COMP1002 - Advanced Python

Final Project Report: Task Tracking System

Group Members:

- Celal Çatal - 232010020019
- Melih Kaan Direk - 232010020015
- Müşerref Ebru Erden - 232010020012

Date: 23.05.2025

---

1. Introduction

Managing tasks in small teams without a proper system often leads to missed deadlines, poor collaboration, and inefficient workflows. This project aims to solve that issue by developing a lightweight command-line based Task Tracking System. The system enables users to create, view, update, and delete tasks, making team task management straightforward and efficient. It is especially designed for teams that require simplicity and ease of access without using complex or resource-heavy tools.

---

2. Technologies Used

- **Programming Language:** Python 3.x
- **Libraries:**
  - `json` – For data persistence using a JSON file
  - `datetime` – For managing and validating due dates
- **Tools:**
  - Command-line interface (CLI)
  - File-based data storage (no external database)

---

3. Implementation Details

The system is implemented as a Python script (`task_manager.py`) which interacts with a JSON file (`tasks.json`) for storing task information. Key functionalities include:

- **Task Creation:**
  - Users can add new tasks with title, description, assignee, due date, and default status.
  - Input validation ensures required fields are not left empty and dates follow `YYYY-MM-DD` format.
- **Task Listing and Filtering:**
  - All tasks are displayed in a list format.
  - Users can filter tasks by assignee or status.

- **Task Updating:**
  - Task status can be updated to "Not Started", "In Progress", or "Completed" by selecting the task index.
- **Task Deletion:**
  - Tasks can be deleted after confirmation, with index validation.
- **Error Handling:**
  - Includes user-friendly prompts and error checks (e.g., invalid index, incorrect formats).

---

4. Challenges & Solutions

- **Challenge:** Handling user input errors such as incorrect date formats or invalid task indices.
  - **Solution:** Implemented input validation and try-except blocks to manage exceptions gracefully.
- **Challenge:** Ensuring data is persistent without a database.
  - **Solution:** Used JSON files to store and update task information between sessions.
- **Challenge:** Keeping the system lightweight and usable via command-line.
  - **Solution:** Avoided external libraries and frameworks, and focused on clear menu-driven CLI navigation.

---

5. Conclusion & Future Work

The Task Tracking System successfully achieves its goal of providing an accessible and easy-to-use task management solution for small teams. It demonstrates efficient use of Python's built-in capabilities for file handling and date management.

Future Improvements:

- Add search and sort capabilities (e.g., by deadline).
- Implement deadline reminders using scheduled tasks.
- Optionally include a basic GUI using Tkinter.

End of Final Report