

**AMAN BHAT**

**RESOLVE**

**TriNIT Hackathon 3.0**

---

***Problem Statement -***

Building a secure healthcare access platform with a focus on frontend development.

***Description -***

1. Building a secure healthcare access platform is deployed to safeguard sensitive patient information and ensuring compliance with regulations.
2. Given the highly confidential nature of medical records and personal data involved, any breach of security could result in severe consequences for patients, healthcare providers, and the organization itself.
3. A secure platform not only protects against unauthorized access, data breaches, and cyber threats but also fosters trust among users, promoting better patient care and confidentiality.
4. Using React to build a secure platform offers several advantages in terms of development efficiency, user experience, and security.
5. React's component-based architecture promotes modular design, making it easier to implement security features consistently across the application.
6. React's virtual DOM efficiently updates the UI, enhancing responsiveness and overall user experience, which can indirectly contribute to security by reducing the likelihood of user errors that could lead to security vulnerabilities.
7. Moreover, React's ecosystem provides a community-contributed libraries and tools for implementing security measures, such as authentication and authorization mechanisms, input validation, and encryption utilities.
8. By leveraging these resources and adhering to React's best practices, developers can streamline the implementation of security features while building a robust and user-friendly healthcare access platform.

## ***Solution –***

### ***1. Building an interface using ReactJS –***

The React code represents a basic healthcare access platform application with authentication functionality. It provides a login interface where users can enter a username and password. Upon successful authentication it displays a welcome message along with a placeholder component for patient information. If the credentials are incorrect, it shows an error message prompting the user to try again.

This code defines a React component named HealthcareAccessPlatform, which represents a healthcare access platform named CareLink . It manages the state for user login status, role, username, password, and security code using React's useState hook. The handleLogin function validates user credentials and sets the user role accordingly upon successful login. The component renders a login form if the user is not logged in, allowing users to input their credentials. Upon successful login, it displays a dashboard showing the user's role and a logout button.

### ***2. Developing a chatbot –***

The role of MedBot is to serve as a virtual assistant for medical queries, allowing users to ask questions or seek assistance related to health concerns, appointments, medication, and more. Through natural language processing and predefined conversation flows, MedBot aims to provide helpful responses and guide users through their inquiries in an interactive and user-friendly manner.

In the code, we've created a virtual chatbot named MedBot using React and the react-simple-chatbot package. The MedBot component defines a series of steps for the chatbot conversation, where it greets the user, asks them for input, and provides an answer. When integrated into the App component, users interact with MedBot through a chat interface displayed on the web page. Overall, this code sets up a simple chatbot interface named MedBot with a predefined conversation flow.

**Codes –**

## 1. CareLink

*App.js*

```

import React, { useState } from 'react';
import './App.css';

const HealthcareAccessPlatform = () => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const [userRole, setUserRole] = useState("");
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const [securityCode, setSecurityCode] = useState("");

  const handleLogin = () => {
    if (username === 'demo' && password === 'password' && securityCode
    === '1234') {
      if (username === 'doctor') {
        setIsLoggedIn(true);
        setUserRole('doctor');
      } else if (username === 'hospital') {
        setIsLoggedIn(true);
        setUserRole('hospital');
      } else {
        setIsLoggedIn(true);
        setUserRole('patient');
      }
    } else {
      alert('Invalid username, password, or security code');
    }
  };

  const handleLogout = () => {
    setIsLoggedIn(false);
    setUserRole("");
    setUsername("");
    setPassword("");
    setSecurityCode("");
  };

  return (

```

```

<div className="healthcare-access-platform">
  {isLoggedIn ? (
    <div className="login-form">
      <h2>Login Page</h2>
      <h3>Welcome to the CareLink, your virtual healthcare
platform!</h3>
      <form onSubmit={handleLogin}>
        <input
          type="text"
          placeholder="Username"
          value={username}
          onChange={(e) => setUsername(e.target.value)}
          required
        />
        <input
          type="password"
          placeholder="Password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          required
        />
        <input
          type="text"
          placeholder="Security Code"
          value={securityCode}
          onChange={(e) => setSecurityCode(e.target.value)}
          required
        />
        <button type="submit">Submit</button>
      </form>
    </div>
  ) : (
    <div className="dashboard">
      <h2>Welcome to the Healthcare Access Platform</h2>
      <p>User Role: {userRole}</p>
      <button onClick={handleLogout}>Logout</button>
    </div>
  )}
</div>
);
};

```

```
export default HealthcareAccessPlatform;
```

*App.css*

```
/* App.css */
```

```
.healthcare-access-platform {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  background-image:  
url('https://etimg.etb2bimg.com/photo/100166038.cms');  
  background-size: cover;  
  background-position: center;  
}
```

```
.login-form {  
  background-color: rgba(255, 255, 255, 0.8); /* Add transparency to the  
background color */  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
  width: 400px;  
}
```

```
.login-form h2 {  
  margin-bottom: 20px;  
  text-align: center;  
}
```

```
.login-form form {  
  display: flex;  
  flex-direction: column;  
}
```

```
.login-form input {  
  padding: 10px;  
  margin-bottom: 10px;  
  border: 1px solid #ccc;  
  border-radius: 4px;
```

```
}
```

```
.login-form button {  
  padding: 10px 20px;  
  background-color: #007bff;  
  color: #fff;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

```
.login-form button:hover {  
  background-color: #0056b3;  
}
```

```
.dashboard {  
  background-color: rgba(255, 255, 255, 0.8); /* Add transparency to the  
background color */  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);  
}
```

```
.dashboard h2 {  
  margin-bottom: 20px;  
  text-align: center;  
}
```

```
.dashboard p {  
  margin-bottom: 20px;  
}
```

```
.dashboard button {  
  padding: 10px 20px;  
  background-color: #dc3545;  
  color: #fff;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

```
.dashboard button:hover {
```

```
background-color: #bd2130;
}
```

## 2. MedBot

*App.js*

```
import React, { useState, useEffect } from 'react';
import './App.css';

const HealthcareAssistant = () => {
  const [userInput, setUserInput] = useState('');
  const [assistantResponses, setAssistantResponses] = useState([]);
  const [currentIndex, setCurrentIndex] = useState(0);

  const handleInputChange = (e) => {
    setUserInput(e.target.value);
  };

  const handleAskAssistant = () => {
    if (userInput.toLowerCase().includes('fever')) {
      setResponse('How long have you been experiencing the fever?');
    } else if (userInput.toLowerCase().includes('past 4 days')) {
      setResponse('Have you noticed any changes in water and food intake??');
    } else if (userInput.toLowerCase().includes('yes')) {
      setResponse('Could you please explain the changes in detail?');
    } else if (userInput.toLowerCase().includes('i took crocin')) {
      setResponse('I understand that you took crocin for the fever, but regarding the changes, please specify what changes you have observed? For example, do you feel dehydrated and drained?');
    } else if (userInput.toLowerCase().includes('thanks for helping me')) {
      setResponse('You are welcome. It is important for us to understand the specific changes to provide the best care.');
```

```

    setResponse('I am here to help. Please let me know by typing in the
box.');
```

```

  }

  setUserInput("");
};

const setResponse = (response) => {
  setAssistantResponses((prevResponses) => [
    ...prevResponses,
    { role: 'user', content: userInput },
    { role: 'assistant', content: response }
  ]);
};

useEffect(() => {
  const timer = setTimeout(() => {
    setCurrentIndex((prevIndex) => prevIndex + 1);
  }, 10);

  return () => clearTimeout(timer);
}, [assistantResponses]);

return (
  <div className="healthcare-assistant">
    <div className="assistant-container">
      <h1>MedBot - Virtual Healthcare Assistant</h1>
      <div className="user-input">
        <label htmlFor="userInput">Hello! I'm MedBot your virtual assistant
and I am here to talk with you about your health and any concerns you
might have. Let's start with some questions to better understand your
situation:</label>
        <input
          type="text"
          value={userInput}
          onChange={handleInputChange}
          placeholder="Enter your message"
        />
        <button onClick={handleAskAssistant}>Ask</button>
      </div>
      <img
```



```

        src="https://nordvpn.com/wp-content/uploads/blog-featured-what-
is-chatbot.svg"
        alt="Chatbot"
        className="chatbot-image"
      />
      <div className="conversation">
        {assistantResponses.slice(0, currentIndex + 1).map((message, index)
=> (
          <div key={index} className={message.role}>
            <strong>{message.role.charAt(0).toUpperCase() +
message.role.slice(1)}:</strong> {message.content}
          </div>
        )}}
      </div>
    </div>
  </div>
);
};

```

```
export default HealthcareAssistant;
```

*App.css*

```
/* src/App.css */
```

```

body {
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f7f7f7;
}

```

```

.healthcare-assistant {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

```

```

.assistant-container {
  max-width: 600px;
  width: 100%;
}

```

```
padding: 20px;
background-color: #fff;
border-radius: 10px;
box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
text-align: center;
}
```

```
h1 {
  color: #3498db;
}
```

```
.user-input {
  margin-top: 20px;
}
```

```
label {
  display: block;
  margin-bottom: 10px;
  color: #555;
}
```

```
input {
  width: 100%;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
```

```
button {
  padding: 10px 20px;
  cursor: pointer;
  background-color: #3498db;
  color: #fff;
  border: none;
  border-radius: 5px;
  transition: background-color 0.3s ease;
}
```

```
button:hover {
  background-color: #2980b9;
}
```

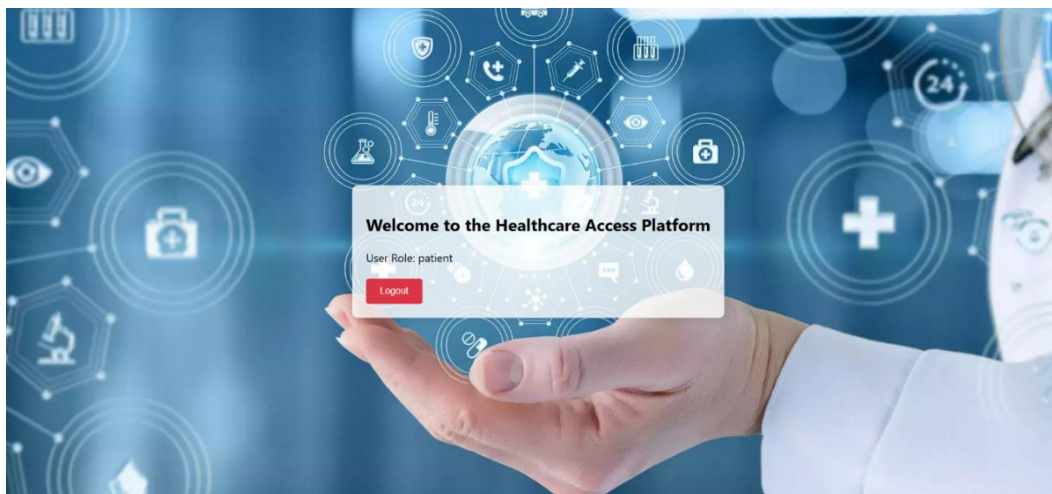
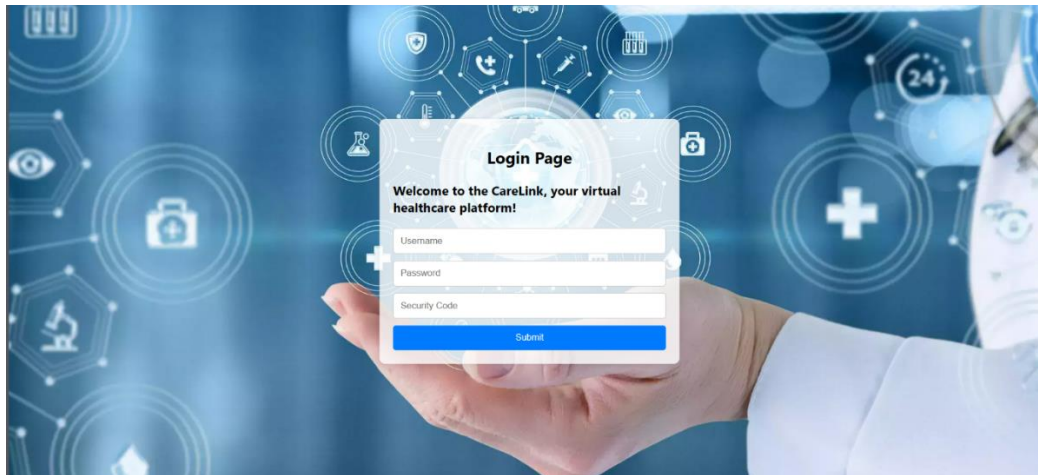
```
.assistant-response {  
  margin-top: 20px;  
  padding: 10px;  
  background-color: #ecf0f1;  
  border-radius: 5px;  
  color: #333;  
}
```

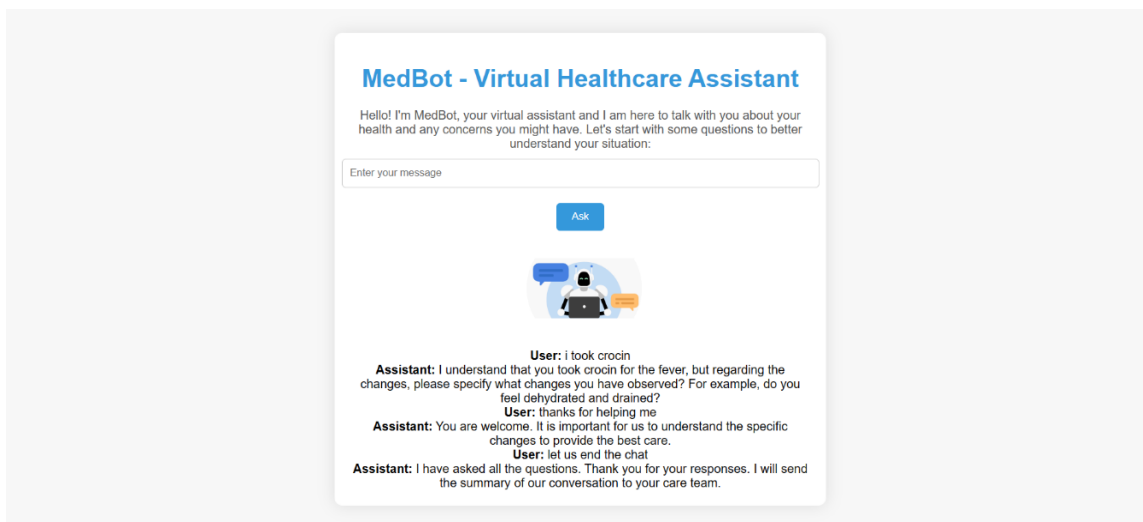
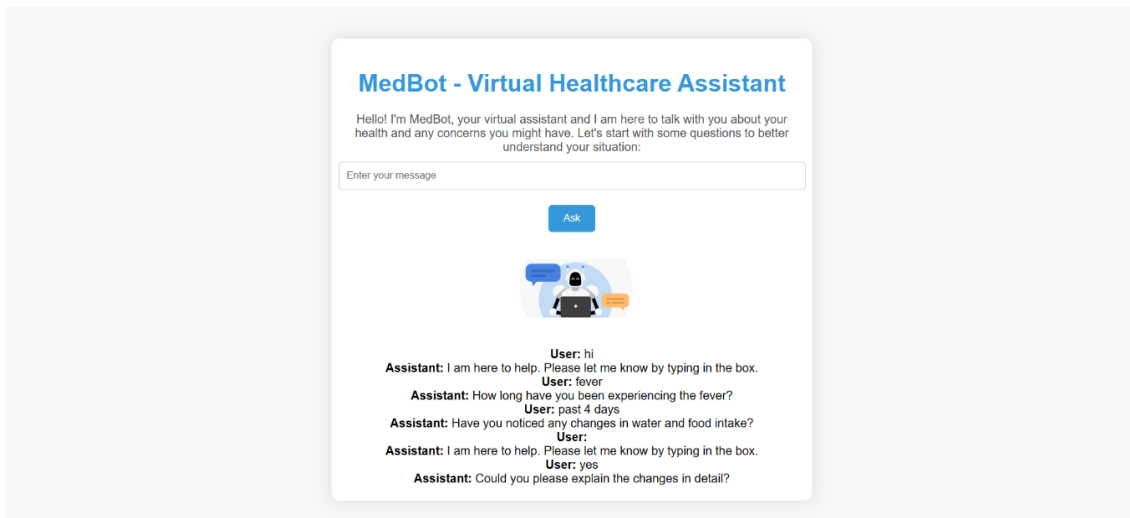
```
.user-input {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```

```
button {  
  margin-top: 10px;  
}
```

```
.chatbot-image {  
  width: 150px;  
  height: 150px;  
  margin-left: 10px;  
  border-radius: 150%;  
}
```

## *Output Screenshots –*





### ***Future Scope –***

In the future, building a secure healthcare access platform could involve building a mobile application for better UX for the patients and doctors and incorporate advanced technologies such as blockchain for immutable patient records, AI detection for early threat identification, and biometric authentication for enhanced security. Integration with wearable devices and IoT sensors could enable real-time health monitoring, while robust encryption and decentralized storage mechanisms would ensure the privacy and integrity of sensitive medical data, advancing patient-centric care and interoperability across healthcare systems.