**Task Manager App**  
A complete WPF application for easy management of tasks that is user-friendly and professional.

**Installation**  
1. Clone the repository or download the source code.  
2. Open the solution file - TaskManagementApp.sln, using Visual Studio 2019 or later.  
3. Ensure that you at least have the .NET Framework 4.7.2 installed on your system.  
4. Block the project to restore NuGet packages and compile the project.

**Running Project**  
1. Open solution in Visual Studio.  
2. Set TaskManagementApp as the startup project.  
3. F5 or click the "Start" button to run in debug mode.  
4. Ctrl + F5 will run without debugging.

**Dependencies**  
This project is dependent on following:  
· .NET Framework 4.7.2 or later  
· WPF - Windows Presentation Foundation  
No additional NuGet packages needs to be added on this project.

**Database**This application does not use any database of SQL Server. All the data is in-memory and managed by Observable Collection<Task> and not persisted across app runs. As an enhanced feature, implement the feature of persistence of data using some local database or file store.

**Settings**  
To run this WPF project, no additional setting is to be done.

 By default, configurations are used in WPF.

Anyhow, following can be modified  
• Window size: Modify the Height and Width properties in the MainWindow.xaml file.  
• Color scheme: Replace colors throughout XAML, especially within the RoundedButtonStyle resource.

**Features**

**1. Task Management:**  
• Create new tasks with title, description, due date, and priority.  
• Edit existing tasks.  
• Delete tasks.  
• Mark tasks complete.

**2. User Interface:**• \_clean and intuitive design.  
• Responsive layout.  
• \_CUSTOM-styled buttons and input fields.

**3. Task Filtering:**  
• Priority-based task filtering - All, Low, Medium, and High.  
• Clear completed tasks.

**4. Data Display:**  
• Shown tasks were in a DataGrid for easy viewing and interaction

• Columns are sortable

**5. Input Validation:**

• Done to prevent adding tasks without a title

• User-friendly error messages Project Structure

• MainWindow.xaml - Here goes UI layout, styles, resources

• MainWindow.xaml.cs - Here goes the application logic, including event handlers and other things related to data management.

•Task.cs - Here would go the class Task

**Code Overview**

• MVVM-like struct without any formal MVVM framework in use

 • Utilize Observable Collection<Task> for real-time UI updates  
• Styles defined in XAML keep consistent the look and feel of custom views

• Event handlers in the code-behind file deal with user input

**Future Improvement**

1. Employ data persistence using a local database or persisting files.
2. Task can be grouped into categories or tagged
3. Searching of tasks
4. Each task has its own edit window
5. Reminders/ Notifications can be set for a task
6. Different users can log in and support multiple users Contribution

**Contributions**

Thisproject involves four group members who have contributed with their invaluable, unique set of skills and expertise.

The following is a list of the members:

**Manpreet Kaur**:  She has implemented core functionalities of the application. She has implemented task management features, such as adding, editing, and deleting of tasks. Her coding made this application user-friendly.

**Jashandeep Singh:** The project proposal was done based on the objectives, features, and scope of the application. He planned and organized everything, which helped give a clear direction in which this project is going.

**Sahil:** He prepared the README file with regard to how to set up the application and running instructions, dependencies, and features included in the project. He did a great job in paying attention to every minute detail such that the user will have everything required for the efficient use of the application.

**Madhia**: Spearheaded the deployment of the project on GitHub and made the code accessible; thus, people were able to work with it with so much ease. Her version control work helped in maintaining an orderly workflow during the development process.

All the members played a highly important role in making this project a reality and showed that teamwork and collaboration exist.

Troubleshooting  
• To troubleshoot the application failing to build, check that you have the appropriate version of the .NET Framework installed.  
• For any errors relating to XAML, try cleaning and rebuilding the solution.