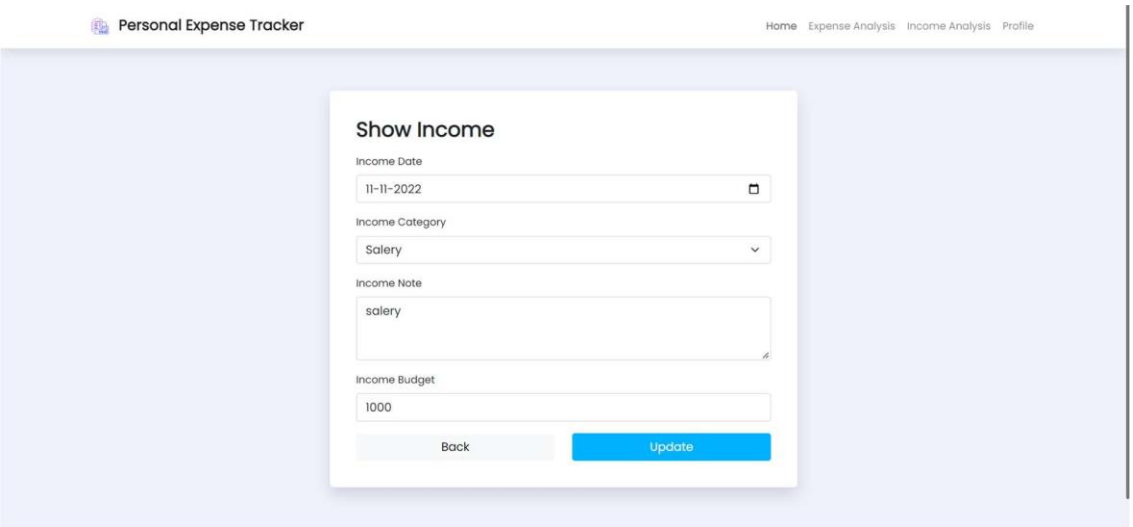
Project Development Phase

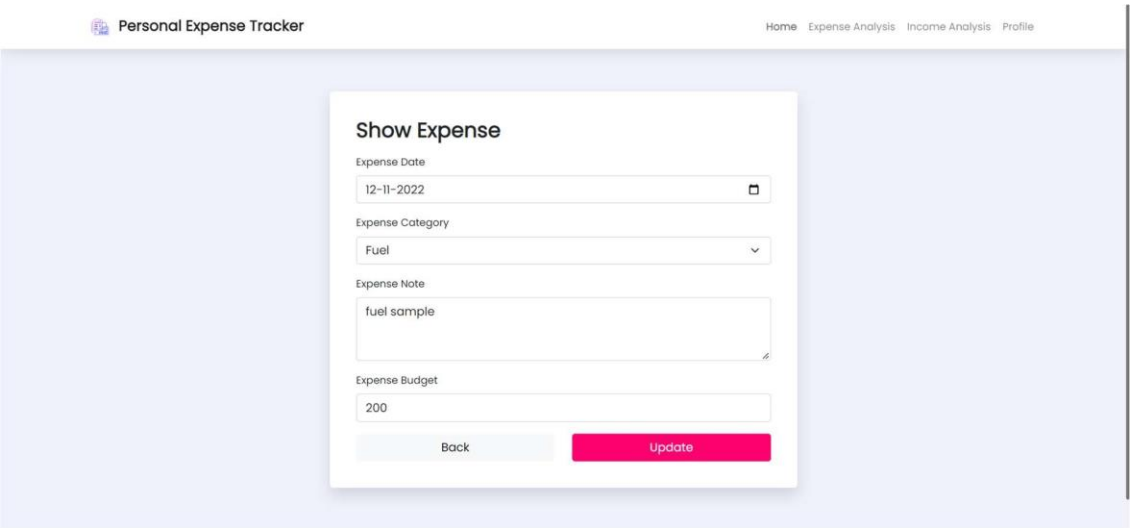
Sprint – 4

Team ID	PNT2022TMID13990
Project Name	Personal Expense Tracker Application

Edit Income Page:



Edit Expense Page:

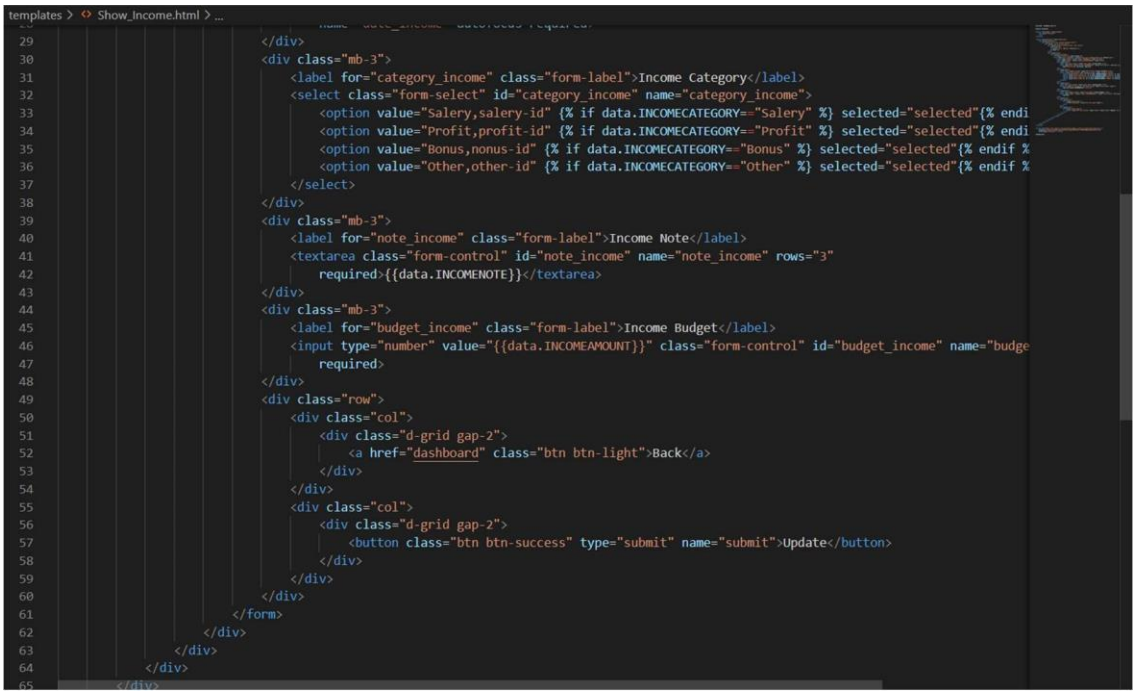


Code:

Base file - HomeBase.html

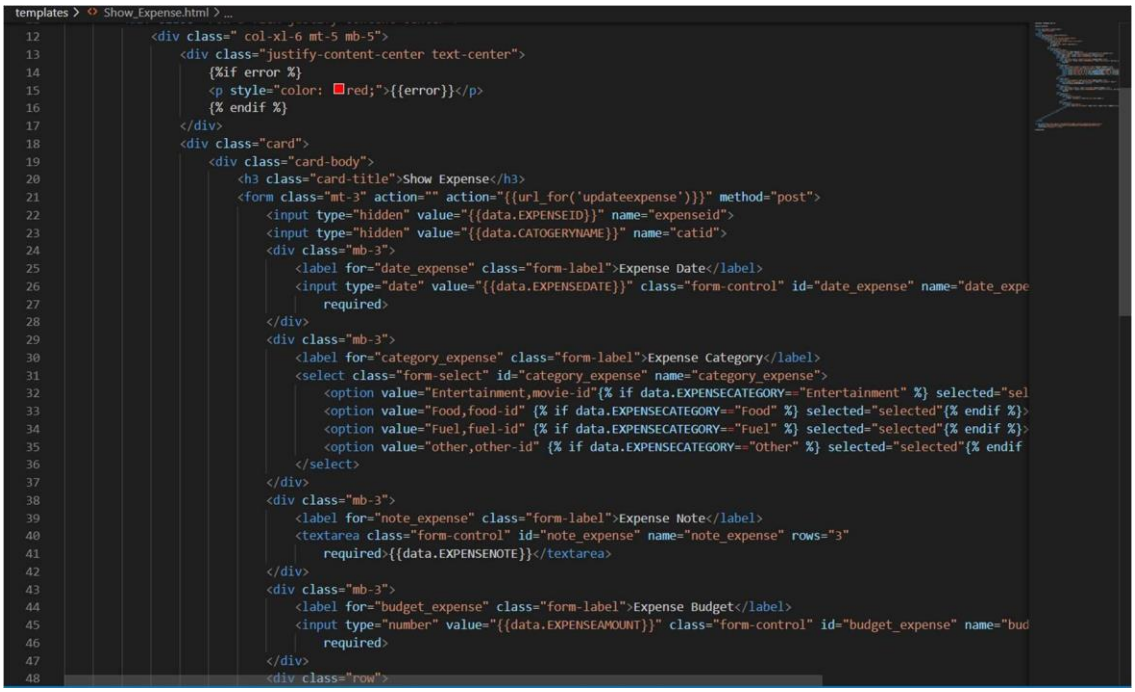
```
templates > HomeBase.html > ...
33
34      <!--Navbar -->
35      <nav class="navbar navbar-expand-lg navbar-light d-none d-md-block bg-light fixed-top">
36          <div class="container-md">
37              <a class="navbar-brand" href="./dashboard">
38                  <img src="{{url_for('static', filename='images/logo.jpg')}}" alt="" width="35" height="35"
39                      class="me-2"><span class="fw-bold">Personal Expense Tracker</span>
40              </a>
41              <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
42                  aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
43                  <span class="navbar-toggler-icon"></span>
44              </button>
45              <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
46                  <div class="navbar-nav ms-auto">
47                      <a class="nav-link actives" href="./dashboard">Home</a>
48                      <a class="nav-link" href="./expenseanalysis">Expense Analysis</a>
49                      <a class="nav-link" href="./incomeanalysis">Income Analysis</a>
50                      <a class="nav-link" href="./profile">Profile</a>
51                  </div>
52              </div>
53          </div>
54      </nav>
55      <!-- End of Navbar -->
56
57      <!-- Bottom Navbar -->
58      <nav class="navbar navbar-light bg-light border-top navbar-expand d-md-none d-lg-none d-xl-none fixed-bottom">
59          <ul class="navbar-nav nav-justified w-100">
60              <li class="nav-item">
61                  <a href="./dashboard" class="nav-link active">
62                      <span>
63                          <i class="fas fa-home"></i>
64                      </span>
65                  </a>
66              </li>
67              <li class="nav-item">
68                  <a href="./expenseanalysis" class="nav-link">
69                      <span>
```

Show & Update Analysis – Show_Income.html



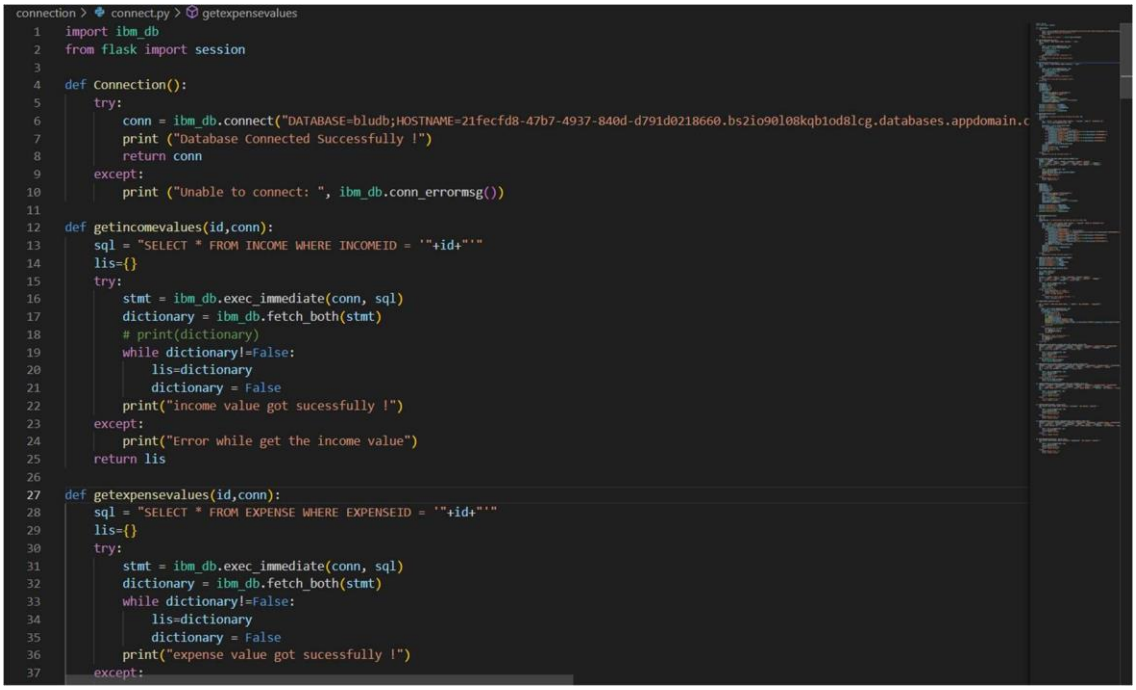
```
templates > Show_Income.html > ...
                        </div>
                        <div class="mb-3">
                            <label for="category_income" class="form-label">Income Category</label>
                            <select class="form-select" id="category_income" name="category_income">
                                <option value="Salery,salery-id" {% if data.INCOMECATEGORY=="Salery" %} selected="selected"{% endi
                                <option value="Profit,profit-id" {% if data.INCOMECATEGORY=="Profit" %} selected="selected"{% endi
                                <option value="Bonus,nonus-id" {% if data.INCOMECATEGORY=="Bonus" %} selected="selected"{% endif %
                                <option value="Other,other-id" {% if data.INCOMECATEGORY=="Other" %} selected="selected"{% endif %
                            </select>
                        </div>
                        <div class="mb-3">
                            <label for="note_income" class="form-label">Income Note</label>
                            <textarea class="form-control" id="note_income" name="note_income" rows="3"
                                required>{{data.INCOMENOTE}}</textarea>
                        </div>
                        <div class="mb-3">
                            <label for="budget_income" class="form-label">Income Budget</label>
                            <input type="number" value="{{data.INCOMEAMOUNT}}" class="form-control" id="budget_income" name="budge
                                required>
                        </div>
                        <div class="row">
                            <div class="col">
                                <div class="d-grid gap-2">
                                    <a href="dashboard" class="btn btn-light">Back</a>
                                </div>
                            </div>
                            <div class="col">
                                <div class="d-grid gap-2">
                                    <button class="btn btn-success" type="submit" name="submit">Update</button>
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
```

Show & Update Expense – Show_Expense.html



```
templates > Show_Expense.html > ...
            <div class=" col-xl-6 mt-5 mb-5">
                <div class="justify-content-center text-center">
                    {%if error %}
                    <p style="color: red;">{{error}}</p>
                    {% endif %}
                </div>
                <div class="card">
                    <div class="card-body">
                        <h3 class="card-title">Show Expense</h3>
                        <form class="mt-3" action="" action="{{url_for('updateexpense')}}" method="post">
                            <input type="hidden" value="{{data.EXPENSEID}}" name="expenseid">
                            <input type="hidden" value="{{data.CATOGERYNAME}}" name="catid">
                            <div class="mb-3">
                                <label for="date_expense" class="form-label">Expense Date</label>
                                <input type="date" value="{{data.EXPENSEDATE}}" class="form-control" id="date_expense" name="date_expe
                                    required>
                            </div>
                            <div class="mb-3">
                                <label for="category_expense" class="form-label">Expense Category</label>
                                <select class="form-select" id="category_expense" name="category_expense">
                                    <option value="Entertainment,movie-id"{% if data.EXPENSECATEGORY=="Entertainment" %} selected="sel
                                    <option value="Food,food-id" {% if data.EXPENSECATEGORY=="Food" %} selected="selected"{% endif %}>
                                    <option value="Fuel,fuel-id" {% if data.EXPENSECATEGORY=="Fuel" %} selected="selected"{% endif %}>
                                    <option value="other,other-id" {% if data.EXPENSECATEGORY=="Other" %} selected="selected"{% endif
                                </select>
                            </div>
                            <div class="mb-3">
                                <label for="note_expense" class="form-label">Expense Note</label>
                                <textarea class="form-control" id="note_expense" name="note_expense" rows="3"
                                    required>{{data.EXPENSENOTE}}</textarea>
                            </div>
                            <div class="mb-3">
                                <label for="budget_expense" class="form-label">Expense Budget</label>
                                <input type="number" value="{{data.EXPENSEAMOUNT}}" class="form-control" id="budget_expense" name="bud
                                    required>
                            </div>
                            <div class="row">
```

Backend file – connect.py



```
connection > connect.py > getexpensevalues
import ibm_db
from flask import session

def Connection():
    try:
        conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.c
        print ("Database Connected Successfully !")
        return conn
    except:
        print ("Unable to connect: ", ibm_db.conn_errormsg())

def getincomevalues(id,conn):
    sql = "SELECT * FROM INCOME WHERE INCOMEID = '"+id+"'"
    lis={}
    try:
        stmt = ibm_db.exec_immediate(conn, sql)
        dictionary = ibm_db.fetch_both(stmt)
        # print(dictionary)
        while dictionary!=False:
            lis=dictionary
            dictionary = False
        print("income value got sucessfully !")
    except:
        print("Error while get the income value")
    return lis

def getexpensevalues(id,conn):
    sql = "SELECT * FROM EXPENSE WHERE EXPENSEID = '"+id+"'"
    lis={}
    try:
        stmt = ibm_db.exec_immediate(conn, sql)
        dictionary = ibm_db.fetch_both(stmt)
        while dictionary!=False:
            lis=dictionary
            dictionary = False
        print("expense value got sucessfully !")
    except:
```

Docker file

Requirements.txt