**Hexaware Foundation Training**

**Python Case Study Report**

**Finance Management System**

**NAME:** Eswara Venkata Sai Raja

**DATE:** 14-10-2024

**1)Sql Schema:**

CREATE DATABASE Finance\_Mgmt;

USE Finance\_Mgmt;

CREATE TABLE Users (

user\_id INT PRIMARY KEY IDENTITY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL

);

CREATE TABLE ExpenseCategories (

category\_id INT PRIMARY KEY IDENTITY,

category\_name VARCHAR(50) NOT NULL

);

CREATE TABLE Expenses (

expense\_id INT PRIMARY KEY IDENTITY,

user\_id INT,

amount DECIMAL(10, 2) NOT NULL,

category\_id INT,

date DATE NOT NULL,

description VARCHAR(255),

FOREIGN KEY (user\_id) REFERENCES Users(user\_id),

FOREIGN KEY (category\_id) REFERENCES ExpenseCategories(category\_id)

);

INSERT INTO ExpenseCategories (category\_name)

VALUES ('Food'), ('Transportation'), ('Utilities'),('Investments');

INSERT INTO ExpenseCategories (category\_name)

VALUES

('Entertainment'),

('Healthcare'),

('Insurance'),

('Education'),

('Clothing'),

('Groceries'),

('Dining Out'),

('Subscriptions'),

('Travel'),

('Home Maintenance'),

('Personal Care'),

('Gifts'),

('Charity'),

('Pet Care'),

('Savings'),

('Debt Payments'),

('Fitness'),

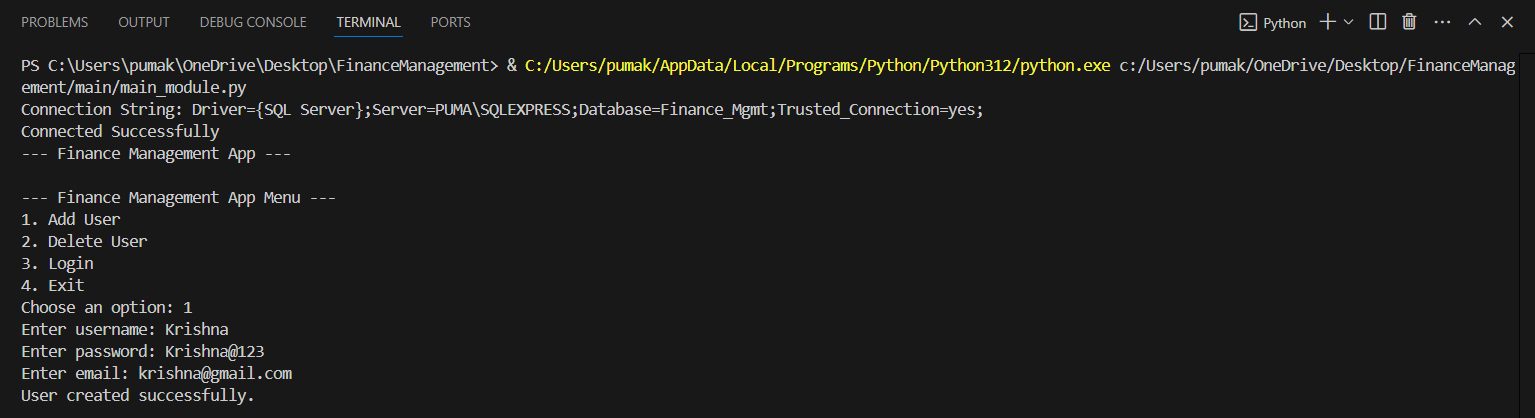
('Electronics'),

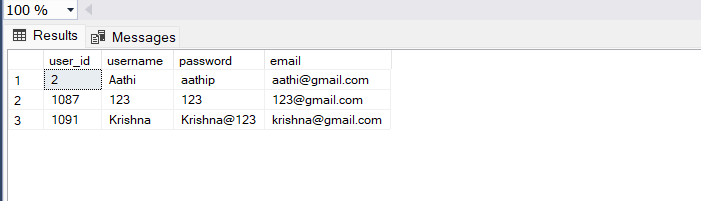
('Rent'),

('Taxes');

**2)Output Screenshots:**

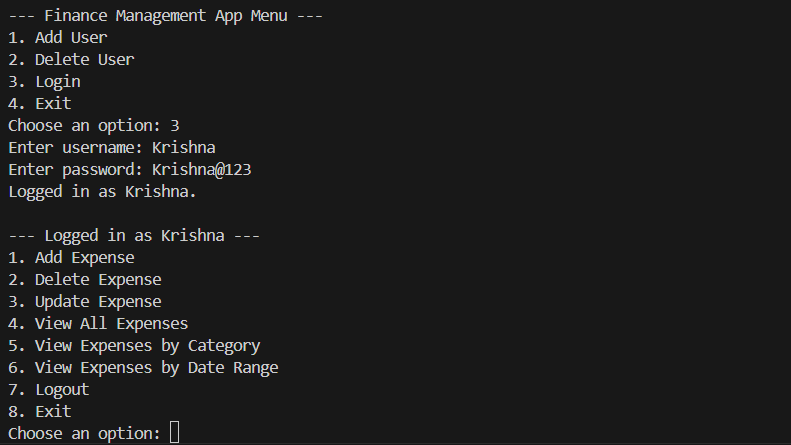
**1.Add User:**

****

****

*The user named ‘Krishna’ has been added to the database.*

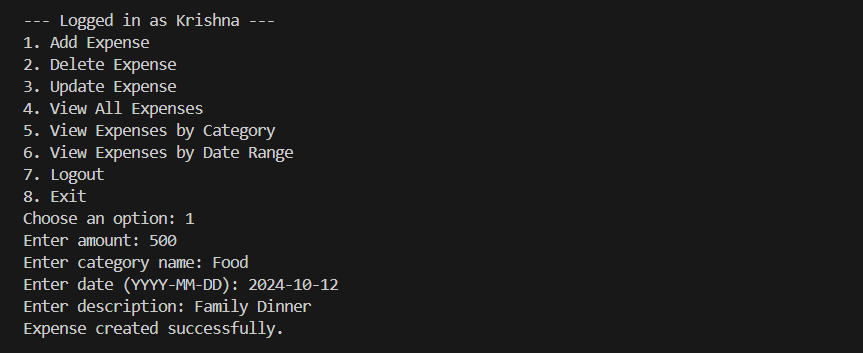
**2. Login User:**

****

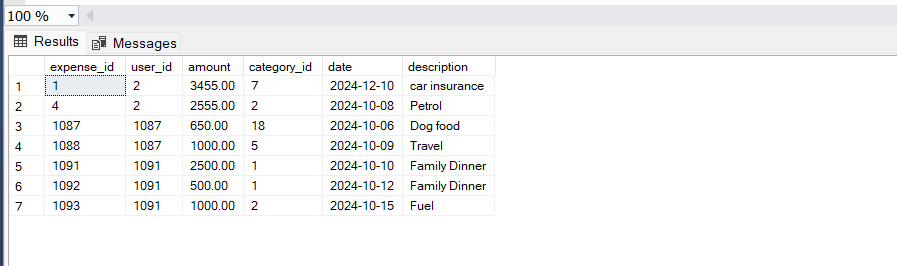
*The user got logged in and he can use the below displayed menu.*

**3.Add Expense:**

****

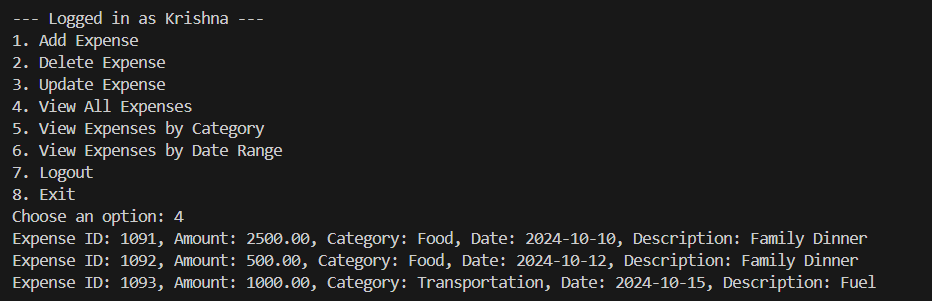
****

****

****

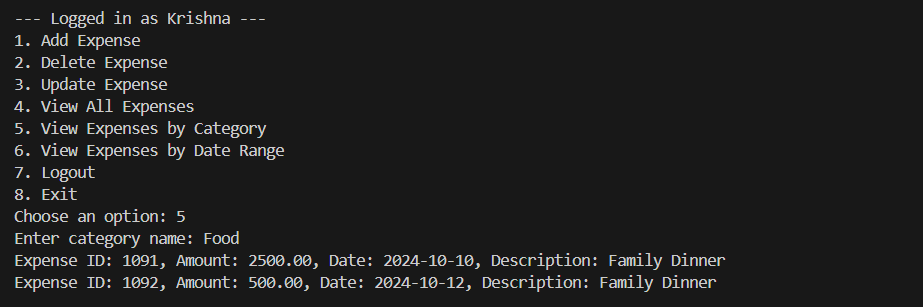
*All three expenses have been added to the Expenses table.*

**4. View All Expenses:**

****

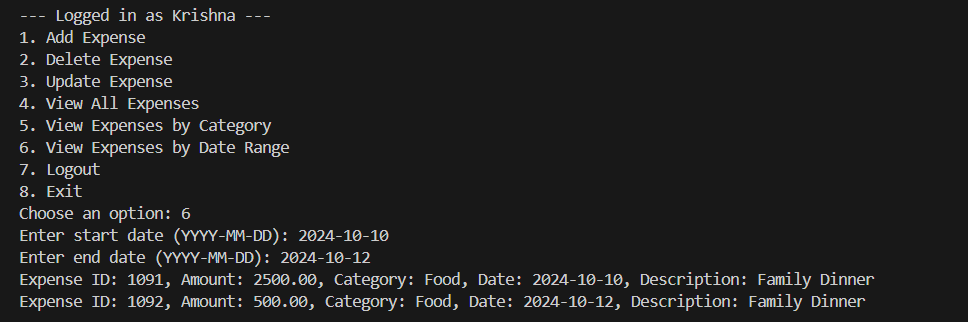
*The user gets to view all the expenses he has made.*

**5. View Expenses by Category:**

****

*Expenses under the category* ***‘Food’*** *is displayed to the user here.*

**6. View Expenses by Date Range:**

****

*Here the user enters dates from* ***2024-10-10 to 2024-10-12*** *to get the expense made between those two dates and the user gets the expenses made.*

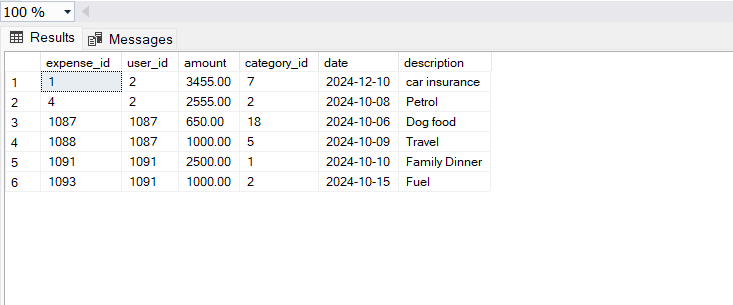
**7.Update Expense:**

****

*Here the updated Expense of the user is successful and can be verified here.*

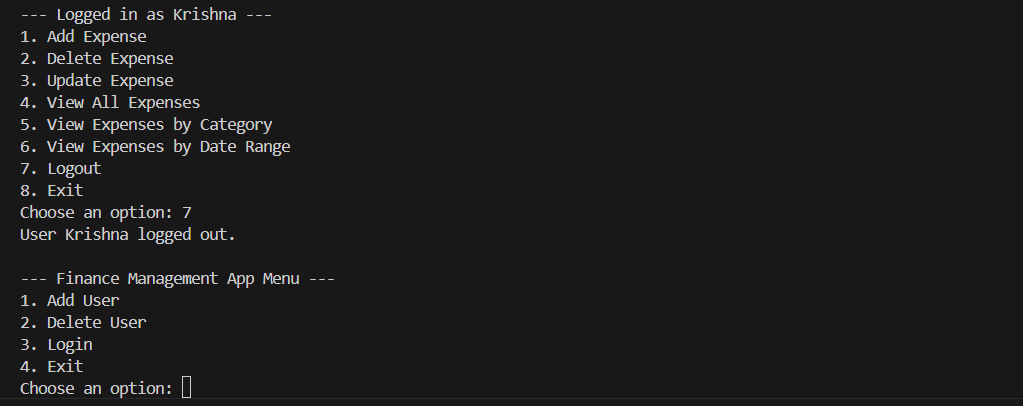
**8. Delete Expense:**

****

****

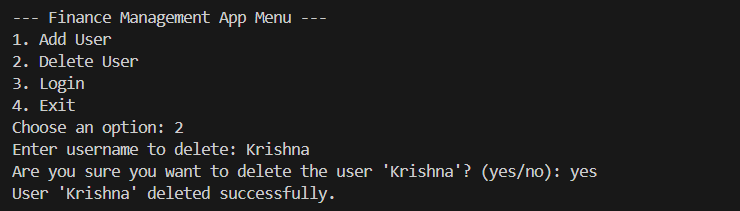
*The expense has been successfully deleted here and is verified.*

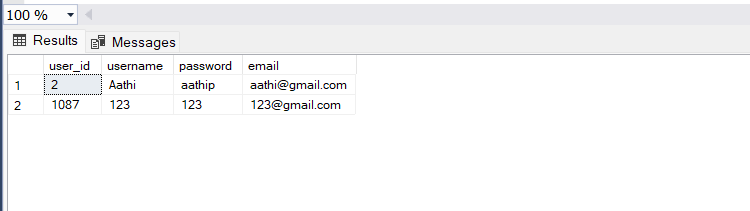
**9. Logout:**

****

*The user has been logged out successfully.*

**10.Delete User:**

****

****

*The user has been deleted successfully.*

**3) Exception Handling:**

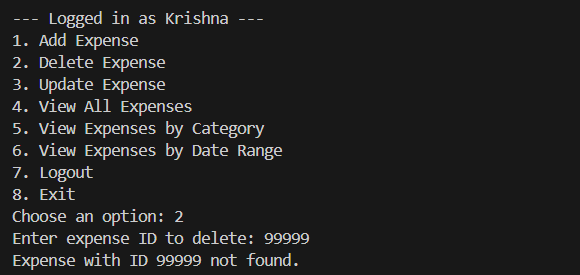
**1. ExpenseNotFoundException:**

class ExpenseNotFoundException(Exception):

    def \_\_init\_\_(self, message="Expense not found"):

        self.message = message

        super().\_\_init\_\_(self.message)

****

*Expection Id is not found.*

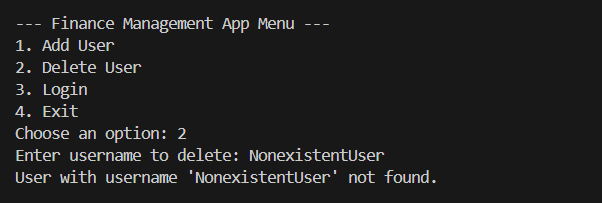
**2.UserNotFoundException:**

class UserNotFoundException(Exception):

    def \_\_init\_\_(self, message="User not found"):

        self.message = message

        super().\_\_init\_\_(self.message)

****

*NonExistentUser is not found.*

**3) Testing:**

**1.Test User Creation:**

import sys

import os

base\_dir = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))

sys.path.append(base\_dir)

import unittest

from dao.FinanceRepositoryImpl import FinanceRepositoryImpl

from entity.User import User

class TestUserCreation(unittest.TestCase):

    def setUp(self):

        self.repo = FinanceRepositoryImpl()

    def test\_user\_creation(self):

        new\_user = User(username="test\_user", password="password123", email="test\_user@example.com")

        success = self.repo.create\_user(new\_user)

        fetched\_user = self.repo.get\_user\_by\_username("test\_user")

        self.assertTrue(success)

        self.assertIsNotNone(fetched\_user)

        self.assertEqual(fetched\_user.get\_username(), "test\_user")

    def tearDown(self):

        test\_user = self.repo.get\_user\_by\_username("test\_user")

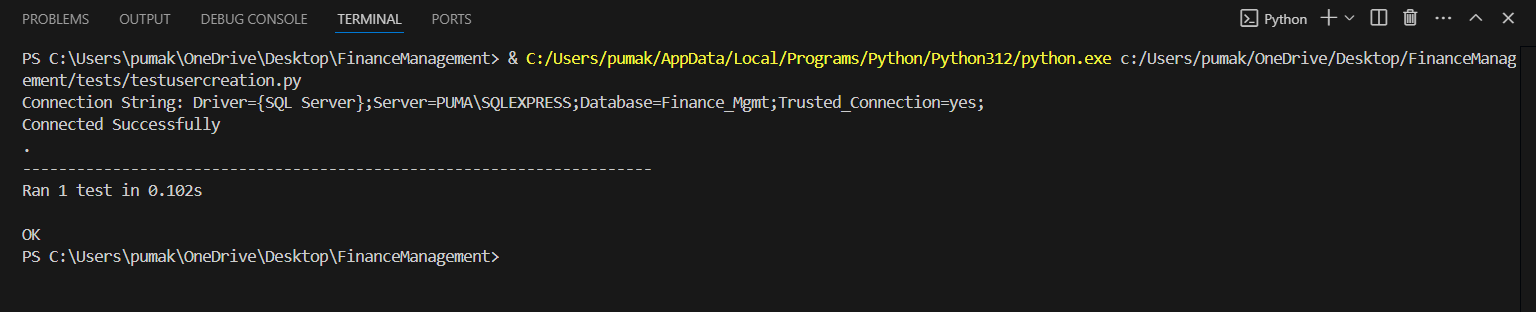
        if test\_user:

            self.repo.delete\_user(test\_user.get\_user\_id())

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

**Output:**

****

**2.Test Search Expense:**

import sys

import os

base\_dir = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))

sys.path.append(base\_dir)

import unittest

from entity.Expense import Expense

from entity.User import User

from dao.FinanceRepositoryImpl import FinanceRepositoryImpl

class TestSearchExpense(unittest.TestCase):

    def setUp(self):

        self.repo = FinanceRepositoryImpl()

        self.test\_user = User(username="search\_user", password="password123", email="search\_user@example.com")

        self.repo.create\_user(self.test\_user)

        self.test\_user = self.repo.get\_user\_by\_username("search\_user")

        category\_id = self.repo.get\_category\_id\_by\_name("Utilities")

        self.expense = Expense(user\_id=self.test\_user.get\_user\_id(), amount=50, category\_id=category\_id, date="2024-10-11", description="Electricity Bill")

        self.repo.create\_expense(self.expense)

    def test\_search\_expense(self):

        expenses = self.repo.get\_all\_expenses(self.test\_user.get\_user\_id())

        self.assertEqual(len(expenses), 1)

        self.assertEqual(expenses[0].get\_description(), "Electricity Bill")

    def tearDown(self):

        expenses = self.repo.get\_all\_expenses(self.test\_user.get\_user\_id())

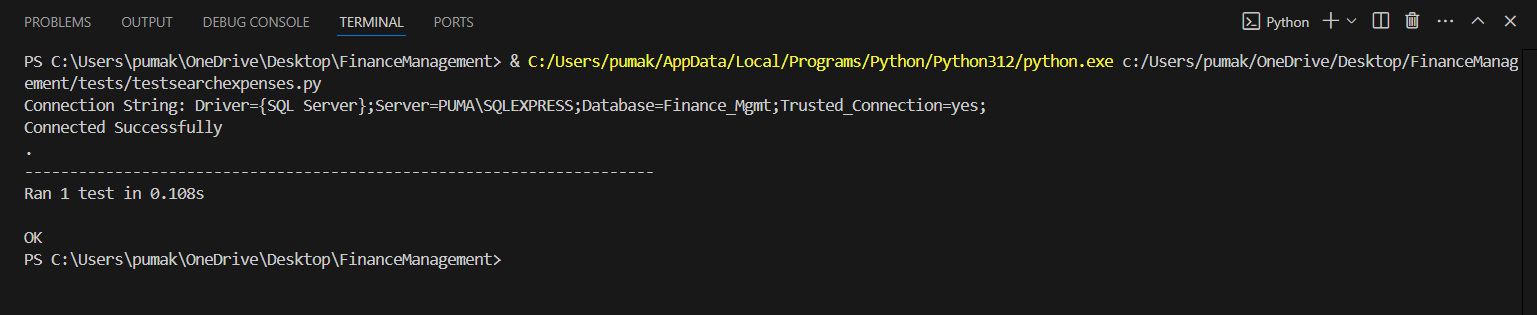
        for expense in expenses:

            self.repo.delete\_expense(expense.get\_expense\_id())

        self.repo.delete\_user(self.test\_user.get\_user\_id())

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

****

**3.Test Expense Creation:**

import sys

import os

base\_dir = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))

sys.path.append(base\_dir)

import unittest

from entity.Expense import Expense

from entity.User import User

from dao.FinanceRepositoryImpl import FinanceRepositoryImpl

class TestExpenseCreation(unittest.TestCase):

    def setUp(self):

        self.repo = FinanceRepositoryImpl()

        self.test\_user = User(username="expense\_user", password="password123", email="expense\_user@example.com")

        self.repo.create\_user(self.test\_user)

        self.test\_user = self.repo.get\_user\_by\_username("expense\_user")

    def test\_expense\_creation(self):

        category\_id = self.repo.get\_category\_id\_by\_name("Food")

        new\_expense = Expense(user\_id=self.test\_user.get\_user\_id(), amount=1000, category\_id=category\_id, date="2024-10-11", description="Dinner")

        success = self.repo.create\_expense(new\_expense)

        expenses = self.repo.get\_all\_expenses(self.test\_user.get\_user\_id())

        self.assertTrue(success)

        self.assertGreater(len(expenses), 0)

        self.assertEqual(expenses[0].get\_description(), "Dinner")

    def tearDown(self):

        expenses = self.repo.get\_all\_expenses(self.test\_user.get\_user\_id())

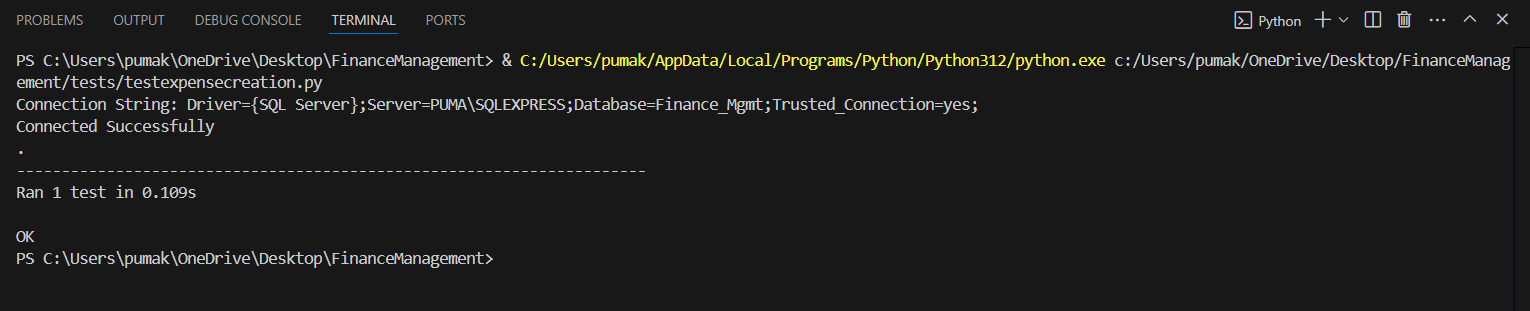
        for expense in expenses:

            self.repo.delete\_expense(expense.get\_expense\_id())

        self.repo.delete\_user(self.test\_user.get\_user\_id())

if \_\_name\_\_ == "\_\_main\_\_":

    unittest.main()

**

**4.Test Exceptions:**

import sys

import os

base\_dir = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))

sys.path.append(base\_dir)

import unittest

from dao.FinanceRepositoryImpl import FinanceRepositoryImpl

from exception.UserNotFoundException import UserNotFoundException

from exception.ExpenseNotFoundException import ExpenseNotFoundException

base\_dir = os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), ".."))

sys.path.append(base\_dir)

class TestExceptions(unittest.TestCase):

    def setUp(self):

        self.repo = FinanceRepositoryImpl()

    def test\_user\_not\_found\_exception(self):

        with self.assertRaises(UserNotFoundException):

            self.repo.get\_user\_by\_username("non\_existent\_user")

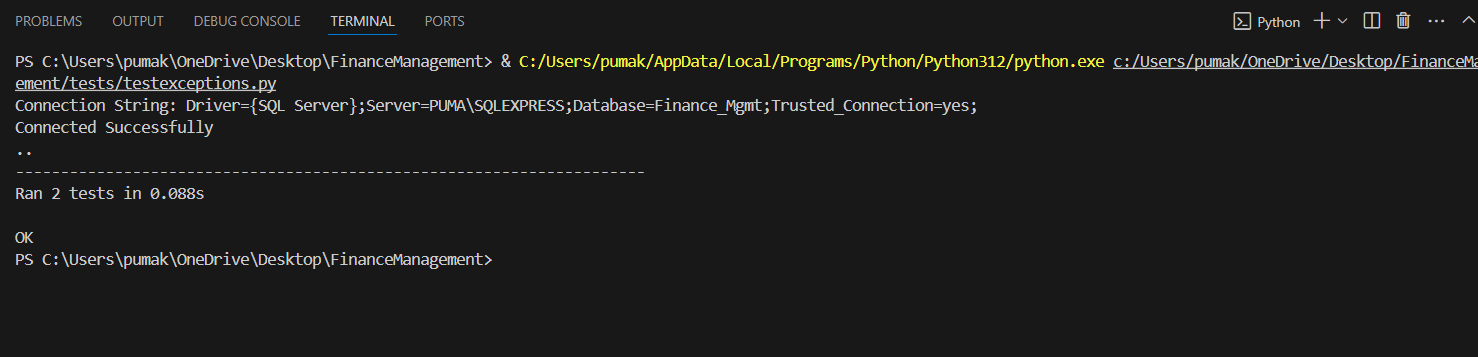
    def test\_expense\_not\_found\_exception(self):

        with self.assertRaises(ExpenseNotFoundException):

            self.repo.delete\_expense(99999)

if \_\_name\_\_ == '\_\_main\_\_':

    unittest.main()

****