Full Stack Development with MERN

Frontend Development Report

Date	08-07-2024
Team ID	SWTID1720104852
Project Name	Banking Management App(MERN)
Maximum Marks	10

Project Title: Banking Management

Date: 08-07-2024

Prepared by: Chandra Kishore Sure

Team Members:

Hithesh S

Venkata Krishna C

Venkata Subrahmanya Deepak N

Objective

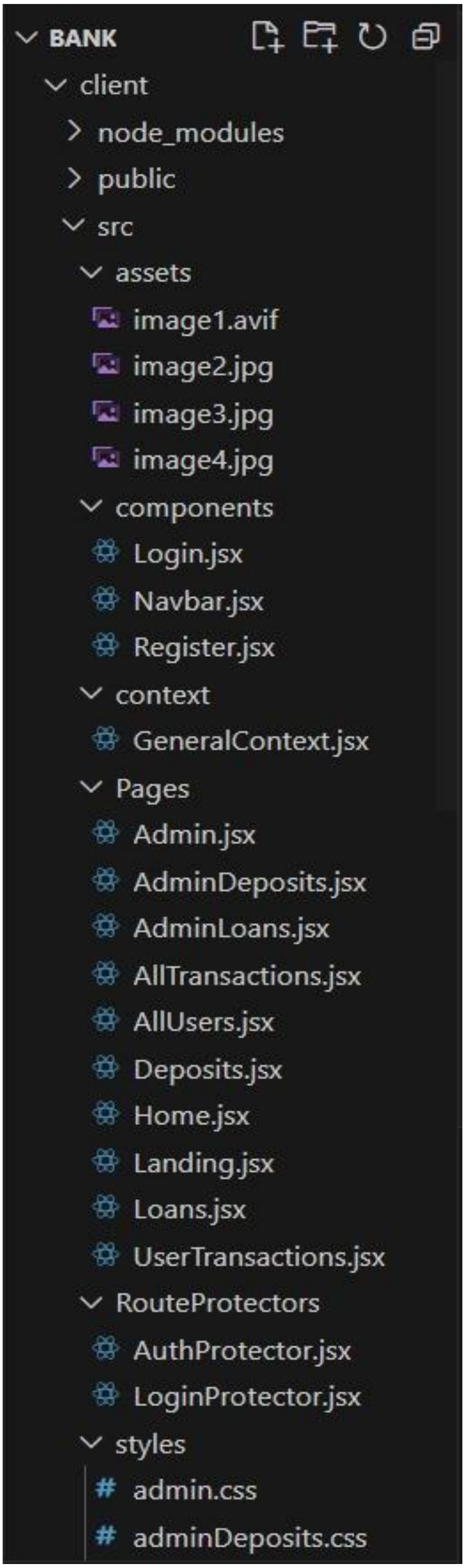
The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the Banking Management project.

Technologies Used

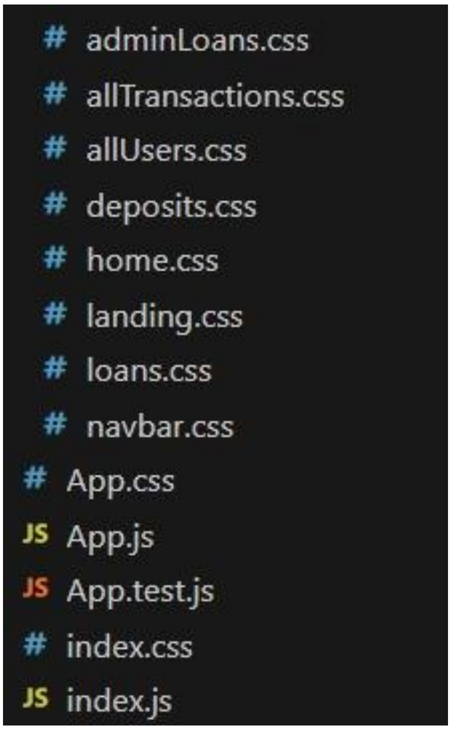
- **Frontend Framework:** React.js
- **State Management:** Redux/Context API
- **UI Framework/Libraries:** Material-UI, Bootstrap
- **API Libraries:** Axios

Project Structure:

Give Screenshot of Frontend Project with explanation

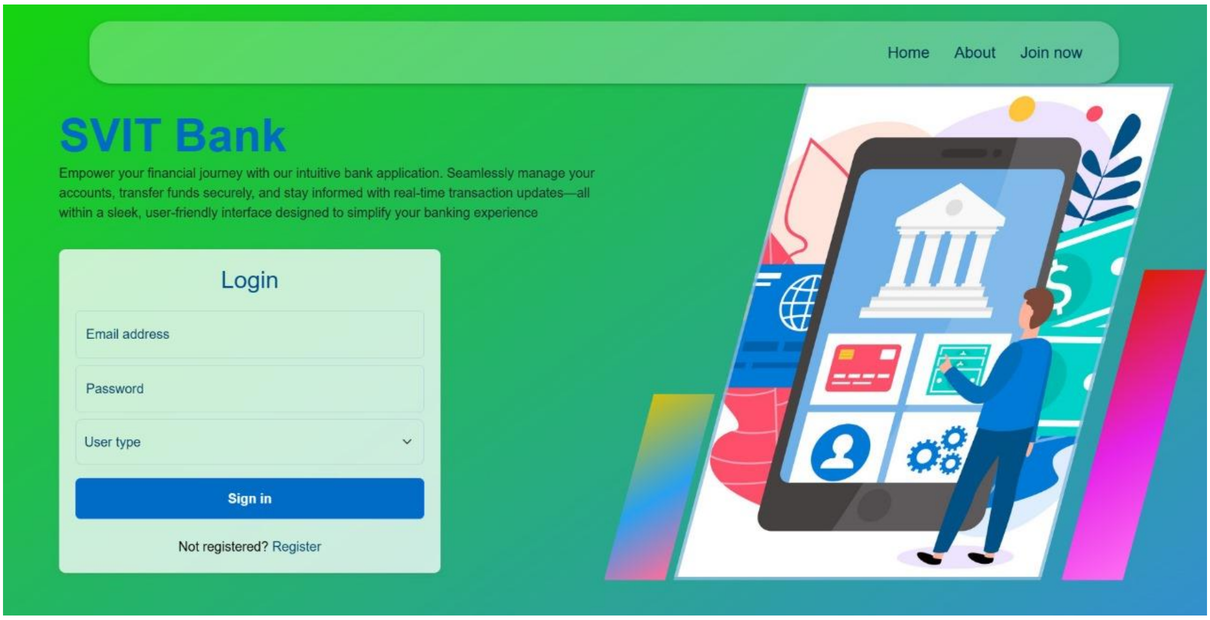


```
BANK
  client
    node_modules
    public
    src
      assets
        image1.avif
        image2.jpg
        image3.jpg
        image4.jpg
      components
        Login.jsx
        Navbar.jsx
        Register.jsx
      context
        GeneralContext.jsx
      Pages
        Admin.jsx
        AdminDeposits.jsx
        AdminLoans.jsx
        AllTransactions.jsx
        AllUsers.jsx
        Deposits.jsx
        Home.jsx
        Landing.jsx
        Loans.jsx
        UserTransactions.jsx
      RouteProtectors
        AuthProtector.jsx
        LoginProtector.jsx
      styles
        admin.css
        adminDeposits.css
```



```
        adminLoans.css
        allTransactions.css
        allUsers.css
        deposits.css
        home.css
        landing.css
        loans.css
        navbar.css
      App.css
      App.js
      App.test.js
      index.css
      index.js
```

Landing Page:

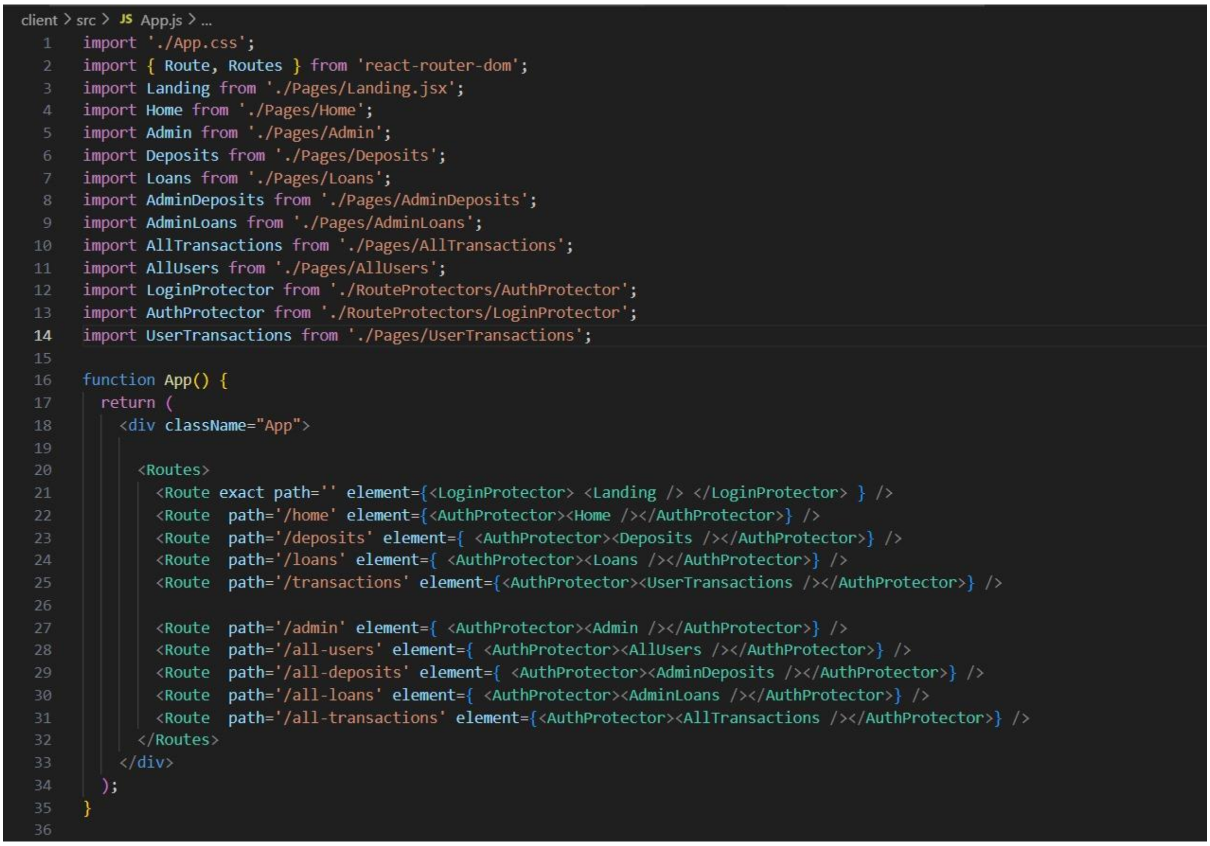


SVIT Bank

Empower your financial journey with our intuitive bank application. Seamlessly manage your accounts, transfer funds securely, and stay informed with real-time transaction updates—all within a sleek, user-friendly interface designed to simplify your banking experience

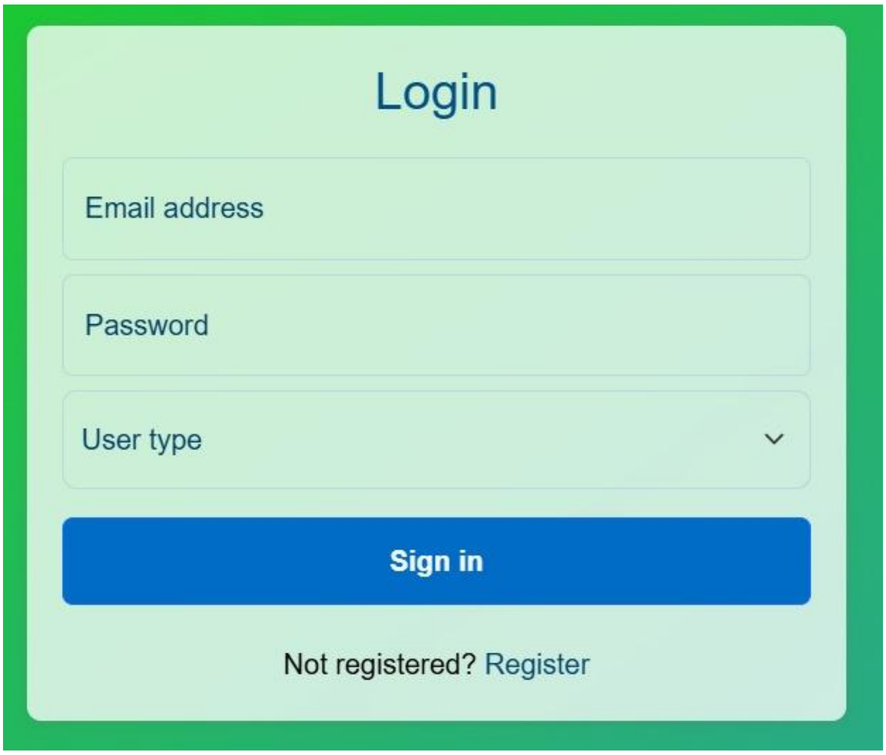
Key Components

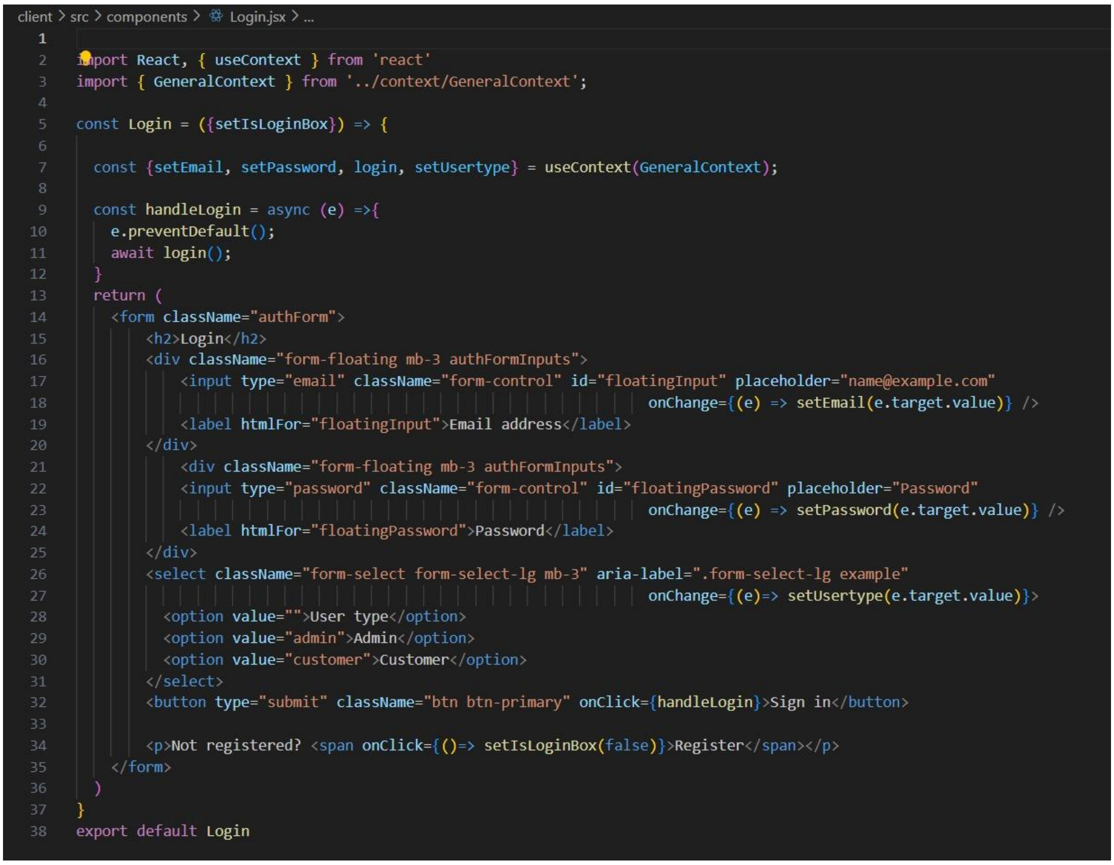
1. **App.js**
   - Responsible for routing and main application layout.



```
import './App.css';
import { Route, Routes } from 'react-router-dom';
import Landing from './Pages/Landing.jsx';
import Home from './Pages/Home';
import Admin from './Pages/Admin';
import Deposits from './Pages/Deposits';
import Loans from './Pages/Loans';
import AdminDeposits from './Pages/AdminDeposits';
import AdminLoans from './Pages/AdminLoans';
import AllTransactions from './Pages/AllTransactions';
import AllUsers from './Pages/AllUsers';
import LoginProtector from './RouteProtectors/AuthProtector';
import AuthProtector from './RouteProtectors/LoginProtector';
import UserTransactions from './Pages/UserTransactions';

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

      <Routes>
        <Route exact path='' element={<LoginProtector> <Landing /> </LoginProtector> } />
        <Route  path='/home' element={<AuthProtector><Home /></AuthProtector>} />
        <Route  path='/deposits' element={ <AuthProtector><Deposits /></AuthProtector>} />
        <Route  path='/loans' element={ <AuthProtector><Loans /></AuthProtector>} />
        <Route  path='/transactions' element={<AuthProtector><UserTransactions /></AuthProtector>} />

        <Route  path='/admin' element={ <AuthProtector><Admin /></AuthProtector>} />
        <Route  path='/all-users' element={ <AuthProtector><AllUsers /></AuthProtector>} />
        <Route  path='/all-deposits' element={ <AuthProtector><AdminDeposits /></AuthProtector>} />
        <Route  path='/all-loans' element={ <AuthProtector><AdminLoans /></AuthProtector>} />
        <Route  path='/all-transactions' element={<AuthProtector><AllTransactions /></AuthProtector>} />
      </Routes>
    </div>
  );
}
```

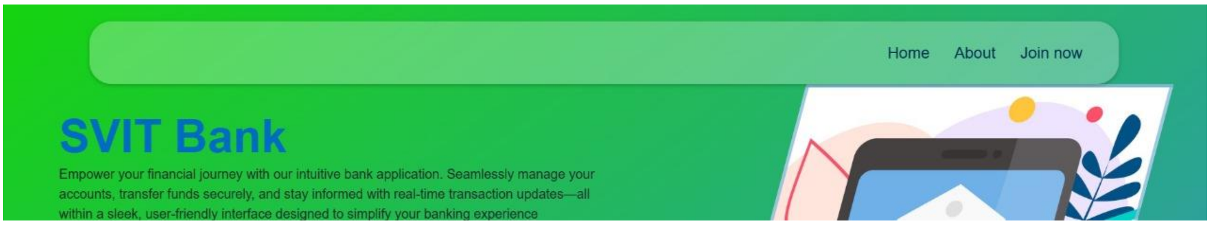
2. **/components**
    - Contains reusable UI components used across the application.
    - Login.jsx




client > src > components > Login.jsx > ...

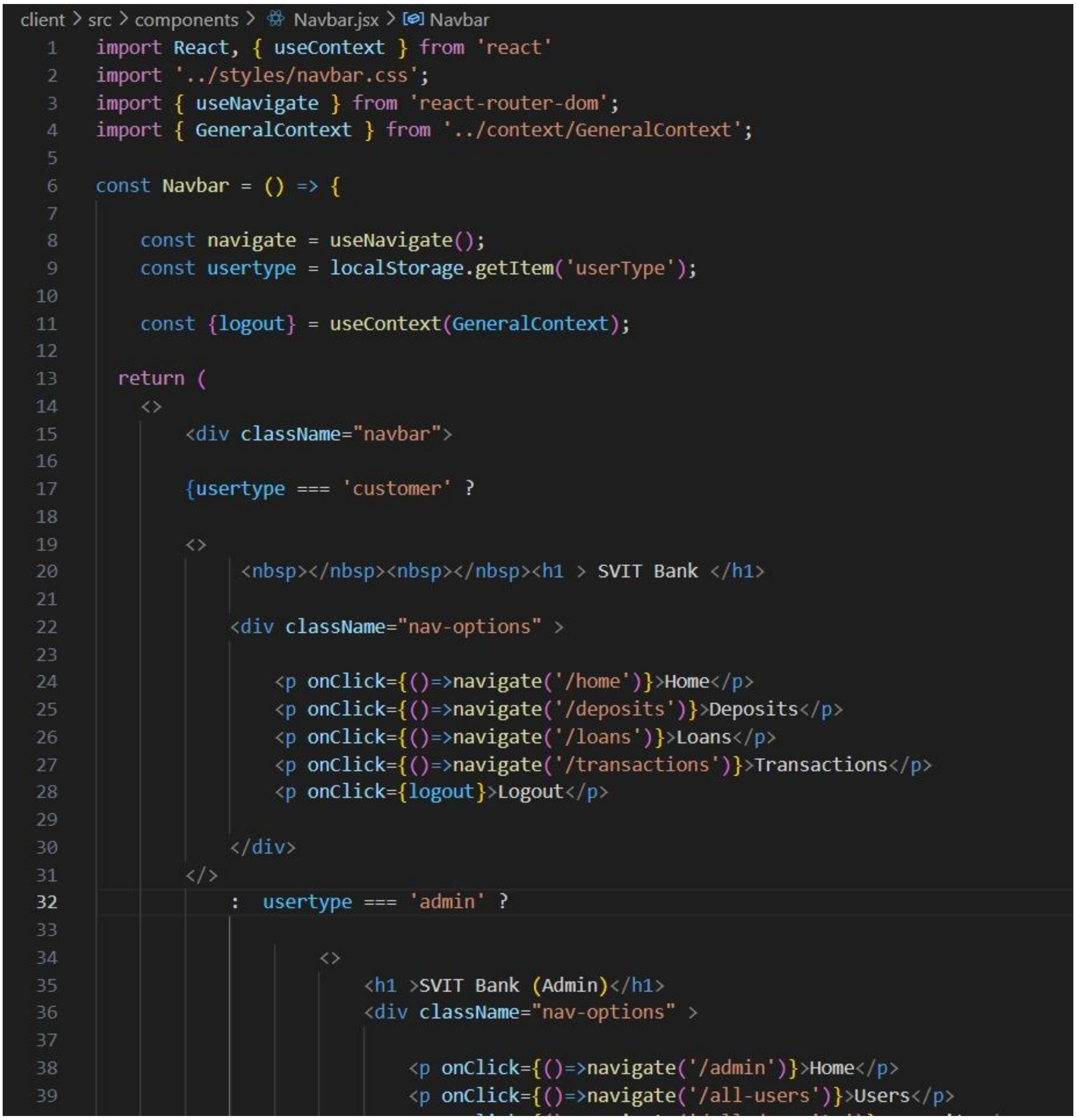
```jsx

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

const Login = ({setIsLoginBox}) => {

  const {setEmail, setPassword, login, setUsertype} = 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>
        </div>
            <div className="form-floating mb-3 authFormInputs">
            <input type="password" className="form-control" id="floatingPassword" placeholder="Password"
                                                                  onChange={(e) => setPassword(e.target.value)} />
            <label htmlFor="floatingPassword">Password</label>
        </div>
        <select className="form-select form-select-lg mb-3" aria-label=".form-select-lg example"
                                                                  onChange={(e)=> setUsertype(e.target.value)}>
          <option value="">User type</option>
          <option value="admin">Admin</option>
          <option value="customer">Customer</option>
        </select>
        <button type="submit" className="btn btn-primary" onClick={handleLogin}>Sign in</button>

        <p>Not registered? <span onClick={()=> setIsLoginBox(false)}>Register</span></p>
    </form>
  )
}
export default Login
```

- Navbar.jsx



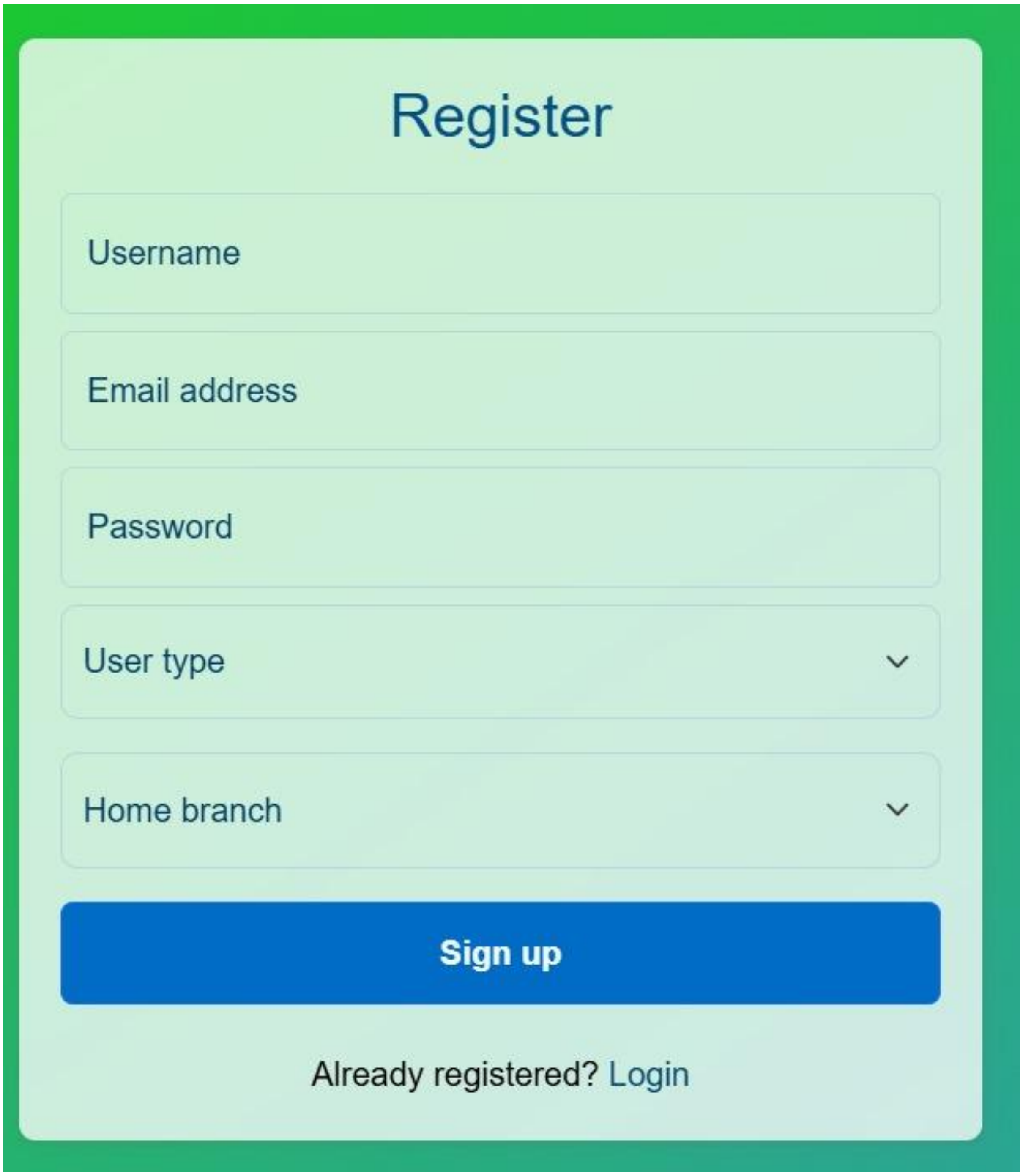
SVIT Bank

Empower your financial journey with our intuitive bank application. Seamlessly manage your accounts, transfer funds securely, and stay informed with real-time transaction updates—all within a sleek, user-friendly interface designed to simplify your banking experience



```
import React, { useContext } from 'react'
import '../styles/navbar.css';
import { useNavigate } from 'react-router-dom';
import { GeneralContext } from '../context/GeneralContext';

const Navbar = () => {

    const navigate = useNavigate();
    const usertype = localStorage.getItem('userType');

    const {logout} = useContext(GeneralContext);

  return (
    <>
        <div className="navbar">

        {usertype === 'customer' ?

        <>
            <nbsp></nbsp><nbsp></nbsp><h1 > SVIT Bank </h1>

            <div className="nav-options" >

                <p onClick={()=>navigate('/home')}>Home</p>
                <p onClick={()=>navigate('/deposits')}>Deposits</p>
                <p onClick={()=>navigate('/loans')}>Loans</p>
                <p onClick={()=>navigate('/transactions')}>Transactions</p>
                <p onClick={logout}>Logout</p>

            </div>
        </>
            :  usertype === 'admin' ?

                    <>
                        <h1 >SVIT Bank (Admin)</h1>
                        <div className="nav-options" >

                            <p onClick={()=>navigate('/admin')}>Home</p>
                            <p onClick={()=>navigate('/all-users')}>Users</p>
```

- Register.jsx



Register

Username

Email address

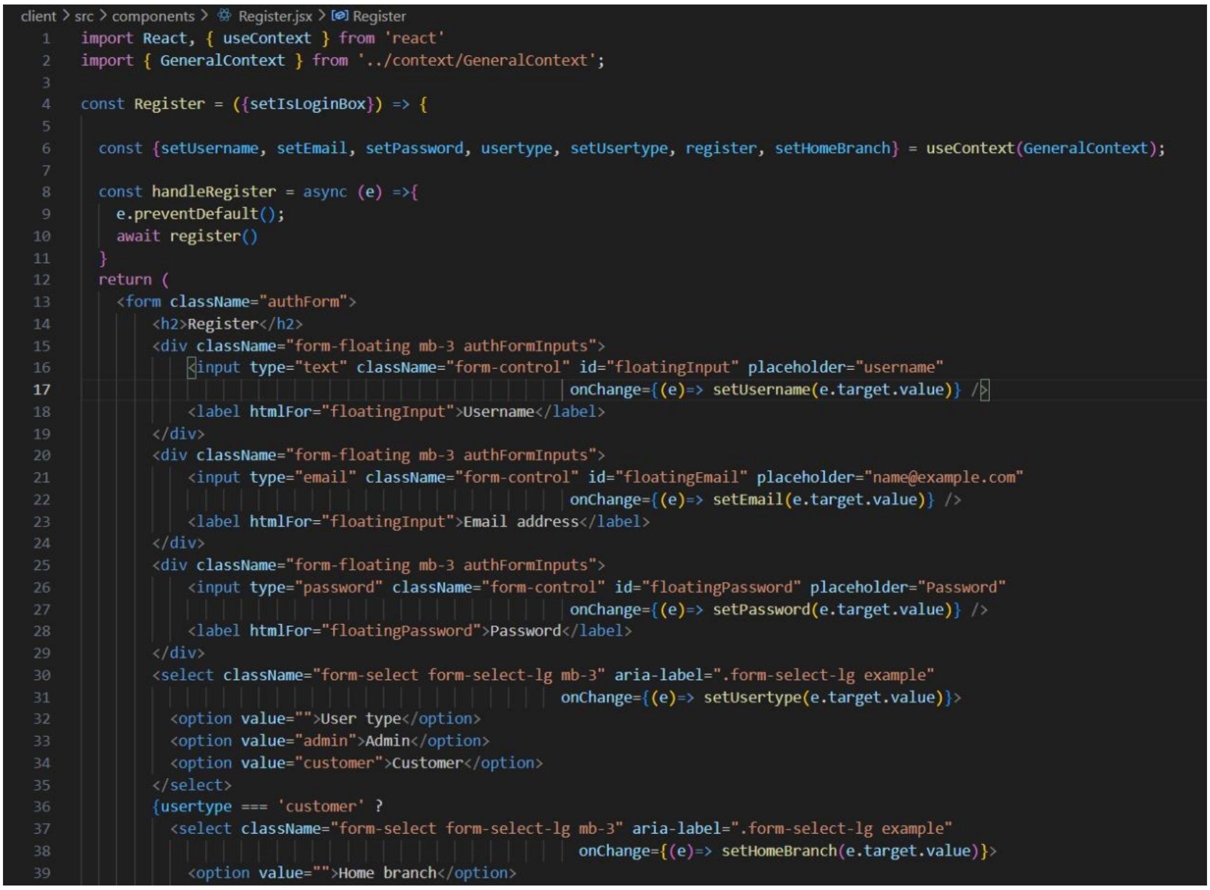
Password

User type

Home branch

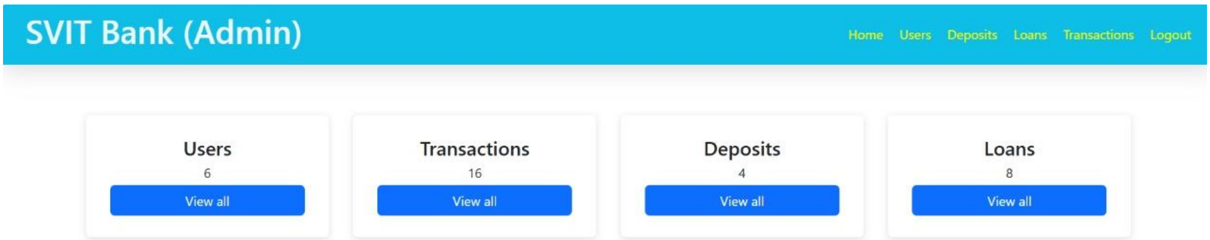
Sign up

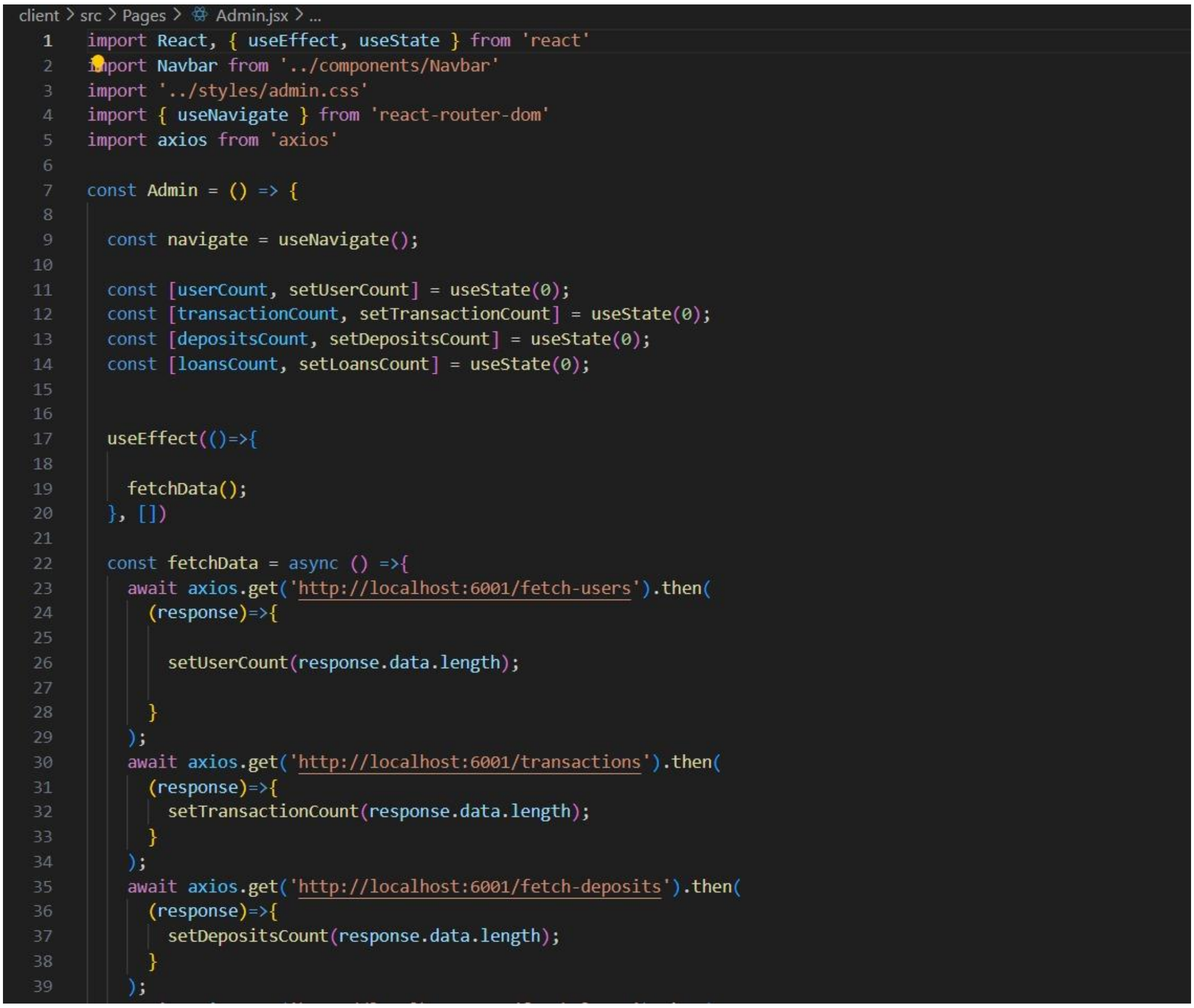
Already registered? Login



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

const Register = ({setIsLoginBox}) => {

  const {setUsername, setEmail, setPassword, usertype, setUsertype, register, setHomeBranch} = useContext(GeneralContext);

  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>
        </div>
        <div className="form-floating mb-3 authFormInputs">
            <input type="password" className="form-control" id="floatingPassword" placeholder="Password"
                                                       onChange={(e)=> setPassword(e.target.value)} />
            <label htmlFor="floatingPassword">Password</label>
        </div>
        <select className="form-select form-select-lg mb-3" aria-label=".form-select-lg example"
                                                      onChange={(e)=> setUsertype(e.target.value)}>
          <option value="">User type</option>
          <option value="admin">Admin</option>
          <option value="customer">Customer</option>
        </select>
        {usertype === 'customer' ?
          <select className="form-select form-select-lg mb-3" aria-label=".form-select-lg example"
                                                        onChange={(e)=> setHomeBranch(e.target.value)}>
            <option value="">Home branch</option>
```

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

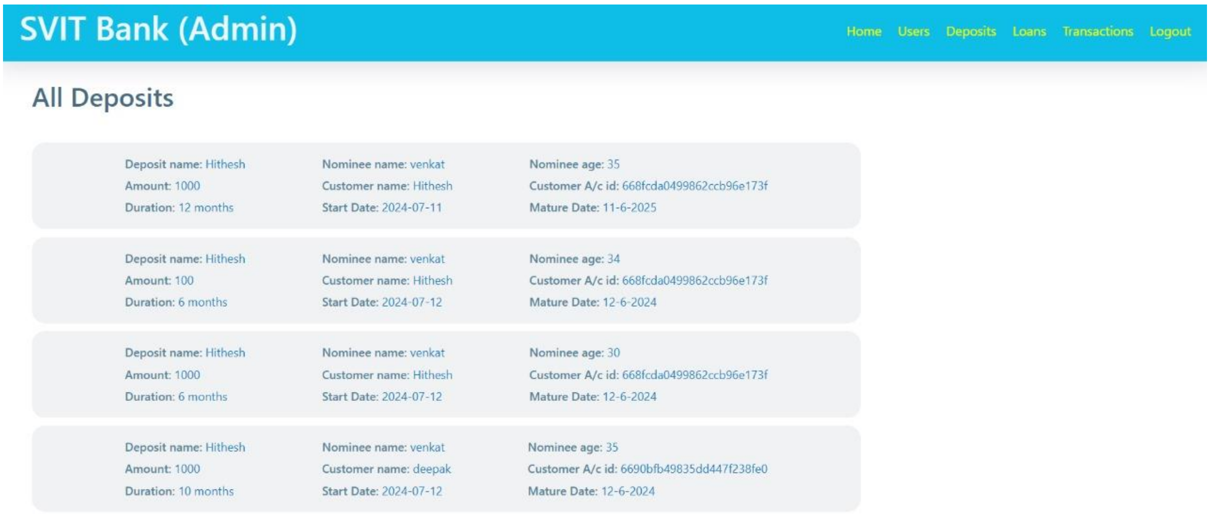




client > src > Pages > Admin.jsx > ...

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

const Admin = () => {

  const navigate = useNavigate();

  const [userCount, setUserCount] = useState(0);
  const [transactionCount, setTransactionCount] = useState(0);
  const [depositsCount, setDepositsCount] = useState(0);
  const [loansCount, setLoansCount] = useState(0);


  useEffect(()=>{

    fetchData();
  }, [])

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

        setUserCount(response.data.length);

      }
    );
    await axios.get('http://localhost:6001/transactions').then(
      (response)=>{
        setTransactionCount(response.data.length);
      }
    );
    await axios.get('http://localhost:6001/fetch-deposits').then(
      (response)=>{
        setDepositsCount(response.data.length);
      }
    );
```

- AdminDeposits.jsx



SVIT Bank (Admin)

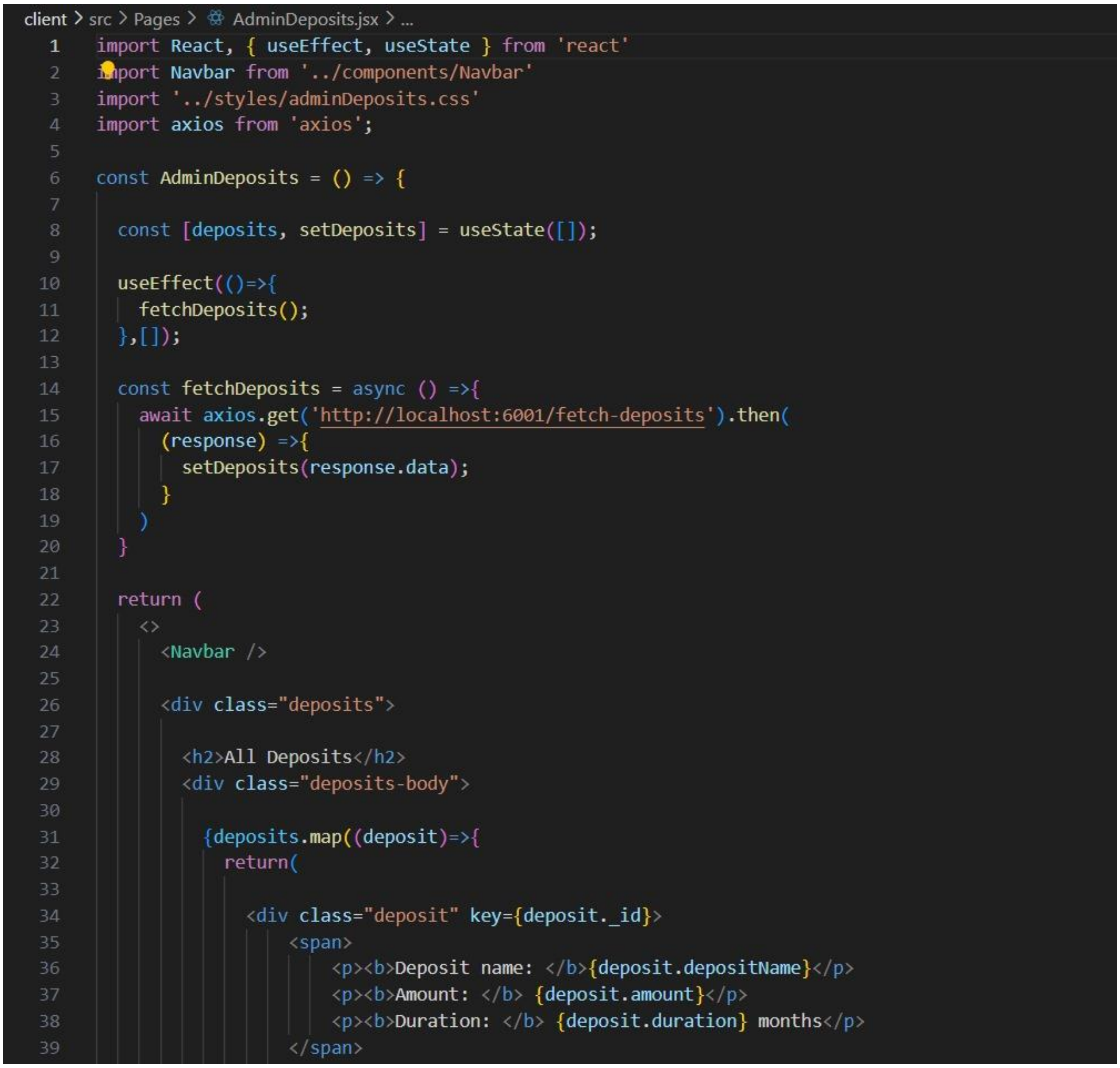
All Deposits

Deposit name: Hithesh
Amount: 1000
Duration: 12 months
Nominee name: venkat
Customer name: Hithesh
Start Date: 2024-07-11
Nominee age: 35
Customer A/c id: 668fcda0499862ccb96e173f
Mature Date: 11-6-2025

Deposit name: Hithesh
Amount: 100
Duration: 6 months
Nominee name: venkat
Customer name: Hithesh
Start Date: 2024-07-12
Nominee age: 34
Customer A/c id: 668fcda0499862ccb96e173f
Mature Date: 12-6-2024

Deposit name: Hithesh
Amount: 1000
Duration: 6 months
Nominee name: venkat
Customer name: Hithesh
Start Date: 2024-07-12
Nominee age: 30
Customer A/c id: 668fcda0499862ccb96e173f
Mature Date: 12-6-2024

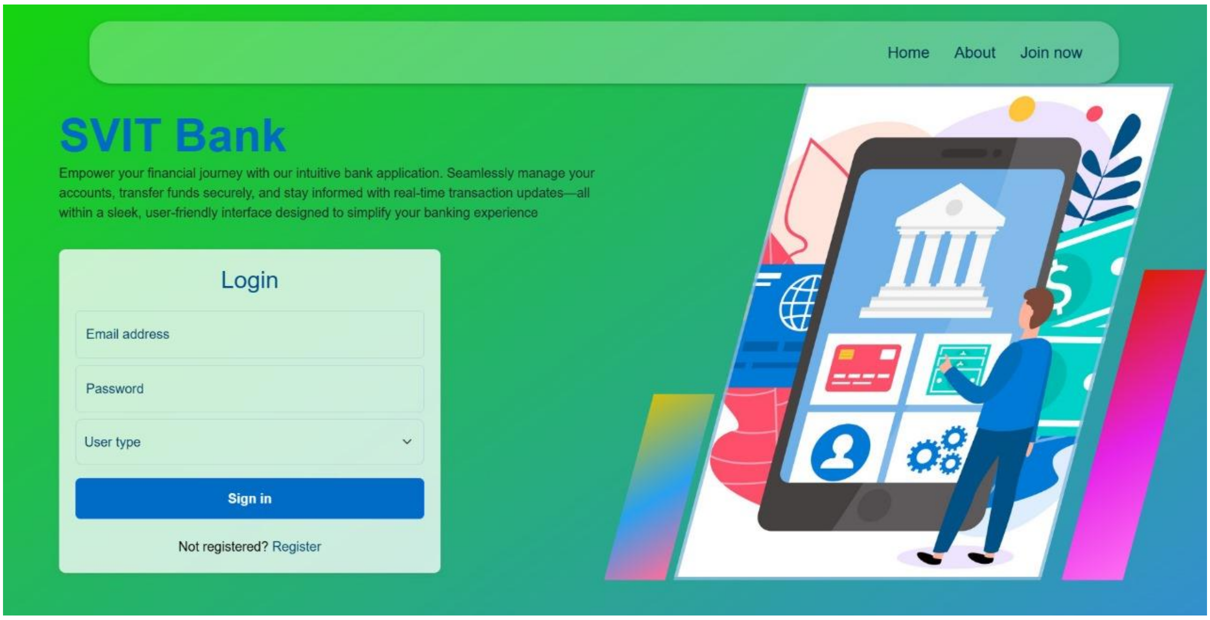
Deposit name: Hithesh
Amount: 1000
Duration: 10 months
Nominee name: venkat
Customer name: deepak
Start Date: 2024-07-12
Nominee age: 35
Customer A/c id: 6690bfb49835dd447f238fe0
Mature Date: 12-6-2024



client > src > Pages > AdminDeposits.jsx > ...

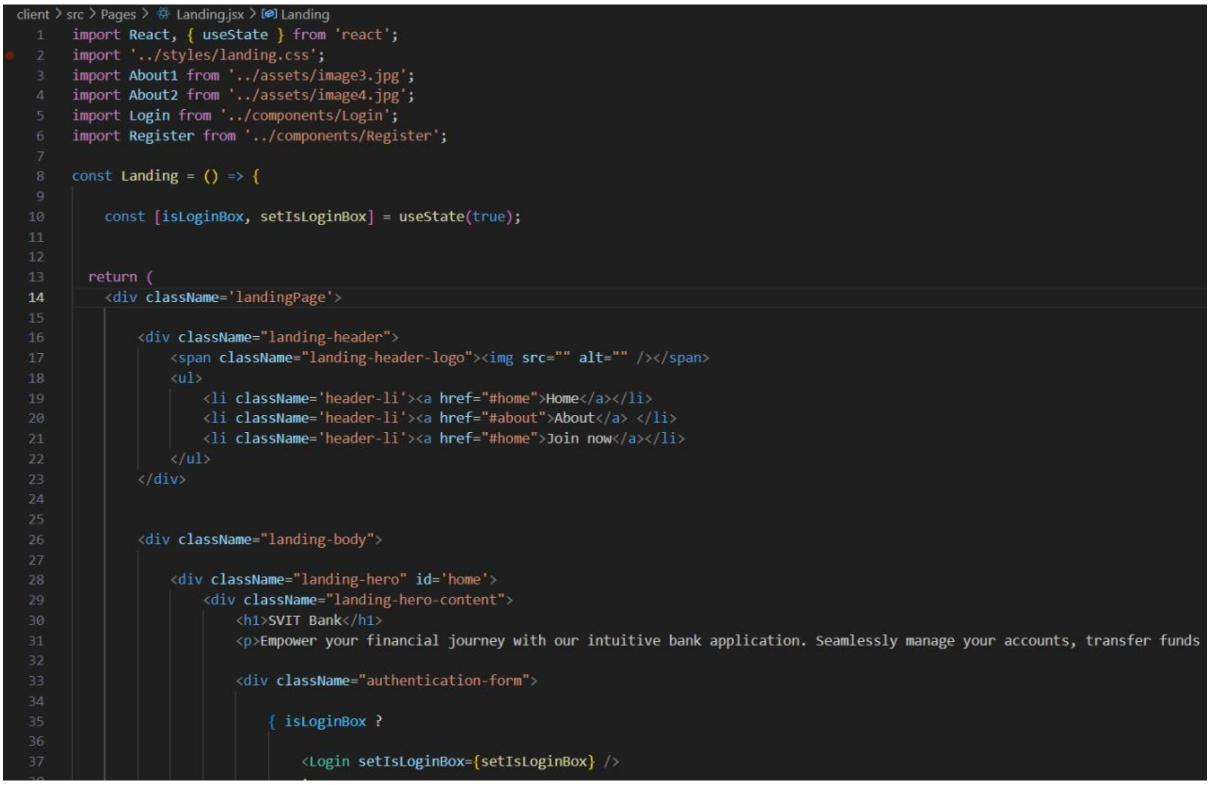
```jsx
import React, { useEffect, useState } from 'react'
import Navbar from '../components/Navbar'
import '../styles/adminDeposits.css'
import axios from 'axios';

const AdminDeposits = () => {

  const [deposits, setDeposits] = useState([]);

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

  const fetchDeposits = async () =>{
    await axios.get('http://localhost:6001/fetch-deposits').then(
      (response) =>{
        setDeposits(response.data);
      }
    )
  }

  return (
    <>
      <Navbar />

      <div class="deposits">

        <h2>All Deposits</h2>
        <div class="deposits-body">

          {deposits.map((deposit)=>{
            return(

              <div class="deposit" key={deposit._id}>
                  <span>
                      <p><b>Deposit name: </b>{deposit.depositName}</p>
                      <p><b>Amount: </b> {deposit.amount}</p>
                      <p><b>Duration: </b> {deposit.duration} months</p>
                  </span>
```

- Landing.jsx



SVIT Bank

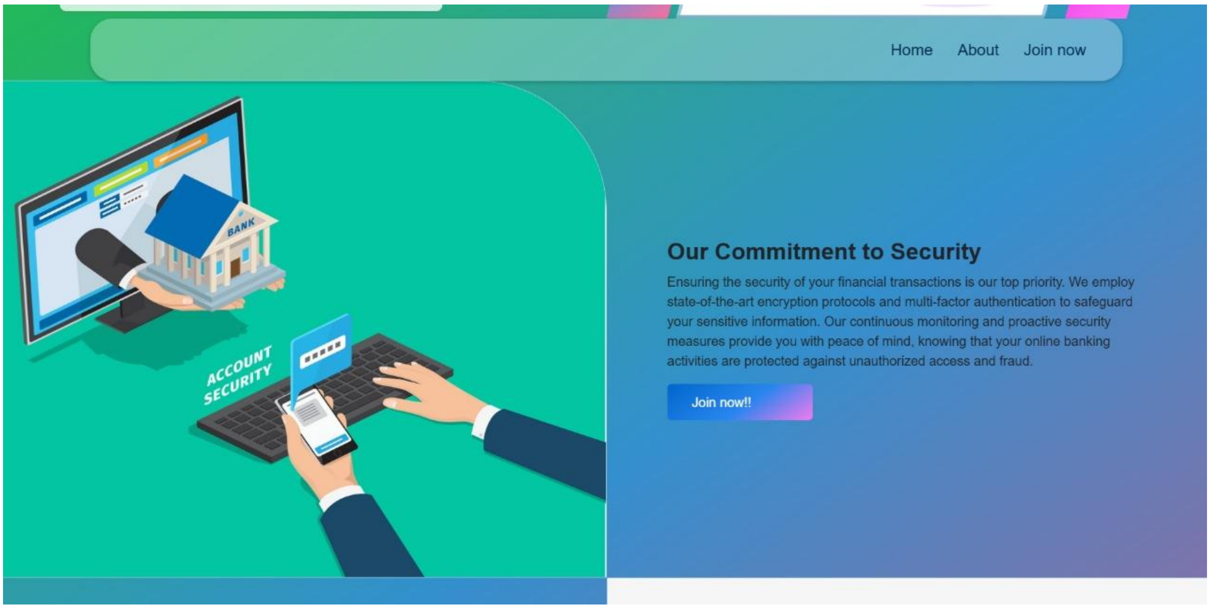
Empower your financial journey with our intuitive bank application. Seamlessly manage your accounts, transfer funds securely, and stay informed with real-time transaction updates—all within a sleek, user-friendly interface designed to simplify your banking experience



client > src > Pages > Landing.jsx > Landing

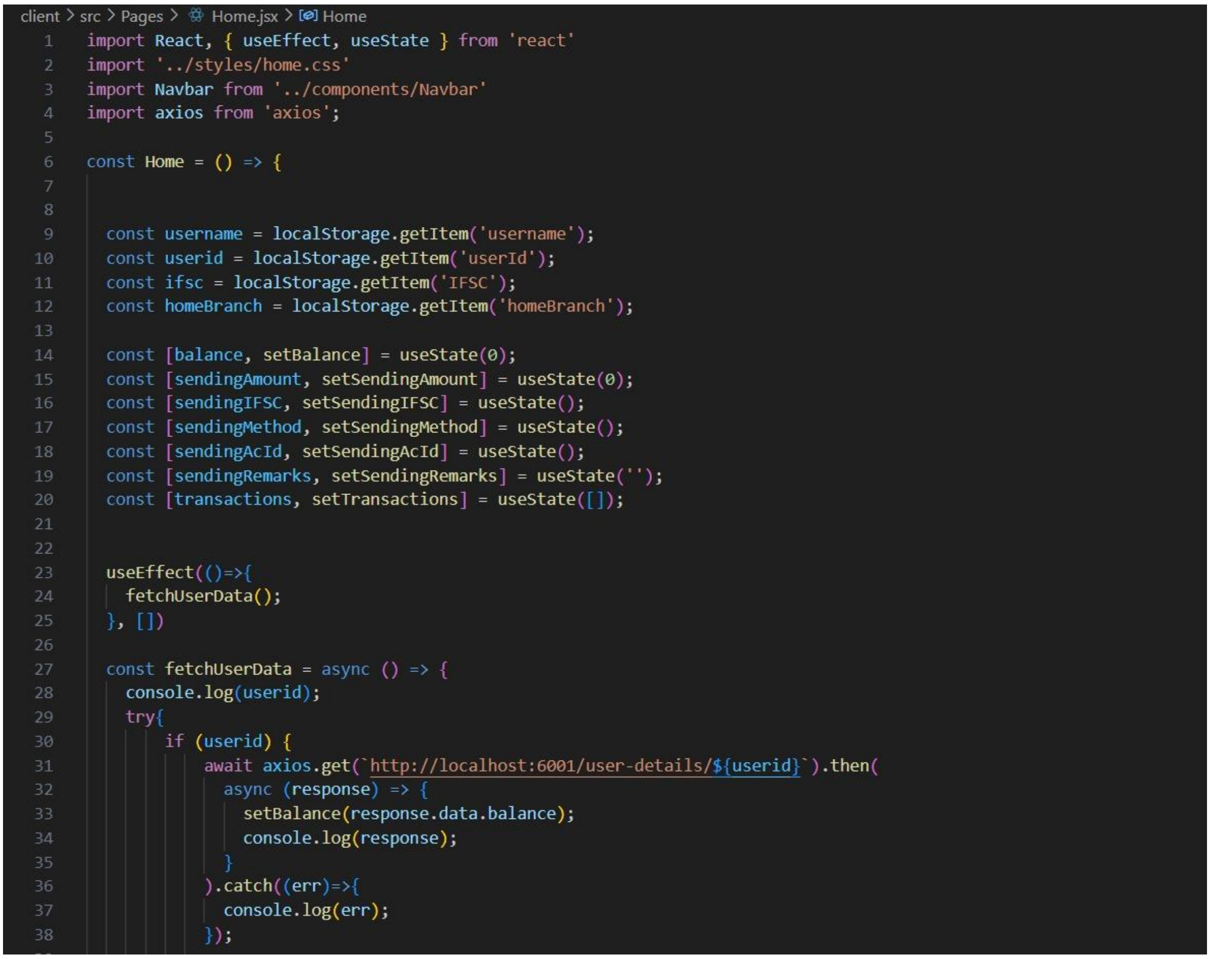
```
import React, { useState } from 'react';
import '../styles/landing.css';
import About1 from '../assets/image3.jpg';
import About2 from '../assets/image4.jpg';
import Login from '../components/Login';
import Register from '../components/Register';

const Landing = () => {

    const [isLoginBox, setIsLoginBox] = useState(true);


  return (
    <div className='landingPage'>

        <div className="landing-header">
            <span className="landing-header-logo"><img src="" alt="" /></span>
            <ul>
                <li className='header-li'><a href="#home">Home</a></li>
                <li className='header-li'><a href="#about">About</a> </li>
                <li className='header-li'><a href="#home">Join now</a></li>
            </ul>
        </div>


        <div className="landing-body">

            <div className="landing-hero" id='home'>
                <div className="landing-hero-content">
                    <h1>SVIT Bank</h1>
                    <p>Empower your financial journey with our intuitive bank application. Seamlessly manage your accounts, transfer funds

                    <div className="authentication-form">

                        { isLoginBox ?

                            <Login setIsLoginBox={setIsLoginBox} />
```

- Home.jsx



Our Commitment to Security

Ensuring the security of your financial transactions is our top priority. We employ state-of-the-art encryption protocols and multi-factor authentication to safeguard your sensitive information. Our continuous monitoring and proactive security measures provide you with peace of mind, knowing that your online banking activities are protected against unauthorized access and fraud.

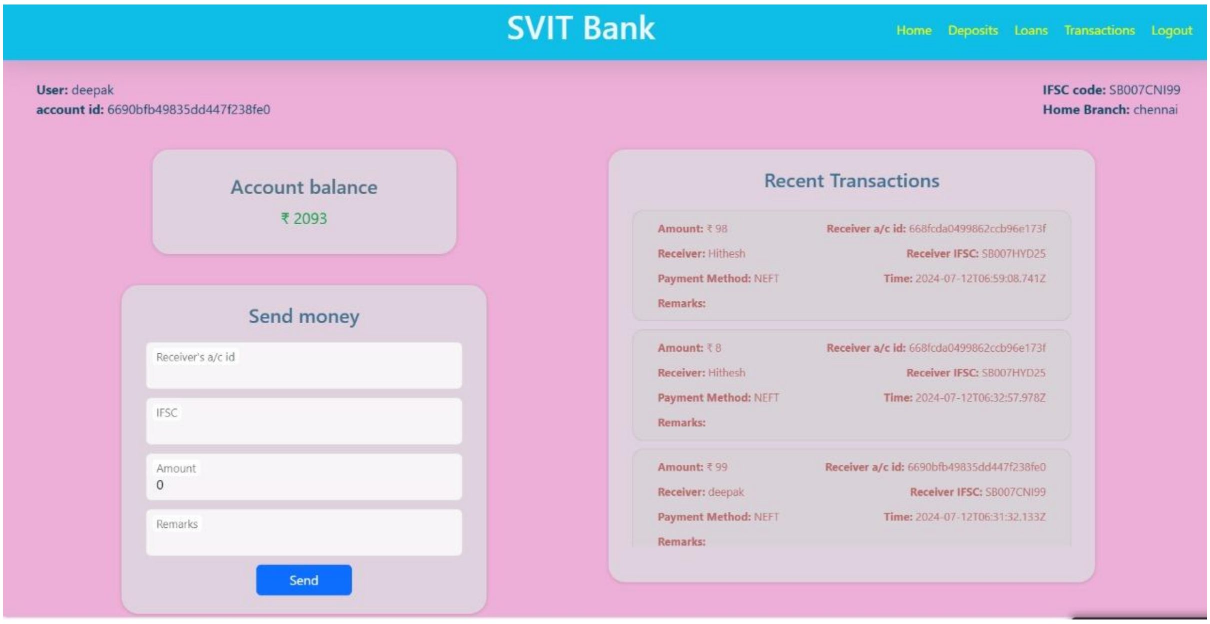


client > src > Pages > Home.jsx > Home

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

const Home = () => {


  const username = localStorage.getItem('username');
  const userid = localStorage.getItem('userId');
  const ifsc = localStorage.getItem('IFSC');
  const homeBranch = localStorage.getItem('homeBranch');

  const [balance, setBalance] = useState(0);
  const [sendingAmount, setSendingAmount] = useState(0);
  const [sendingIFSC, setSendingIFSC] = useState();
  const [sendingMethod, setSendingMethod] = useState();
  const [sendingAcId, setSendingAcId] = useState();
  const [sendingRemarks, setSendingRemarks] = useState('');
  const [transactions, setTransactions] = useState([]);


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

  const fetchUserData = async () => {
    console.log(userid);
    try{
        if (userid) {
            await axios.get(`http://localhost:6001/user-details/${userid}`).then(
              async (response) => {
                setBalance(response.data.balance);
                console.log(response);
              }
            ).catch((err)=>{
              console.log(err);
            });
```

- UserTransactions.jsx

To send money:



User: deepak
account id: 6690bfb49835dd447f238fe0

IFSC code: SB007CNI99
Home Branch: chennai

Account balance
₹ 2093

Send money
Receiver's a/c id
IFSC
Amount
0
Remarks

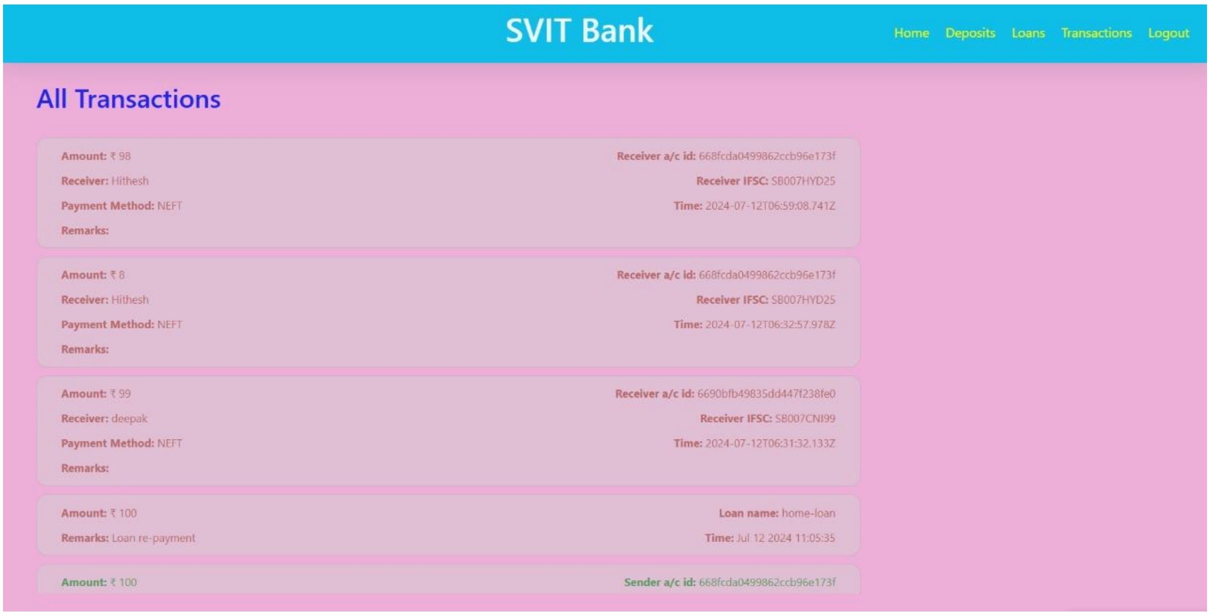
Recent Transactions

Amount: ₹ 98 **Receiver a/c id:** 668fcda0499862ccb96e173f
Receiver: Hithesh **Receiver IFSC:** SB007HYD25
Payment Method: NEFT **Time:** 2024-07-12T06:59:08.741Z
Remarks:

Amount: ₹ 8 **Receiver a/c id:** 668fcda0499862ccb96e173f
Receiver: Hithesh **Receiver IFSC:** SB007HYD25
Payment Method: NEFT **Time:** 2024-07-12T06:32:57.978Z
Remarks:

Amount: ₹ 99 **Receiver a/c id:** 6690bfb49835dd447f238fe0
Receiver: deepak **Receiver IFSC:** SB007CNI99
Payment Method: NEFT **Time:** 2024-07-12T06:31:32.133Z
Remarks:

To see all transactions:



All Transactions

Amount: ₹ 98 **Receiver a/c id:** 668fcda0499862ccb96e173f
Receiver: Hithesh **Receiver IFSC:** SB007HYD25
Payment Method: NEFT **Time:** 2024-07-12T06:59:08.741Z
Remarks:

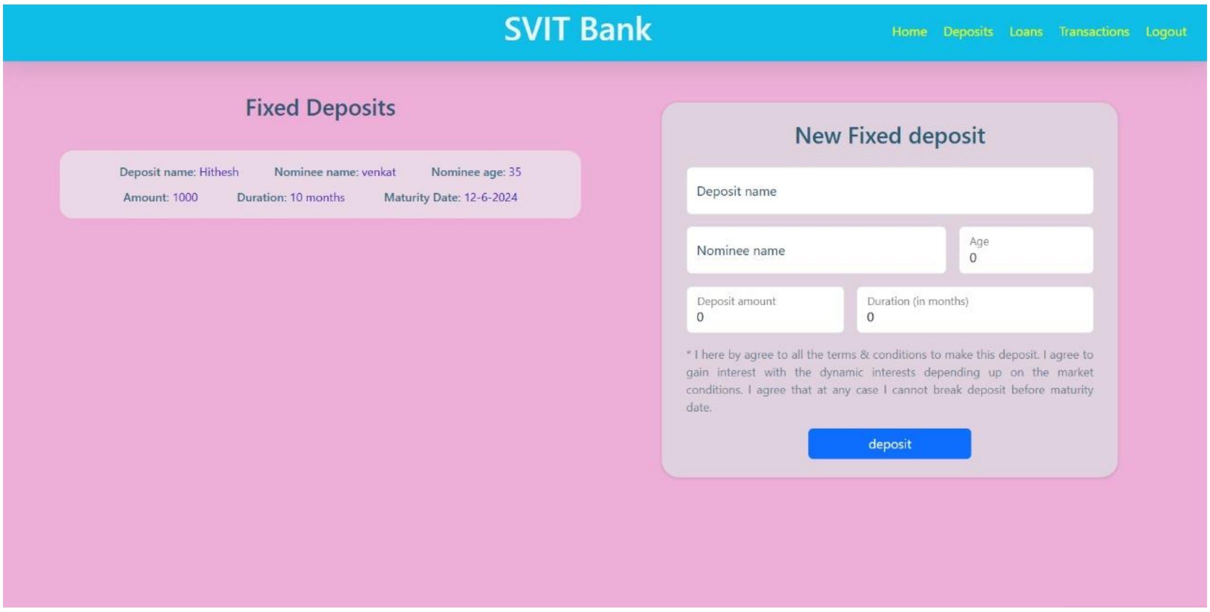
Amount: ₹ 8 **Receiver a/c id:** 668fcda0499862ccb96e173f
Receiver: Hithesh **Receiver IFSC:** SB007HYD25
Payment Method: NEFT **Time:** 2024-07-12T06:32:57.978Z
Remarks:

Amount: ₹ 99 **Receiver a/c id:** 6690bfb49835dd447f238fe0
Receiver: deepak **Receiver IFSC:** SB007CNI99
Payment Method: NEFT **Time:** 2024-07-12T06:31:32.133Z
Remarks:

Amount: ₹ 100 **Loan name:** home-loan
Remarks: Loan re-payment **Time:** Jul 12 2024 11:05:35

Amount: ₹ 100 **Sender a/c id:** 668fcda0499862ccb96e173f

To make an Fixed Deposit(FD):



Fixed Deposits

Deposit name: Hithesh   Nominee name: venkat   Nominee age: 35
Amount: 1000   Duration: 10 months   Maturity Date: 12-6-2024

New Fixed deposit

Deposit name

Nominee name

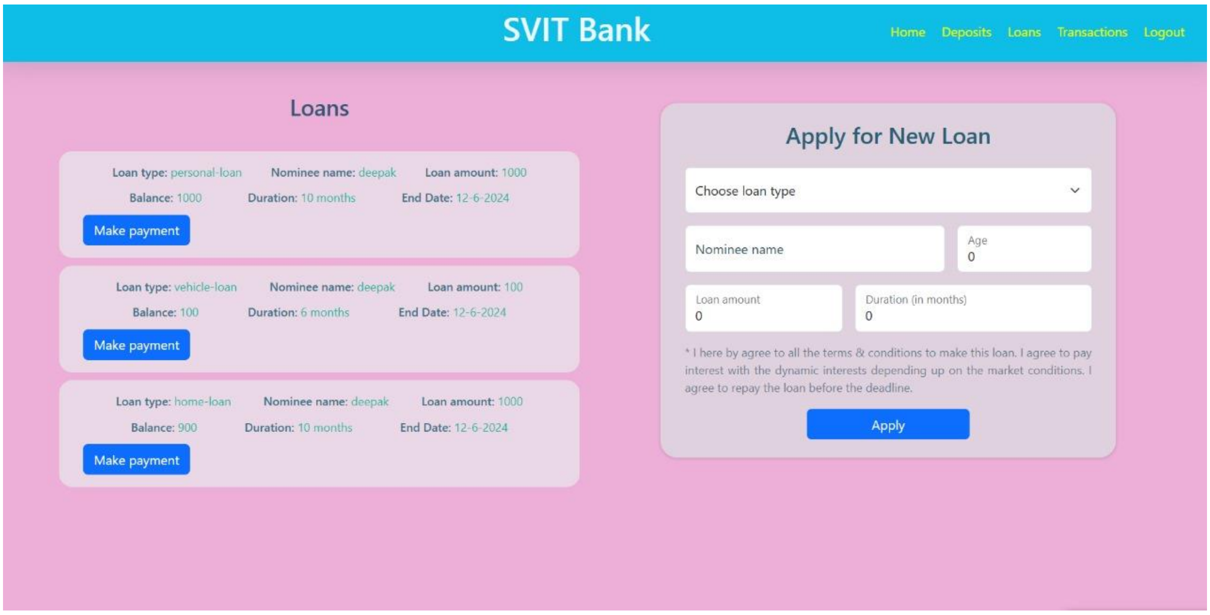
Age
0

Deposit amount
0

Duration (in months)
0

* I here by agree to all the terms & conditions to make this deposit. I agree to gain interest with the dynamic interests depending up on the market conditions. I agree that at any case I cannot break deposit before maturity date.

To apply for a LOAN:



Loans

Loan type: personal-loan   Nominee name: deepak   Loan amount: 1000
Balance: 1000   Duration: 10 months   End Date: 12-6-2024

Loan type: vehicle-loan   Nominee name: deepak   Loan amount: 100
Balance: 100   Duration: 6 months   End Date: 12-6-2024

Loan type: home-loan   Nominee name: deepak   Loan amount: 1000
Balance: 900   Duration: 10 months   End Date: 12-6-2024

Apply for New Loan

Choose loan type

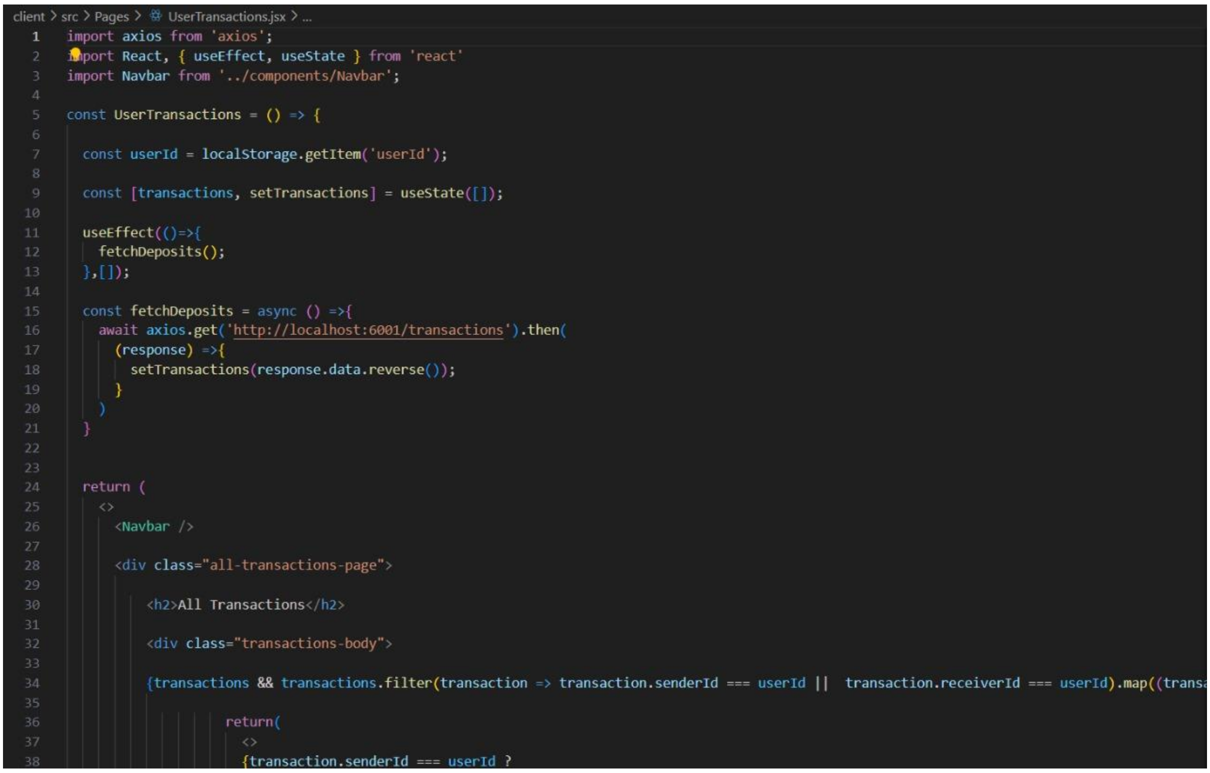
Nominee name

Age
0

Loan amount
0

Duration (in months)
0

* I here by agree to all the terms & conditions to make this loan. I agree to pay interest with the dynamic interests depending up on the market conditions. I agree to repay the loan before the deadline.



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

const UserTransactions = () => {

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

  const [transactions, setTransactions] = useState([]);

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

  const fetchDeposits = async () =>{
    await axios.get('http://localhost:6001/transactions').then(
      (response) =>{
        setTransactions(response.data.reverse());
      }
    )
  }


  return (
    <>
      <Navbar />

      <div class="all-transactions-page">

          <h2>All Transactions</h2>

          <div class="transactions-body">

          {transactions && transactions.filter(transaction => transaction.senderId === userId ||  transaction.receiverId === userId).map((transa

                    return(
                      <>
                      {transaction.senderId === userId ?
```

Routing

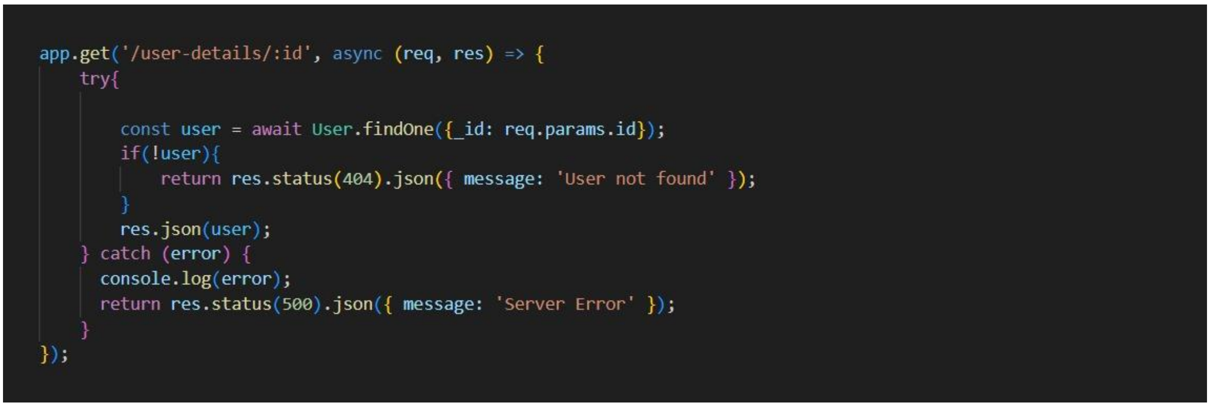
Routing is managed using React Router. Here are the main routes:

- **/home** - Landing page of the application.
- **/dashboard** - Dashboard displaying user data and statistics.
- **/profile** - User profile management.

Integration with Backend

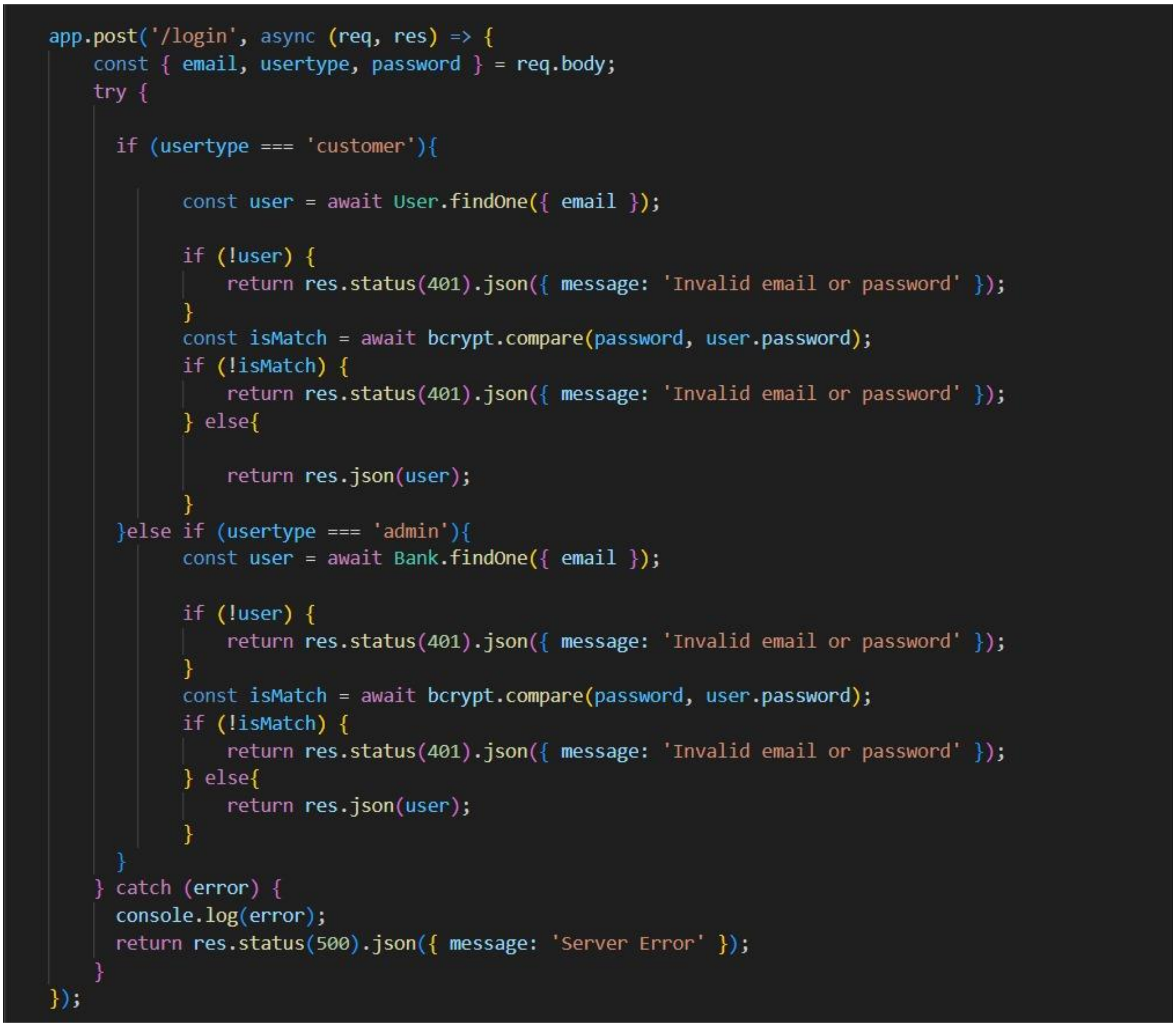
The frontend communicates with the backend APIs hosted on [backend URL]. Key endpoints include:

- **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 UI is designed to be simple and intuitive, minimizing the user's learning curve and ensuring ease of use. Elements are presented clearly to avoid clutter and confusion.
- **Consistency**: Consistent layout, color schemes, typography, and interaction patterns across the application to provide a cohesive user experience.
- **Responsiveness**: The UI is responsive and adapts seamlessly to various screen sizes and devices, ensuring accessibility and usability on desktops, tablets, and mobile devices.
- **User-Centric**: The design focuses on user needs, ensuring that key actions are easily accessible and user flows are logical. Feedback is provided promptly for user actions.
- **Accessibility**: Consideration for users with disabilities, incorporating features such as keyboard navigation, screen reader compatibility, and sufficient color contrast.

- Implemented using [UI framework/library].

  - **UI Framework/Library**: The UI is implemented using **React**, a popular JavaScript library for building user interfaces. React enables the development of reusable components and facilitates efficient rendering of the UI.
  - **CSS Framework**: For styling, **Bootstrap** is used, providing a comprehensive set of pre-defined classes and components for building responsive and modern UIs. Custom CSS is also incorporated for specific styling needs that go beyond the framework's capabilities.
  - **Routing**: **React Router** is used for handling navigation and routing within the application, enabling smooth transitions between different views and maintaining the single-page application (SPA) experience.
  - **Form Handling**: For handling user input in forms, **Formik** along with **Yup** for validation, is used, ensuring that form submissions are managed efficiently and validations are applied consistently.
  - **HTTP Requests**: **Axios** is used for making HTTP requests to the backend API, providing a simple and clean syntax for interacting with the server and handling responses and errors.