To-Do List Application

Key Features and Functionalities

1. Task Management

- **Adding Tasks:** Users can input a task's title, description, date, and time, and add it to the list.
- **Editing Tasks:** Tasks can be updated by selecting an existing one, editing its details, and re-adding it.
- **Deleting Tasks:** Selected tasks can be removed from the list.
- **Marking Tasks as Completed:** Tasks can be flagged as completed, with the description updated to indicate completion.

2. User Interface

- **Input Panel:**
  - Includes entry fields for the task title, description, date selection (via a calendar), and time input.
  - Buttons for adding tasks.
- **Task List:**
  - Displays tasks in a tabular format with columns for "Description" and "Date."
- **Action Buttons:** Buttons for deleting tasks, tracking upcoming tasks, editing tasks, and marking tasks as completed.

3. Tracking and Alerts

- **Track Upcoming Tasks:** Identifies and displays tasks with deadlines later than the current date and time.
- **Notifications:** Uses message boxes to provide feedback on actions such as successful task addition, deletion, or errors in input.

4. Data Persistence

- **Saving Tasks:** Tasks are saved to a `tasks.json` file in JSON format for persistence across sessions.
- **Loading Tasks:** On startup, the app reads tasks from the JSON file and populates the task list.

5. Error Handling

- Alerts the user in case of:
  - Empty input fields.

- Invalid date or time formats.
- Selection errors when performing actions on tasks.

How the Code Works

1. **Task Class:**
   - Represents individual tasks with attributes for title, description, and deadline.
   - Includes a `to_dict` method to convert task details into a dictionary for JSON serialization.
2. **ToDoApp Class:**
   - Manages the entire application, handling the UI and task operations.
   - Initializes UI components and defines methods for various functionalities.
3. **Graphical User Interface:**
   - Built using `tkinter` and includes elements like labels, entry fields, buttons, a calendar widget (`tkcalendar`), and a treeview for displaying tasks.
4. **Data Persistence:**
   - Tasks are serialized into JSON format and saved in a file (`tasks.json`).
   - Upon application launch, tasks are loaded from the file if it exists.
5. **Main Application Loop:**
   - The `mainloop` of the `tkinter` root window keeps the application running, waiting for user interactions.

How to Use

1. **Run the Code:**
   - Execute the script to launch the to-do list application.
2. **Add Tasks:**
   - Fill in the task details and click "Add Task."
3. **Manage Tasks:**
   - Use the buttons to delete, edit, or mark tasks as completed.
4. **Track Tasks:**
   - Click "Track Task" to view upcoming tasks.
5. **Close and Reopen:**
   - Tasks will persist, saved in `tasks.json`, and automatically reload on the next launch.

This program provides an intuitive and efficient way to organize and manage tasks with a user-friendly GUI.