

Project Development phase - Sprint 2

Team ID	PNT2022TMID04644
Project Name	Personal Expense Tracker

Add Income & Expense

H

500000
Income

-

15000
Expense

=

485000
Available Balance

Expense

Income

Amount

Category

Select option

Description

Describe your transaction

Add Expense

Transactions

D

2022-11-19

food

5000

C

2022-11-19

500000

D

2022-11-19

medicine

10000

H

500000
Income

-

15000
Expense

=

485000
Available Balance

Expense

Income

Amount

Add Income

Transactions

D

2022-11-19

food

5000

C

2022-11-19

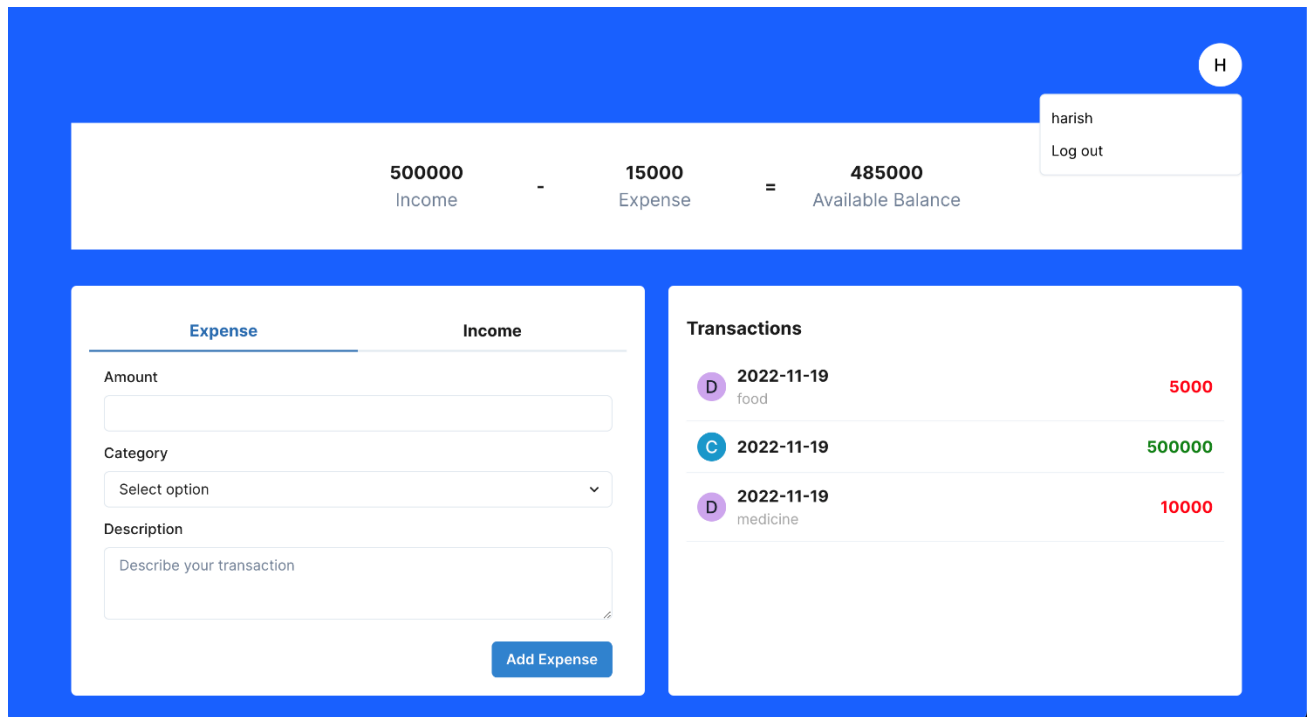
500000

D

2022-11-19

medicine

10000



Front End Code

```

Dashboard.tsx U  EntryPoint.tsx U  TopCard.tsx U  Authenticate.tsx U  user.ts U
1  import { Button, Flex, FormControl, FormLabel, Input, InputGroup, InputLeftElement, Select, Tab, TabList, TabPanel,
2    TabPanels, Tabs, Text, Textarea } from '@chakra-ui/react'
3  import React from 'react'
4  import { useRecoilValue } from 'recoil'
5  import { currency as currencyAtom } from '../../state/user'
6
7  const EntryPoint = () => {
8    const category = [
9      'food',
10     'entertainment',
11     'medicine',
12     'transport',
13     'utilities',
14     'bills',
15     'subscription',
16     'others'
17   ]
18   // const currency = useRecoilValue(currencyAtom)
19
20   return (
21     <Flex flexDir="column" width="45%" bg="#232323" px={5} py={1}>
22       <Tabs mt={6} isFitted>
23         <TabList>
24           <Tab><Text fontSize="lg" fontWeight="bold">Expense</Text></Tab>
25           <Tab><Text fontSize="lg" fontWeight="bold">Income</Text></Tab>
26         </TabList>
27         <TabPanels>
28           <TabPanel>
29             <FormControl>
30               <FormLabel>Amount</FormLabel>
31               <Input type="text" />
32               {/* <InputGroup size='md'>
33                 <Select placeholder='INR' width="150px" rounded="sm">
34                   {currency.map(item => {
35                     return <option value={item}>{item}</option>
36                   })}
37                 </Select>
38               </InputGroup>
39             </FormLabel>
40             <Textarea />
41             <Button color="blue" mt={5}>Add Expense</Button>
42           </FormLabel>
43         </FormLabel>
44       </FormLabel>
45     </FormLabel>
46   )
47 }

```

```
Dashboard.tsx U EntryPoint.tsx U TopCard.tsx U Authenticate.tsx U user.ts U
43 <FormLabel>Category</FormLabel>
44 <Select placeholder='Select option'>
45   {category.map(item => {
46     return <option value={item}>{item}</option>
47   })}
48 </Select>
49 </FormControl>
50 <FormControl mt={3}>
51   <FormLabel>Description</FormLabel>
52   <Textarea placeholder='Describe your transaction' />
53 </FormControl>
54 <Flex width="100%" alignItems="center" justifyContent="end" mt={6}>
55   <Button size='md' colorScheme="blue">
56     Add Expense
57   </Button>
58 </Flex>
59
60 </TabPanel>
61 <TabPanel>
62
63   <FormControl>
64     <FormLabel>Amount</FormLabel>
65     <Input type='text' />
66   </FormControl>
67   <Flex width="100%" alignItems="center" justifyContent="end" mt={6}>
68     <Button size='md' colorScheme="blue">
69       Add Income
70     </Button>
71   </Flex>
72
73 </TabPanel>
74 </TabPanels>
75 </Tabs>
76 </Flex>
77
78
79
```

Ln 18, Col 1 Spaces: 4 UTF-8 LF () TypeScript React Go Live 57m Flow Prettier

```
Dashboard.tsx U EntryPoint.tsx U TopCard.tsx U Authenticate.tsx U user.ts U
1 import { Flex, Text } from '@chakra-ui/react'
2 import React from 'react'
3
4 const TopCard = (props: any) => {
5   return (
6     <Flex flexDir="row" alignItems="center" justifyContent="center" width="30%" bg="#232323" px={10} py={8}>
7       {props.icon}
8       <Flex flexDir="column" justifyContent="center" ml={5}>
9         <Text fontSize="lg" color="whiteAlpha.600">{props.text}</Text>
10        <Text fontSize="xl" fontWeight="bold">{props.data}</Text>
11      </Flex>
12    </Flex>
13  )
14 }
15
16 export default TopCard
```

Ln 6, Col 81 (12 selected) Spaces: 4 UTF-8 LF () TypeScript React Go Live 59m Flow Prettier

```
Dashboard.tsx U x EntryPoint.tsx U TopCard.tsx U Authenticate.tsx U user.ts U
11 const user = useRecoilValue(userAtom);
12
13
14 return (
15   <Flex flexDir="column" width="100%" alignItems="center" justifyContent="center">
16     <Flex flexDir="row" width="90%" alignItems="center" justifyContent="space-between" mt={10}>
17       <Text fontSize="2xl" fontWeight="bold">Welcome back, {user.name}</Text>
18       <Menu>
19         <MenuButton><Avatar name={user.name} /></MenuButton>
20         <Portal>
21           <MenuList color='black'>
22             <MenuItem>Profile</MenuItem>
23             <MenuItem>Log out</MenuItem>
24           </MenuList>
25         </Portal>
26       </Menu>
27     </Flex>
28     <Flex flexDir="row" width="90%" alignItems="center" justifyContent="space-between" mt={10}>
29       <TopCard icon={MdOutlineAccountBalanceWallet size={35} /> text="Available Balance" data="250000" />
30       <TopCard icon={GiMoneyStack size={35} /> text="Income" data="500000" />
31       <TopCard icon={GiPayMoney size={35} /> text="Expense" data="250000" />
32     </Flex>
33     <Flex flexDir="row" width="90%" justifyContent="space-between" mt={10}>
34       <heading>
35         <EntryPoint />
36       </Flex>
37     </Flex>
38   )
39 }
40 export default Dashboard
```

Flask Code

```
controllers > transactions.py > Transactions > get
69
70
71 def get(self):
72     try:
73         self.__middleware()
74         result = ibm_db.exec_immediate(db, "SELECT * FROM TRANSACTIONS WHERE TRANSACTIONS.USER={}".format(self.user[
75         transactions = ibm_db.fetch_tuple(result)
76         payload = []
77         while (transactions):
78             payload.append({
79                 "amount": transactions[1],
80                 "date": str(transactions[2]),
81                 "category": transactions[3],
82                 "description": transactions[4],
83                 "type": transactions[5]
84             })
85             transactions = ibm_db.fetch_tuple(result)
86         if len(payload) == 0:
87             return json.dumps({"status": "Error", "data": None, "message": "No Transaction Availabe"})
88         return json.dumps({"status": "Success", "data": payload, "message": "Transaction Retrived Successfully"})
```

API Response

