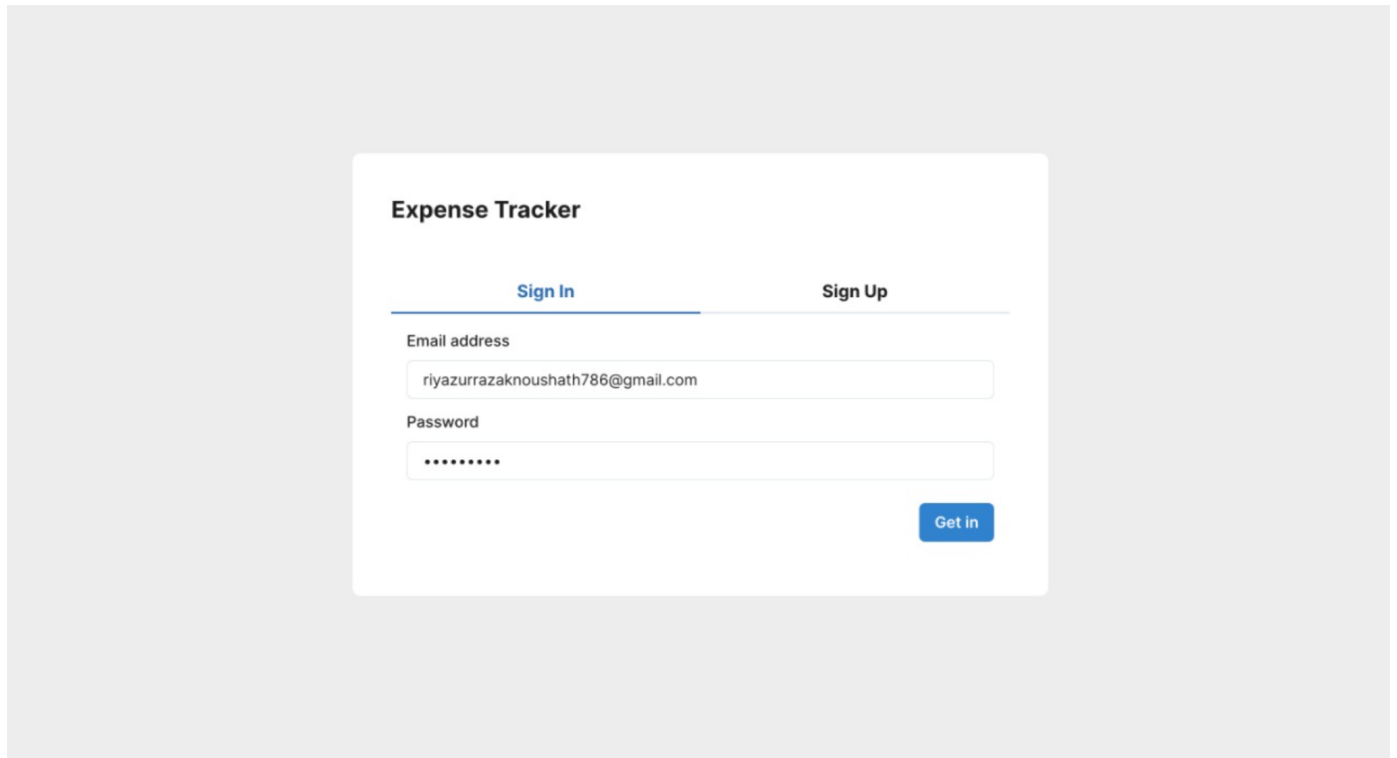


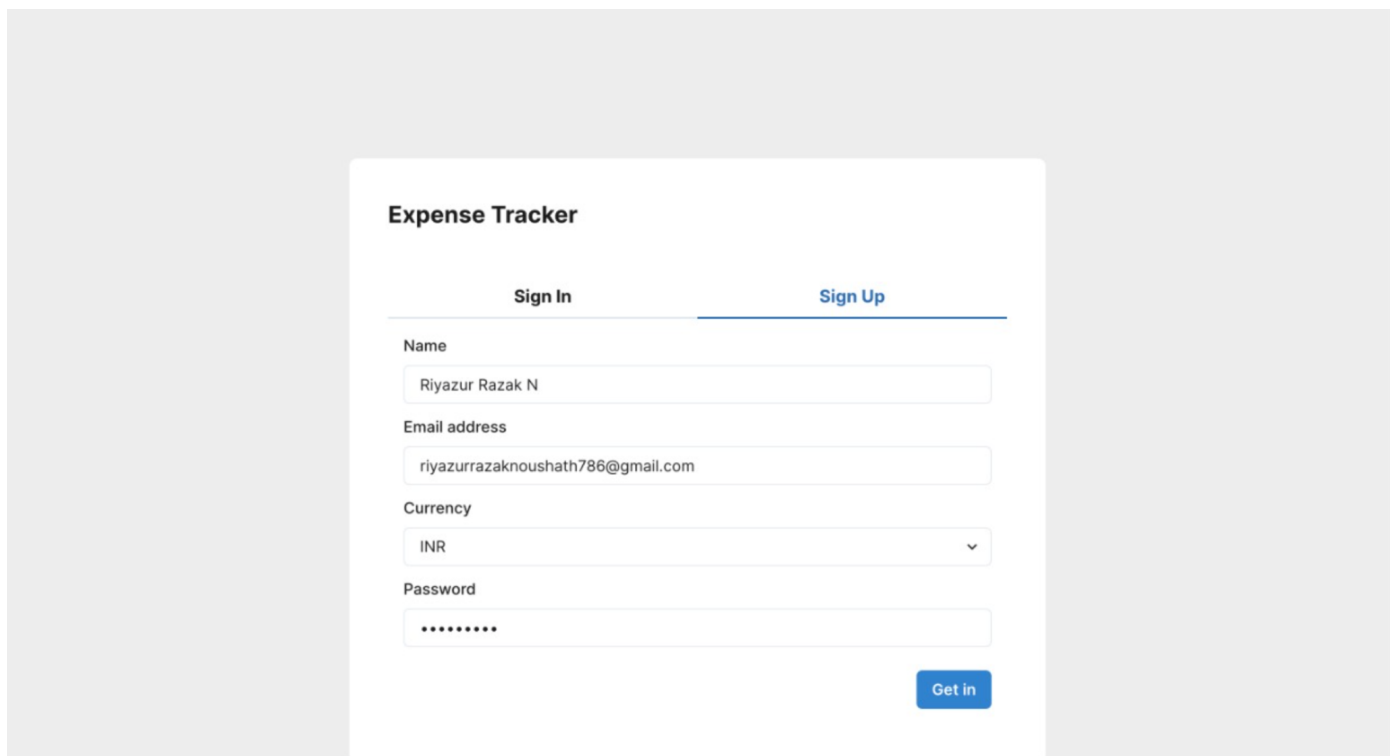
Sprint 1

Team ID	PNT2022TMID04671
Project Name	Personal Expense Tracker

LOGIN & REGISTER PAGE SNAPSHOTS



A screenshot of the 'Expense Tracker' login page. The page has a white background with a light gray border. At the top, the title 'Expense Tracker' is centered. Below it, there are two tabs: 'Sign In' (active, underlined) and 'Sign Up'. The 'Sign In' section contains two input fields: 'Email address' with the value 'riyazurrazaknoushath786@gmail.com' and 'Password' with masked characters '*****'. A blue 'Get in' button is located at the bottom right of the form.



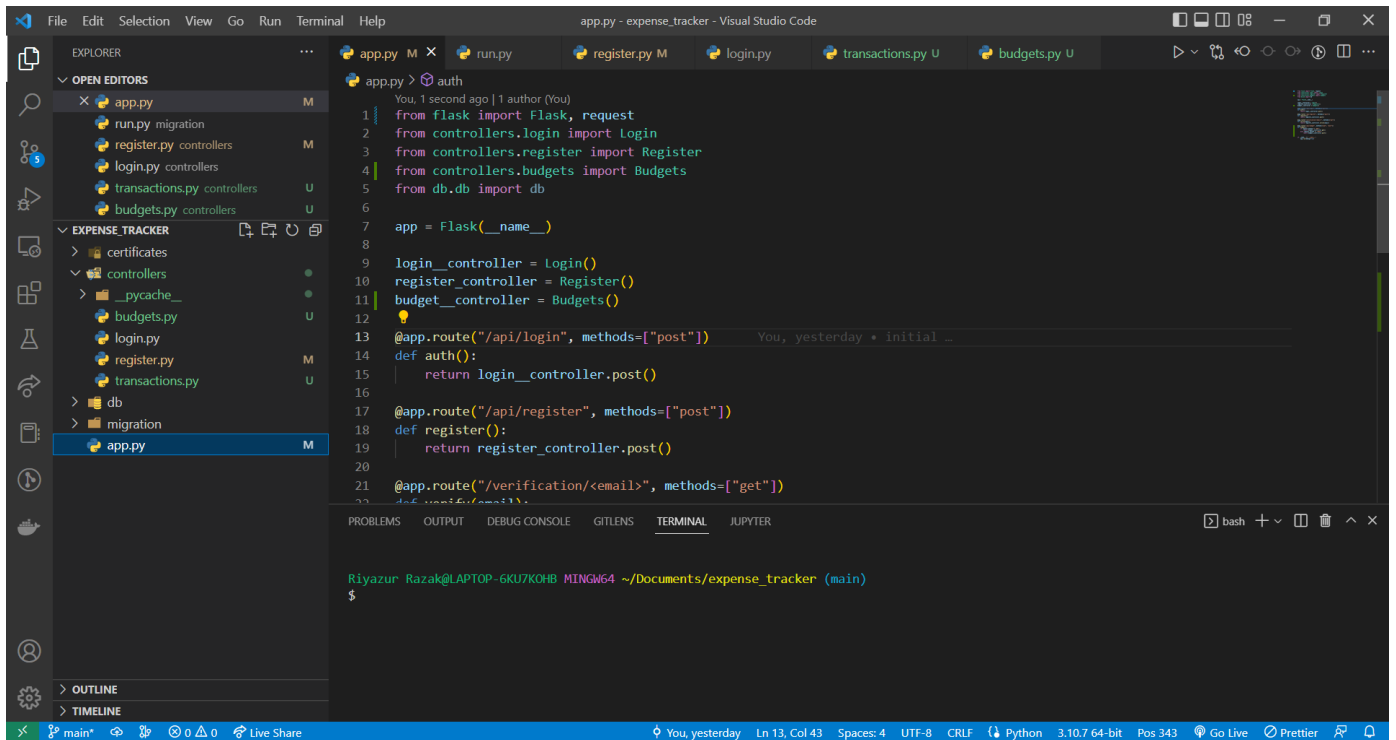
A screenshot of the 'Expense Tracker' register page. The page has a white background with a light gray border. At the top, the title 'Expense Tracker' is centered. Below it, there are two tabs: 'Sign In' and 'Sign Up' (active, underlined). The 'Sign Up' section contains four input fields: 'Name' with the value 'Riyazur Razak N', 'Email address' with the value 'riyazurrazaknoushath786@gmail.com', 'Currency' with a dropdown menu showing 'INR', and 'Password' with masked characters '*****'. A blue 'Get in' button is located at the bottom right of the form.

FRONTEND CODE SNAPSHOTS

```
1 import { Button, Flex, Image, Input, InputGroup, InputRightElement, Select, Text } from '@chakra-ui/react'
2 import React, { useEffect } 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 import axios from 'axios';
14
15 const Authenticate = () => {
16   const setUser = useSetRecoilState(userAtom);
17   const navigate = useNavigate();
18
19   const [show, setShow] = React.useState(false)
20   const handleClick = () => setShow(!show)
21
22   const [currency, setCurrency] = React.useState<string[]>([]);
23
24   const handleLogin = () => {
25     navigate('/');
26   }
27
28   useEffect(() => {
29     const fetch = async () => {
30       const data = await axios({
31         method: 'GET',
32         url: 'https://api.apilayer.com/exchangerates_data/symbols',
33         headers: {
34           apikey: 'FA1Nha1PRf5VrFv9BMNRLoNqXw08MmHS',
35         },
36       });
37       console.log(data.data.symbols);
38     }
39     fetch();
40   });
41 }
```

```
Dashboard.tsx U  Authenticate.tsx U x
49 return (
50   <Flex flexDir="column" width="100%" height="100%" mt={40} alignItems="center">
51     <Flex flexDir="column" width="50%" rounded="lg" bgColor="#232323" px={10} py={10}>
52       <Text fontSize="2xl" fontWeight="bold">Expense Tracker</Text>
53       <Text fontSize="sm" color="whiteAlpha.600">Tracking your expense and starting smart way to spend money is
54         just one click ahead</Text>
55       <Tabs mt={6} isFitted>
56         <TabList>
57           <Tab><Text fontSize="lg" fontWeight="bold">Sign In</Text></Tab>
58           <Tab><Text fontSize="lg" fontWeight="bold">Sign Up</Text></Tab>
59         </TabList>
60
61         <TabPanels>
62           <TabPanel>
63             <FormControl>
64               <FormLabel>Email address</FormLabel>
65               <Input type="email" />
66             </FormControl>
67             <FormControl mt={3}>
68               <FormLabel>Password</FormLabel>
69               <InputGroup size="md">
70                 <Input
71                   pr="4.5rem"
72                   type={show ? 'text' : 'password'}
73                   placeholder="Enter password"
74                 />
75                 <InputRightElement width="4.5rem">
76                   <Button h="1.75rem" size="sm" onClick={handleClick} colorScheme="white">
77                     {show ? 'Hide' : 'Show'}
78                   </Button>
79                 </InputRightElement>
80               </InputGroup>
81             </FormControl>
82             <Flex width="100%" alignItems="center" justifyContent="end" mt={6}>
83               <Button size="md" colorScheme="blue" onClick={handleLogin}>
84                 Get in
85             </Flex>
86           </TabPanel>
87         </TabPanels>
88       </Tabs>
89     </Flex>
90   </Flex>
91 )
```

BACKEND CODE SNAPSHOTS

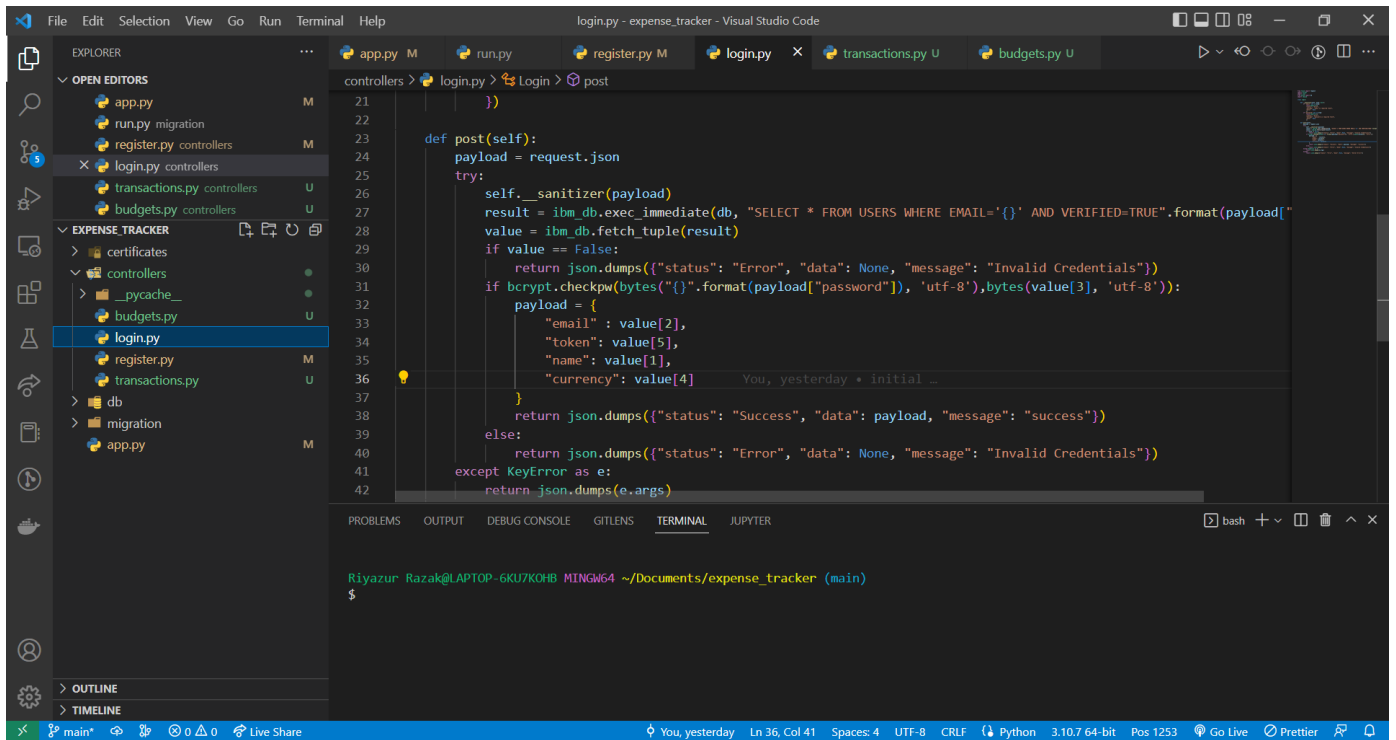


```
File Edit Selection View Go Run Terminal Help
app.py - expense_tracker - Visual Studio Code

EXPLORER
  OPEN EDITORS
    app.py M
    run.py migration
    register.py controllers M
    login.py controllers
    transactions.py controllers U
    budgets.py controllers U
  EXPENSE_TRACKER
    certificates
    controllers
    _pycache_
    budgets.py U
    login.py
    register.py M
    transactions.py U
    db
    migration
    app.py M

app.py
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"])
22 def verification(email):
23     pass

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL JUPYTER
Riyazur Razak@LAPTOP-6KU7K0HB MINGW64 ~/Documents/expense_tracker (main)
$
```



```
File Edit Selection View Go Run Terminal Help
login.py - expense_tracker - Visual Studio Code

EXPLORER
  OPEN EDITORS
    app.py M
    run.py migration
    register.py controllers M
    login.py controllers
    transactions.py controllers U
    budgets.py controllers U
  EXPENSE_TRACKER
    certificates
    controllers
    _pycache_
    budgets.py U
    login.py
    register.py M
    transactions.py U
    db
    migration
    app.py M

login.py
21
22
23
24 def post(self):
25     payload = request.json
26     try:
27         self._sanitizer(payload)
28         result = ibm_db.exec_immediate(db, "SELECT * FROM USERS WHERE EMAIL='{0}' AND VERIFIED=TRUE".format(payload["email"]))
29         value = ibm_db.fetch_tuple(result)
30         if value == False:
31             return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
32         if bcrypt.checkpw(bytes(payload["password"], 'utf-8'), bytes(value[3], 'utf-8')):
33             payload = {
34                 "email": value[2],
35                 "token": value[5],
36                 "name": value[1],
37                 "currency": value[4]
38             }
39             return json.dumps({"status": "Success", "data": payload, "message": "success"})
40         else:
41             return json.dumps({"status": "Error", "data": None, "message": "Invalid Credentials"})
42     except KeyError as e:
43         return json.dumps(e.args)

PROBLEMS OUTPUT DEBUG CONSOLE GITLENS TERMINAL JUPYTER
Riyazur Razak@LAPTOP-6KU7K0HB MINGW64 ~/Documents/expense_tracker (main)
$
```