

**Name:- Vidya CS**

**TRACK:JAVA FSE**

Create a React Application named “ticketbookingapp” where the guest user can browse the page where the flight details are displayed whereas the logged in user only can book tickets.

The Login and Logout buttons should accordingly display different pages. Once the user is logged in the User page should be displayed. When the user clicks on Logout, the Guest page should be displayed.

**Step1 :- Create a react project(ticketbookingapp):-**

```
PS C:\Users\DELL\OneDrive\Documents\cogni handson> npx create-react-app ticketbookingapp
>>

Creating a new React app in C:\Users\DELL\OneDrive\Documents\cogni handson\ticketbookingapp.

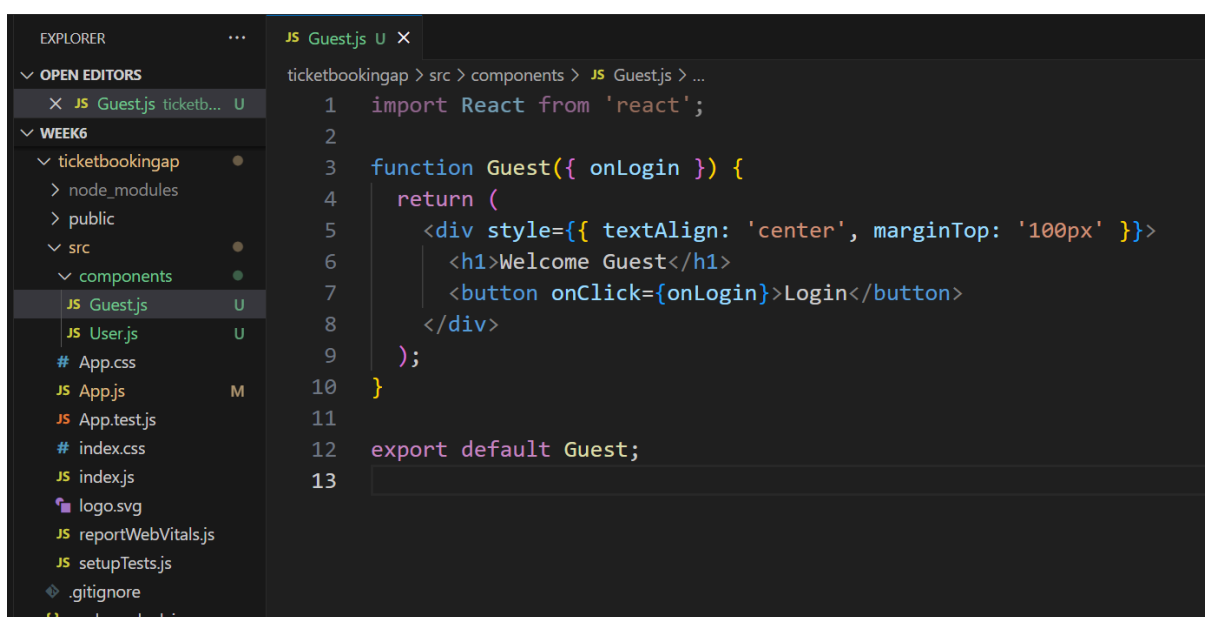
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1323 packages in 4m

269 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.
```

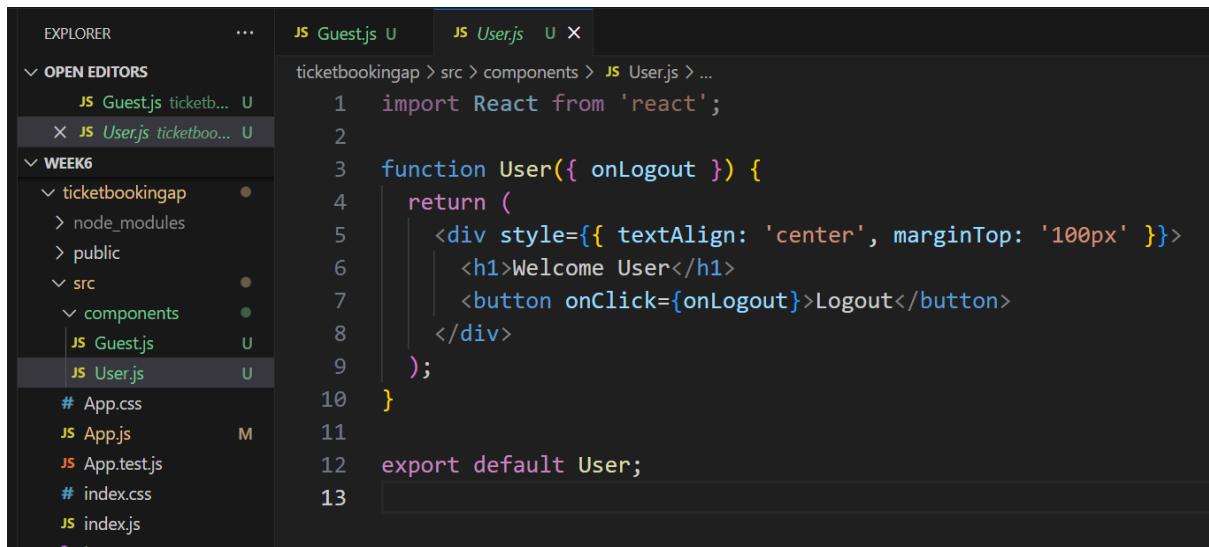
**Step 2:- code components/Guest.js**



The screenshot shows the VS Code editor with the Explorer sidebar on the left and the code editor on the right. The Explorer sidebar shows the file structure of the 'ticketbookingapp' project, with the 'components' folder expanded and 'Guest.js' selected. The code editor displays the following code for 'Guest.js':

```
1  import React from 'react';
2
3  function Guest({ onLogin }) {
4    return (
5      <div style={{ textAlign: 'center', marginTop: '100px' }}>
6        <h1>Welcome Guest</h1>
7        <button onClick={onLogin}>Login</button>
8      </div>
9    );
10 }
11
12 export default Guest;
```

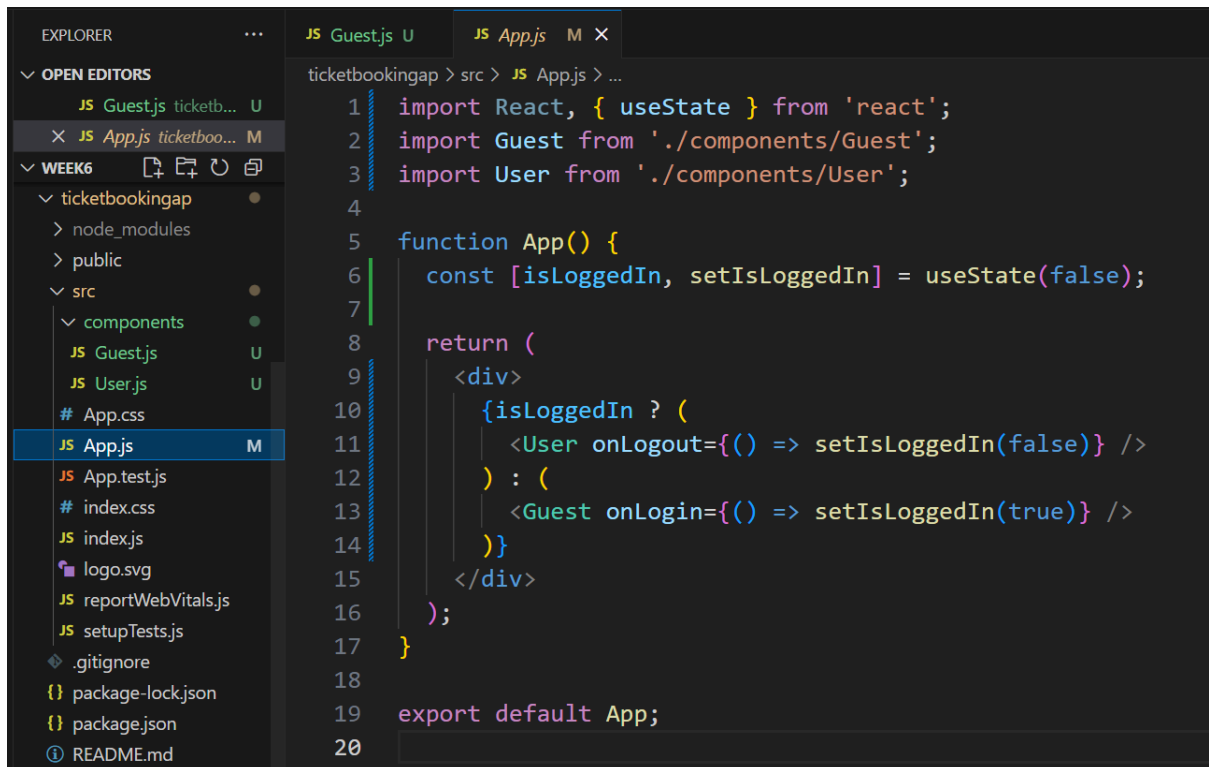
### Step 3:- code components/User.js



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the project structure with 'ticketbookingapp' expanded, showing 'src' > 'components'. The Editor shows the code for 'User.js'.

```
1 import React from 'react';
2
3 function User({ onLogout }) {
4   return (
5     <div style={{ textAlign: 'center', marginTop: '100px' }}>
6       <h1>Welcome User</h1>
7       <button onClick={onLogout}>Logout</button>
8     </div>
9   );
10 }
11
12 export default User;
```

### Step 4:- code App.js



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the project structure with 'ticketbookingapp' expanded, showing 'src' > 'components'. The Editor shows the code for 'App.js'.

```
1 import React, { useState } from 'react';
2 import Guest from './components/Guest';
3 import User from './components/User';
4
5 function App() {
6   const [isLoggedIn, setIsLoggedIn] = useState(false);
7
8   return (
9     <div>
10       {isLoggedIn ? (
11         <User onLogout={() => setIsLoggedIn(false)} />
12       ) : (
13         <Guest onLogin={() => setIsLoggedIn(true)} />
14       )}
15     </div>
16   );
17 }
18
19 export default App;
```

## OUTPUT:-

```
> ticketbookingap@0.1.0 start
> react-scripts start

(node:11244) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:11244) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view ticketbookingap in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.1.43:3000
```

