

PROJECT DEPLOYMENT PHASE

SPRINT - II :

Team Id : PNT2022TMID26966

Project Name : PERSONAL EXPENSE TRACKER

TABLES :

The screenshot shows the IBM Db2 on Cloud web interface. The 'Tables' tab is selected in the top navigation bar. On the left, the 'Schemas' panel shows a table with columns 'Name', 'Type', and 'Tables'. It lists one schema: 'MYZ74370' of type 'User' with 3 tables. On the right, the 'Tables' panel shows a table with columns 'Name', 'Schema', and 'Properties'. It lists three tables: 'EXPENDITURE', 'TRANSACTIONS', and 'USERS', all under the 'MYZ74370' schema. A 'New table' button is visible in the top right of the 'Tables' panel.

Name	Type	Tables
MYZ74370	User	3

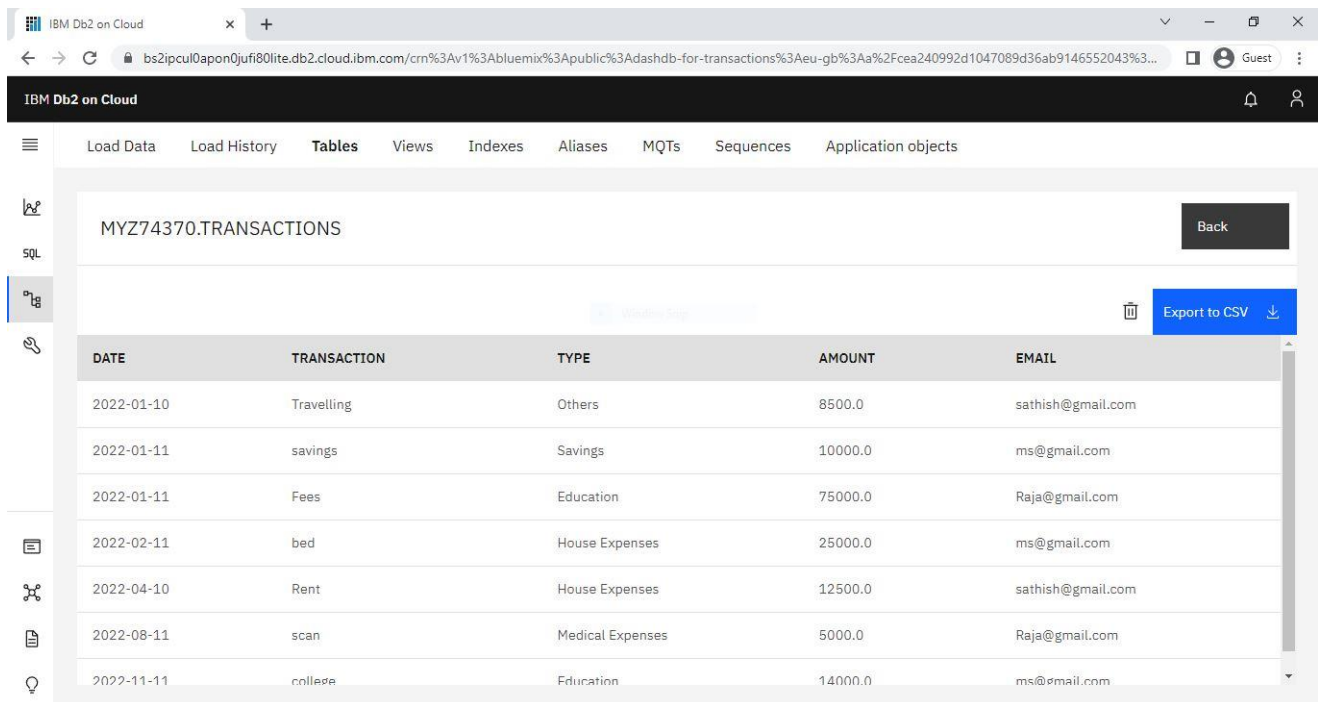
Name	Schema	Properties
EXPENDITURE	MYZ74370	...
TRANSACTIONS	MYZ74370	...
USERS	MYZ74370	...

USER TABLE :

The screenshot shows the IBM Db2 on Cloud web interface with the 'MYZ74370.USERS' table selected. The table has three columns: 'NAME', 'EMAIL', and 'PASSWORD'. It contains three rows of user data. A 'Back' button is in the top right, and an 'Export to CSV' button is in the bottom right.

NAME	EMAIL	PASSWORD
Rajagopalan	Raja@gmail.com	raja
Sathish	sathish@gmail.com	sathish
sai	ms@gmail.com	sai

TRANSACTION TABLE :



The screenshot shows the IBM Db2 on Cloud web interface. The top navigation bar includes 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected, displaying a table named 'MYZ74370.TRANSACTIONS'. The table has five columns: 'DATE', 'TRANSACTION', 'TYPE', 'AMOUNT', and 'EMAIL'. The data is as follows:

DATE	TRANSACTION	TYPE	AMOUNT	EMAIL
2022-01-10	Travelling	Others	8500.0	sathish@gmail.com
2022-01-11	savings	Savings	10000.0	ms@gmail.com
2022-01-11	Fees	Education	75000.0	Raja@gmail.com
2022-02-11	bed	House Expenses	25000.0	ms@gmail.com
2022-04-10	Rent	House Expenses	12500.0	sathish@gmail.com
2022-08-11	scan	Medical Expenses	5000.0	Raja@gmail.com
2022-11-11	college	Education	14000.0	ms@gmail.com

SOURCE CODE :

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re

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=B73d0VDHJ6nuSg33",'',''
)

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(prepare_stmt, 1, name)
    ibm_db.bind_param(prepare_stmt, 2, email)
    ibm_db.bind_param(prepare_stmt, 3, password)
    ibm_db.execute(prepare_stmt)

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

    message = ""

    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)

        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)

```