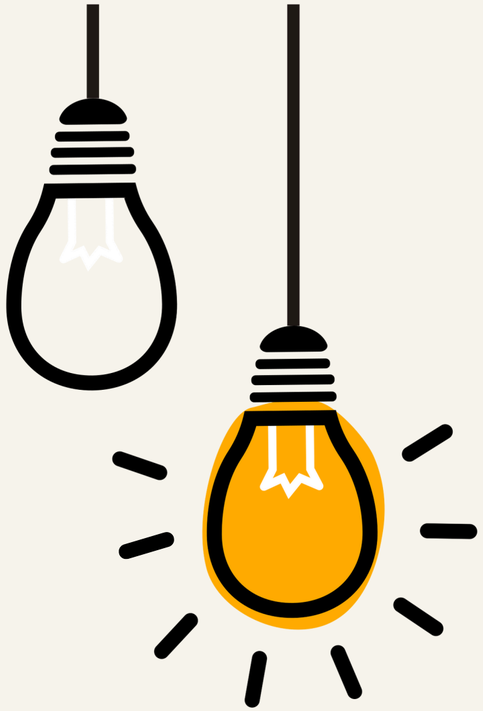

Filoger Comprehensive Python For AI Course 2024

FinalProject

Score: 1000 + 200 (Optional Points)+ 50 (GitHub)

Time Tracker Program Description

1000

Create a Python program that functions as a time tracker for managing tasks and projects using object-oriented programming (OOP). The program will allow you to manage projects and their associated tasks, track time spent on tasks. The entire implementation should follow OOP, including defining appropriate classes, attributes, and methods.

Overall Structure:

- Project Management: Each project has a unique name and can contain multiple tasks.
- Task Management: Each task belongs to one project, has specific attributes, and must have a unique identifier (ID) within the project.

Time Tracker Program Description

Phase I: Basic Project and Task Management

- **Objectives:**
  - Build classes for managing projects and tasks.
  - Implement basic CRUD operations for projects and tasks.

- **Requirements:**

  I. Project Management:

  - Add a Project: Create and add a new project to the system. Ensure that the project name is unique.
  - Edit a Project: Allow modification of project details, such as renaming the project.
  - Remove a Project: remove a project and all tasks associated with it.
  - Display Projects: Display a list of all projects, showing project names and the number of tasks associated with each project.

Time Tracker Program Description

2. Task Management:

- Add a Task to a Project: Add a task to a specified project by entering its details.
- Edit a Task: Modify the name, description, status, start time, or end time of a task.
- Remove a Task: Remove a task from a project using its unique ID.
- Mark a Task as Done: Update a task's status to 'Done' (True)

- **Task Attributes:**
  - Name: The name of the task (not necessarily unique within the project).
  - ID: A unique identifier for each task.
  - Description: A brief explanation of the task.
  - Status: Indicates whether the task is done (True) or not done (False).
  - Start Time: The time the task began (entered by the user).
  - End Time: The time the task was completed (entered by the user).
  - Duration: The time spent on the task (difference between the end time and start time, in hours).
  - Project Association: Each task must be associated with a project.

Time Tracker Program Description

Phase 2: Display and Search Functionality

1. **Display:**
   - **Display All Projects: Show a list of all projects, including their names and the total number of tasks.**
   - **Display Tasks for a Project: Display all tasks associated with a specified project, including each task's name, ID, description, status (done or not done), start time, end time, and duration.**
2. **Search:**
   - **Search for a Project: Allow searching for a project by its name. Display details about the project and its tasks.**
   - **Search for a Task: Implement the ability to search for a task by its name within a specific project. Display the task's details (name, description, status, start time, end time, and duration).**

Time Tracker Program Description

Additional Points:

- The program must be OOP, defining all necessary classes, attributes, and methods to accomplish the requirements.
- The program should continuously prompt the user for input and perform requested operations until the user enters "exit".
- Input validation and error handling should be implemented where appropriate.
- Advanced features such as data export to a text file and simple user authentication are optional (+200).

Did you know *?*

You automatically
lose the chances you don't take.
Trust yourself. You can do this.

Don't give up on your dreams :)