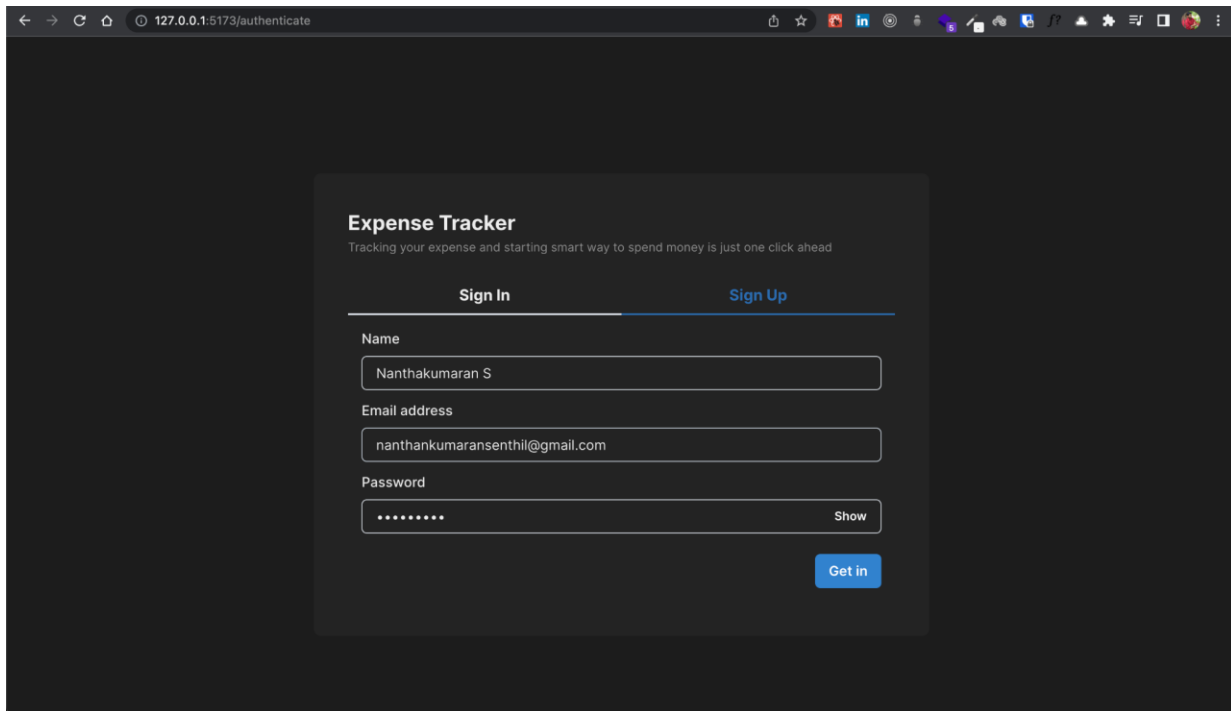


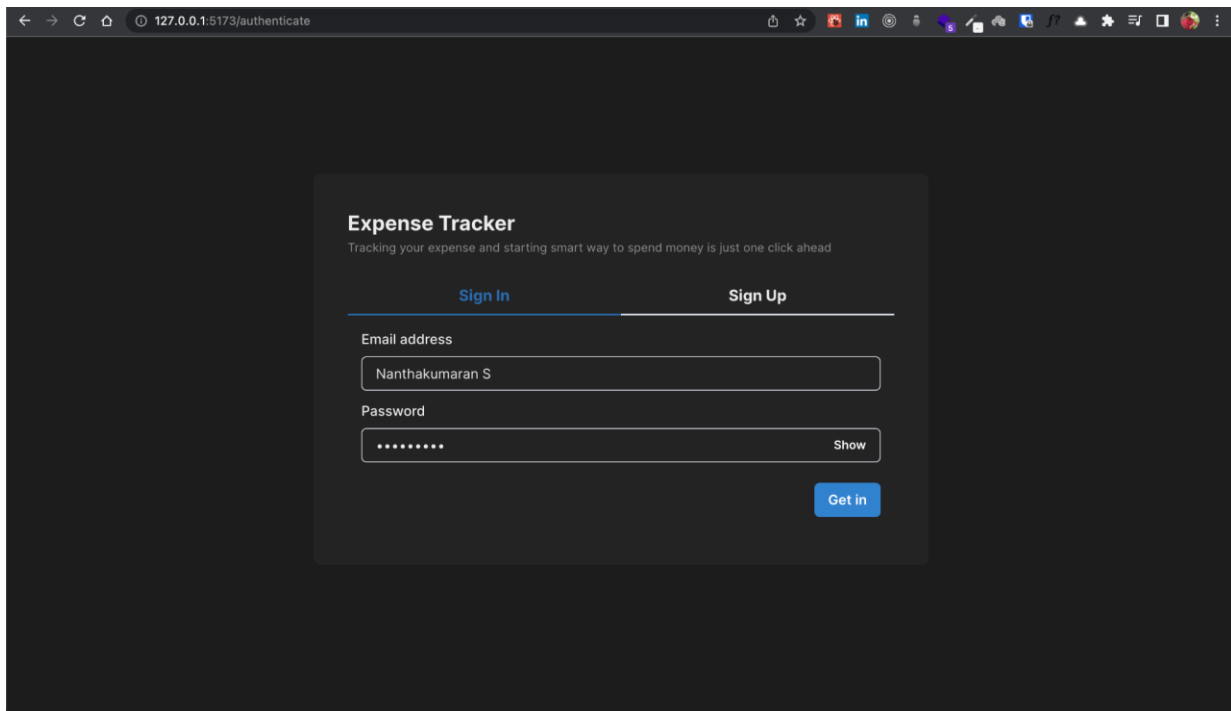
Project Development phase - Sprint 1

Team ID	PNT2022TMID20278
Project Name	Personal Expense Tracker

Login & Registration Pages

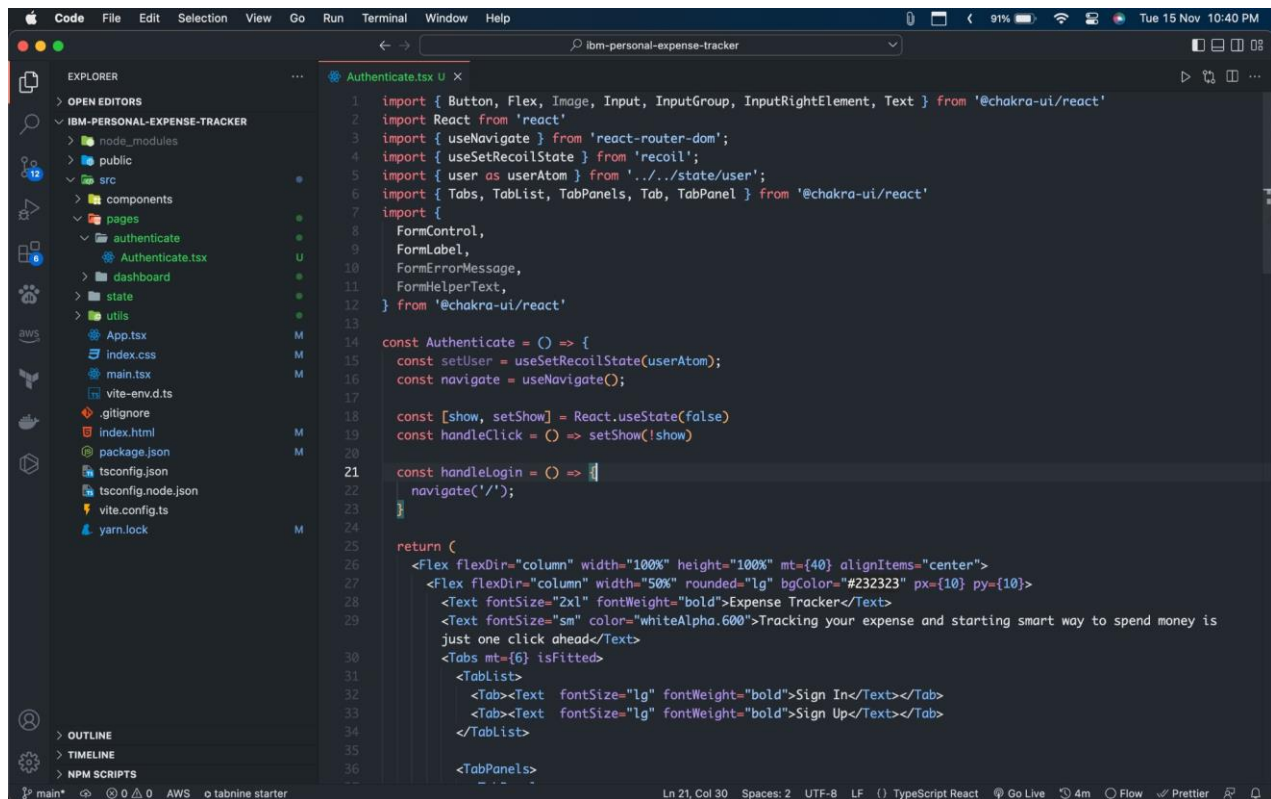


A screenshot of a web browser displaying the 'Expense Tracker' application. The browser's address bar shows '127.0.0.1:5173/authenticate'. The page has a dark background with a central light gray box containing the login form. The form is titled 'Expense Tracker' with a subtitle 'Tracking your expense and starting smart way to spend money is just one click ahead'. Below the title, there are two tabs: 'Sign In' (active) and 'Sign Up'. The 'Sign In' tab contains three input fields: 'Name' (with the value 'Nanthakumaran S'), 'Email address' (with the value 'nanthakumaransenthil@gmail.com'), and 'Password' (with masked characters '.....'). A 'Show' button is next to the password field. A blue 'Get in' button is at the bottom right of the form.

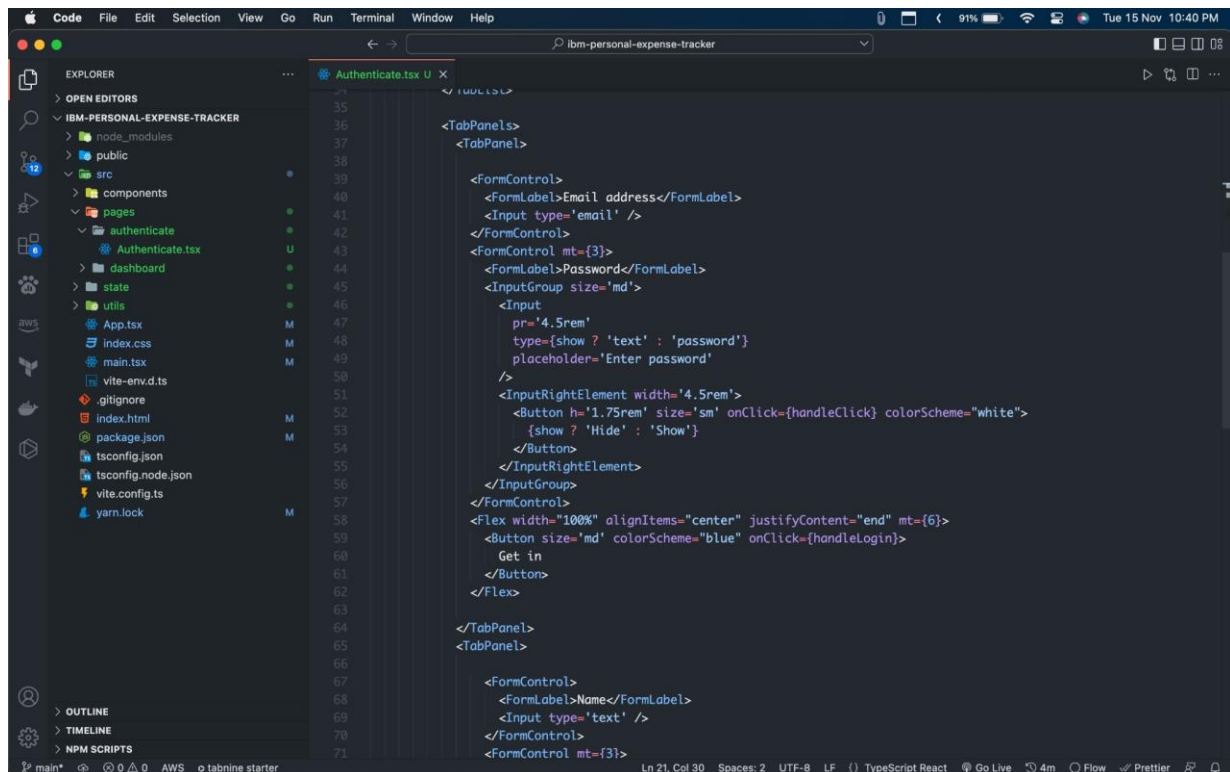


A screenshot of the same 'Expense Tracker' application, but with the 'Sign Up' tab selected. The browser's address bar remains '127.0.0.1:5173/authenticate'. The 'Sign Up' tab is active, and the 'Sign In' tab is inactive. The form contains three input fields: 'Email address' (with the value 'Nanthakumaran S'), 'Password' (with masked characters '.....'), and a 'Show' button next to the password field. A blue 'Get in' button is at the bottom right of the form.

FrontEnd Code



```
1 import { Button, Flex, Image, Input, InputGroup, InputRightElement, Text } from '@chakra-ui/react'
2 import React from 'react'
3 import { useNavigate } from 'react-router-dom';
4 import { useSetRecoilState } from 'recoil';
5 import { user as userAtom } from '../../state/user';
6 import { Tabs, TabList, TabPanels, Tab, TabPanel } from '@chakra-ui/react'
7 import {
8   FormControl,
9   FormLabel,
10   FormErrorMessage,
11   FormHelperText,
12 } from '@chakra-ui/react'
13
14 const Authenticate = () => {
15   const setUser = useSetRecoilState(userAtom);
16   const navigate = useNavigate();
17
18   const [show, setShow] = React.useState(false)
19   const handleClick = () => setShow(!show)
20
21   const handleLogin = () => {
22     navigate('/');
23   }
24
25   return (
26     <Flex flexDir="column" width="100%" height="100%" mt={40} alignItems="center">
27       <Text fontSize="2xl" fontWeight="bold">Expense Tracker</Text>
28       <Text fontSize="sm" color="whiteAlpha.600">Tracking your expense and starting smart way to spend money is
29         just one click ahead</Text>
30       <Tabs mt={6} isFitted>
31         <TabList>
32           <Tab><Text fontSize="lg" fontWeight="bold">Sign In</Text></Tab>
33           <Tab><Text fontSize="lg" fontWeight="bold">Sign Up</Text></Tab>
34         </TabList>
35         <TabPanels>
36           <TabPanel>
```



```
37 <TabPanel>
38   <FormControl>
39     <FormLabel>Email address</FormLabel>
40     <Input type='email' />
41   </FormControl>
42   <FormControl mt={3}>
43     <FormLabel>Password</FormLabel>
44     <InputGroup size='md'>
45       <Input
46         pr='4.5rem'
47         type={show ? 'text' : 'password'}
48         placeholder='Enter password'
49       />
50       <InputRightElement width='4.5rem'>
51         <Button h='1.75rem' size='sm' onClick={handleClick} colorScheme='white'>
52           {show ? 'Hide' : 'Show'}
53         </Button>
54       </InputRightElement>
55     </InputGroup>
56   </FormControl>
57   <Flex width="100%" alignItems="center" justifyContent="end" mt={6}>
58     <Button size='md' colorScheme='blue' onClick={handleLogin}>
59       Get in
60     </Button>
61   </Flex>
62 </TabPanel>
63 </TabPanel>
64 </TabPanels>
65 </TabList>
66 </Flex>
67
68 <FormLabel>Name</FormLabel>
69 <Input type='text' />
70 </FormLabel>
71 <FormLabel>Password</FormLabel>
```

The screenshot shows the VS Code editor interface with the 'Authenticate.tsx' file open. The Explorer sidebar on the left displays the project structure, including folders like 'src', 'components', 'pages', 'dashboard', 'state', and 'utils', and files like 'App.tsx', 'index.css', 'main.tsx', 'vite-env.d.ts', '.gitignore', 'index.html', 'package.json', 'tsconfig.json', 'tsconfig.node.json', 'vite.config.ts', and 'yarn.lock'. The main editor area shows the following code:

```
65 <TabPanel>
66
67 <FormControl>
68   <FormLabel>Name</FormLabel>
69   <Input type='text' />
70 </FormControl>
71 <FormControl mt={3}>
72   <FormLabel>Email address</FormLabel>
73   <Input type='email' />
74 </FormControl>
75 <FormControl mt={3}>
76   <FormLabel>Password</FormLabel>
77   <InputGroup size='md'>
78     <Input
79       pr='4.5rem'
80       type={show ? 'text' : 'password'}
81       placeholder='Enter password'
82     />
83     <InputRightElement width='4.5rem'>
84       <Button h='1.75rem' size='sm' onClick={handleClick} colorScheme='white'>
85         {show ? 'Hide' : 'Show'}
86       </Button>
87     </InputRightElement>
88   </InputGroup>
89 </FormControl>
90 <Flex width='100%' alignItems='center' justifyContent='end' mt={6}>
91   <Button size='md' colorScheme='blue'>
92     Get in
93   </Button>
94 </Flex>
95
96 </TabPanel>
97 </TabPanels>
98 </Tabs>
99 </Flex>
100 </Flex>
101 )
```

The status bar at the bottom indicates the file is at Line 21, Column 30, with 2 spaces, UTF-8 encoding, and LF line endings. It also shows the TypeScript React extension and other tools like Go Live, 4m, Flow, Prettier, and R.

Flask Code

The screenshot shows the VS Code editor interface with the 'app.py' file open in the 'EXPENSE TRACKER' project. The Explorer sidebar on the left displays the project structure, including folders like 'certificates', 'controllers', '__pycache__', 'budgets.py', 'login.py', 'register.py', 'transactions.py', 'db', 'migration', and 'app.py'. The main editor area shows the following code:

```
1 from flask import Flask, request
2 from controllers.login import Login
3 from controllers.register import Register
4 from controllers.budgets import Budgets
5 from db.db import db
6
7 app = Flask(__name__)
8
9 login_controller = Login()
10 register_controller = Register()
11 budget_controller = Budgets()
12
13 @app.route("/api/login", methods=["post"])
14 def auth():
15     return login_controller.post()
16
17 @app.route("/api/register", methods=["post"])
18 def register():
19     return register_controller.post()
20
21 @app.route("/verification/<email>", methods=["get"])
```

```
21    ))
22
23    def post(self):
24        payload = request.json
25        try:
26            self._sanitizer(payload)
27            result = ibm_db.exec_immediate(db, "SELECT * FROM USERS WHERE EMAIL='{}' AND VERIFIED=TRUE".format(payload["email"]))
28            value = ibm_db.fetch_tuple(result)
29            if value == False:
30                return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
31            if bcrypt.checkpw(bytes("{}{}{}".format(payload["password"], 'utf-8'), bytes(value[3], 'utf-8'))):
32                payload = {
33                    "email": value[2],
34                    "token": value[5],
35                    "name": value[1],
36                    "currency": value[4]    You, yesterday * initial _
37                }
38                return json.dumps({"status": "Success", "data": payload, "message": "success"})
39            else:
40                return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
41        except KeyError as e:
42            return json.dumps(e.args)
```