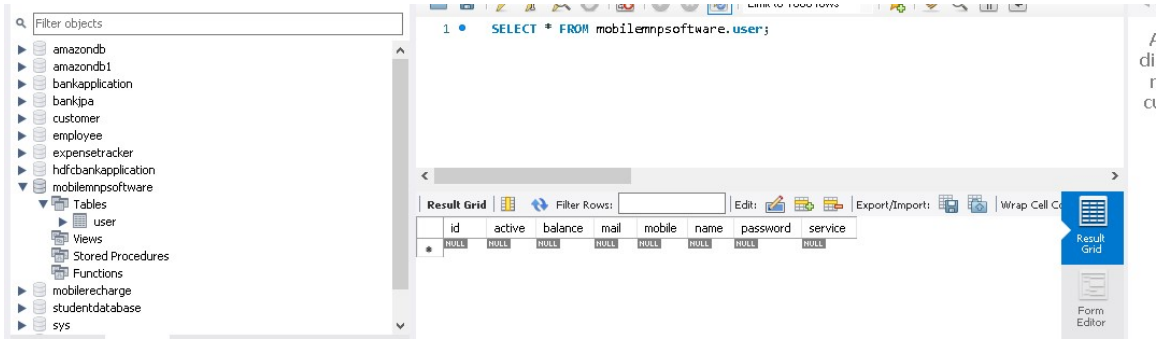
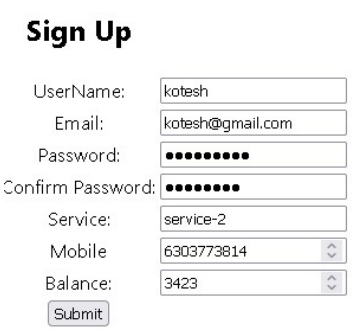
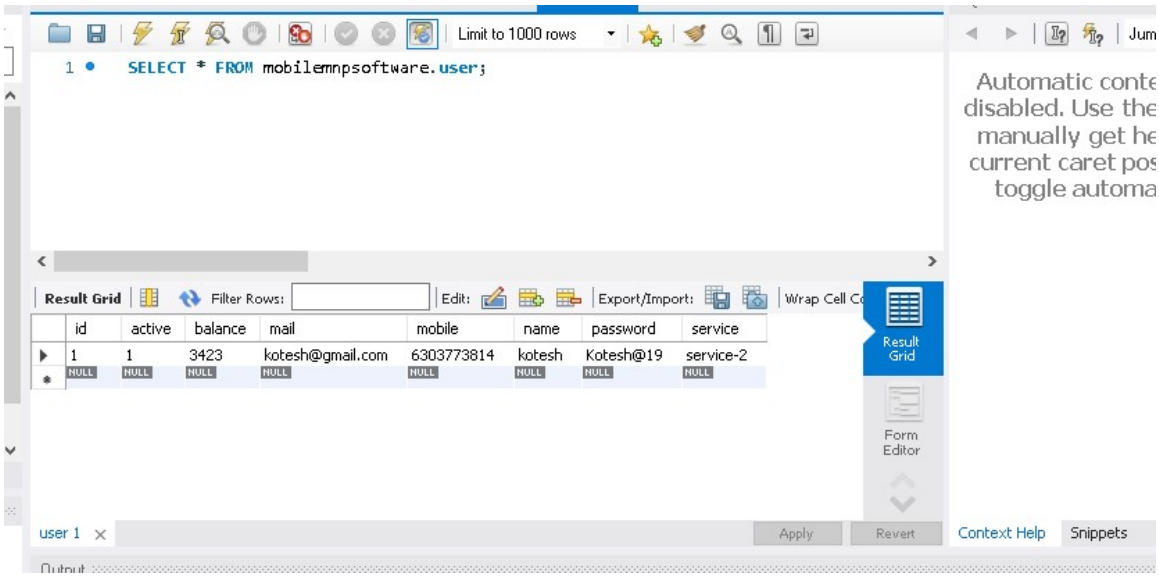
Mobile-mnp software

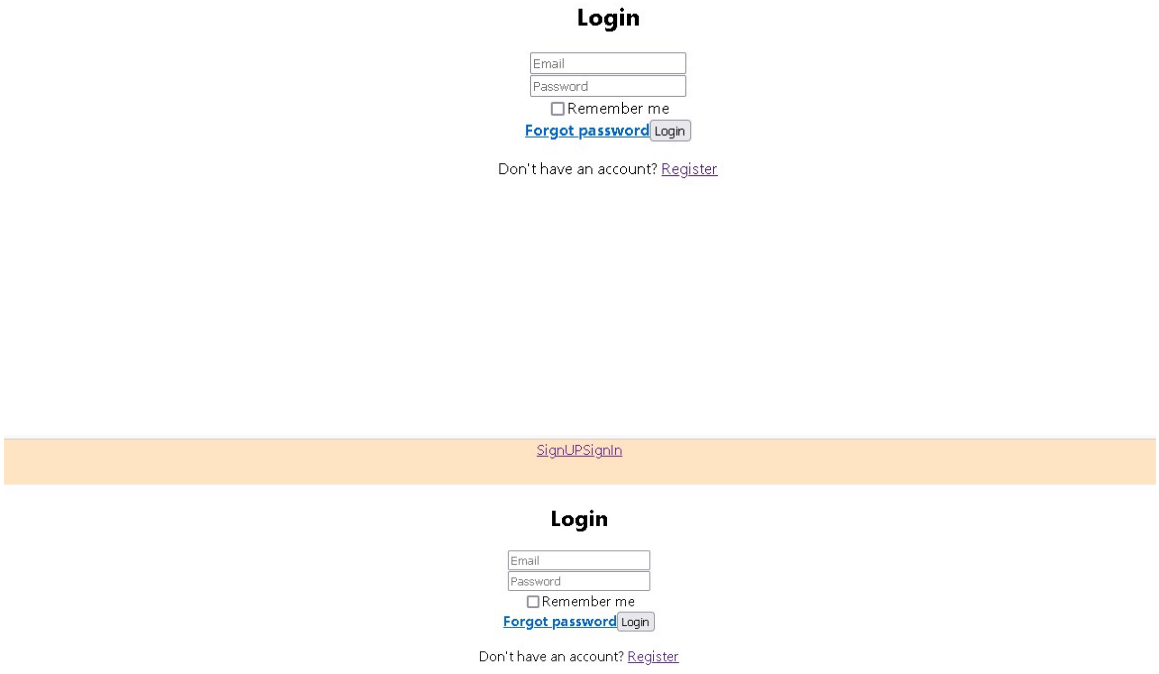
User-Table

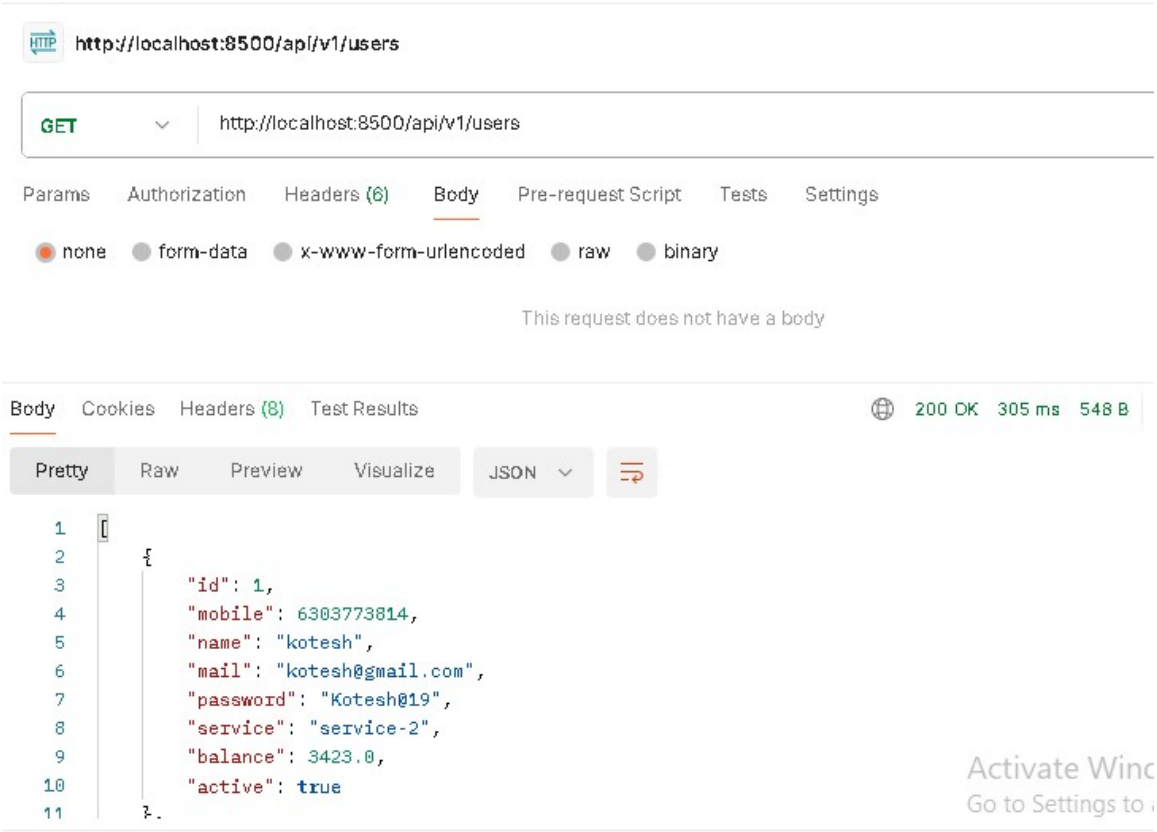


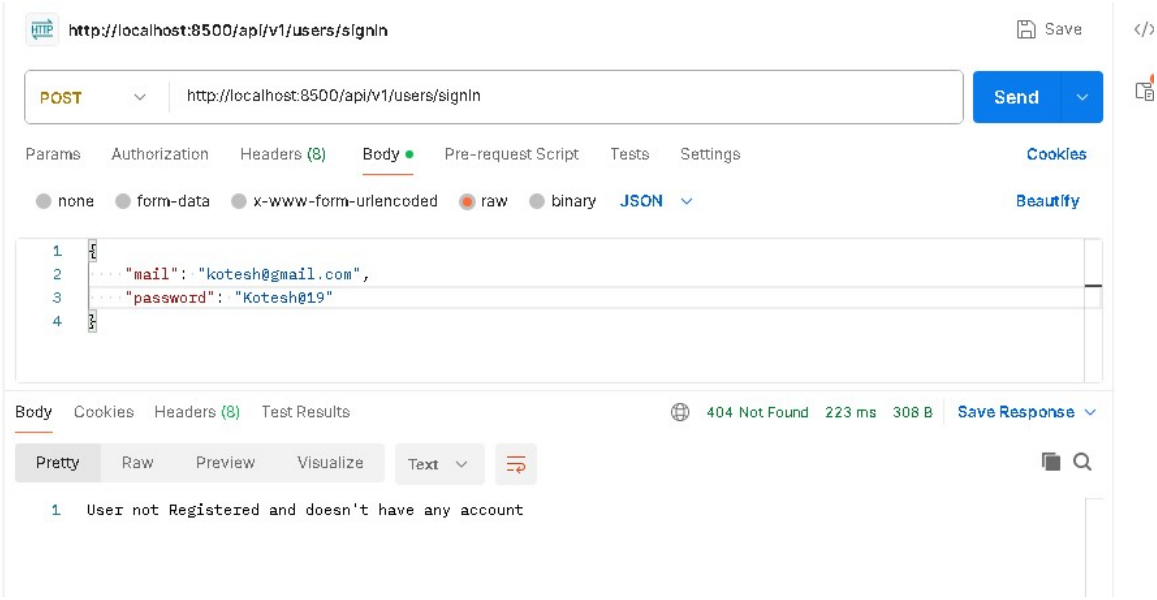
By registering itself we are making user as active



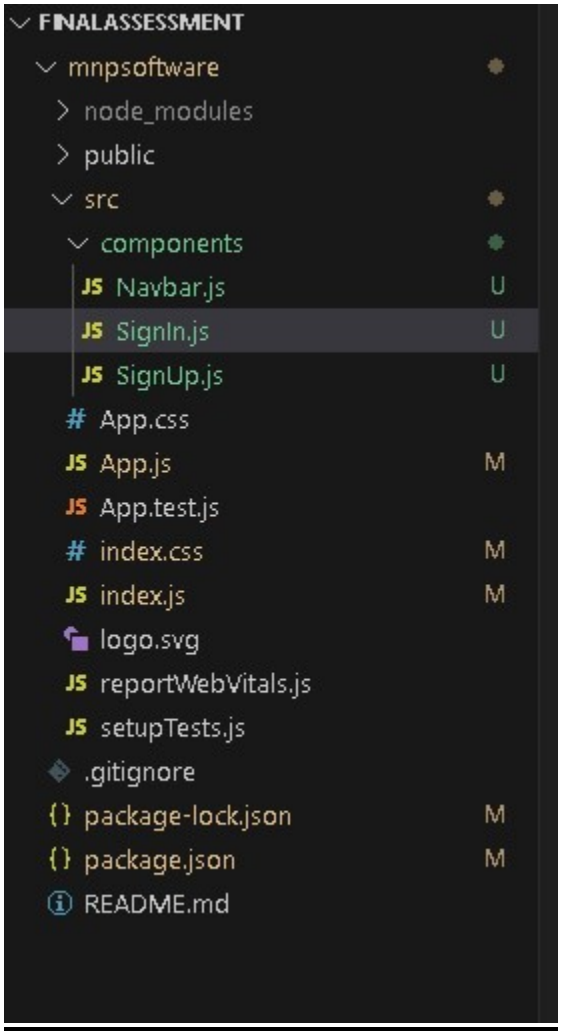

Limit to 1000 rows
1 SELECT * FROM mobilemnpsoftware.user;
Result Grid
Filter Rows:
Edit:
Export/Import:
id active balance mail mobile name password service
1 1 3423 kotesh@gmail.com 6303773814 kotesh Kotesh@19 service-2
NULL NULL NULL NULL NULL NULL NULL NULL
Result Grid
Form Editor
user 1
Apply
Revert
Context Help
Snippets
Output


Login
Email
Password
Remember me
Forgot password
Login
Don't have an account? Register
SignUPSignIn
Login
Email
Password
Remember me
Forgot password
Login
Don't have an account? Register


http://localhost:8500/apl/v1/users
GET
http://localhost:8500/api/v1/users
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
This request does not have a body
Body
Cookies
Headers (8)
Test Results
200 OK
305 ms
548 B
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"mobile": 6303773814,
"name": "kotesh",
"mail": "kotesh@gmail.com",
"password": "Kotesh@19",
"service": "service-2",
"balance": 3423.0,
"active": true
},
Activate Winc
Go to Settings to


http://localhost:8500/apl/v1/users/signIn
Save
POST
http://localhost:8500/api/v1/users/signIn
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
JSON
Beautify
{
"mail": "kotesh@gmail.com",
"password": "Kotesh@19"
}
Body
Cookies
Headers (8)
Test Results
404 Not Found
223 ms
308 B
Save Response
Pretty
Raw
Preview
Visualize
Text
User not Registered and doesn't have any account

Front-end



SignUp.js

```
import axios from 'axios';

import React, { useState } from 'react';


  const SignUp = () => {
```

```
const [name, setName] = useState('');
const [mail, setEmail] = useState('');
const [password, setPassword] = useState('');
const [confirmPassword, setConfirmPassword] = useState('');
const [service, setService] = useState('');
const [balance, setBalance] = useState('');
const [mobile,setMobileNo]=useState('');
const [active, setActive] = useState();




const handleSubmit = async (e) => {
   e.preventDefault();
   setActive(true);
   var user={name,mail,password,service,balance,mobile,active};

   const postUrl='http://localhost:8500/api/v1/users/signUp'
   const response= await axios.post(postUrl,user);
   console.log(response.data);



};
```

```
  return (
    <div className='form'>
      <form onSubmit={handleSubmit}>
          <table className='form-1'>
            <thead>
              <tr>
                <th>
                  <h2>Sign Up</h2>
                </th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td>
                  <label>UserName:</label>
                </td>
                <td>
                  <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} required />
                </td>
              </tr>
              <tr>
                <td>
                  <label>Email:</label>
                </td>
```

```
                    <td>
                        <input type="email" value={mail} onChange={(e) =>
setEmail(e.target.value)} required />
                    </td>
                </tr>

                <tr>
                    <td>
                        <label>Password:</label>
                    </td>
                    <td>
                        <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
                    </td>
                </tr>
                <tr>
                    <td>
                        <label>Confirm Password:</label>
                    </td>
                    <td>
                        <input type="password" value={confirmPassword} onChange={(e)
=> setConfirmPassword(e.target.value)} required />
                    </td>
                </tr>
                <tr>
                    <td>
```

```
                    <label>Service:</label>
                  </td>
                  <td>
                    <input type="text" value={service} onChange={(e) =>
setService(e.target.value)} required />
                  </td>
                </tr>

                <tr>
                  <td>
                    <label>Mobile</label>
                  </td>
                  <td>
                    <input type="number" maxLength={10} value={mobile}
onChange={(e) => setMobileNo(e.target.value)} required />
                  </td>
                </tr>
                <tr>
                  <td>
                    <label>Balance:</label>
                  </td>
                  <td>
                    <input type="number" maxLength={1} value={balance}
onChange={(e) => setBalance(e.target.value)} required />
                  </td>
```

```
                    </tr>


                </tbody>
                <input class="btn btn-primary" type="submit" value="Submit"></input>

            </table>

        </form>

    </div>
  );
};

export default SignUp;
```

signIn.js

```
import React from 'react';

import { Link } from 'react-router-dom';
import { useState } from 'react';
```

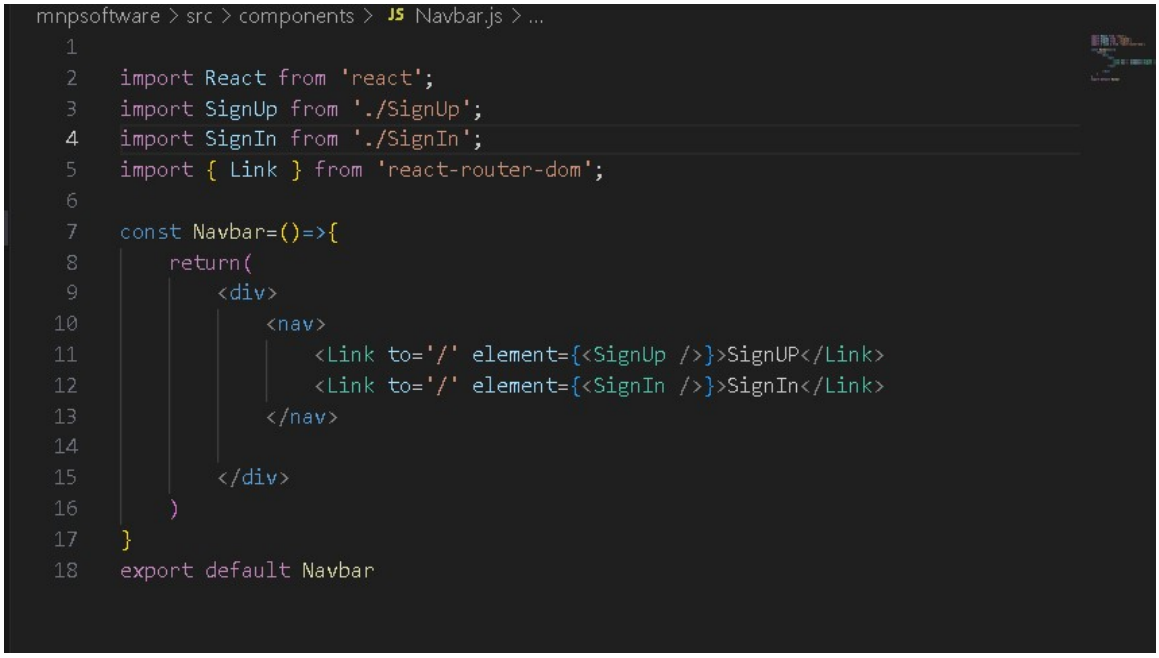
```
import axios from 'axios'



const SignIn = () => {


   const [mail, setMail] = useState('');

   const [password, setPassword] = useState('');


 const  handleLogin = async (e) => {

       e.preventDefault();

       const user = {mail, password };

       const response = await axios.post('http://localhost:8500/api/v1/users/signIn',
user);

       console.log(response.data);


 }


     return (

         <>




             <form action="">

             <h2>Login</h2>
```

```
            <div className='input-box'>
              <input type='email' placeholder='Email' required onChange={(e)=>
(setMail(e.target.value))}/>

            </div>
            <div className='input-box'>
              <input type='password' placeholder='Password' required onChange={(e)=>
(setPassword(e.target.value))}/>

            </div>
            <div className='remember-forgot'>
              <label>
                <input type="checkbox"/>
                Remember me
              </label>

            </div>
            <b><Link to="/reset" id="forget">Forgot password</Link></b>
            <button type="submit" onClick={handleLogin}>Login</button>
            <div className="register-link">
              <p>Don't have an account? <Link to="/">Register</Link></p>
            </div>
          </form>
        </>
    );
```

};

export default SignIn;

Navbar.js



import React from 'react';

import SignUp from './SignUp';

import SignIn from './SignIn';

import { Link } from 'react-router-dom';

const Navbar=()=>{

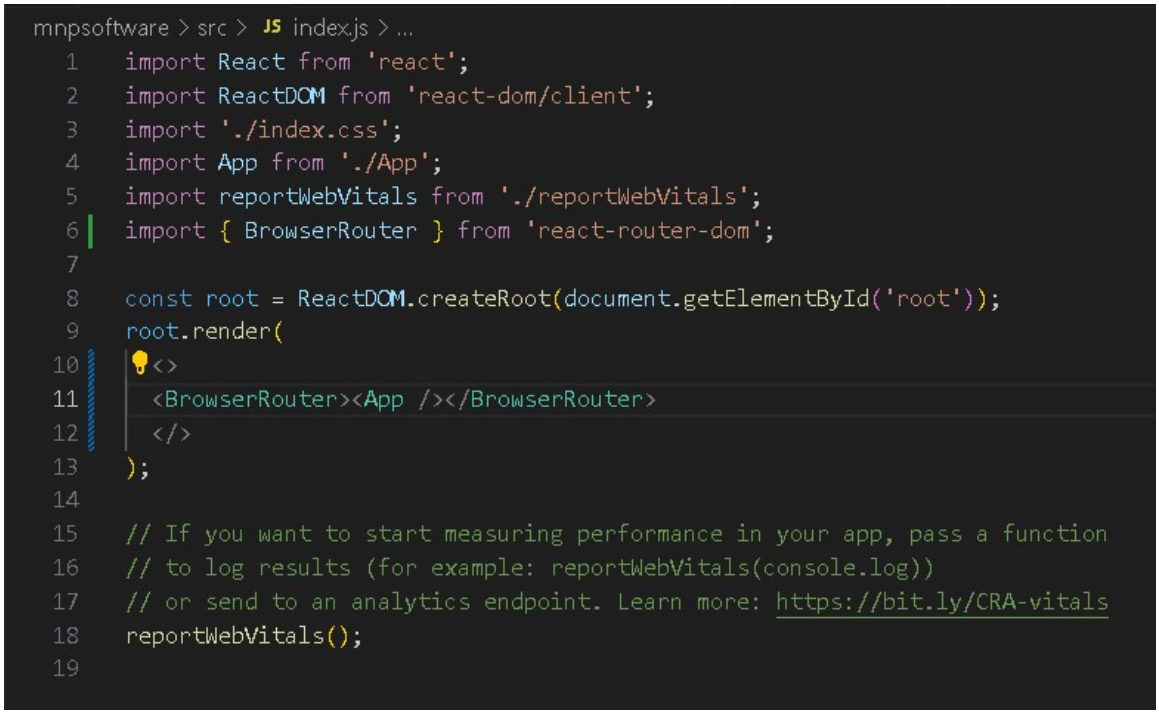
return(

```
      <div>
        <nav>
          <Link to='/' element={<SignUp />}>SignUP</Link>
          <Link to='/' element={<SignIn />}>SignIn</Link>
        </nav>

      </div>
    )
}
export default Navbar
```

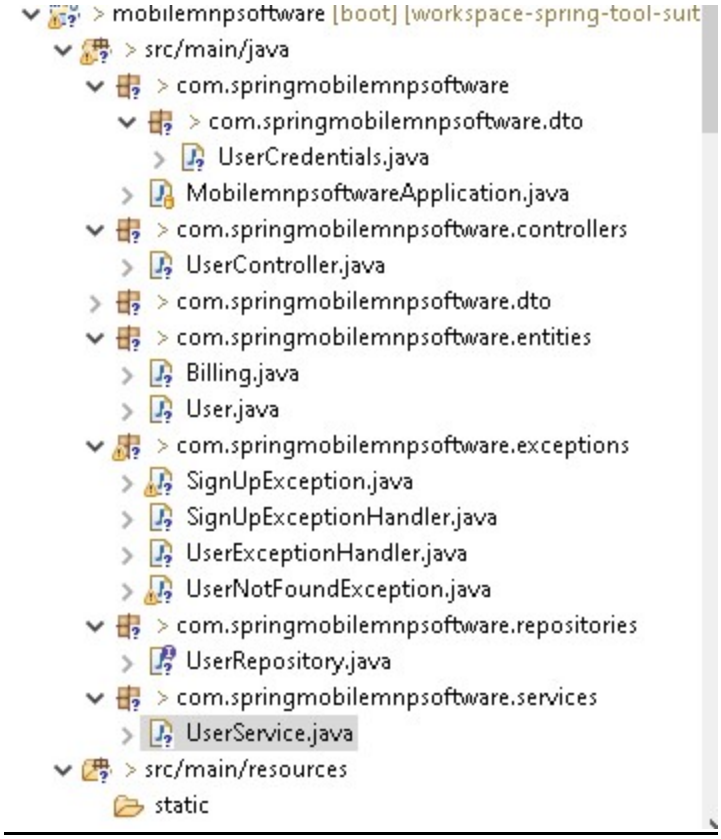
App.js

```
import logo from './logo.svg';
import './App.css';
import SignUp from './components/SignUp';
import {Routes,Route} from 'react-router-dom'
import SignIn from './components/SignIn';
import Navbar from './components/Navbar';

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

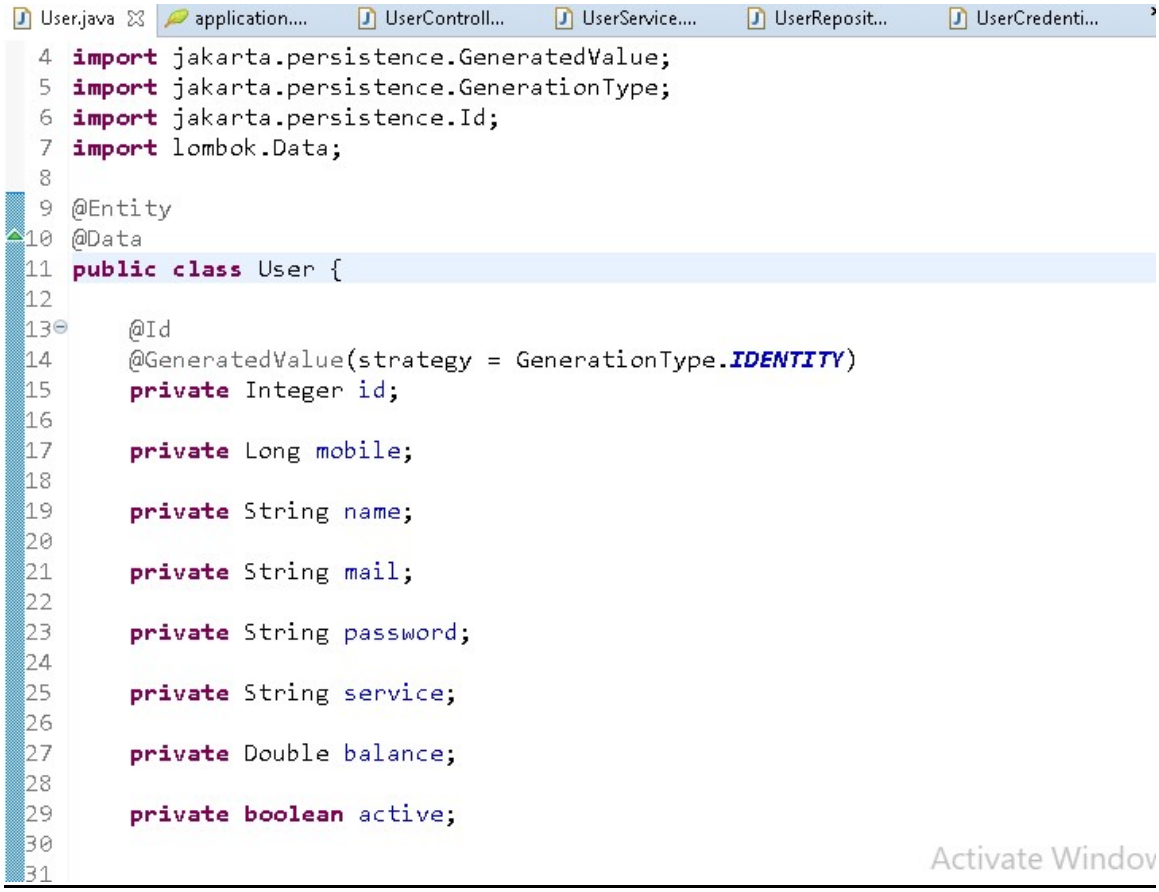
```
    <Routes>
      <Route path='/' element={<SignUp />}></Route>
      <Route path='/signIn' element={<SignIn />}></Route>
    </Routes>
   </div>
  );
}

export default App;
```



Sprin-boot


mobilemnpsoftware [boot] [workspace-spring-tool-suit
src/main/java
com.springmobilemnpsoftware
com.springmobilemnpsoftware.dto
UserCredentials.java
MobilemnpsoftwareApplication.java
com.springmobilemnpsoftware.controllers
UserController.java
com.springmobilemnpsoftware.dto
com.springmobilemnpsoftware.entities
Billing.java
User.java
com.springmobilemnpsoftware.exceptions
SignUpException.java
SignUpExceptionHandler.java
UserExceptionHandler.java
UserNotFoundException.java
com.springmobilemnpsoftware.repositories
UserRepository.java
com.springmobilemnpsoftware.services
UserService.java
src/main/resources
static

User Entity



Billing Entity

User.java | UserControll... | UserService.... | UserReposit... | Billing.java | UserCredenti...

```java
package com.springmobilemnpsoftware.entities;

import java.sql.Date;
import java.util.UUID;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.PrePersist;
import lombok.Data;

@Entity
@Data
public class Billing {

    @Id
    private String billId;

    @PrePersist
    public void generateBillID() {
        this.billId= UUID.randomUUID().toString().substring(2,13);
    }


    private Date billingDate;


}
```

UserController.java

```java
package com.springmobilemnpsoftware.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
import com.springmobilemnpsoftware.dto.UserCredentials;
import com.springmobilemnpsoftware.entities.User;
import com.springmobilemnpsoftware.services.UserService;

@RestController
@RequestMapping("api/v1/users")
@CrossOrigin("*")
public class UserController {

	@Autowired
	UserService service;

	@PostMapping("/signUp")
	public String addUser(@RequestBody User user) {
		return service.addUser(user);
	}

	@PostMapping("/signIn")
	public String signIn(UserCredentials userCredentials) {

		return
service.signIn(userCredentials.getMail(),userCredentials.getPassw
ord());
	}

	@GetMapping
	public Iterable<User> allUsers()
```

```
	{
		return service.allUsers();
	}
}
```

UserService.jaava

```
package com.springmobilemnpsoftware.services;

import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.springmobilemnpsoftware.entities.User;
import com.springmobilemnpsoftware.exceptions.SignUpException;
import
com.springmobilemnpsoftware.exceptions.UserNotFoundException;
import com.springmobilemnpsoftware.repositories.UserRepository;

@Service
public class UserService {

	@Autowired
	private UserRepository repository;

	public String addUser(User user) {
		Optional<User> existedUser =
repository.findByMail(user.getMail());
		if (existedUser.isPresent()) {
```

```
                throw new SignUpException();
            } else {
                repository.save(existedUser.get());
                return "User Saved Successfully";
            }

        }

        public String signIn(String mail, String password) {
            // TODO Auto-generated method stub

            Optional<User> user=repository.findByMail(mail);
            if(user.isPresent()) {
                User existingUser = user.get();
                if(existingUser.getPassword().equals(password)){
                    return "signin successfull";
                }
            }
            throw new UserNotFoundException();
        }

        public Iterable<User> allUsers() {
            return repository.findAll();
        }

}
```

UserRepository

```
package com.springmobilemnpsoftware.repositories;


import java.util.Optional;


import org.springframework.data.repository.CrudRepository;


import com.springmobilemnpsoftware.entities.User;


public interface UserRepository extends CrudRepository<User,
Integer> {


Optional<User> findByMail(String mail);

}
```

UserDto

```
package com.springmobilemnpsoftware.dto;


public class UserCredentials {


	private String mail;


	private String password;


	public UserCredentials(String mail, String password) {

		super();
```

```
		this.mail = mail;
		this.password = password;
	}

	@Override
	public String toString() {
		return "UserCredentials [mail=" + mail + ", password="
+ password + "]";
	}

	public String getMail() {
		return mail;
	}

	public void setMail(String mail) {
		this.mail = mail;
	}

	public String getPassword() {
		return password;
	}

	public void setPassword(String password) {
		this.password = password;
	}
	public UserCredentials() {
```

```
            // TODO Auto-generated constructor stub
        }

}
```

Exceptions

```
1  package com.springmobilemnpsoftware.exceptions;
2
3  public class SignUpException extends RuntimeException {
4
5  }
6
```

```
1  com.springmobilemnpsoftware.exceptions;
2
3  rg.springframework.http.HttpStatus;
4  rg.springframework.http.ResponseEntity;
5  rg.springframework.web.bind.annotation.ControllerAdvice;
6  rg.springframework.web.bind.annotation.ExceptionHandler;
7
8  lerAdvice
9  lass SignUpExceptionHandler {
10
11 eptionHandler(value = SignUpException.class)
12 ic ResponseEntity<Object> exception(SignUpException signUpException){
13 return new ResponseEntity<>("User Already Registered and try to register with new ma
14
15
16
```

```java
package com.springmobilemnpsoftware.exceptions;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice
public class UserExceptionHandler {

    @ExceptionHandler(value = UserNotFoundException.class)
    public ResponseEntity<Object> exception(UserNotFoundException userNotFoundExcepti
        return new ResponseEntity<>("User not Registered and doesn't have any account
    }
}

```