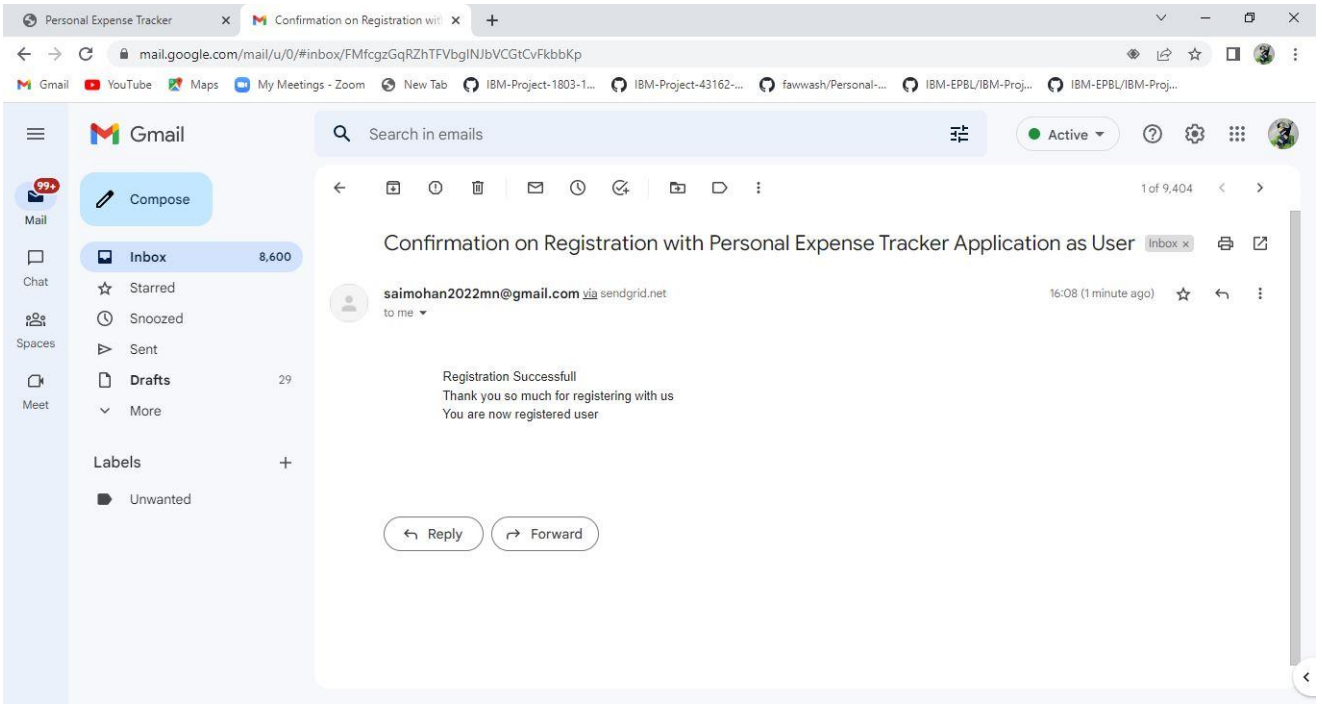
PROJECT DEPLOYMENT PHASE

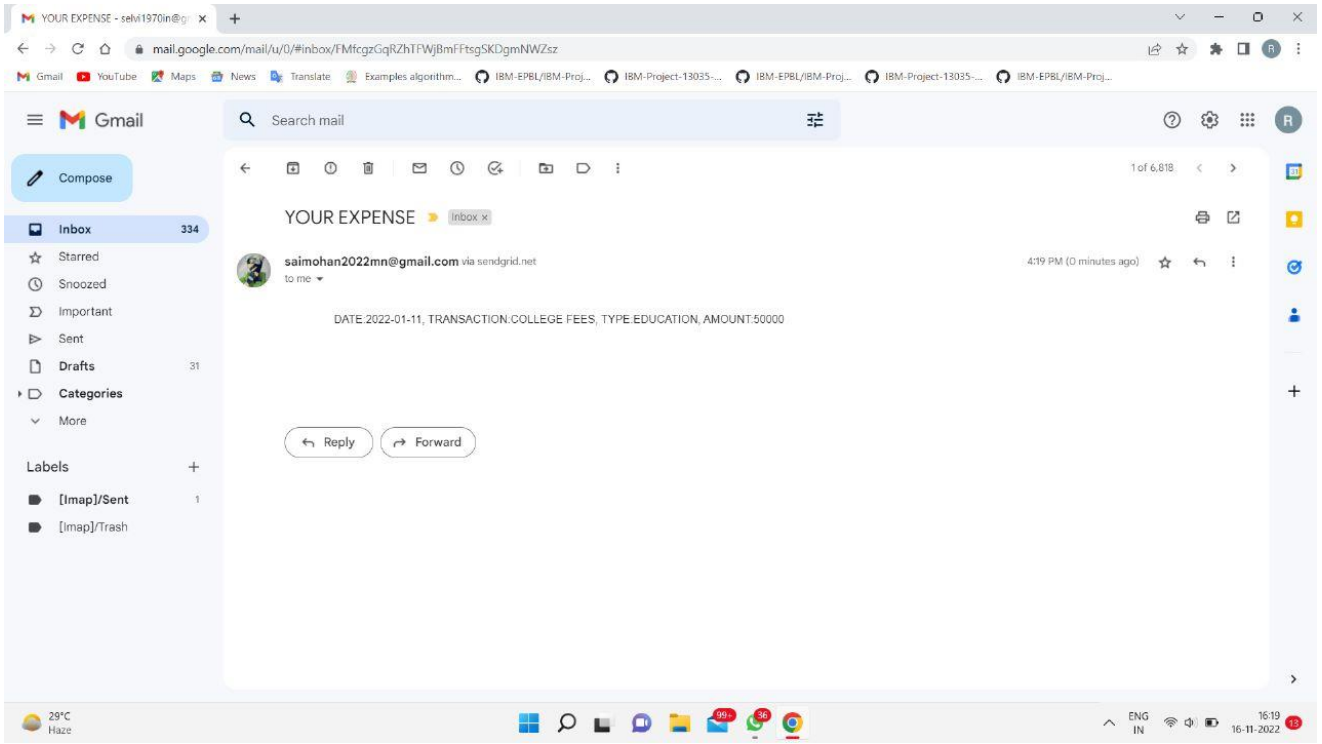
SPRINT - III :

Team Id : PNT2022TMID26966
Project Name : PERSONAL EXPENSE TRACKER

REGISTRATION MAIL :



EXPENSE MAIL :



Source Code :

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
from sendmail import sendMailUsingSendGrid
from dotenv import load_dotenv
import configparser
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
config=configparser.ConfigParser()
config.read("config.ini")

load_dotenv()

app = Flask(__name__)
app.secret_key = "ibm"



conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-
a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURITY=SSL;S
SLServerCertificate=DigiCertGlobalRootCA.crt;UID=myz74370;PWD=B73dOVDHJ6nuSg33",'','
)



message = ""


@app.route('/', methods=['GET', 'POST'])
def home():
    print(session)
    print("Message - " + message)
    if session:
        if session["loggedin"]:
            return redirect(url_for('tracker'))
    else:
        login_page = True
        print(request.values.get('page'))
        if request.values.get('page') == "register":
            login_page = False
        return render_template('index.html', login=login_page, message=message)


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == "POST":
        global message


        user = request.form
```

```
        print(user)
        email = user["email"]
        password = user["password"]

        print("Email - " + email + ", Password - " + password)

        sql = "SELECT * FROM users WHERE email = ? AND password = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)
        print("Account - ")
        print(account)

        if account:
            session['loggedin'] = True
            session['id'] = account['EMAIL']
            user_email = account['EMAIL']
            session['email'] = account['EMAIL']
            session['name'] = account['NAME']

            return redirect(url_for('tracker'))

        else:
            message = "Incorrect Email or Password"
            return redirect(url_for('home'))


@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == "POST":
        global message

        user = request.form
        print(user)
        name = user["name"]
        email = user["email"]
        password = user["password"]

        sql = "SELECT * FROM USERS WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)
        print("Account - ", end="")
        print(account)

        if account:
            message = "Account already exists"
            return redirect(url_for('home', page="register"))
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            message = "Invalid email address"
            return redirect(url_for('home', page="register"))
```

```
        elif not re.match(r'[A-Za-z0-9]+', name):
            message = "Name must contain only characters and numbers"
            return redirect(url_for('home', page="register"))
        else:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)

            session['loggedin'] = True
            session['id'] = email
            user_email = email
            session['email'] = email
            session['name'] = name

            message = ""




            settings=config["SETTINGS"]




            API= settings.get("APIKEY", None)
            from_email = settings.get("FROM", None)
            to_email = email
            subject = "Confirmation on Registration with Personal Expense Tracker
Application as User"
            html_content ='''

                 Registration Successfull
                 Thank you so much for registering with us
                 You are now registered user

                '''


            sendMailUsingSendGrid(API, from_email, to_email, subject,html_content)


            return redirect(url_for('tracker'))


@app.route('/tracker')
def tracker():
    global message
    data = []
    expenses = {"Medical Expenses": 0, "House Expenses": 0, "Education": 0,
"Savings": 0, "Others": 0}

    if session:
```

```
        if session["loggedin"]:
            sql = "SELECT date, transaction, type, amount FROM TRANSACTIONS WHERE
email = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, session["email"])
            ibm_db.execute(stmt)

            row = ibm_db.fetch_assoc(stmt)
            while row:
                data.append(row)
                expenses[row["TYPE"]] += row["AMOUNT"]
                row = ibm_db.fetch_assoc(stmt)

            print(data)
            print(expenses)

            message = ""

            return render_template('home.html', name=session['name'], data=data[::-
1], expenses=expenses)
    else:
        message = "Session Expired"
    return redirect(url_for("home"))


@app.route('/add-expenditure', methods=['GET', 'POST'])
def add_expenditure():
    if request.method == "POST":
        details = request.form
        print(details)

        date = details["date"][-2:] + "/" + details["date"][5:7] + "/" +
details["date"][:4]
        transaction = details["transaction"]
        type = details["type"]
        amount = details["amount"]
        print(date, transaction, type, amount)

        sql = "INSERT INTO transactions VALUES (?, ?, ?, ?, ?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, date)
        ibm_db.bind_param(stmt, 2, transaction)
        ibm_db.bind_param(stmt, 3, type)
        ibm_db.bind_param(stmt, 4, amount)
        ibm_db.bind_param(stmt, 5, session["email"])


        ibm_db.execute(stmt)

        settings=config["SETTINGS"]




        API= settings.get("APIKEY", None)
```

```
        from_email = settings.get("FROM", None)
        to_email =  settings.get("TO", None)
        subject = "Confirmation on Registration with Personal Expense Tracker
Application as User"
        html_content
=["date:",date,"transaction:",transaction,"type:",type,"amount:",amount]


        sendMailUsingSendGrid(API, from_email, to_email, subject,html_content)

        return redirect(url_for('tracker'))


@app.route('/logout')
def logout():
    print("Logging Out")
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('email', None)
    session.pop('name', None)
    return redirect(url_for('home'))


if __name__ == '__main__':
    app.run(debug=True)
```