Assignment 2 – Expense Tracker Using Python

```
import random


class Expense:
    def __init__(self, expense_id, date, category, description,
amount):
        self.expense_id = expense_id
        self.date = date
        self.category = category
        self.description = description
        self.amount = amount

    def __str__(self):
        return f"expense_id:{self.expense_id}, date:{self.date},
category:{self.category}, description:{self.description}, amount:
{self.amount}"



# 3.
# empty list to store expenses
expenses = []

# function to add new expenses
def add_expense(expense):
    expenses.append(expense)

# function to update existing expenses
def update_expense(expense_id, new_expense):
    for item in expenses:
        if item.expense_id == expense_id:
            item.date = new_expense.date
            item.category = new_expense.category
            item.description = new_expense.description
            item.amount = new_expense.amount

# function to delete some existing expense
def delete_expense(expense_id):
    found = False
    for item in expenses:
        if item.expense_id == expense_id:
            expenses.remove(item)
            found = True
    if found==False:
        print("Specified expense_id not found")
    else:
        print("Expense deleted")
```

```
# function to display all expenses
def display_expenses():
    print("ID\tDATE\tCATEGORY\tDESCRIPTION\tAMOUNT")
    for item in expenses:
        print(item)



# 4. - dictionary storing username and password for authentication
users = {'user1':'password1', 'user2':'password2'}

# function to perform authentication
def authenticate_user(username, password):
    if username not in users:
        print("User not found")
        return False

    if users[username] == password:
        print("Successfully authenticated")
        return True
    else:
        print("Incorrect password")
        return False



# 5. function to split expenses based on category
def categorize_expenses():
    categories = {}
    for expense in expenses:
        category_temp = (expense.category).lower()
        if category_temp in categories:
            categories[category_temp] += float(expense.amount)
        else:
            categories[category_temp] = float(expense.amount)
    return categories


def summarize_expenses():
    total = 0
    for expense in expenses:
        total += float(expense.amount)
    return total



# 6. functino to find total expense
def calculate_total_expenses():
    total_sum = summarize_expenses()
    return total_sum
```

```
# function to create a summary
def generate_summary_report():
    category_wise_expense = categorize_expenses()
    print("Category wise expense:")
    for item, amount in category_wise_expense.items():
        print(f"{item}: {amount}")
    print(f"Sum of all expenses: {calculate_total_expenses()}")



# generate a unique 3-digit expense ID
def generate_expense_id():
    new_unique_id = 0
    while True:
        new_unique_id = random.randint(100, 999)
        new_unique_id_possible = True
        for expense in expenses:
            if new_unique_id == expense.expense_id:
                new_unique_id_possible = False
                break
        if new_unique_id_possible:
            return new_unique_id


# 7 - function that creates interface for user to interact with
def cli():
    print("Options:")
    print("1. Add new expense")
    print("2. Update existing expense")
    print("3. Delete an expense")
    print("4. Display all expenses")
    print("5. Generate summary report")
    print("6. Exit application")

    while True:
        user_choice = int(input("\nPlease input choice: "))
        if user_choice == 1:
            expense_id_new_input = generate_expense_id()
            date_new_input = input("Enter date in DD/MM/YYYY: ")
            category_new_input = input("Enter new category: ")
            description_new_input = input("Enter description: ")
            amount_new_input = float(input("Enter amount [float
datatype]: "))
            new_expense = Expense(expense_id_new_input,
date_new_input, category_new_input, description_new_input,
amount_new_input)
            add_expense(new_expense)
            print("New expense added")
        elif user_choice == 2:
```

```
            expense_id_to_modify = int(input("Enter expense id to
modify: "))
            date_modify_input = input("Enter date in DD/MM/YYYY: ")
            category_modify_input = input("Enter new category: ")
            description_modify_input = input("Enter description: ")
            amount_modify_input = float(input("Enter amount [float
datatype]: "))
            modified_expense = Expense(expense_id_to_modify,
date_modify_input, category_modify_input, description_modify_input,
amount_modify_input)
            update_expense(expense_id_to_modify, modified_expense)
        elif user_choice == 3:
            expense_id_to_delete = int(input("Enter expense_id to be
deleted: "))
            delete_expense(expense_id_to_delete)
        elif user_choice == 4:
            display_expenses()
        elif user_choice == 5:
            generate_summary_report()
        elif user_choice == 6:
            print("Exiting application")
            return
        else:
            print("Incorrect choice, retry!!")



# main code, entry point into program
username_input = input("Enter username: ")
password_input = input("Enter password: ")
if authenticate_user(username_input, password_input):
    cli()

Enter username:  user1
Enter password:  password1

Successfully authenticated
Options:
1. Add new expense
2. Update existing expense
3. Delete an expense
4. Display all expenses
5. Generate summary report
6. Exit application


Please input choice:  1
Enter date in DD/MM/YYYY:  06/04/2024
Enter new category:  food
```

```
Enter description:  pizza
Enter amount [float datatype]:  563

New expense added


Please input choice:  1
Enter date in DD/MM/YYYY:  07/04/2024
Enter new category:  clothes
Enter description:  shirt
Enter amount [float datatype]:  673

New expense added


Please input choice:  1
Enter date in DD/MM/YYYY:  07/04/2024
Enter new category:  clothes
Enter description:  pants
Enter amount [float datatype]:  590

New expense added


Please input choice:  4

ID    DATE  CATEGORY   DESCRIPTION     AMOUNT
expense_id:397, date:06/04/2024, category:food, description:pizza,
amount:563.0
expense_id:970, date:07/04/2024, category:clothes, description:shirt,
amount:673.0
expense_id:198, date:07/04/2024, category:clothes, description:pants,
amount:590.0


Please input choice:  5

Category wise expense:
food: 563.0
clothes: 1263.0
Sum of all expenses: 1826.0


Please input choice:  6

Exiting application
```