# AltSchool of Data Engineering Tinyuka 2024 First Semester Project Exam

## Introduction

The goal of this project is to assess your understanding of **object-oriented programming** (OOP) concepts in Python. You will be tasked with implementing two classes, **Expense** and **ExpenseDatabase**, to model and manage financial expenses.

The project will test your proficiency in defining classes, utilizing class attributes and methods,and handling time-related functionalities.

## Classes Overview:

### Expense Class:

Represents an individual financial expense.

**Attributes**:

1. **id**: A unique identifier generated as a UUID string.
2. **title**: A string representing the title of the expense.
3. **amount**: A float representing the amount of the expense.

4. **created_at**: A timestamp indicating when the expense was created (UTC).
5. **updated_at**: A timestamp indicating the last time the expense was updated (UTC).

**Methods:**

1. **__init__**: Initializes the attributes.
2. **update**: Allows updating the title and/or amount, updating the updated_at timestamp.
3. **to_dict**: Returns a dictionary representation of the expense.

## ExpenseDB class

Manages a collection of Expense objects.

**Attributes:**

1. **expenses:** A list storing Expense instances.

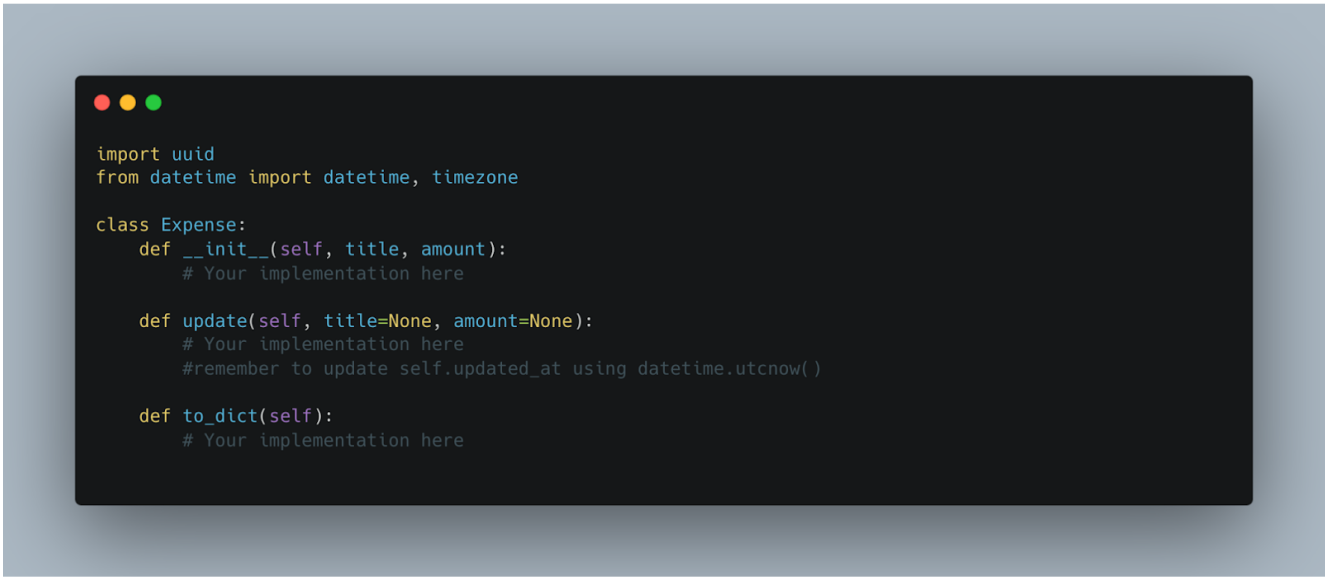
Methods:

1. **__init__**: Initializes the list.
2. **add_expense**: Adds an expense.
3. **remove_expense**: Removes an expense.
4. **get_expense_by_id**: Retrieves an expense by ID.
5. **get_expense_by_title**: Retrieves expenses by title.
6. **to_dict**: Returns a list of dictionaries representing expenses.

# Instructions

## Expense class

1. Implement an **__init__** method to initialize the attributes.
2. Implement an **update** method that allows updating the title and/or amount of the expense. The updated_at attribute should be automatically set to the current UTC timestamp whenever an update occurs.
3. Implement a **to_dict** method that returns a dictionary representation of the expense.



```python
import uuid
from datetime import datetime, timezone

class Expense:
    def __init__(self, title, amount):
        # Your implementation here

    def update(self, title=None, amount=None):
        # Your implementation here
        #remember to update self.updated_at using datetime.utcnow()

    def to_dict(self):
        # Your implementation here
```

## Expense Database class

1. Implement an __init__ method to initialize the expenses list.
2. Implement methods to:
    1. Add an expense to the database.
    2. Remove an expense from the database.

    3. Get an expense by ID.
    4. Get expenses by title (returning a list).
3. Create a **to_dict** method that returns a list of dictionaries representing each expense in the database.

```python
class ExpenseDatabase:
    def __init__(self):
        # Your implementation here

    def add_expense(self, expense):
        # Your implementation here

    def remove_expense(self, expense_id):
        # Your implementation here

    def get_expense_by_id(self, expense_id):
        # Your implementation here

    def get_expense_by_title(self, expense_title):
        # Your implementation here

    def to_dict(self):
        # Your implementation here
```

## Github

After you have completed the implementation for you classes, create a github repository and share your project there. You are expected to add a comprehensive readme.md file that:

- Describes your project
- Explains how to clone it
- Explains how to run the code.

Please share a link to your github repository in this form