Project Development Phase

Sprint – 4

Team ID	PNT2022TMID14045
Project Name	Personal Expense Tracker Application

Edit Income Page:

Edit Expense Page:

Code:

Base file - *HomeBase.html*

```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*

```html
templates > <> Show_Income.html > ...
29                            </div>
30                            <div class="mb-3">
31                                <label for="category_income" class="form-label">Income Category</label>
32                                <select class="form-select" id="category_income" name="category_income">
33                                    <option value="Salery,salery-id" {% if data.INCOMECATEGORY=="Salery" %} selected="selected"{% endi
34                                    <option value="Profit,profit-id" {% if data.INCOMECATEGORY=="Profit" %} selected="selected"{% endi
35                                    <option value="Bonus,nonus-id" {% if data.INCOMECATEGORY=="Bonus" %} selected="selected"{% endif %
36                                    <option value="Other,other-id" {% if data.INCOMECATEGORY=="Other" %} selected="selected"{% endif %
37                                </select>
38                            </div>
39                            <div class="mb-3">
40                                <label for="note_income" class="form-label">Income Note</label>
41                                <textarea class="form-control" id="note_income" name="note_income" rows="3"
42                                    required>{{data.INCOMENOTE}}</textarea>
43                            </div>
44                            <div class="mb-3">
45                                <label for="budget_income" class="form-label">Income Budget</label>
46                                <input type="number" value="{{data.INCOMEAMOUNT}}" class="form-control" id="budget_income" name="budge
47                                    required>
48                            </div>
49                            <div class="row">
50                                <div class="col">
51                                    <div class="d-grid gap-2">
52                                        <a href="dashboard" class="btn btn-light">Back</a>
53                                    </div>
54                                </div>
55                                <div class="col">
56                                    <div class="d-grid gap-2">
57                                        <button class="btn btn-success" type="submit" name="submit">Update</button>
58                                    </div>
59                                </div>
60                            </div>
61                        </form>
62                    </div>
63                </div>
64            </div>
65        </div>
```

Show & Update Expense – *Show_Expense.html*

```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*

```python
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

```
Dockerfile > ...
1    FROM python:latest
2    WORKDIR /app
3    COPY requirements.txt ./
4    RUN pip install -r requirements.txt
5    COPY . .
6    EXPOSE 5000
7    CMD ["python","./app.py"]
```
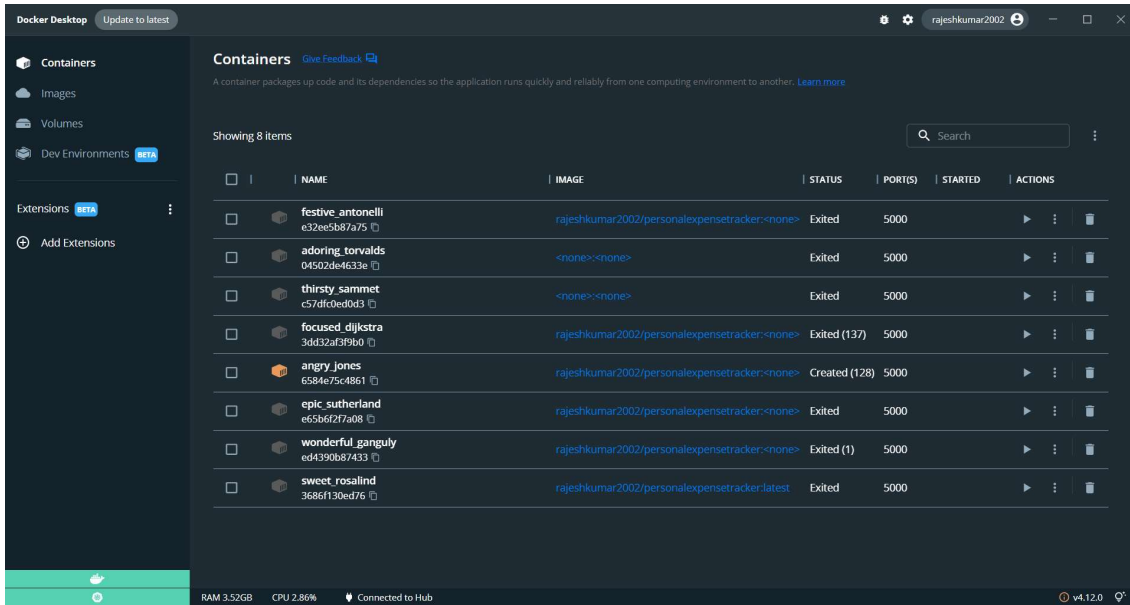
Requirements.txt

```
requirements.txt
1    flask
2    ibm_db
3    sendgrid
4    flask_session
```

Kubernetes - *Kubernetes.yaml*

```yaml
kubernetes.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app

spec:
  replicas: 3
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app

    spec:
      containers:
        - name: repo2
          image: docker.io/rajeshkumar2002/personalexpensetracker
          ports:
            - containerPort: 5000
              protocol: TCP
```

Docker :

Docker Desktop:

Docker – Personal Expense Tracker:

IBM Kubernetes:

Kubernetes Shell:

Kubernetes Dashboard:

http://169.51.194.34:31794/

Hosted Application: