

## Project Development phase - Sprint 1

Team ID	PNT2022TMID04644
Project Name	Personal Expense Tracker

### Login & Registration Pages

### Expense Tracker

[Sign In](#)[Sign Up](#)

Name

Harish M

Email address

harishm.19ece@kongu.edu

Password

\*\*\*\*\*

Get in

### Expense Tracker

[Sign In](#)[Sign Up](#)

Email address

harishm.19ece@kongu.edu

Password

\*\*\*\*\*|

Get in

## FrontEnd Code

```
Authenticate.tsx U X
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       <Flex flexDir="column" width="50%" rounded="lg" bgColor="#232323" px={10} py={10}>
28         <Text fontSize="2xl" fontWeight="bold">Expense Tracker</Text>
29         <Text fontSize="sm" color="whiteAlpha.600">Tracking your expense and starting smart way to spend money is
30           just one click ahead</Text>
31         <Tabs mt={6} isFitted>
32           <TabList>
33             <Tab><Text fontSize="lg" fontWeight="bold">Sign In</Text></Tab>
34             <Tab><Text fontSize="lg" fontWeight="bold">Sign Up</Text></Tab>
35           </TabList>
36           <TabPanels>
37             <TabPanel>
38               <FormLabel>Email address</FormLabel>
39               <Input type='email' />
40             </TabPanel>
41             <TabPanel>
42               <FormLabel>Password</FormLabel>
43               <InputGroup size='md'>
44                 <Input
45                   pr='4.5rem'
46                   type={show ? 'text' : 'password'}
47                   placeholder='Enter password'
48                 />
49                 <InputRightElement width='4.5rem'>
50                   <Button h='1.75rem' size='sm' onClick={handleClick} colorScheme="white">
51                     {show ? 'Hide' : 'Show'}
52                   </Button>
53                 </InputRightElement>
54               </InputGroup>
55             </TabPanel>
56             <TabPanel>
57               <FormLabel>Name</FormLabel>
58               <Input type='text' />
59             </TabPanel>
60           </TabPanels>
61         </Tabs>
62       </Flex>
63     </Flex>
64   )
65 }
```

```
Authenticate.tsx U X
35
36 <TabPanels>
37   <TabPanel>
38     <FormLabel>Email address</FormLabel>
39     <Input type='email' />
40   </TabPanel>
41   <TabPanel>
42     <FormLabel>Password</FormLabel>
43     <InputGroup size='md'>
44       <Input
45         pr='4.5rem'
46         type={show ? 'text' : 'password'}
47         placeholder='Enter password'
48       />
49       <InputRightElement width='4.5rem'>
50         <Button h='1.75rem' size='sm' onClick={handleClick} colorScheme="white">
51           {show ? 'Hide' : 'Show'}
52         </Button>
53       </InputRightElement>
54     </InputGroup>
55   </TabPanel>
56   <TabPanel>
57     <FormLabel>Name</FormLabel>
58     <Input type='text' />
59   </TabPanel>
60 </TabPanels>
61 </FormLabel>
62 </FormLabel>
63 </FormLabel>
64 </FormLabel>
65 </FormLabel>
66 </FormLabel>
67 </FormLabel>
68 </FormLabel>
69 </FormLabel>
70 </FormLabel>
71 </FormLabel>
```

```
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 </TabPanel>
96 </TabPanels>
97 </Tabs>
98 </Flex>
99 </Flex>
100
101
```

## Flask Code

```
app.py > auth
You, 1 second ago | 1 author (You)
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"])
```

controllers > login.py > Login > post

```
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='{0}' AND VERIFIED=TRUE".format(payload["
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]
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)
```