul class="list-group list-group-flush" id="user-list" style="cursor: pointer;"></ul>
            </div>/.card-body
          </div>/.card
        </div>/.row.justify-content-center
      </div>/.container
    </main>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zj" data-bbox="118 378 891 391"/>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmf" data-bbox="118 393 891 406"/>
  </body>
</html>
```

31°C Mostly sunny

Screen Reader Optimized Ln 17, Col 62 Spaces: 4 UTF-8 CRLF ⚡ HTML Go Live Prettier ENG IN 09-09-2023 06:46 PM



```
if(localStorage.getItem('loggedin_user_id') === "" || localStorage.getItem('loggedin_user_id') === undefined || localStorage.getItem('loggedin_user_id') === null) {
  location.href = "index.html";
}

const tooltipTriggerList = document.querySelectorAll('[data-bs-toggle="tooltip"]');
const tooltipList = [...tooltipTriggerList].map(tooltipTriggerEl => new bootstrap.Tooltip(tooltipTriggerEl))

function triggerLogout(){
  localStorage.removeItem('loggedin_user_id')
  localStorage.removeItem('chat_user_id')
  localStorage.removeItem('chat_user_email')
  location.href = "index.html";
}

</script>
<script type="module">
  import { initializeApp } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-app.js";
  import { getAuth } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-auth.js";
  import { getDatabase, ref, onValue } from "https://www.gstatic.com/firebasejs/10.3.1.firebaseio-database.js";
  const firebaseConfig = {
    apiKey: "AIzaSyBg9XJWm4LOn-ZTlSuqXOWS-Se6wp-sFhg",
    authDomain: "chatapp-e602f.firebaseio.com",
    databaseURL: "https://chatapp-e602f-default-firebaseio.com",
    projectId: "chatapp-e602f",
    storageBucket: "chatapp-e602f.appspot.com",
    messagingSenderId: "150115543267",
    appId: "1:150115543267:web:0322faf04cfbecce57dd6",
    measurementId: "6-Z5K276QYET"
  }
</script>
```

31°C Mostly sunny

Screen Reader Optimized Ln 17, Col 62 Spaces: 4 UTF-8 CRLF ⚡ HTML Go Live Prettier ENG IN 09-09-2023 06:46 PM

```
54     messagingSenderId: "150115543267",
55     appId: "1:150115543267:web:0322faf04cfbeccee57dd6",
56     measurementId: "G-Z5K276QYET"
57   };
58   const app = initializeApp(firebaseConfig);
59   var db = getDatabase();
60   const userRef = ref(db, "/registered_users")
61   onValue(userRef, (snapShot)=>{
62     const data = snapShot.val();
63     showUsers(data)
64   })
65   function showUsers(users){
66     var html="";
67     for(let user in users){
68       html = html + '<li class="list-group-item loadchat" data-user_id="'+users[user].userid+'" data-user_email="'+users[user].email+'</li>';
69     }
70     $("#user-list").html(html);
71   }
72
73   $(document).on('click','.loadchat',function(e){
74     localStorage.setItem('chat_user_id',$(this).attr('data-user_id'))
75     localStorage.setItem('chat_user_email',$(this).attr('data-user_email'))
76     setTimeout(location.href="chat.html",2000)
77   })
78
79 </script>
80 </body>
</html>
```

CHAT:-

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width,initial-scale=1">
6      <title>Chat | onestop.ai | Simple Chat App</title>
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzA5LmzJnVqfZjwvDyHkzXWzPjzCzBzJLcJLdJLmzJnVqfZjwvDyHkzXWzPjzCzBzJLcJLdJ"/>
8      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" integrity="sha512-iecdLmaskSOFAFZqKPCtLNlOgq33AqTlGpaWpLaF3GqtWxzi5SfxSpNGZeoE3P/pft0mquxV3mOpPAS7dZo0w"/>
9      <link href="style.css" rel="stylesheet">
10
11 </head>
12 <body>
13   <main>
14     <div class="container">
15       <div class="row justify-content-center">
16         <div class="card" style="width: 700px;height: 600px;">
17           <div class="card-header text-center">
18             <span class="float-start" data-bs-toggle="tooltip" data-bs-placement="right" title="Go Back to Start Chat" onclick="goBackToStartPage()"></span>
19             <span id="email-label"></span>
20             <span class="float-end" data-bs-toggle="tooltip" data-bs-placement="right" title="Logout" onclick="triggerLogout()"></span>
21           </div> /.card-header.text-center
22           <div class="card-body">
23             <div class="chats" id="chatBox">
24               <ul class="chat-bubble-container">
25                 <li class="sent">
26                   <p class="chat-bubble from-message">Hi How are you?</p>
27                 </li> /.sent
28                 <li class="reply">
29                   <p class="chat-bubble to-message">Hi! I'm doing good thank you!</p>
30                 </li> /.reply
31               </ul> /.chat-bubble-container
32             </div> #chatBox
33           </div> #card-body
34         </div> #card
35       </div> #row
36     </div> #container
37   </main>
38
39 </body>
40 </html>
```

```
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/ZJ</script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmf</script>
    <script>
        if(localStorage.getItem('loggedin_user_id')=="") || localStorage.getItem('loggedin_user_id')==undefined || localStorage.getItem('chat_user_email')==""
            location.href = "index.html";
        }
        document.getElementById('email-label').innerHTML = "Chatting with <b>" + localStorage.getItem('chat_user_email') + "</b>";
        const tooltipTriggerList = document.querySelectorAll('[data-bs-toggle="tooltip"]');
        const tooltipList = [...tooltipTriggerList].map(tooltipTriggerEl => new bootstrap.Tooltip(tooltipTriggerEl))

        function triggerLogout(){
            localStorage.removeItem('loggedin_user_id')
            localStorage.removeItem('chat_user_id')
            localStorage.removeItem('chat_user_email')
            location.href = "index.html";
        }

        function goBackToStartPage(){
            localStorage.removeItem('chat_user_id')
            localStorage.removeItem('chat_user_email')
            location.href = "start-chat.html";
        }
    </script>
    <script type="module">
        import { initializeApp } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-app.js";
        import { getAuth } from "https://www.gstatic.com/firebasejs/10.3.1.firebaseio-auth.js";
        import { getDatabase, ref, set, push, onDataChange } from "https://www.gstatic.com/firebasejs/10.3.1.firebaseio-database.js";
        const firebaseConfig = {
            apiKey: "AIzaSyB9XJWm4LOn-ZTlSUqXOWS-Se6wp-sFhg",
            authDomain: "chatapp-e602f.firebaseio.com",
            databaseURL: "https://chatapp-e602f-default-rtbd.firebaseio.com",
            projectId: "chatapp-e602f",
            storageBucket: "chatapp-e602f.appspot.com",
            messagingSenderId: "150115543267",
            appId: "1:150115543267:web:0322faf04cfbecce57dd6",
        }
    </script>
</head>
<body>
    <script>
        goBackToStartPage();
    </script>
    <div class="chat-bubble-container">
        <ul class="list-group reply">
            <li class="list-item">
                <p class="chat-bubble to-message">Hi! I'm doing good thank you!</p>
            </li>
        </ul>
    </div>
    <div class="card-body">
        <div class="card-footer">
            <div class="chat-input">
                <div class="input-group mb-3">
                    <input type="text" class="form-control" placeholder="Write your message">
                    <div class="input-group-append">
                        <button class="btn btn-primary submit"><i class="fa fa-paper-plane"></i></button>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="row justify-content-center">
        <div class="container">
            <main>
                <script>
                    goBackToStartPage();
                </script>
                <div class="chat">
                    <div class="chat-box">
                        <div class="chat-header">
                            <div class="user-profile">
                                <img alt="User profile picture" />
                            </div>
                            <div class="user-info">
                                <span>John Doe</span>
                                <span>Online</span>
                            </div>
                        </div>
                        <div class="chat-messages">
                            <ul class="list-group reply">
                                <li class="list-item">
                                    <p class="chat-bubble to-me">Hello! How can I help you today?</p>
                                </li>
                            </ul>
                        </div>
                    </div>
                    <div class="chat-form">
                        <div class="input-group mb-3">
                            <input type="text" class="form-control" placeholder="Type your message...">
                            <div class="input-group-append">
                                <button class="btn btn-primary send"><i class="fa fa-paper-plane"></i></button>
                            </div>
                        </div>
                    </div>
                </div>
            </main>
        </div>
    </div>
</body>

```

```
<html>
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/ZJ</script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmf</script>
    <script>
        if(localStorage.getItem('loggedin_user_id')=="") || localStorage.getItem('loggedin_user_id')==undefined || localStorage.getItem('chat_user_email')==""
            location.href = "index.html";
        }
        document.getElementById('email-label').innerHTML = "Chatting with <b>" + localStorage.getItem('chat_user_email') + "</b>";
        const tooltipTriggerList = document.querySelectorAll('[data-bs-toggle="tooltip"]');
        const tooltipList = [...tooltipTriggerList].map(tooltipTriggerEl => new bootstrap.Tooltip(tooltipTriggerEl))

        function triggerLogout(){
            localStorage.removeItem('loggedin_user_id')
            localStorage.removeItem('chat_user_id')
            localStorage.removeItem('chat_user_email')
            location.href = "index.html";
        }

        function goBackToStartPage(){
            localStorage.removeItem('chat_user_id')
            localStorage.removeItem('chat_user_email')
            location.href = "start-chat.html";
        }
    </script>
    <script type="module">
        import { initializeApp } from "https://www.gstatic.com/firebasejs/10.3.1/firebase-app.js";
        import { getAuth } from "https://www.gstatic.com/firebasejs/10.3.1.firebaseio-auth.js";
        import { getDatabase, ref, set, push, onDataChange } from "https://www.gstatic.com/firebasejs/10.3.1.firebaseio-database.js";
        const firebaseConfig = {
            apiKey: "AIzaSyB9XJWm4LOn-ZTlSUqXOWS-Se6wp-sFhg",
            authDomain: "chatapp-e602f.firebaseio.com",
            databaseURL: "https://chatapp-e602f-default-rtbd.firebaseio.com",
            projectId: "chatapp-e602f",
            storageBucket: "chatapp-e602f.appspot.com",
            messagingSenderId: "150115543267",
            appId: "1:150115543267:web:0322faf04cfbecce57dd6",
        }
    </script>
</head>
<body>
    <script>
        goBackToStartPage();
    </script>
    <div class="chat-bubble-container">
        <ul class="list-group reply">
            <li class="list-item">
                <p class="chat-bubble to-message">Hi! I'm doing good thank you!</p>
            </li>
        </ul>
    </div>
    <div class="card-body">
        <div class="card-footer">
            <div class="chat-input">
                <div class="input-group mb-3">
                    <input type="text" class="form-control" placeholder="Write your message...">
                    <div class="input-group-append">
                        <button class="btn btn-primary submit"><i class="fa fa-paper-plane"></i></button>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="row justify-content-center">
        <div class="container">
            <main>
                <script>
                    goBackToStartPage();
                </script>
                <div class="chat">
                    <div class="chat-box">
                        <div class="chat-header">
                            <div class="user-profile">
                                <img alt="User profile picture" />
                            </div>
                            <div class="user-info">
                                <span>John Doe</span>
                                <span>Online</span>
                            </div>
                        </div>
                        <div class="chat-messages">
                            <ul class="list-group reply">
                                <li class="list-item">
                                    <p class="chat-bubble to-me">Hello! How can I help you today?</p>
                                </li>
                            </ul>
                        </div>
                    </div>
                    <div class="chat-form">
                        <div class="input-group mb-3">
                            <input type="text" class="form-control" placeholder="Type your message...">
                            <div class="input-group-append">
                                <button class="btn btn-primary send"><i class="fa fa-paper-plane"></i></button>
                            </div>
                        </div>
                    </div>
                </div>
            </main>
        </div>
    </div>
</body>

```

```
81     storageBucket: "chatapp-e662f.appspot.com",
82     messagingSenderId: "150115543267",
83     appId: "1:150115543267:web:0322faf04cfbecce57dd6",
84     measurementId: "G-Z5K276QYET"
85   };
86   const app = initializeApp(firebaseConfig);
87   const db = getDatabase();
88
89   $(window).on('keydown',function(e){
90     if(e.which == 13){
91       newChat();
92       return false;
93     }
94   })
95
96   $(".submit").click(function(){
97     newChat();
98   })
99
100  function newChat(){
101    var chat = $(".chat").val();
102    if($.trim(chat)==""){
103      return false;
104    }
105    insertChatData(chat)
106  }
107
108  function insertChatData(chat){
109    push(ref(db,"/chats"),{
```

```
107    function insertChatData(chat){
108      push(ref(db,"/chats"),{
109        from : localStorage.getItem('loggedin_user_id'),
110        to : localStorage.getItem('chat_user_id'),
111        message : chat
112      })
113    }
114
115    const chatRef = ref(db,"/chats");
116    onValue(chatRef,(snapshot)=>{
117      var arr = [];
118      snapshot.forEach((childSnapshot)=>{
119        const childData = childSnapshot.val();
120        if((childData.from == localStorage.getItem('loggedin_user_id') || childData.to == localStorage.getItem('loggedin_user_id')) && childData.message != null){
121          arr.push(childData);
122        }
123      })
124      writeChatsOnScreen(arr);
125    })
126  }
127
128  function writeChatsOnScreen(chats){
129    $(".chats ul").html("");
130    for(let chat in chats){
131      var from_user_id = chats[chat].from
132      var to_user_id = chats[chat].to
133      var stored_message = chats[chat].message
134      var message_type;
135      var bubble;
136      if(from_user_id == localStorage.getItem('loggedin_user_id')){
```

The screenshot shows a code editor interface with the title bar "chat application". The left sidebar contains icons for file operations like Open, Save, Find, and Refresh. The top navigation bar includes File, Edit, Selection, View, Go, Run, and a Help menu. The main workspace displays a script file named "chat.html" with the following content:

```
124      })
125      writeChatsOnScreen(arr);
126  }
127
128  function writeChatsOnScreen(chats){
129    $(".chats ul").html("");
130    for(let chat in chats){
131      var from_user_id = chats[chat].from
132      var to_user_id = chats[chat].to
133      var stored_message = chats[chat].message
134      var message_type;
135      var bubble;
136      if(from_user_id == localStorage.getItem('loggedin_user_id')){
137        message_type = "sent ";
138        bubble = "from-message"
139      }else{
140        message_type = "reply ";
141        bubble = "to-message"
142      }
143      $('- <p class="chat-bubble '+bubble+'">' +stored_message+'
').appendTo($(".chats ul"))
144      $(".chat").val(null);
145    }
146  }
147  </script>
148 </body>
149 </html>
```

The status bar at the bottom shows various system and application icons, including a weather icon (31°C, Mostly sunny), a search bar, and a date/time indicator (09-09-2023, 06:48 PM).

CSS:-

The screenshot shows a code editor window titled "chat application". The tabs at the top include index.html, start-chat.html, chat.html, signup.html, # style.css (which is the active tab), and validations.js. The CSS code in the editor is as follows:

```
# style.css > ↗ to-message
1 body{
2     display: flex;
3     align-items: center;
4     justify-content: center;
5     min-height: 100vh;
6     background: #00FAF2;
7     font-size: 1em;
8     letter-spacing: 0.1px;
9     color: black;
10    text-rendering: optimizeLegibility;
11    text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.004);
12 }
13 .chats{
14     max-height: 450px;
15     overflow-y: auto;
16 }
17 .chat-bubble-container{
18     list-style: none;
19 }
20 .chat-bubble{
21     max-width: 450px;
22     padding: 15px;
23     margin: 2px;
24     background: grey;
25     border-radius: 15px;
26     text-align: left;
27 }
28 .from-message{
29     background-color: #03FFDD;
30     border-bottom-left-radius: 0;
31 }
32 .to-message{
33     background-color: #5FFF03;
34     border-bottom-right-radius: 0;
35     right: -100px !important;
36     position: relative;
37 }
38 }
```

The status bar at the bottom displays "Screen Reader Optimized", "Ln 36, Col 24", "Spaces: 4", "UTF-8", "CRLF", "CSS", "Go Live", "Prettier", the date "09-09-2023", and the time "06:50 PM".

The screenshot shows a code editor window titled "chat application". The tabs at the top include index.html, start-chat.html, chat.html, signup.html, # style.css (which is the active tab), and validations.js. The CSS code in the editor has been updated:

```
# style.css > ↗ .chat-bubble
17 .chat-bubble-container{
18     list-style: none;
19 }
20 .chat-bubble{
21     max-width: 450px;
22     padding: 15px;
23     margin: 2px;
24     background: grey;
25     border-radius: 15px;
26     text-align: left;
27 }
28 .from-message{
29     background-color: #03FFDD;
30     border-bottom-left-radius: 0;
31     position: relative;
32 }
33 .to-message{
34     background-color: #5FFF03;
35     border-bottom-right-radius: 0;
36     right: -100px !important;
37     position: relative;
38 }
```

The status bar at the bottom displays "Screen Reader Optimized", "Ln 22, Col 19", "Spaces: 4", "UTF-8", "CRLF", "CSS", "Go Live", "Prettier", the date "09-09-2023", and the time "06:50 PM".

JAVASCRIPT:-

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the "CHAT APPLICATION" folder: index.html, start-chat.html, chat.html, signup.html, style.css, and validation.js.
- Editor:** The "validation.js" file is open, displaying JavaScript code for validating user input. The code includes functions for validating sign-up and login credentials, using regular expressions for email validation, and printing errors to the console.
- Bottom Bar:** Includes icons for weather (31°C, mostly sunny), search, screen reader optimization, code analysis (Prettier), and system status (Screen Reader Optimized, Ln 26, Col 1, Spaces: 4, CRLF, JavaScript, Go Live, ENG IN, 06:51 PM, 09-09-2023).

```
if((nameErr || emailErr || passwordErr || confirmPasswordErr) == true) {
    return false;
} else{
    return true;
}

function validateSignUpInputCredentials(){
    //event.preventDefault();
    email=document.getElementById("email").value;
    password=document.getElementById("password").value;
    var emailErr = passwordErr = true;

    // Validate email address
    if(email == "") {
        printError("emailErr", "Please enter your email address");
    } else {
        // Regular expression for basic email validation
        var regex = /^[\w+@\w+\.\w+$/;
        if(regex.test(email) === false) {
            printError("emailErr", "Please enter a valid email address");
        } else{
            printError("emailErr", "");
            emailErr = false;
        }
    }

    if(password == "") {
```

The screenshot shows a code editor interface with the following details:

- File Explorer (Left):** Shows a project structure under "CHAT APPLICATION" containing files: chat.html, index.html, signup.html, start-chat.html, style.css, and validation.js (selected).
- Search Bar (Top):** Contains the text "chat application".
- Code Editor (Main Area):** Displays the content of validation.js. The code handles user input validation for email, password, and confirmPassword fields. It uses regular expressions for email validation and prints errors to "emailErr", "passwordErr", and "confirmPasswordErr" if inputs are empty or invalid.
- Bottom Status Bar:** Includes icons for screen reader optimization, file save, and various developer tools. The status bar also shows the current date and time: "09-09-2023 06:51 PM".

```
// Validation logic for sign-up input credentials
if(email == "") {
    printError("emailErr", "Please enter your email address");
} else {
    // Regular expression for basic email validation
    var regex = /^[\S+@\S+\.\S+$/;
    if(regex.test(email) === false) {
        printError("emailErr", "Please enter a valid email address");
    } else{
        printError("emailErr", "");
        emailErr = false;
    }
}

if(password == "") {
    printError("passwordErr", "Please enter your password");
} else{
    printError("passwordErr", "");
    passwordErr = false;
}

if(confirmpassword == "") {
    printError("confirmPasswordErr", "Please enter your password");
} else{
    printError("confirmPasswordErr", "");
    confirmPasswordErr = false;
}

if((nameErr || emailErr || passwordErr || confirmPasswordErr ) == true) {
    return false;
}
```

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

- File Explorer:** Shows a folder named "CHAT APPLICATION" containing files: chat.html, index.html, signup.html, start-chat.html, and style.css. The file "validation.js" is currently selected.
- Editor:** Displays the content of validation.js. The code is written in JavaScript and focuses on validating sign-up input credentials. It includes functions for printing errors and validating name, email, and password fields using regular expressions.
- Bottom Bar:** Includes icons for weather (31°C, Mostly sunny), search, and various extensions like Prettier and ESLint.

```
function printError(elemId, hintMsg) {
    document.getElementById(elemId).innerHTML = hintMsg;
}

function validateSignUpInputCredentials() {
    username=document.getElementById("name").value;
    email=document.getElementById("email_address").value;
    password=document.getElementById("password").value;
    confirmPassword=document.getElementById("confirm_password").value;
    var nameErr = emailErr = passwordErr = confirmPasswordErr = true;

    // Validate email address
    if(username == "") {
        printError("nameErr", "Please enter your Name");
    } else {
        // Regular expression for basic email validation
        var regex = /^[A-Za-z ]+$/;
        if(regex.test(username) === false) {
            printError("nameErr", "Please enter a valid Name");
        } else{
            printError("nameErr", "");
            nameErr = false;
        }
    }

    // Validate email address
    if(email == "") {
        printError("emailErr", "Please enter your email address");
    } else {
        var regex = /^[\S+@\S+\.\S+$/;
        if(regex.test(email) === false) {
            printError("emailErr", "Please enter a valid email address");
        } else{
            printError("emailErr", "");
            emailErr = false;
        }
    }

    if(password == "") {
        printError("passwordErr", "Please enter your password");
    } else{
        printError("passwordErr", "");
        passwordErr = false;
    }

    if((emailErr || passwordErr ) == true) {
        return false;
    }else{
        return true;
    }
}
```

This screenshot is nearly identical to the first one, showing the same validation.js code in VS Code. The main difference is the timestamp at the bottom right of the interface, which has changed from "06:51 PM 09-09-2023" to "06:52 PM 09-09-2023".

```
function printError(elemId, hintMsg) {
    document.getElementById(elemId).innerHTML = hintMsg;
}

function validateSignUpInputCredentials() {
    username=document.getElementById("name").value;
    email=document.getElementById("email_address").value;
    password=document.getElementById("password").value;
    confirmPassword=document.getElementById("confirm_password").value;
    var nameErr = emailErr = passwordErr = confirmPasswordErr = true;

    // Validate email address
    if(username == "") {
        printError("nameErr", "Please enter your Name");
    } else {
        // Regular expression for basic email validation
        var regex = /^[A-Za-z ]+$/;
        if(regex.test(username) === false) {
            printError("nameErr", "Please enter a valid Name");
        } else{
            printError("nameErr", "");
            nameErr = false;
        }
    }

    // Validate email address
    if(email == "") {
        printError("emailErr", "Please enter your email address");
    } else {
        var regex = /^[\S+@\S+\.\S+$/;
        if(regex.test(email) === false) {
            printError("emailErr", "Please enter a valid email address");
        } else{
            printError("emailErr", "");
            emailErr = false;
        }
    }

    if(password == "") {
        printError("passwordErr", "Please enter your password");
    } else{
        printError("passwordErr", "");
        passwordErr = false;
    }

    if((emailErr || passwordErr ) == true) {
        return false;
    }else{
        return true;
    }
}
```

CONCLUSION

OUTPUT:-

The screenshot shows a web browser window with the URL `127.0.0.1:5500/signup.html`. A modal dialog box is displayed, stating "127.0.0.1:5500 says successfully Registered" with an "OK" button. Below the modal is a "Sign Up" form with fields for User name (Varun Kesarwani), Email Address (shivarun2003@gmail.com), Password, and Confirm Password. A "Signup" button is present, and a link "I already have an Account [Login instead](#)" is at the bottom.

The screenshot shows a web browser window with the URL `127.0.0.1:5500/index.html`. A "Login" form is displayed with fields for Email Address (shivarun2003@gmail.com) and Password. There is a "Remember Me" checkbox checked, a "Login" button, and a link "I do not have an Account [Sign me up](#)".

