Full Stack Development with MERN

Frontend Development Report

Date	8 July 2024
Team ID	SWTID1720170691
Project Name	Flight Booking APP
Maximum Marks	10

Project Title: Book Nest

Date: 8 July 2024

Prepared by: Surya Teja A

Team Members:

Chandra kanth J, Koushik S, Anurag Reddy A

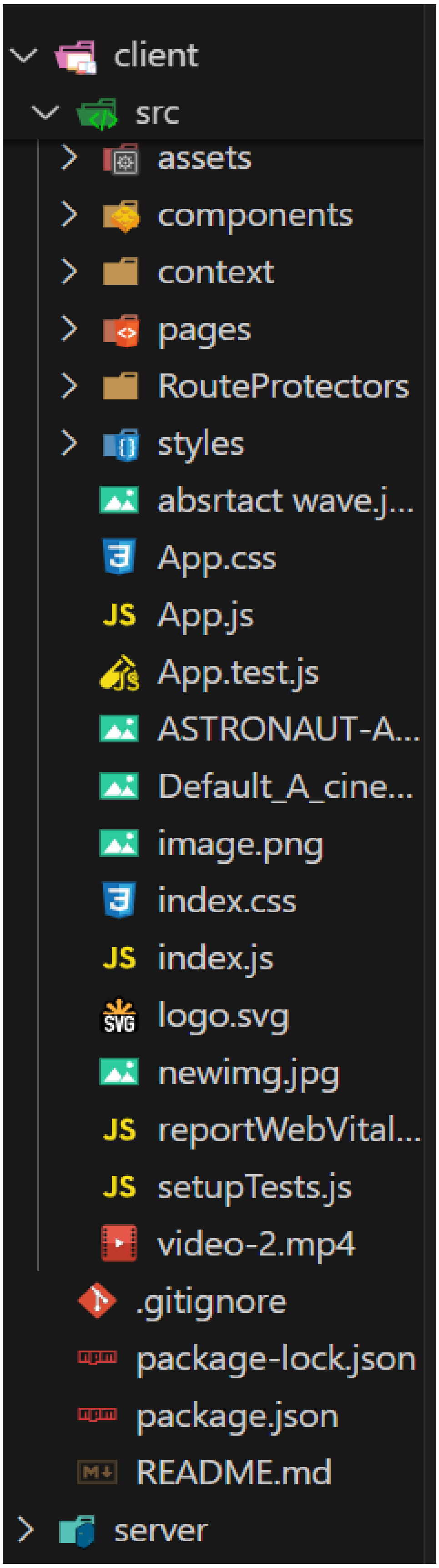
Objective

The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the **Flight Booking APP** project.

Technologies Used

- **Frontend Framework:** React.js
- **State Management:** [Redux/Context API, if applicable]
- **UI Framework/Libraries:** [Bootstrap]
- **API Libraries:** [e.g., Axios,Fetch]

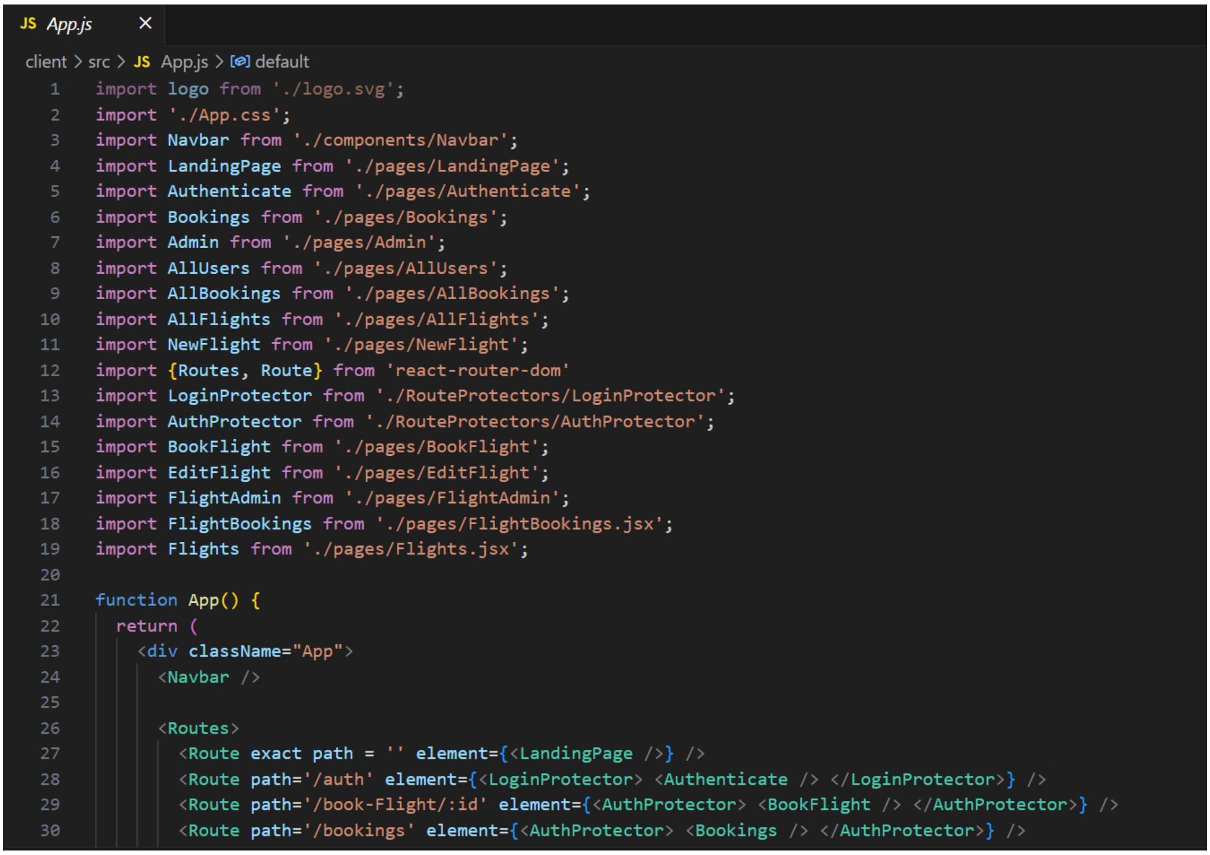
Project Structure



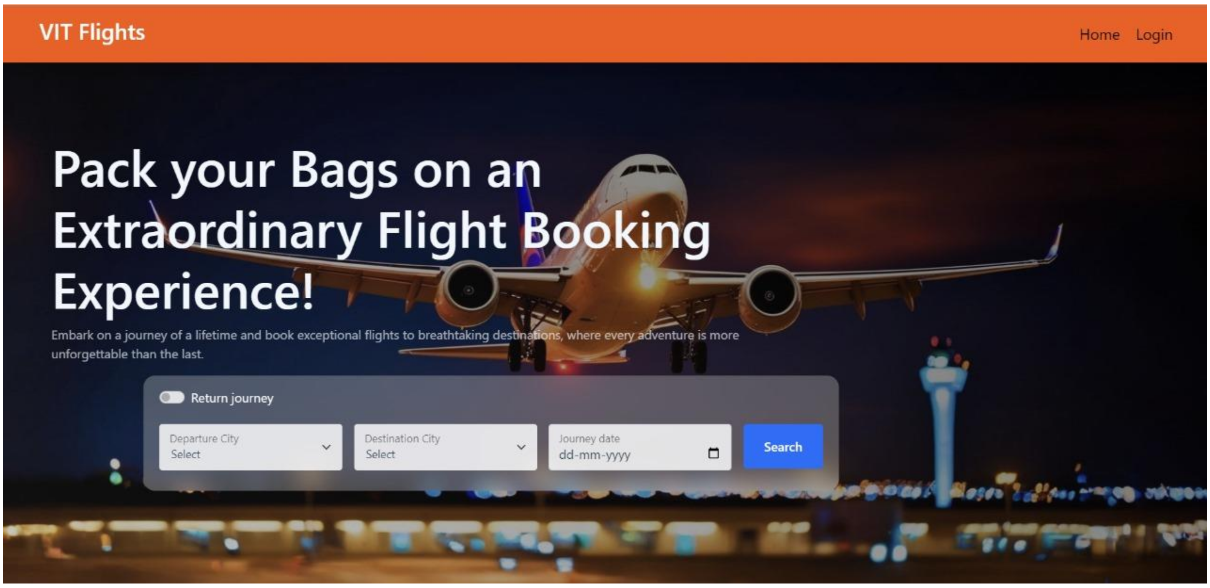
```
client
  src
    assets
    components
    context
    pages
    RouteProtectors
    styles
    absrtact wave.j...
    App.css
    App.js
    App.test.js
    ASTRONAUT-A...
    Default_A_cine...
    image.png
    index.css
    index.js
    logo.svg
    newimg.jpg
    reportWebVital...
    setupTests.js
    video-2.mp4
  .gitignore
  package-lock.json
  package.json
  README.md
server
```

Key Components

1. **App.js**: Responsible for routing and main application layout.



```
import logo from './logo.svg';
import './App.css';
import Navbar from './components/Navbar';
import LandingPage from './pages/LandingPage';
import Authenticate from './pages/Authenticate';
import Bookings from './pages/Bookings';
import Admin from './pages/Admin';
import AllUsers from './pages/AllUsers';
import AllBookings from './pages/AllBookings';
import AllFlights from './pages/AllFlights';
import NewFlight from './pages/NewFlight';
import {Routes, Route} from 'react-router-dom'
import LoginProtector from './RouteProtectors/LoginProtector';
import AuthProtector from './RouteProtectors/AuthProtector';
import BookFlight from './pages/BookFlight';
import EditFlight from './pages/EditFlight';
import FlightAdmin from './pages/FlightAdmin';
import FlightBookings from './pages/FlightBookings.jsx';
import Flights from './pages/Flights.jsx';

function App() {
  return (
    <div className="App">
      <Navbar />

      <Routes>
        <Route exact path = '' element={<LandingPage />} />
        <Route path='/auth' element={<LoginProtector> <Authenticate /> </LoginProtector>} />
        <Route path='/book-Flight/:id' element={<AuthProtector> <BookFlight /> </AuthProtector>} />
        <Route path='/bookings' element={<AuthProtector> <Bookings /> </AuthProtector>} />
```



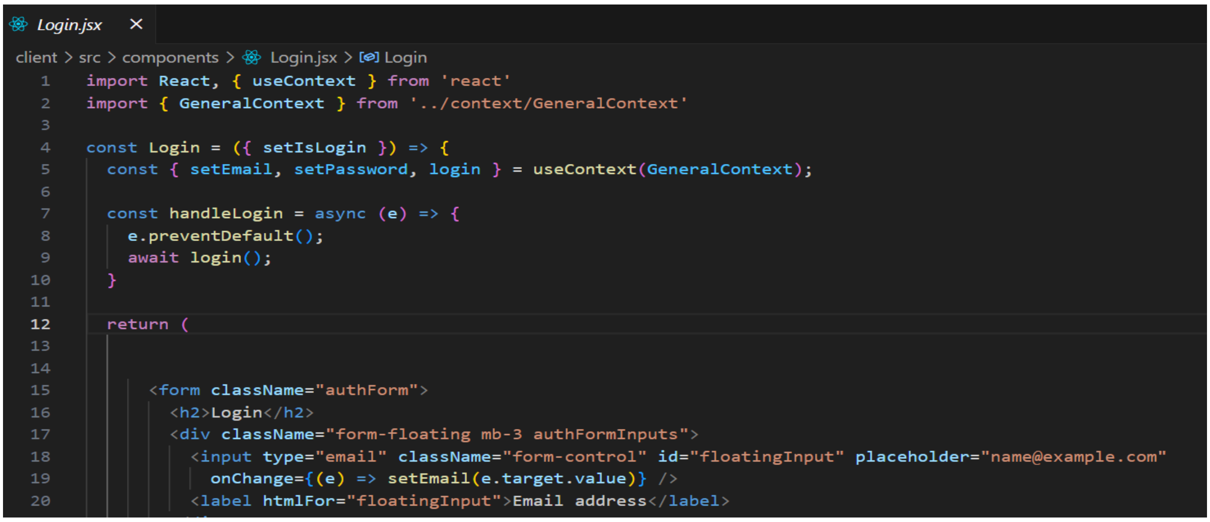
Pack your Bags on an Extraordinary Flight Booking Experience!

Embark on a journey of a lifetime and book exceptional flights to breathtaking destinations, where every adventure is more unforgettable than the last.

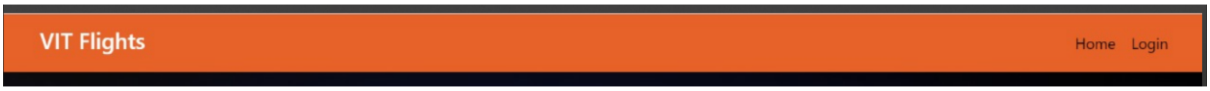
2. **/components**

- **Contains reusable UI components used across the application.**

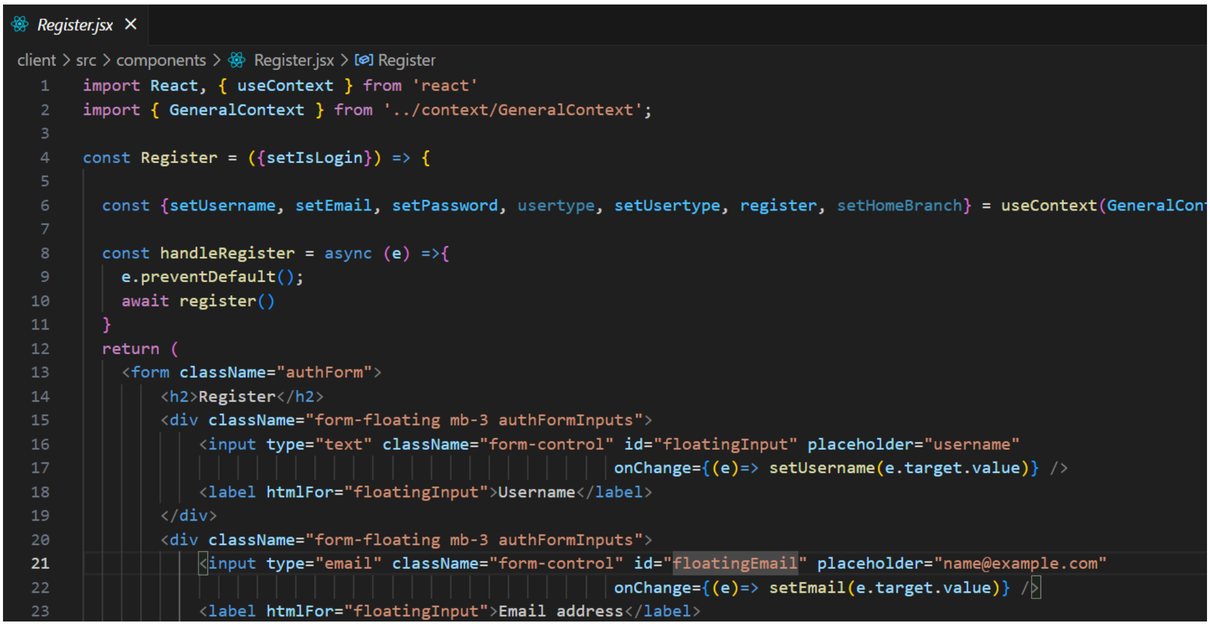
Navbar.jsx



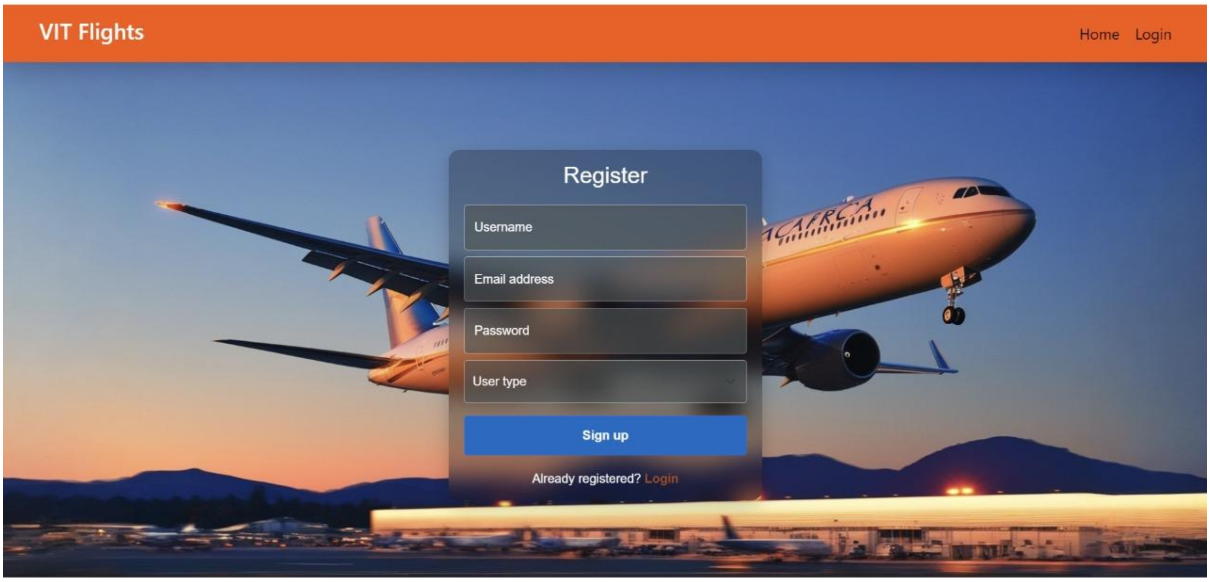
```
import React, { useContext } from 'react'
import { GeneralContext } from '../context/GeneralContext'

const Login = ({ setIsLogin }) => {
  const { setEmail, setPassword, login } = useContext(GeneralContext);

  const handleLogin = async (e) => {
    e.preventDefault();
    await login();
  }

  return (


      <form className="authForm">
        <h2>Login</h2>
        <div className="form-floating mb-3 authFormInputs">
          <input type="email" className="form-control" id="floatingInput" placeholder="name@example.com"
            onChange={(e) => setEmail(e.target.value)} />
          <label htmlFor="floatingInput">Email address</label>
```



Register.jsx

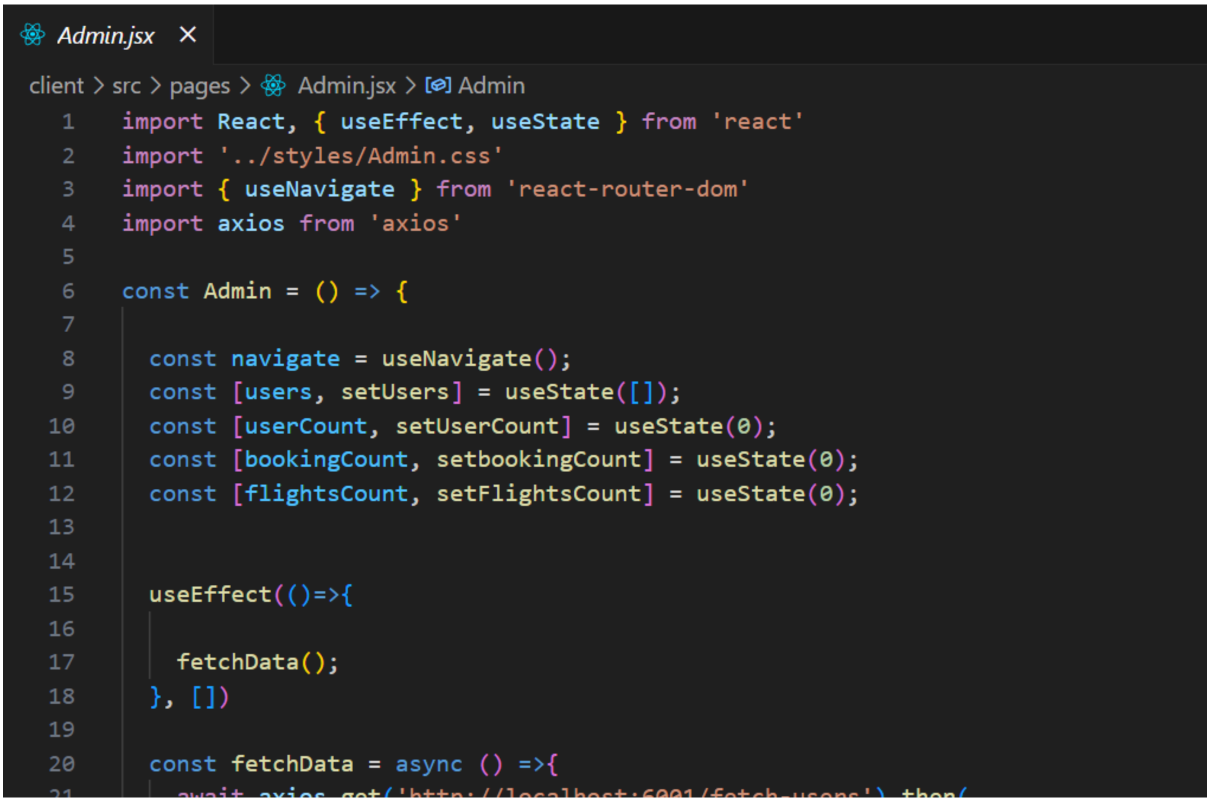


```
import React, { useContext } from 'react'
import { GeneralContext } from '../context/GeneralContext';

const Register = ({setIsLogin}) => {

  const {setUsername, setEmail, setPassword, usertype, setUsertype, register, setHomeBranch} = useContext(GeneralCon

  const handleRegister = async (e) =>{
    e.preventDefault();
    await register()
  }
  return (
    <form className="authForm">
        <h2>Register</h2>
        <div className="form-floating mb-3 authFormInputs">
            <input type="text" className="form-control" id="floatingInput" placeholder="username"
                                                  onChange={(e)=> setUsername(e.target.value)} />
            <label htmlFor="floatingInput">Username</label>
        </div>
        <div className="form-floating mb-3 authFormInputs">
            <input type="email" className="form-control" id="floatingEmail" placeholder="name@example.com"
                                                  onChange={(e)=> setEmail(e.target.value)} />
            <label htmlFor="floatingInput">Email address</label>
```

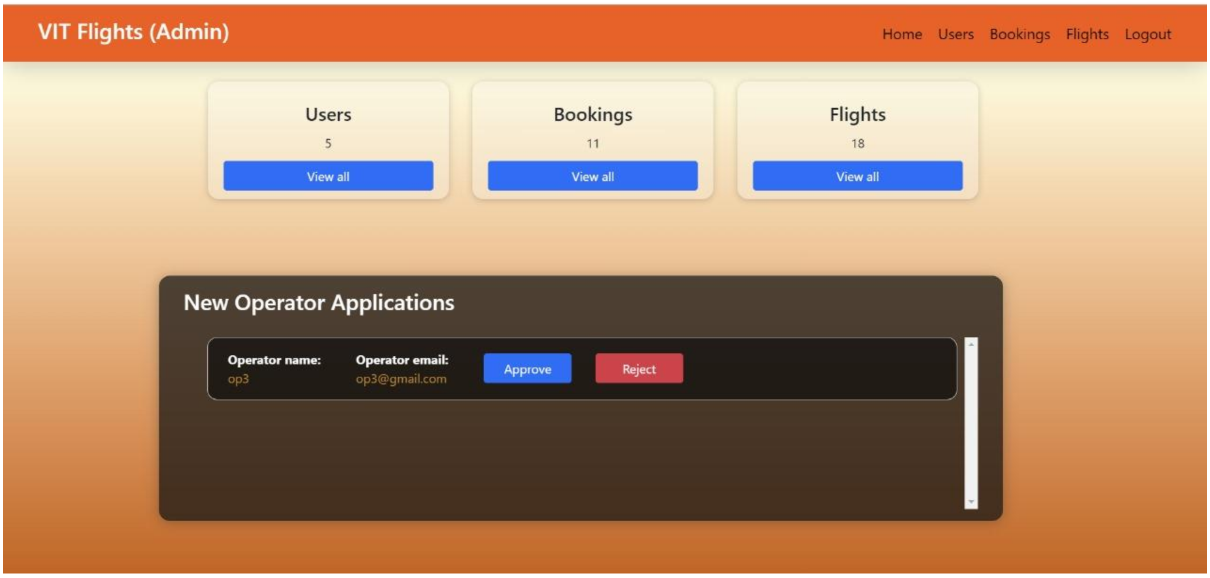


3. **/pages**
   - Includes different pages for Web App.

Admin.jsx



```
import React, { useEffect, useState } from 'react'
import '../styles/Admin.css'
import { useNavigate } from 'react-router-dom'
import axios from 'axios'

const Admin = () => {

  const navigate = useNavigate();
  const [users, setUsers] = useState([]);
  const [userCount, setUserCount] = useState(0);
  const [bookingCount, setbookingCount] = useState(0);
  const [flightsCount, setFlightsCount] = useState(0);


  useEffect(()=>{

    fetchData();
  }, [])

  const fetchData = async () =>{
    await axios.get('http://localhost:6001/fetch-users').then(
```



Users
5

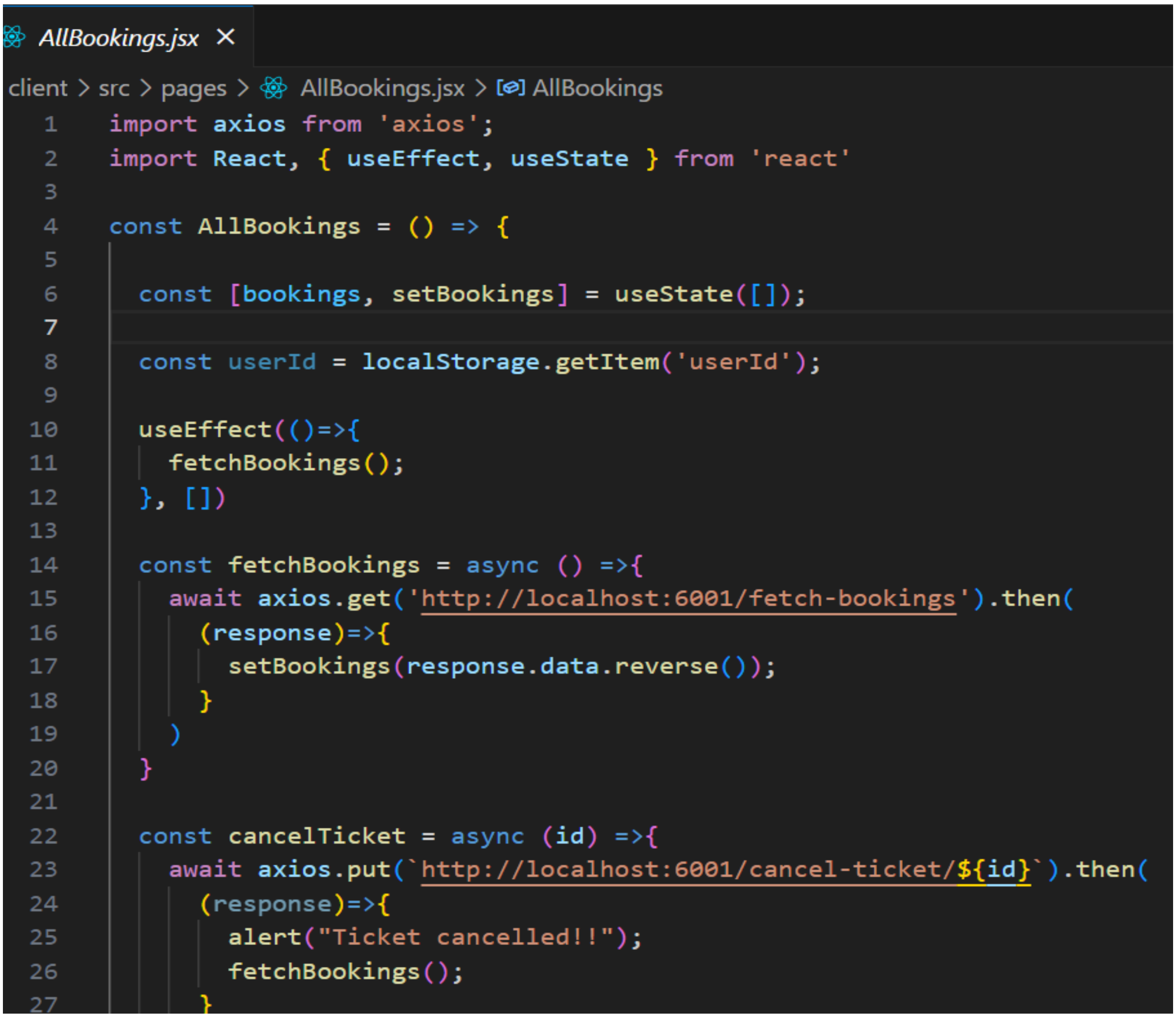
Bookings
11

Flights
18

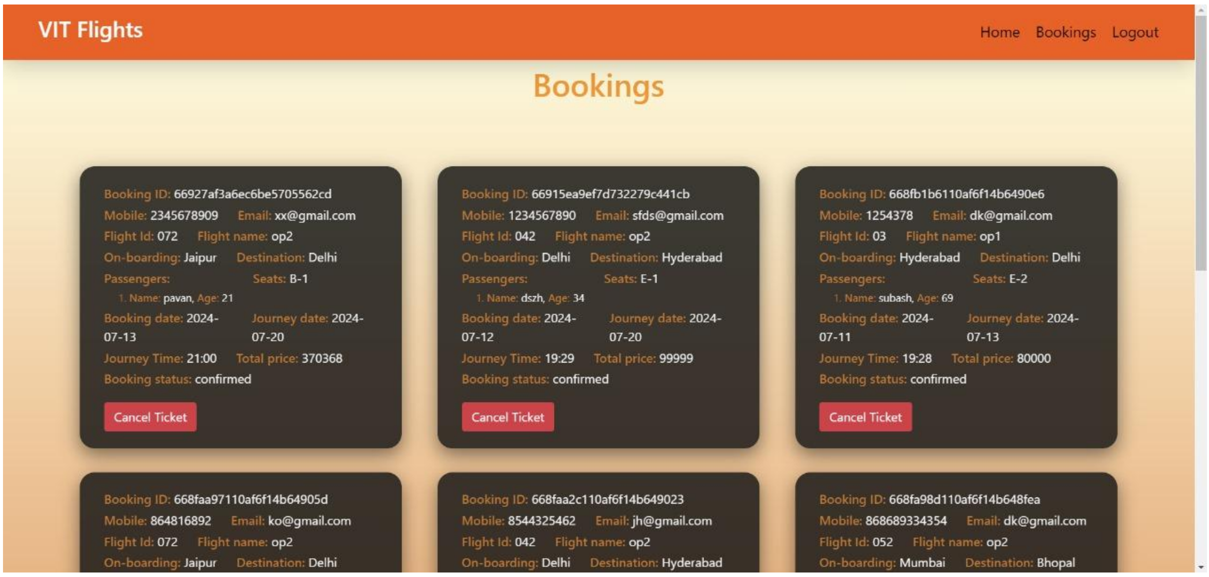
New Operator Applications

Operator name: op3
Operator email: op3@gmail.com

AllBookings.jsx



```
import axios from 'axios';
import React, { useEffect, useState } from 'react'

const AllBookings = () => {

  const [bookings, setBookings] = useState([]);

  const userId = localStorage.getItem('userId');

  useEffect(()=>{
    fetchBookings();
  }, [])

  const fetchBookings = async () =>{
    await axios.get('http://localhost:6001/fetch-bookings').then(
      (response)=>{
        setBookings(response.data.reverse());
      }
    )
  }

  const cancelTicket = async (id) =>{
    await axios.put(`http://localhost:6001/cancel-ticket/${id}`).then(
      (response)=>{
        alert("Ticket cancelled!!");
        fetchBookings();
      }
```



Bookings

Booking ID: 66927af3a6ec6be5705562cd
Mobile: 2345678909 Email: xx@gmail.com
Flight Id: 072 Flight name: op2
On-boarding: Jaipur Destination: Delhi
Passengers: Seats: B-1
1. Name: pavan, Age: 21
Booking date: 2024-07-13 Journey date: 2024-07-20
Journey Time: 21:00 Total price: 370368
Booking status: confirmed
Cancel Ticket

Booking ID: 66915ea9ef7d732279c441cb
Mobile: 1234567890 Email: sfds@gmail.com
Flight Id: 042 Flight name: op2
On-boarding: Delhi Destination: Hyderabad
Passengers: Seats: E-1
1. Name: dszh, Age: 34
Booking date: 2024-07-12 Journey date: 2024-07-20
Journey Time: 19:29 Total price: 99999
Booking status: confirmed
Cancel Ticket

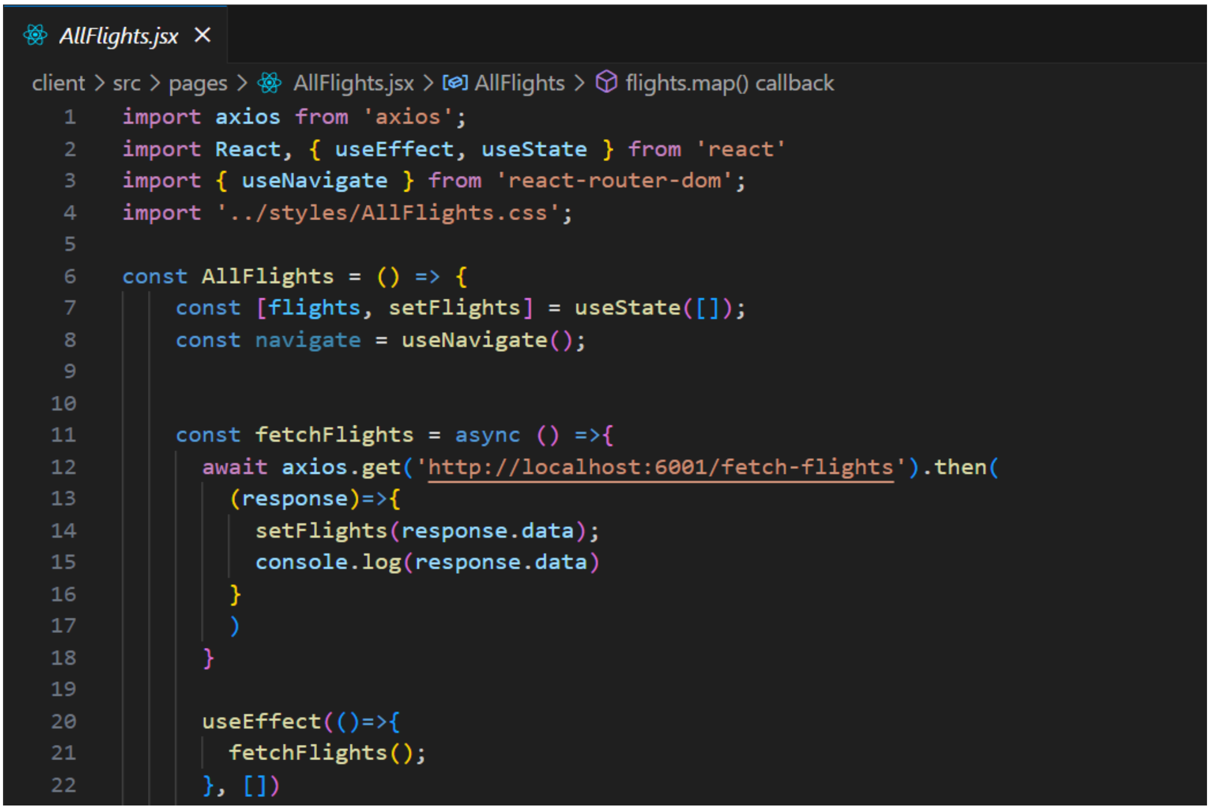
Booking ID: 668fb1b6110af6f14b6490e6
Mobile: 1254378 Email: dk@gmail.com
Flight Id: 03 Flight name: op1
On-boarding: Hyderabad Destination: Delhi
Passengers: Seats: E-2
1. Name: subash, Age: 69
Booking date: 2024-07-11 Journey date: 2024-07-13
Journey Time: 19:28 Total price: 80000
Booking status: confirmed
Cancel Ticket

Booking ID: 668faa97110af6f14b64905d
Mobile: 864816892 Email: ko@gmail.com
Flight Id: 072 Flight name: op2
On-boarding: Jaipur Destination: Delhi

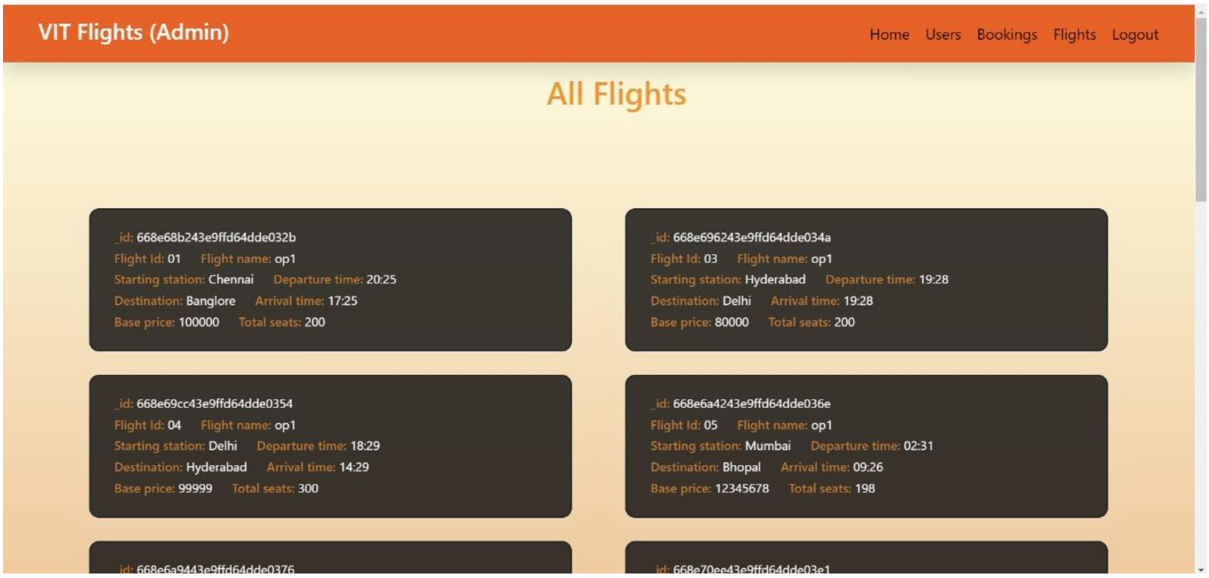
Booking ID: 668faa2c110af6f14b649023
Mobile: 8544325462 Email: jh@gmail.com
Flight Id: 042 Flight name: op2
On-boarding: Delhi Destination: Hyderabad

Booking ID: 668fa98d110af6f14b648fea
Mobile: 868689334354 Email: dk@gmail.com
Flight Id: 052 Flight name: op2
On-boarding: Mumbai Destination: Bhopal

AllFlights.jsx


AllFlights.jsx
client > src > pages > AllFlights.jsx > AllFlights > flights.map() callback

```jsx
import axios from 'axios';
import React, { useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom';
import '../styles/AllFlights.css';

const AllFlights = () => {
    const [flights, setFlights] = useState([]);
    const navigate = useNavigate();


    const fetchFlights = async () =>{
      await axios.get('http://localhost:6001/fetch-flights').then(
        (response)=>{
          setFlights(response.data);
          console.log(response.data)
        }
        )
      }

      useEffect(()=>{
        fetchFlights();
      }, [])
```



_id: 668e68b243e9ffd64dde032b
Flight Id: 01 Flight name: op1
Starting station: Chennai Departure time: 20:25
Destination: Banglore Arrival time: 17:25
Base price: 100000 Total seats: 200

_id: 668e696243e9ffd64dde034a
Flight Id: 03 Flight name: op1
Starting station: Hyderabad Departure time: 19:28
Destination: Delhi Arrival time: 19:28
Base price: 80000 Total seats: 200

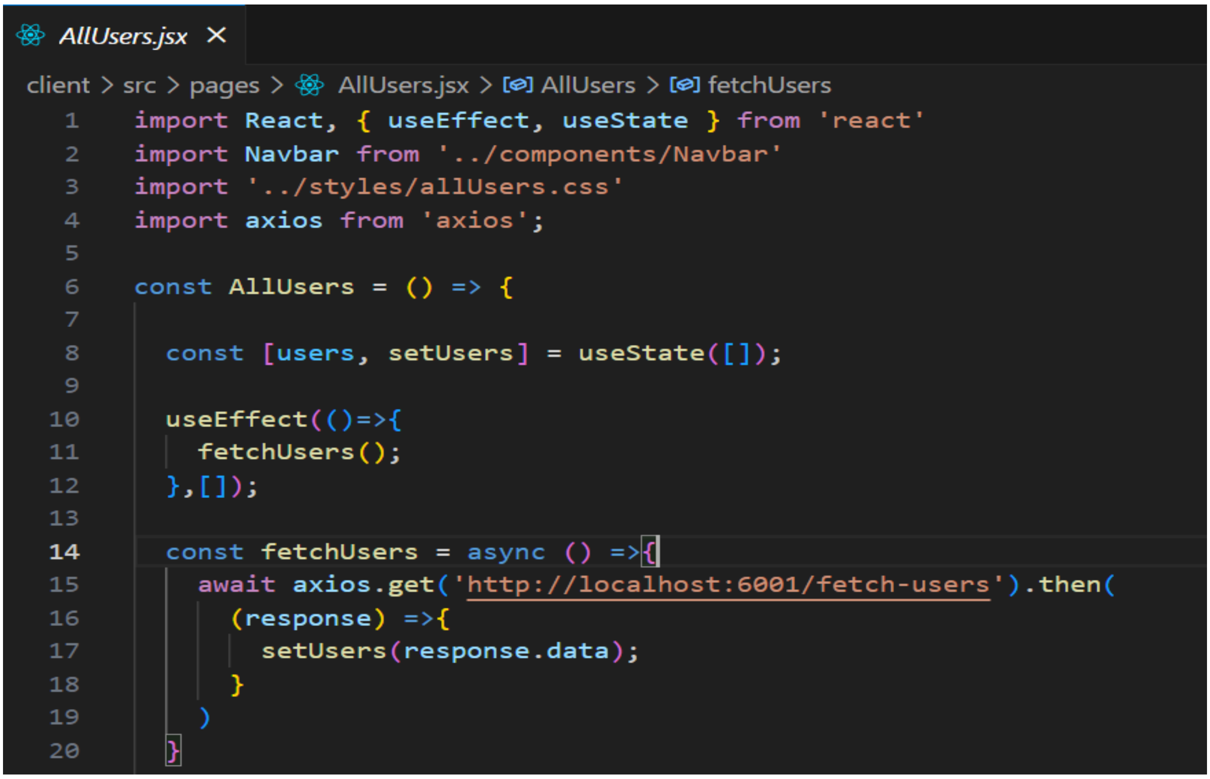
_id: 668e69cc43e9ffd64dde0354
Flight Id: 04 Flight name: op1
Starting station: Delhi Departure time: 18:29
Destination: Hyderabad Arrival time: 14:29
Base price: 99999 Total seats: 300

_id: 668e6a4243e9ffd64dde036e
Flight Id: 05 Flight name: op1
Starting station: Mumbai Departure time: 02:31
Destination: Bhopal Arrival time: 09:26
Base price: 12345678 Total seats: 198

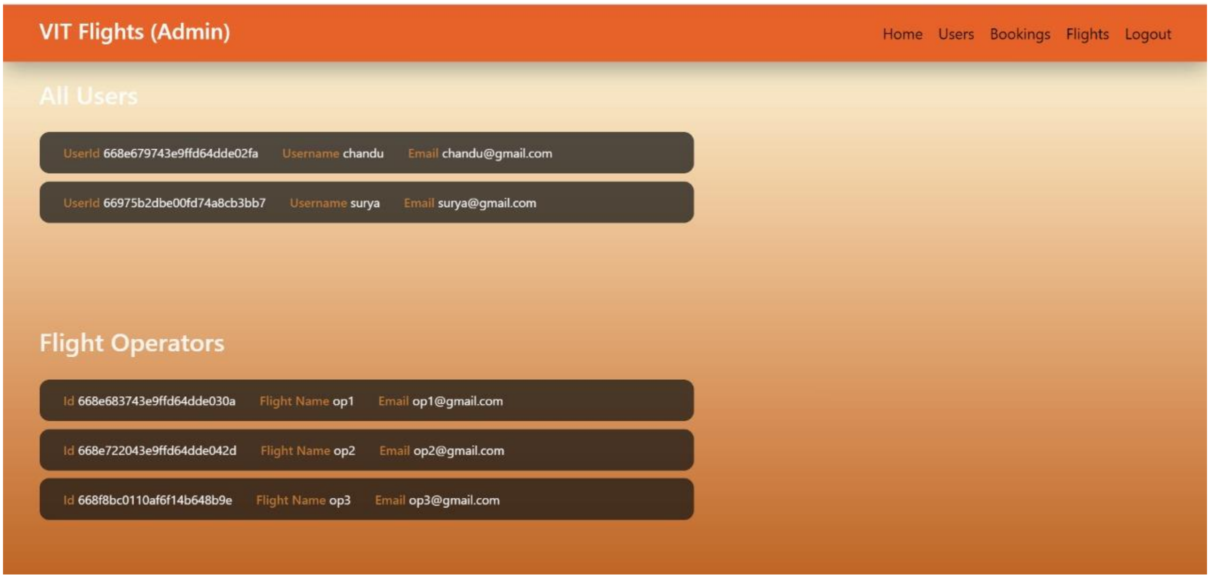
_id: 668e6a9443e9ffd64dde0376

_id: 668e70ee43e9ffd64dde03e1

AllUsers.jsx



```jsx
import React, { useEffect, useState } from 'react'
import Navbar from '../components/Navbar'
import '../styles/allUsers.css'
import axios from 'axios';

const AllUsers = () => {

  const [users, setUsers] = useState([]);

  useEffect(()=>{
    fetchUsers();
  },[]);

  const fetchUsers = async () =>{
    await axios.get('http://localhost:6001/fetch-users').then(
      (response) =>{
        setUsers(response.data);
      }
    )
  }
```



All Users

UserId 668e679743e9ffd64dde02fa Username chandu Email chandu@gmail.com

UserId 66975b2dbe00fd74a8cb3bb7 Username surya Email surya@gmail.com

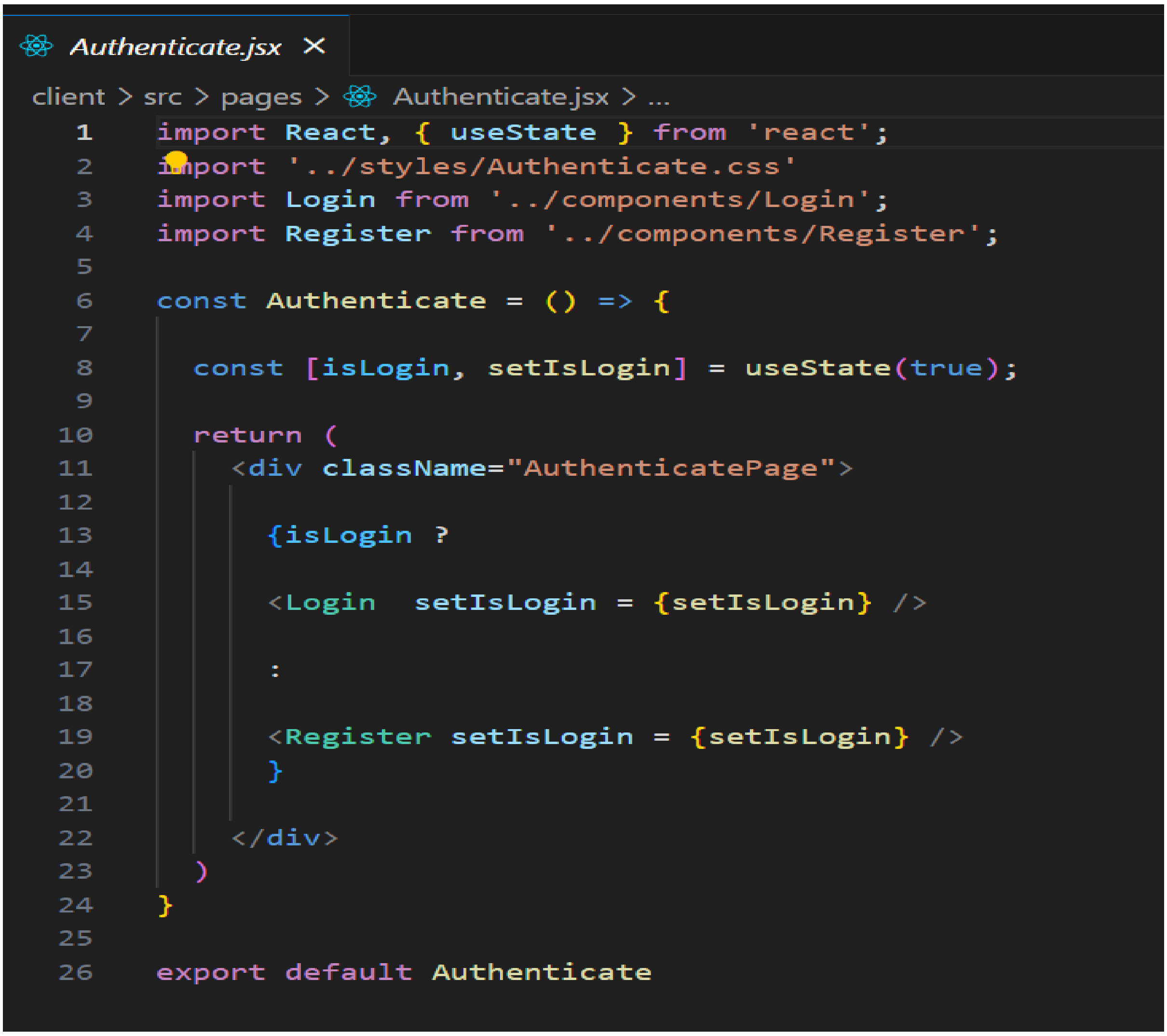
Flight Operators

Id 668e683743e9ffd64dde030a Flight Name op1 Email op1@gmail.com

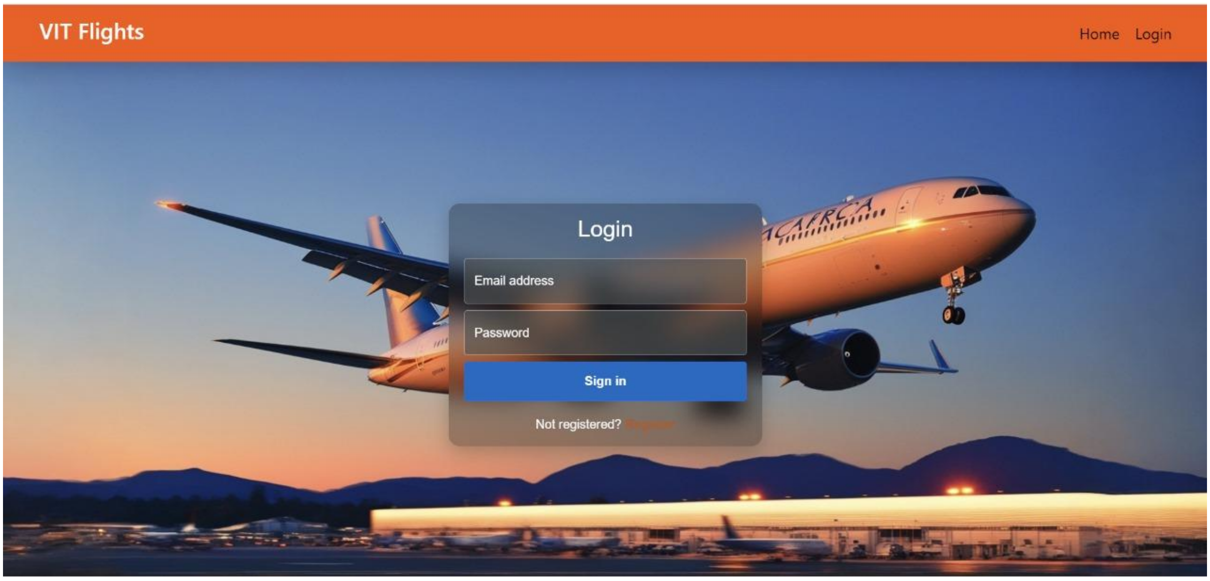
Id 668e722043e9ffd64dde042d Flight Name op2 Email op2@gmail.com

Id 668f8bc0110af6f14b648b9e Flight Name op3 Email op3@gmail.com

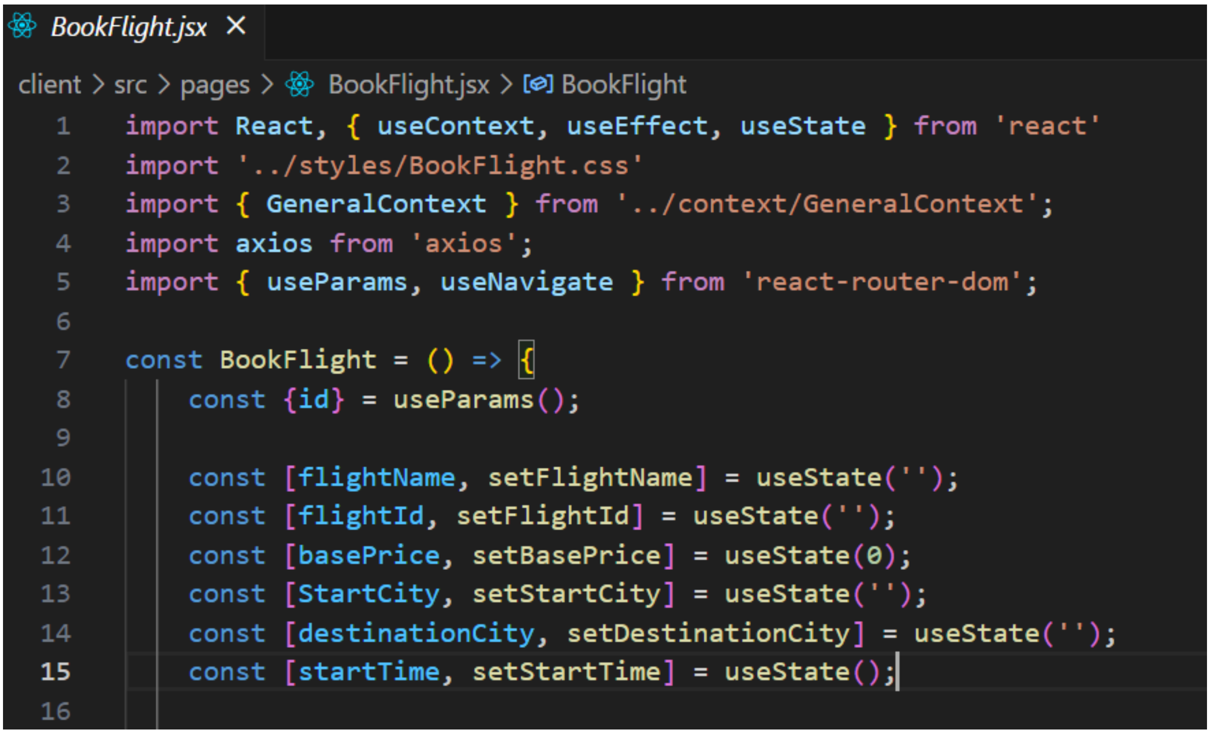
Authenticate.jsx



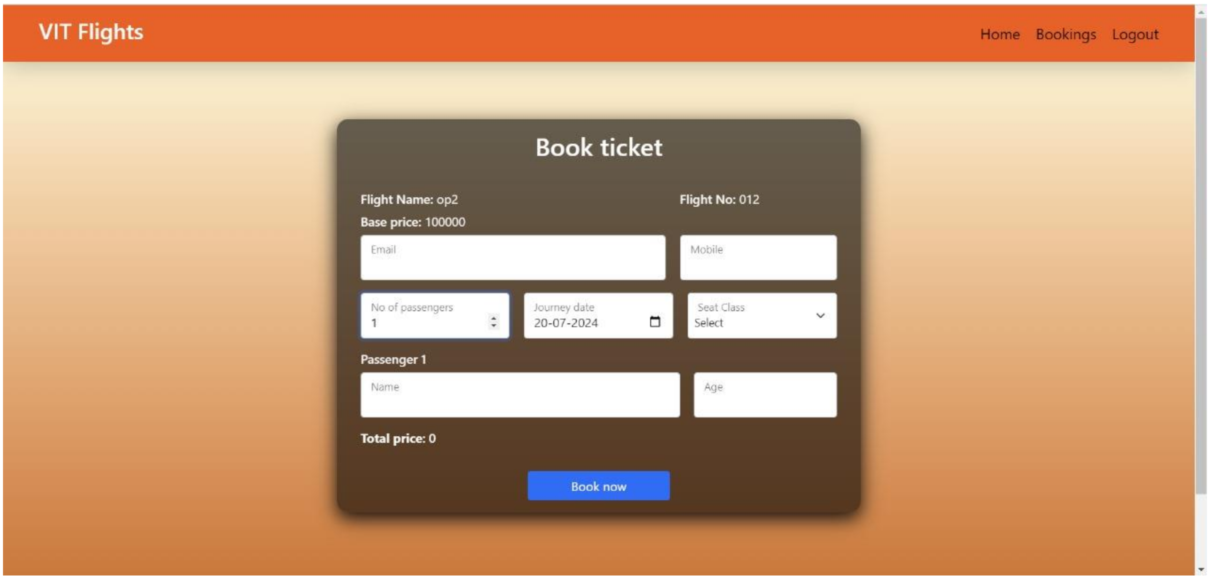
```jsx
import React, { useState } from 'react';
import '../styles/Authenticate.css'
import Login from '../components/Login';
import Register from '../components/Register';

const Authenticate = () => {

  const [isLogin, setIsLogin] = useState(true);

  return (
    <div className="AuthenticatePage">

      {isLogin ?

      <Login  setIsLogin = {setIsLogin} />

      :

      <Register setIsLogin = {setIsLogin} />
      }

    </div>
  )
}

export default Authenticate
```



BookFlight.jsx



```
import React, { useContext, useEffect, useState } from 'react'
import '../styles/BookFlight.css'
import { GeneralContext } from '../context/GeneralContext';
import axios from 'axios';
import { useParams, useNavigate } from 'react-router-dom';

const BookFlight = () => {
    const {id} = useParams();

    const [flightName, setFlightName] = useState('');
    const [flightId, setFlightId] = useState('');
    const [basePrice, setBasePrice] = useState(0);
    const [StartCity, setStartCity] = useState('');
    const [destinationCity, setDestinationCity] = useState('');
    const [startTime, setStartTime] = useState();

```



Book ticket

Flight Name: op2 Flight No: 012

Base price: 100000

Email

Mobile

No of passengers: 1

Journey date: 20-07-2024

Seat Class: Select

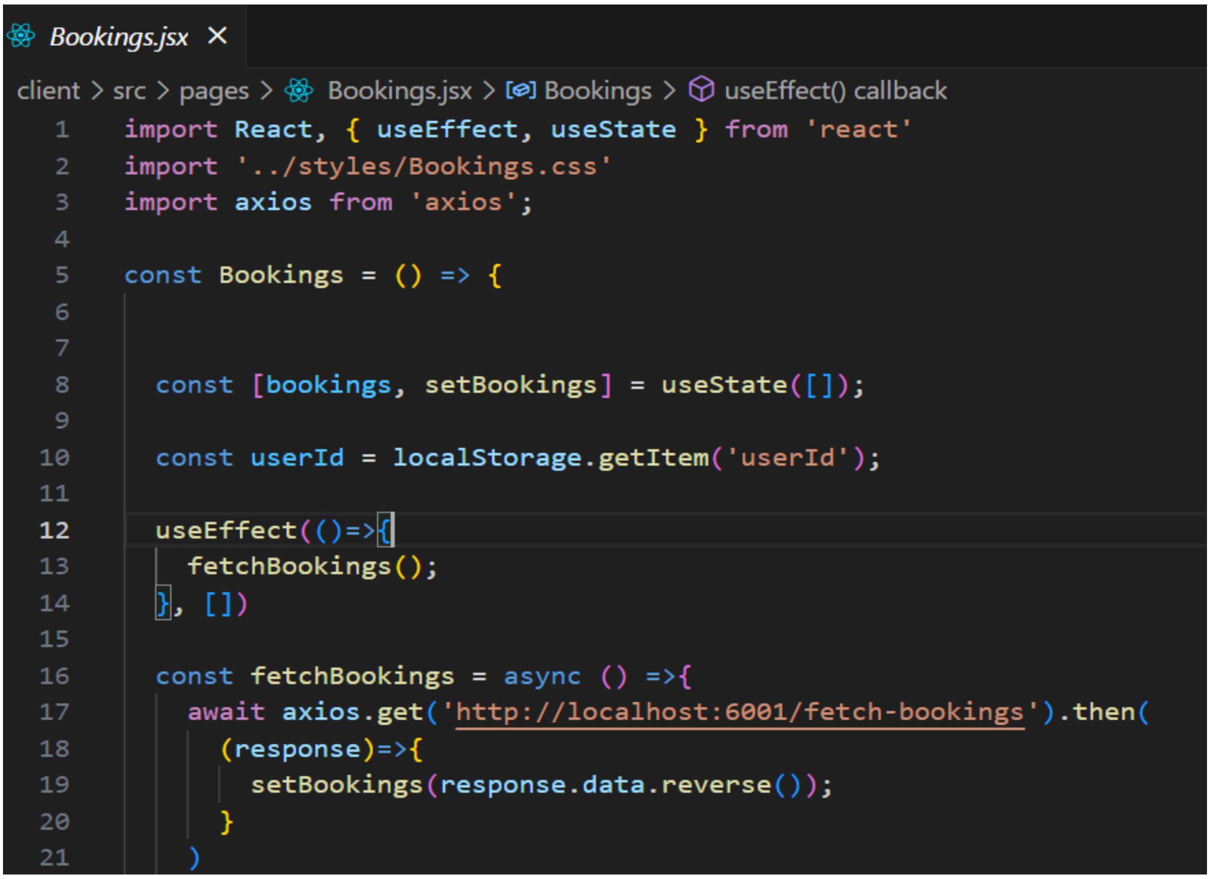
Passenger 1

Name

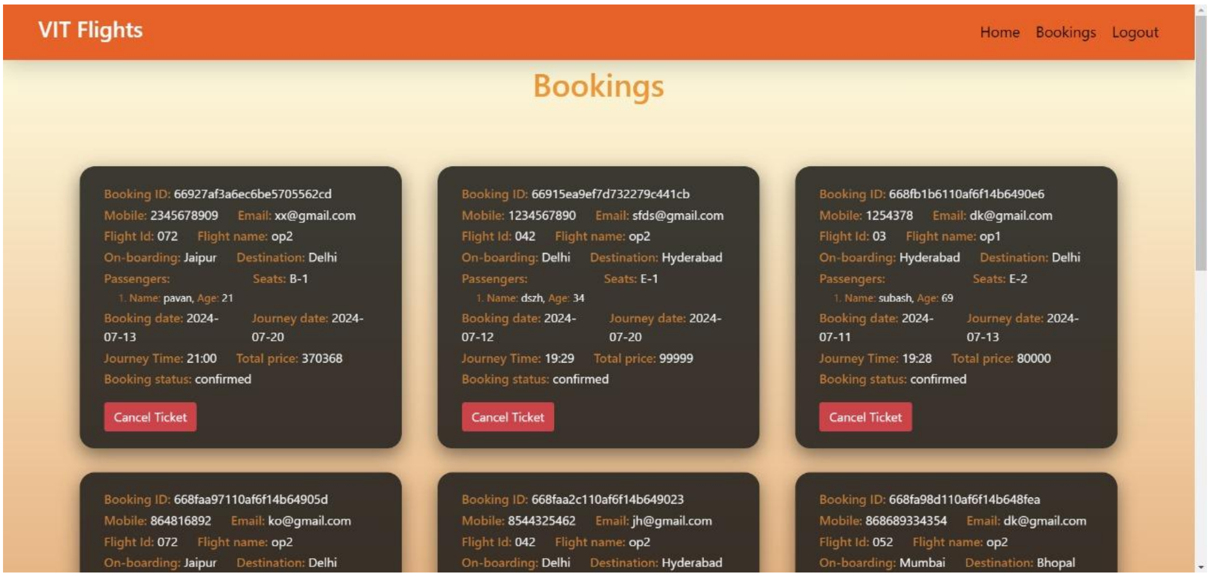
Age

Total price: 0

Bookings.jsx


Bookings.jsx
client > src > pages > Bookings.jsx > Bookings > useEffect() callback

```
import React, { useEffect, useState } from 'react'
import '../styles/Bookings.css'
import axios from 'axios';

const Bookings = () => {


  const [bookings, setBookings] = useState([]);

  const userId = localStorage.getItem('userId');

  useEffect(()=>{
    fetchBookings();
  }, [])

  const fetchBookings = async () =>{
    await axios.get('http://localhost:6001/fetch-bookings').then(
      (response)=>{
        setBookings(response.data.reverse());
      }
    )
```



Bookings

Booking ID: 66927af3a6ec6be5705562cd
Mobile: 2345678909 Email: xx@gmail.com
Flight Id: 072 Flight name: op2
On-boarding: Jaipur Destination: Delhi
Passengers: Seats: B-1
1. Name: pavan, Age: 21
Booking date: 2024-07-13 Journey date: 2024-07-20
Journey Time: 21:00 Total price: 370368
Booking status: confirmed

Booking ID: 66915ea9ef7d732279c441cb
Mobile: 1234567890 Email: sfds@gmail.com
Flight Id: 042 Flight name: op2
On-boarding: Delhi Destination: Hyderabad
Passengers: Seats: E-1
1. Name: dszh, Age: 34
Booking date: 2024-07-12 Journey date: 2024-07-20
Journey Time: 19:29 Total price: 99999
Booking status: confirmed

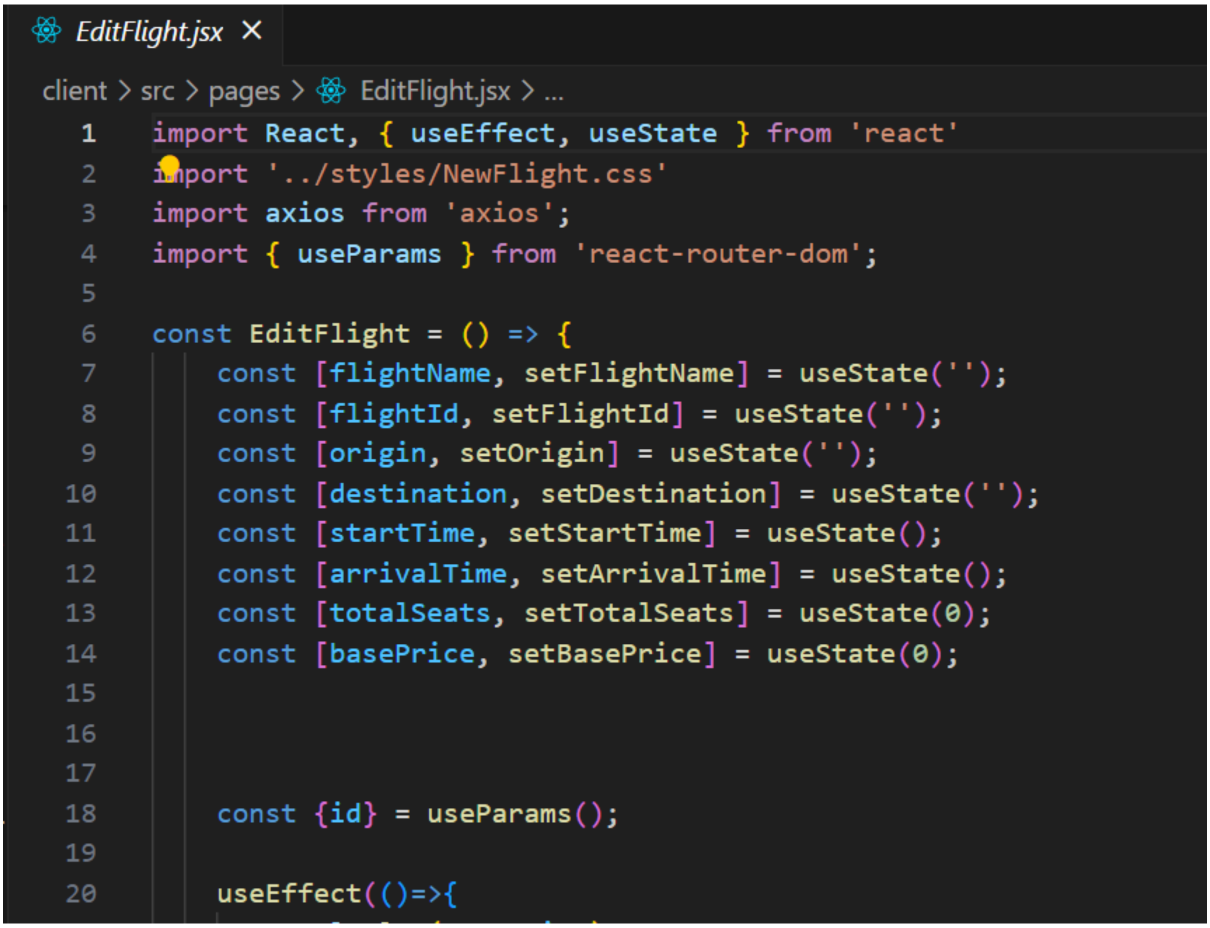
Booking ID: 668fb1b6110af6f14b6490e6
Mobile: 1254378 Email: dk@gmail.com
Flight Id: 03 Flight name: op1
On-boarding: Hyderabad Destination: Delhi
Passengers: Seats: E-2
1. Name: subash, Age: 69
Booking date: 2024-07-11 Journey date: 2024-07-13
Journey Time: 19:28 Total price: 80000
Booking status: confirmed

Booking ID: 668faa97110af6f14b64905d
Mobile: 864816892 Email: ko@gmail.com
Flight Id: 072 Flight name: op2
On-boarding: Jaipur Destination: Delhi

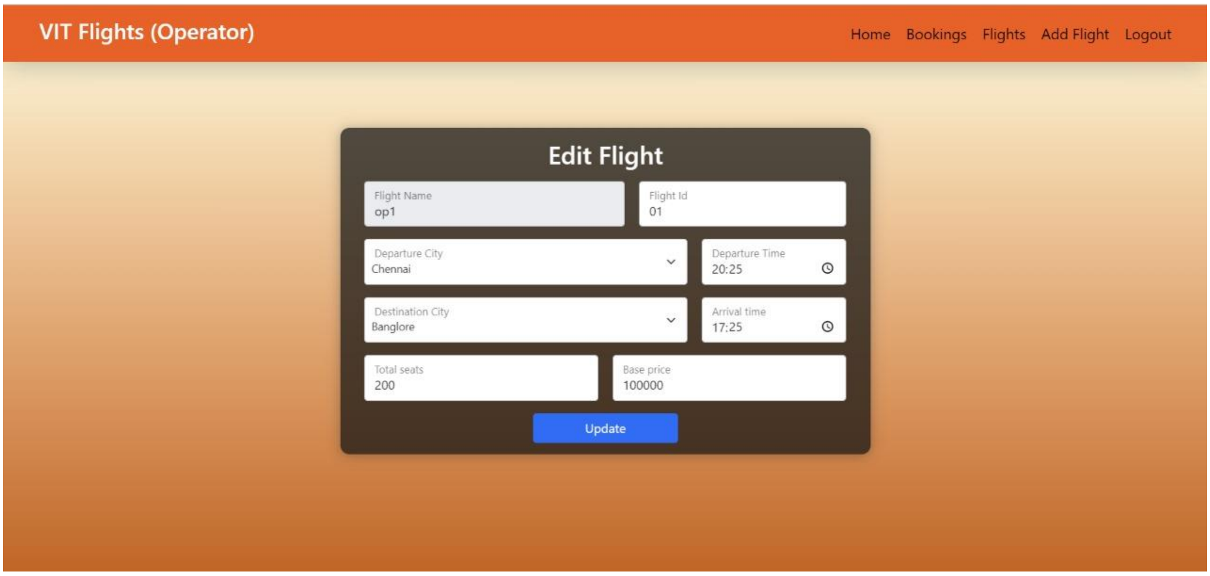
Booking ID: 668faa2c110af6f14b649023
Mobile: 8544325462 Email: jh@gmail.com
Flight Id: 042 Flight name: op2
On-boarding: Delhi Destination: Hyderabad

Booking ID: 668fa98d110af6f14b648fea
Mobile: 868689334354 Email: dk@gmail.com
Flight Id: 052 Flight name: op2
On-boarding: Mumbai Destination: Bhopal

EditFlight.jsx



```
import React, { useEffect, useState } from 'react'
import '../styles/NewFlight.css'
import axios from 'axios';
import { useParams } from 'react-router-dom';

const EditFlight = () => {
    const [flightName, setFlightName] = useState('');
    const [flightId, setFlightId] = useState('');
    const [origin, setOrigin] = useState('');
    const [destination, setDestination] = useState('');
    const [startTime, setStartTime] = useState();
    const [arrivalTime, setArrivalTime] = useState();
    const [totalSeats, setTotalSeats] = useState(0);
    const [basePrice, setBasePrice] = useState(0);



    const {id} = useParams();

    useEffect(()=>{
```



Edit Flight

Flight Name: op1

Flight Id: 01

Departure City: Chennai

Departure Time: 20:25

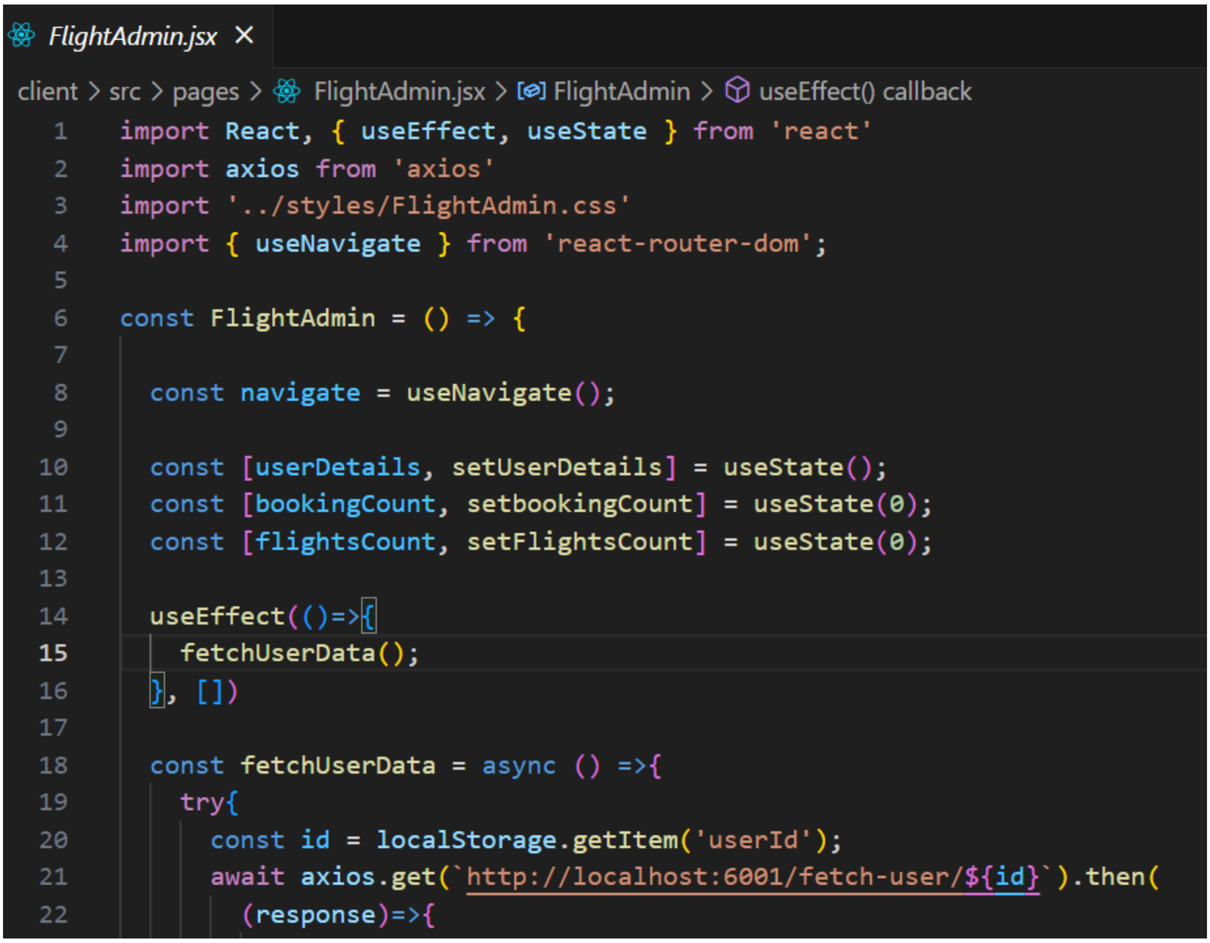
Destination City: Banglore

Arrival time: 17:25

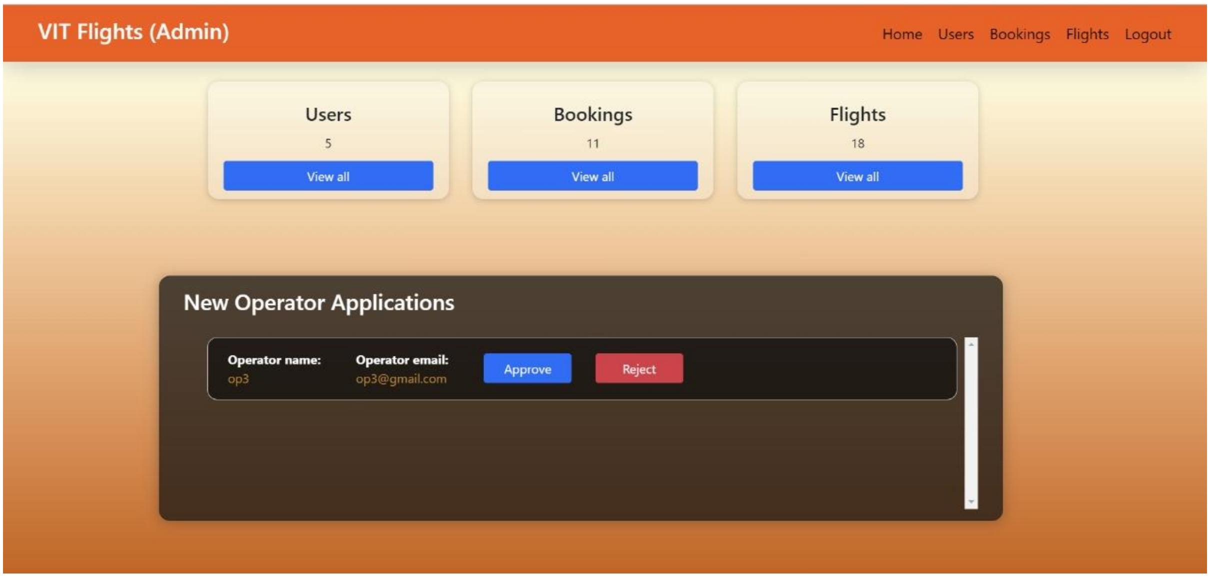
Total seats: 200

Base price: 100000

FlightAdmin.jsx


FlightAdmin.jsx
client > src > pages > FlightAdmin.jsx > FlightAdmin > useEffect() callback

```
import React, { useEffect, useState } from 'react'
import axios from 'axios'
import '../styles/FlightAdmin.css'
import { useNavigate } from 'react-router-dom';

const FlightAdmin = () => {

  const navigate = useNavigate();

  const [userDetails, setUserDetails] = useState();
  const [bookingCount, setbookingCount] = useState(0);
  const [flightsCount, setFlightsCount] = useState(0);

  useEffect(()=>{
    fetchUserData();
  }, [])

  const fetchUserData = async () =>{
    try{
      const id = localStorage.getItem('userId');
      await axios.get(`http://localhost:6001/fetch-user/${id}`).then(
        (response)=>{
```



Users
5

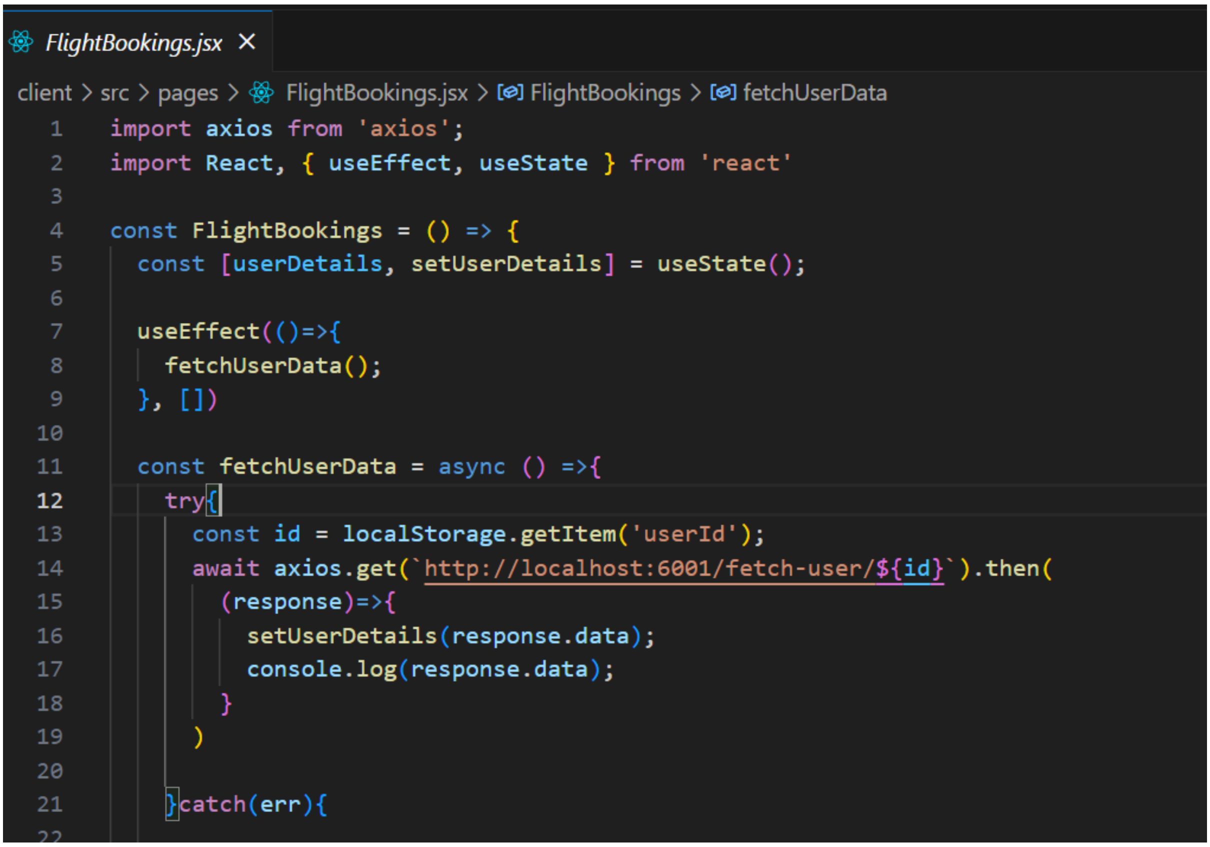
Bookings
11

Flights
18

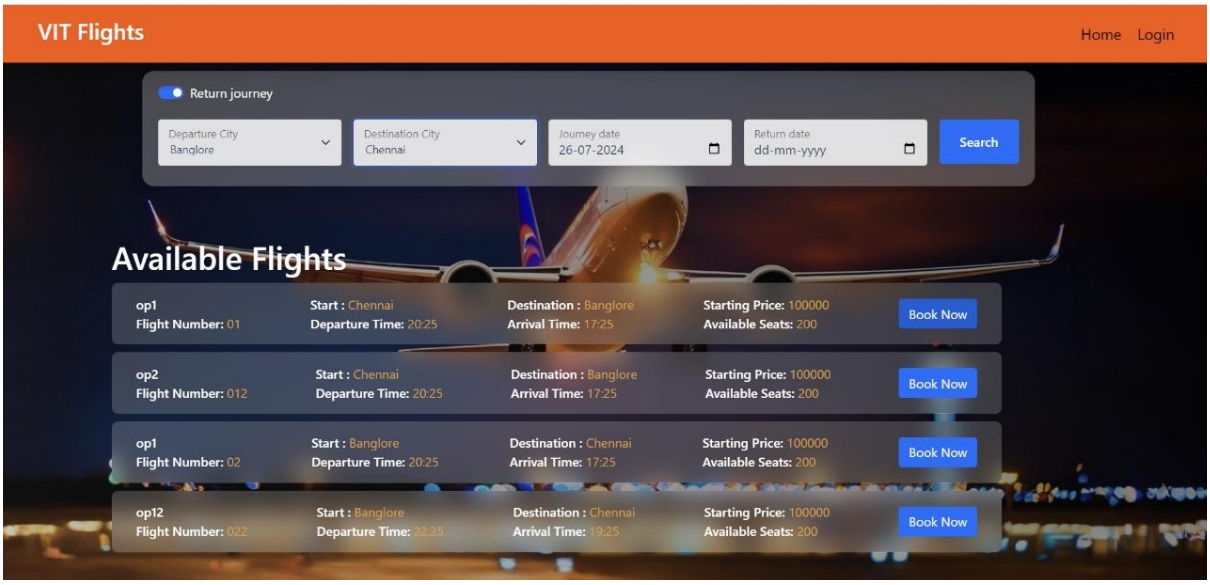
New Operator Applications

Operator name: op3
Operator email: op3@gmail.com

FlightBooking.jsx


FlightBookings.jsx ✕
client > src > pages > FlightBookings.jsx > FlightBookings > fetchUserData

```
import axios from 'axios';
import React, { useEffect, useState } from 'react'

const FlightBookings = () => {
  const [userDetails, setUserDetails] = useState();

  useEffect(()=>{
    fetchUserData();
  }, [])

  const fetchUserData = async () =>{
    try{
      const id = localStorage.getItem('userId');
      await axios.get(`http://localhost:6001/fetch-user/${id}`).then(
        (response)=>{
          setUserDetails(response.data);
          console.log(response.data);
        }
      )

    }catch(err){
```



Return journey

Departure City: Banglore

Destination City: Chennai

Journey date: 26-07-2024

Return date: dd-mm-yyyy

Available Flights

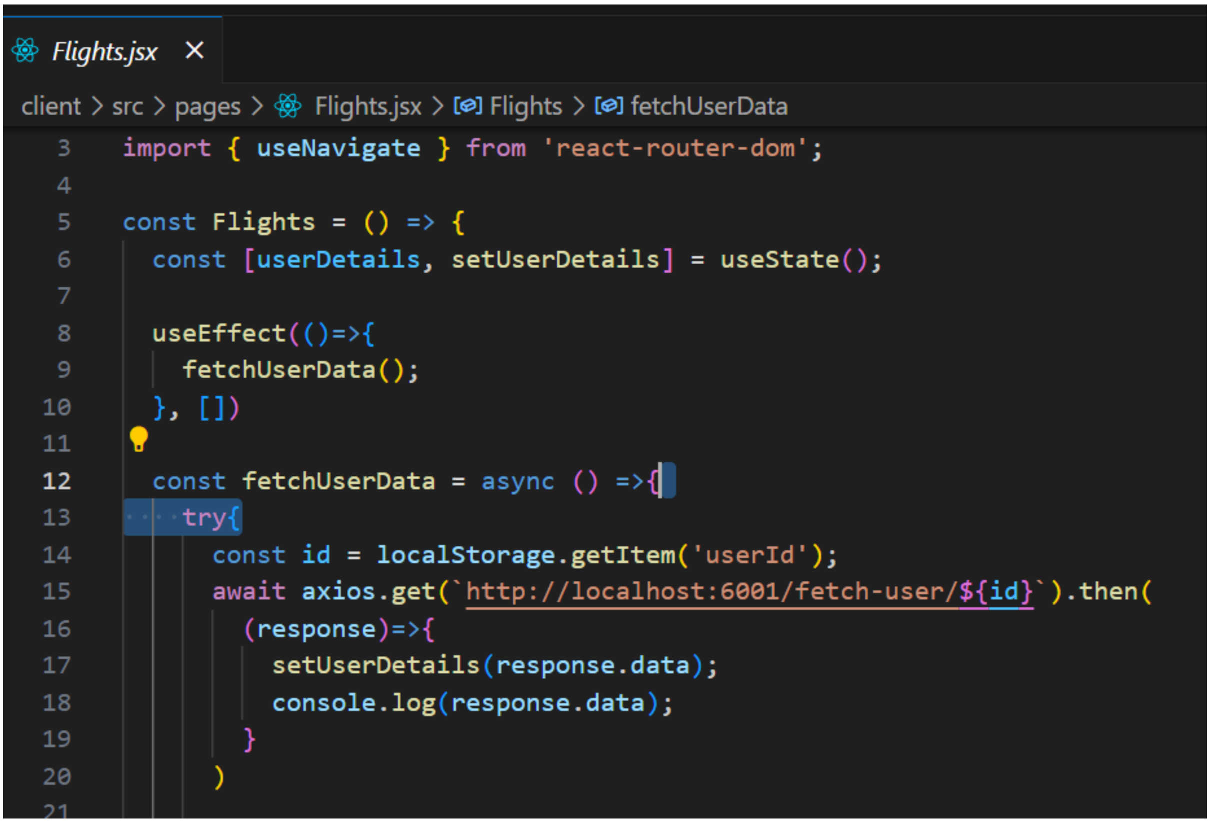
op1
Flight Number: 01
Start : Chennai
Departure Time: 20:25
Destination : Banglore
Arrival Time: 17:25
Starting Price: 100000
Available Seats: 200

op2
Flight Number: 012
Start : Chennai
Departure Time: 20:25
Destination : Banglore
Arrival Time: 17:25
Starting Price: 100000
Available Seats: 200

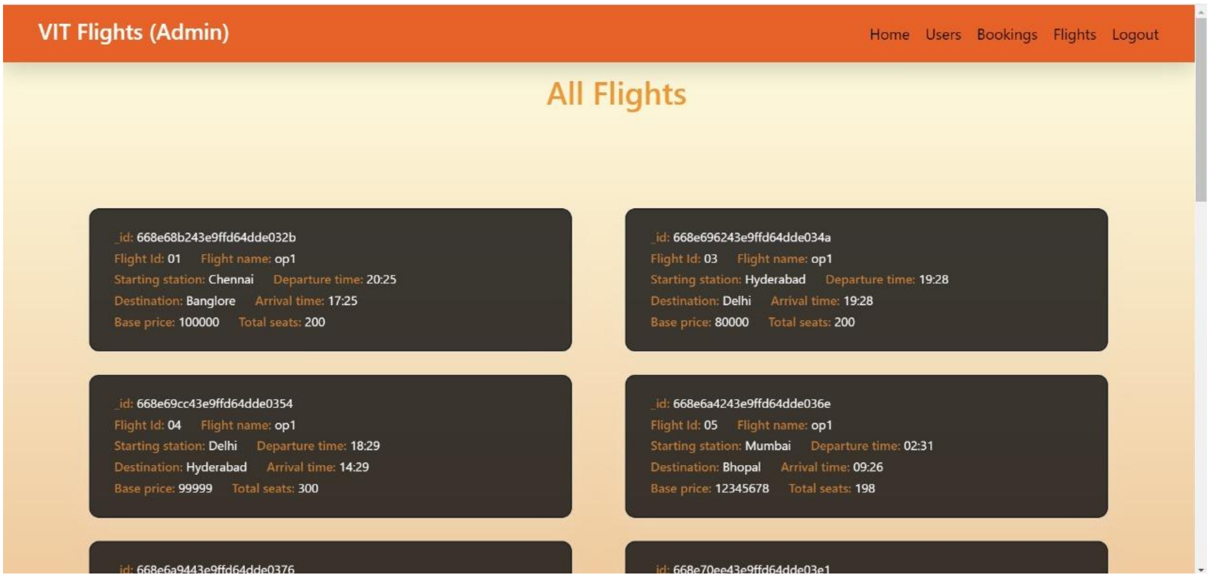
op1
Flight Number: 02
Start : Banglore
Departure Time: 20:25
Destination : Chennai
Arrival Time: 17:25
Starting Price: 100000
Available Seats: 200

op12
Flight Number: 022
Start : Banglore
Departure Time: 22:25
Destination : Chennai
Arrival Time: 19:25
Starting Price: 100000
Available Seats: 200

Flights.jsx


Flights.jsx
client > src > pages > Flights.jsx > Flights > fetchUserData

```jsx
import { useNavigate } from 'react-router-dom';

const Flights = () => {
  const [userDetails, setUserDetails] = useState();

  useEffect(()=>{
    fetchUserData();
  }, [])

  const fetchUserData = async () =>{
    try{
      const id = localStorage.getItem('userId');
      await axios.get(`http://localhost:6001/fetch-user/${id}`).then(
        (response)=>{
          setUserDetails(response.data);
          console.log(response.data);
        }
      )
```



All Flights

_id: 668e68b243e9ffd64dde032b
Flight Id: 01 Flight name: op1
Starting station: Chennai Departure time: 20:25
Destination: Banglore Arrival time: 17:25
Base price: 100000 Total seats: 200

_id: 668e696243e9ffd64dde034a
Flight Id: 03 Flight name: op1
Starting station: Hyderabad Departure time: 19:28
Destination: Delhi Arrival time: 19:28
Base price: 80000 Total seats: 200

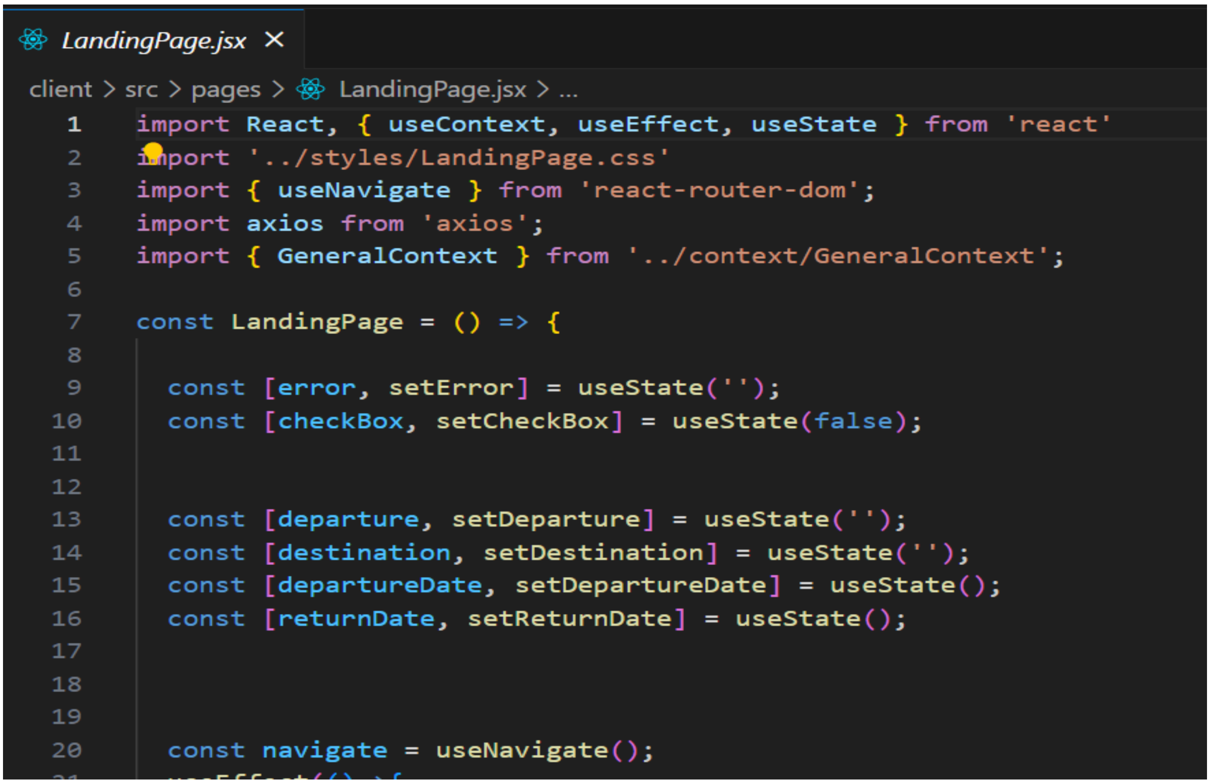
_id: 668e69cc43e9ffd64dde0354
Flight Id: 04 Flight name: op1
Starting station: Delhi Departure time: 18:29
Destination: Hyderabad Arrival time: 14:29
Base price: 99999 Total seats: 300

_id: 668e6a4243e9ffd64dde036e
Flight Id: 05 Flight name: op1
Starting station: Mumbai Departure time: 02:31
Destination: Bhopal Arrival time: 09:26
Base price: 12345678 Total seats: 198

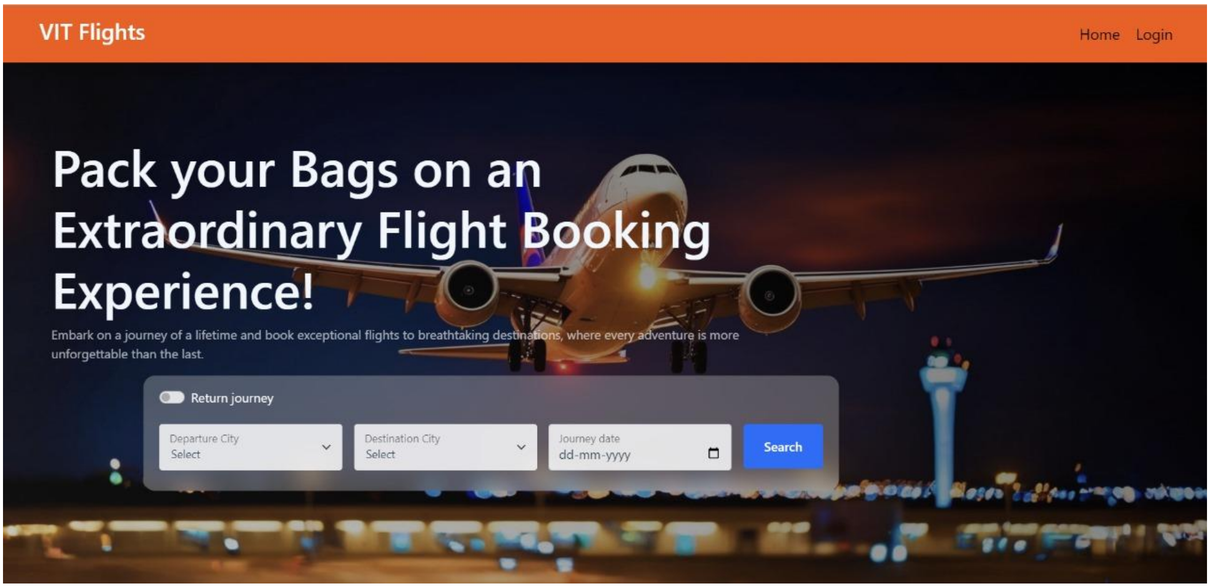
_id: 668e6a9443e9ffd64dde0376

_id: 668e70ee43e9ffd64dde03e1

LandingPage.jsx

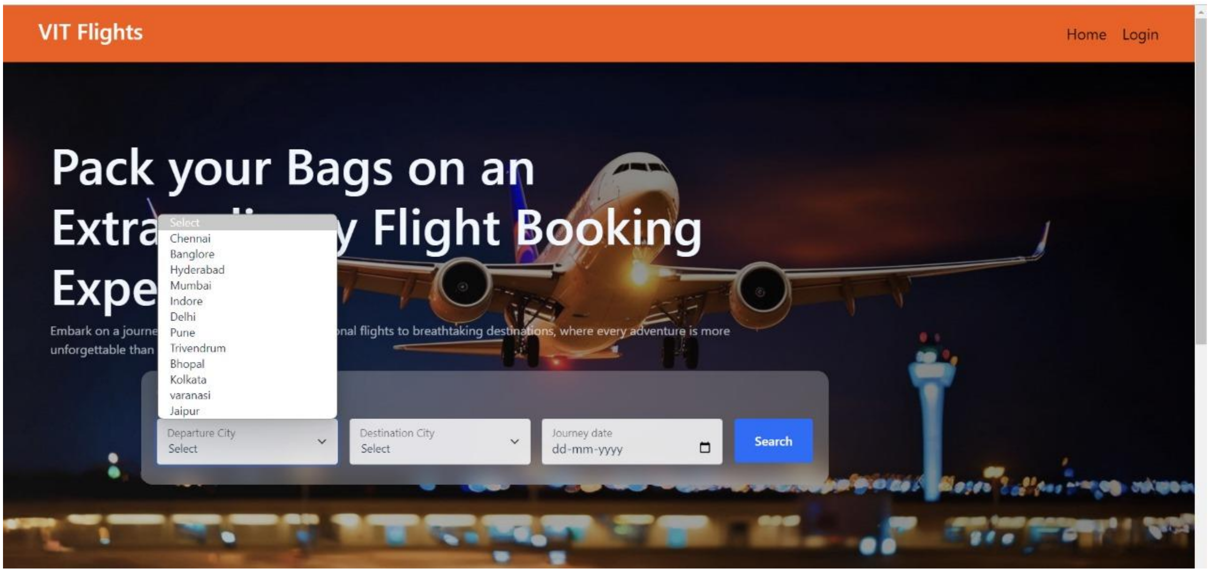


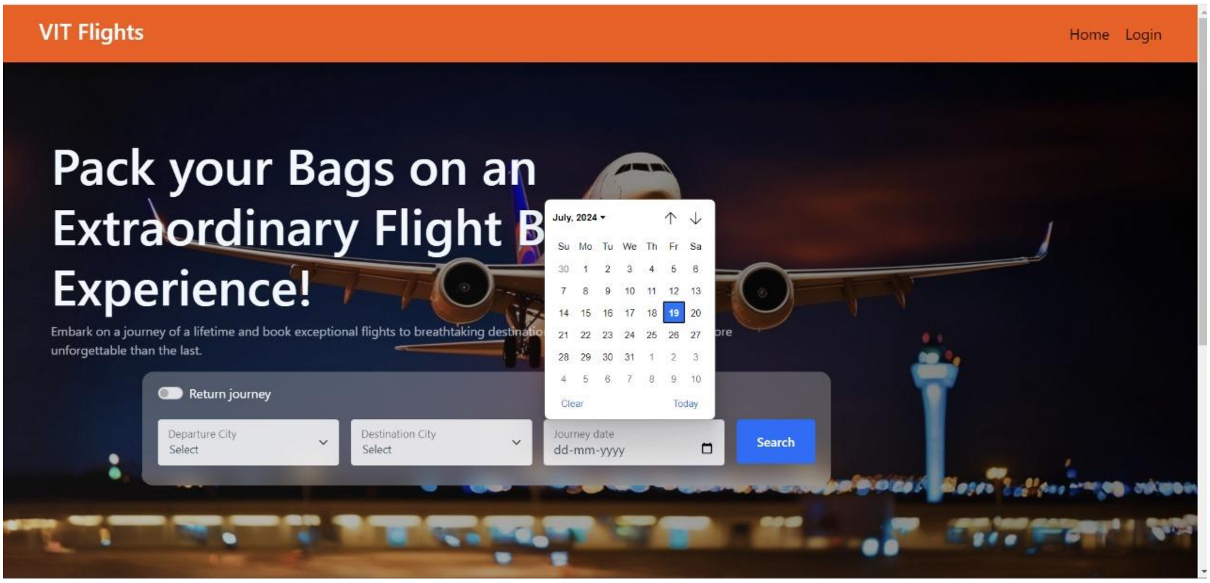
```jsx
import React, { useContext, useEffect, useState } from 'react'
import '../styles/LandingPage.css'
import { useNavigate } from 'react-router-dom';
import axios from 'axios';
import { GeneralContext } from '../context/GeneralContext';

const LandingPage = () => {

  const [error, setError] = useState('');
  const [checkBox, setCheckBox] = useState(false);


  const [departure, setDeparture] = useState('');
  const [destination, setDestination] = useState('');
  const [departureDate, setDepartureDate] = useState();
  const [returnDate, setReturnDate] = useState();



  const navigate = useNavigate();
  useEffect(() =>{
```



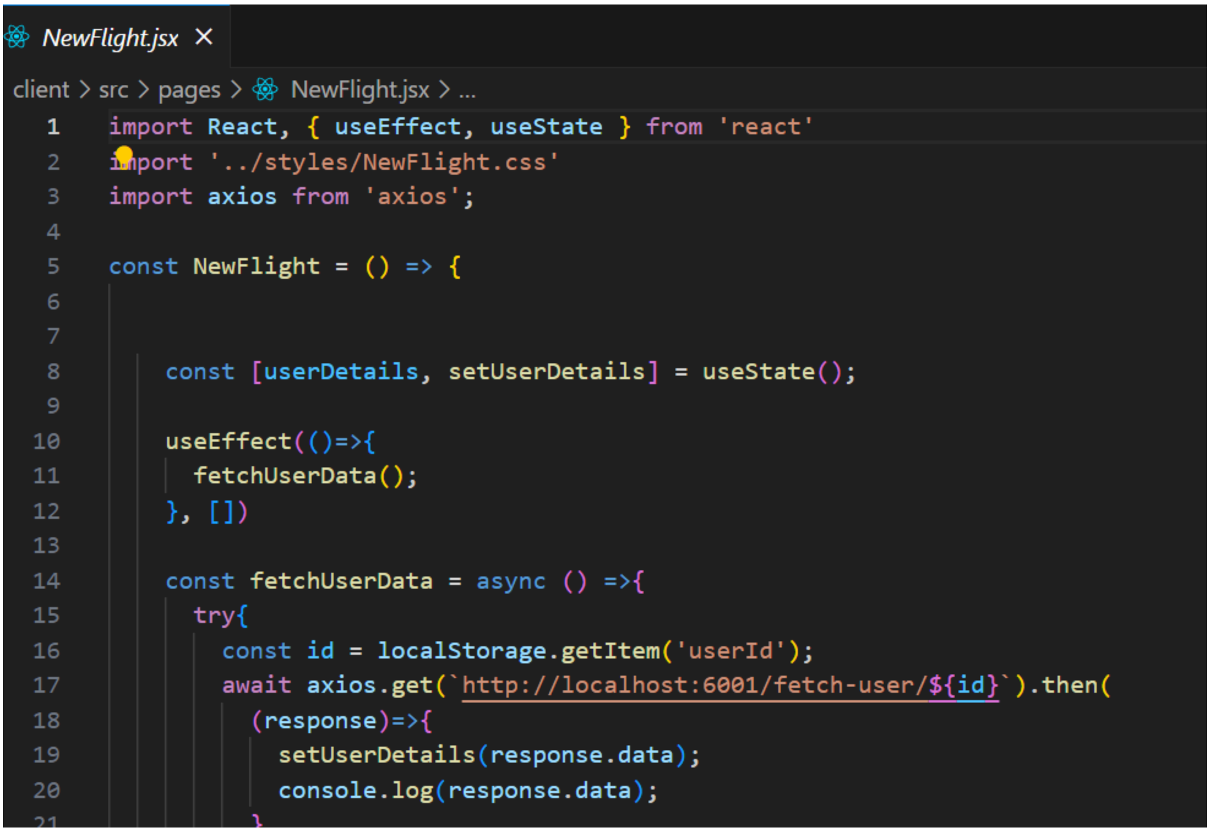
Pack your Bags on an Extraordinary Flight Booking Experience!

Embark on a journey of a lifetime and book exceptional flights to breathtaking destinations, where every adventure is more unforgettable than the last.

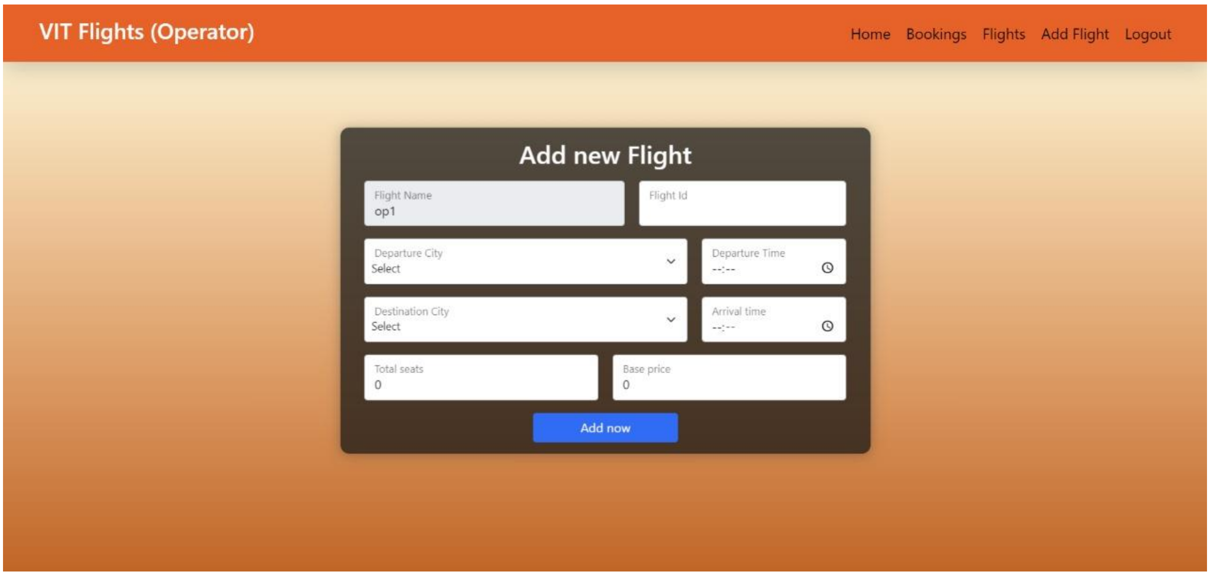




NewFlight.jsx



```
import React, { useEffect, useState } from 'react'
import '../styles/NewFlight.css'
import axios from 'axios';

const NewFlight = () => {


    const [userDetails, setUserDetails] = useState();

    useEffect(()=>{
      fetchUserData();
    }, [])

    const fetchUserData = async () =>{
      try{
        const id = localStorage.getItem('userId');
        await axios.get(`http://localhost:6001/fetch-user/${id}`).then(
          (response)=>{
            setUserDetails(response.data);
            console.log(response.data);
          }
```



Add new Flight

Flight Name: op1

Flight Id

Departure City: Select

Departure Time: --:--

Destination City: Select

Arrival time: --:--

Total seats: 0

Base price: 0

Routing

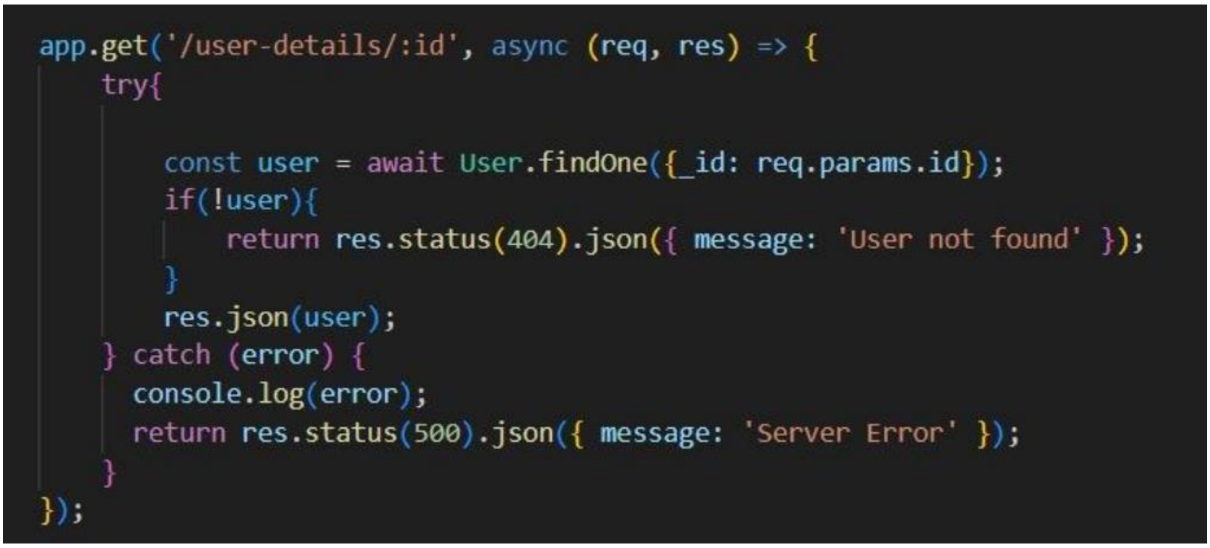
Routing is managed using React Router. Here are the main routes:

- **/home**: Landing page of the application.
- **/auth:** Navigate to the authentication page for users to log in or sign up.
- **/book-Flight/:id:** Navigate to the page where users can book a specific flight.
- **/bookings:** Navigate to the page displaying all the user's bookings.
- **/admin:** Admin dashboard to control the application.
- **/all-users:** Navigate to the page displaying all registered users.
- **/all-bookings:** Navigate to the page displaying all bookings.
- **/all-flights:** Navigate to the page displaying all available flights.
- **/flight-admin:** Navigate to the flight administration page.
- **/flight-bookings:** Navigate to the page displaying bookings for flights.
- **/flights:** Navigate to the page displaying flight information.
- **/new-flight:** Navigate to the page for adding a new flight.
- **/edit-flight/:id:** Navigate to the page for editing flight details.

Integration with Backend

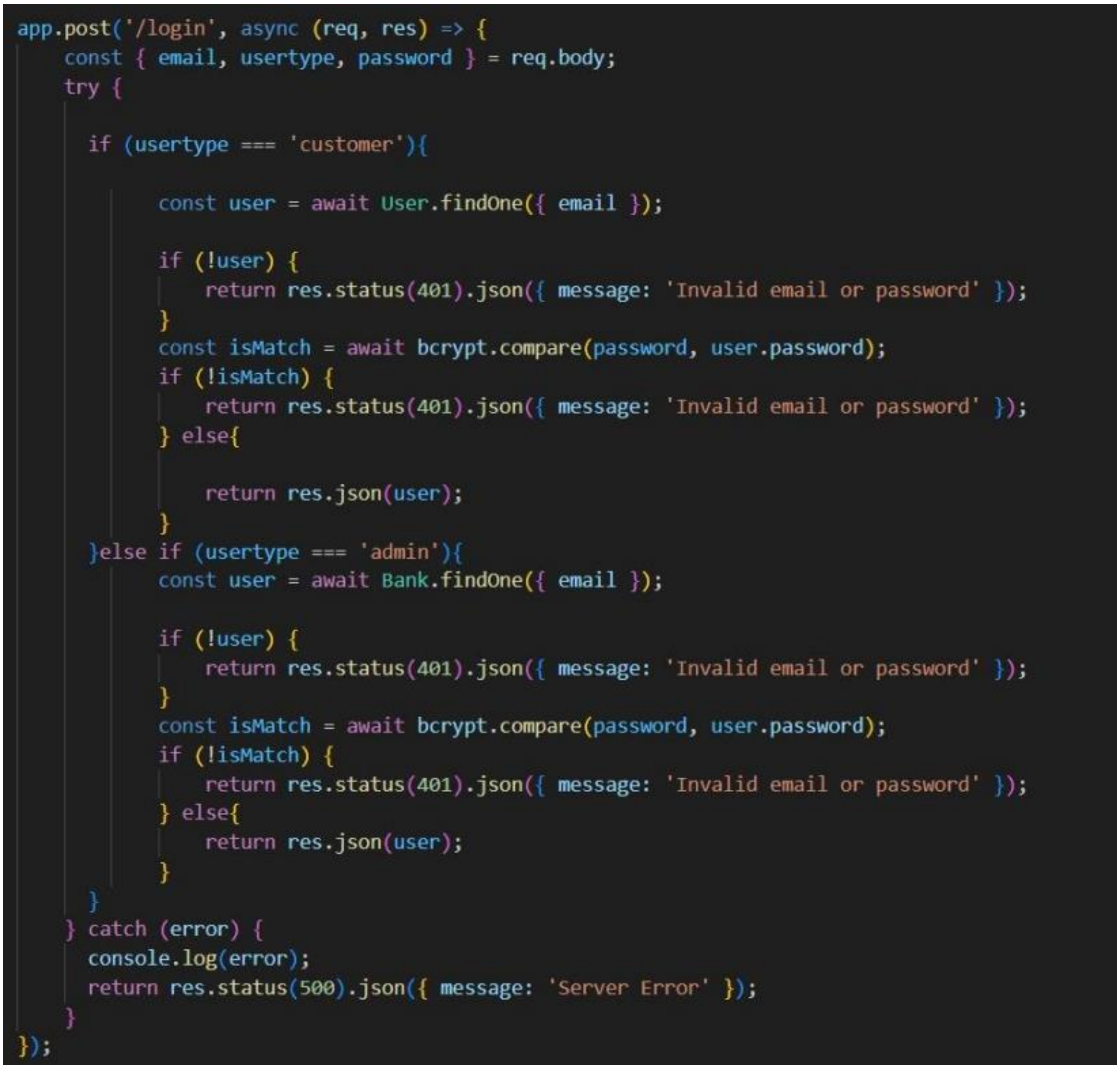
The frontend communicates with the backend APIs hosted on **http://localhost:5173/**. Key endpoints include:

- **GET /api/data** - Retrieves data for display.



```
app.get('/user-details/:id', async (req, res) => {
    try{

        const user = await User.findOne({_id: req.params.id});
        if(!user){
            return res.status(404).json({ message: 'User not found' });
        }
        res.json(user);
    } catch (error) {
        console.log(error);
        return res.status(500).json({ message: 'Server Error' });
    }
});
```

- **POST /api/user/login** - Handles user authentication.



```
app.post('/login', async (req, res) => {
    const { email, usertype, password } = req.body;
    try {

      if (usertype === 'customer'){

            const user = await User.findOne({ email });

            if (!user) {
                return res.status(401).json({ message: 'Invalid email or password' });
            }
            const isMatch = await bcrypt.compare(password, user.password);
            if (!isMatch) {
                return res.status(401).json({ message: 'Invalid email or password' });
            } else{

                return res.json(user);
            }
      }else if (usertype === 'admin'){
            const user = await Bank.findOne({ email });

            if (!user) {
                return res.status(401).json({ message: 'Invalid email or password' });
            }
            const isMatch = await bcrypt.compare(password, user.password);
            if (!isMatch) {
                return res.status(401).json({ message: 'Invalid email or password' });
            } else{
                return res.json(user);
            }
      }
    } catch (error) {
      console.log(error);
      return res.status(500).json({ message: 'Server Error' });
    }
});
```

User Interface (UI) Design

- The UI design follows a [describe design principles].
  - **Simplicity:** The user interface is crafted to be simple and intuitive, reducing the user's learning curve and ensuring ease of use. Elements are clearly presented to avoid clutter and confusion.
  - **Responsiveness:** The UI is designed to be responsive, adapting seamlessly to various screen sizes and devices. This ensures accessibility and usability on desktops, tablets, and mobile devices.

- **Consistency:** The application maintains a consistent layout, color scheme, typography, and interaction patterns throughout, providing a cohesive user experience.
- **Accessibility:** The design considers users with disabilities, incorporating features such as keyboard navigation, screen reader compatibility, and sufficient color contrast.
- **User-Centric:** The design prioritizes user needs, making key actions easily accessible and user flows logical. Users receive prompt feedback for their actions.

- Implemented using [UI framework/library].

    - **UI Framework/Library**: The user interface is built using React, a widely-used JavaScript library for constructing user interfaces. React allows for the creation of reusable components and ensures efficient rendering of the UI.
    - **HTTP Requests**: Axios handles HTTP requests to the backend API, providing a straightforward and clean syntax for server interactions and managing responses and errors effectively.
    - **CSS Framework**: Bootstrap is utilized for styling, offering a comprehensive set of predefined classes and components for developing responsive and modern UIs. Custom CSS is also employed to address specific styling needs beyond the framework's capabilities.
    - **Routing**: React Router manages navigation and routing within the application, ensuring smooth transitions between different views and maintaining the single-page application (SPA) experience.
    - **Form Handling**: Formik is used for managing user input in forms, paired with Yup for validation. This combination ensures efficient form submissions and consistent application of validations.