**Assignment: Build a Simple To-Do List Application**

**Objective:**

Develop a **To-Do List** application using Python. This application will allow users to manage their daily tasks by adding, updating, and organizing their to-do items. The project will emphasize modular and functional programming principles and will include an interactive interface built with Streamlit.

**Problem Statement:**

You want to keep track of your daily tasks to stay organized and ensure nothing is forgotten. Your goal is to create a Python application that lets users easily manage their tasks. The application should allow users to add new tasks, mark tasks as completed, categorize tasks, and view their tasks in an organized manner.

**Application Features:**

1. **Task Management:**
   * Allow users to add new tasks, including:
     + **Task Description**: A brief description of the task.
     + **Category**: The category or type of task (e.g., work, personal, shopping).
     + **Due Date**: When the task is due.
   * Provide the ability to edit or delete tasks.
   * Allow users to mark tasks as completed.
2. **Task Organization:**
   * Organize tasks by:
     + **Category**: View tasks grouped by their category.
     + **Due Date**: View tasks ordered by their due dates.
     + **Completion Status**: View completed vs. pending tasks.
   * Implement filtering options so users can quickly find specific tasks (e.g., show only work tasks or only tasks due today).
3. **Visualizations:**
   * Provide visual representations of task completion:
     + A progress bar showing the percentage of tasks completed.
     + Pie charts or bar graphs displaying the distribution of tasks across different categories or completion statuses.
4. **Data Persistence:**
   * Enable users to save their task list to a file (e.g., CSV or JSON) and load it later to continue managing their tasks.

**How to Build It:**

* **Modular Programming:**
  + Divide the application into modules:
    - **Task Management Module**: Handles adding, editing, deleting, and marking tasks as completed.
    - **Data Management Module**: Handles saving/loading tasks from a file.
    - **Visualization Module**: Generates visual elements like progress bars and charts.
* **Functional Programming:**
  + Use functions that return consistent results for the same inputs without side effects.
  + Apply higher-order functions where appropriate (e.g., filtering tasks based on criteria).
* **Streamlit Interface:**
  + Create a simple and intuitive Streamlit app that allows users to:
    - Input new tasks.
    - View, filter, and organize tasks.
    - See visual progress and statistics on their tasks.

**Example Scenario:**

A user wants to organize their day by creating a to-do list. They enter tasks like "Buy groceries," "Finish project report," and "Go for a run." They categorize them as "Shopping," "Work," and "Personal." As they complete tasks, they mark them as done and see their progress in a progress bar.

**Requirements:**

* **Code Quality**: Ensure the application is modular and follows best practices in functional programming.
* **Streamlit Interface**: The interface should be user-friendly and intuitive, allowing users to interact with their tasks easily.
* **Error Handling**: Implement error handling to manage invalid inputs or issues with data files.
* **Documentation**: Include a README file explaining how to use the application, install dependencies, and an overview of each module.

**Deliverables:**

* Python code files organized into modules.
* Streamlit app file.
* A README file with instructions.
* A requirements.txt file listing any dependencies.

**Evaluation Criteria:**

* **Functionality**: Does the application allow users to manage tasks effectively?
* **Code Quality**: Is the code modular, readable, and well-organized?
* **User Interface**: Is the Streamlit interface clear and easy to use?
* **Documentation**: Is the README comprehensive and helpful?

**Deadline:**

* The assignment is due by [18/08/2024].