Requirement ID	Description of Requirement	Story Points	Priority	Sprint No.	story point per sprint	estimated hours:
1	Create a basic GUI layout using Tkinter or PyQt with separate sections for the calendar and to-do list. (UI only)	5	1			
2	Design the calendar interface that displays days, weeks, or months. with a way to access events (UI only)	5	3			
3	Build an event screen that allows you to add, view, edit, and delete events (whether they are to do list items or events) (UI onl	5	3			
4	Design the to-do list section that allows users to add, view, edit, and delete tasks. (UI only)	5	3			
5	Implement a unified data structure to handle both calendar events and to-do tasks. should handle: (title, time, location, notes, priority (High, Medium, Low), deadline, event/task, active?, catagories)	2	2			
6	Implement local data storage using SQLite for storing calendar events and to-do tasks. that ensures that data persists between app sessions by loading stored data when the app is launched.	3	1		25	37.5
8	Add input fields and dialog boxes for creating and editing calendar events and to-do tasks.	3	4			
9	Implement the ability to add new calendar events with details (title, time, location, notes).	3	5			
11	Implement the ability to add new to-do list tasks with details (title, priority, deadline).	3	5			
12	Add functionality to prioritize tasks in the to-do list (e.g., High, Medium, Low).	2	7			
13	Implement the ability to edit and delete calendar events and to-do tasks. (and mark to do list items as complete)	3	6			
14	Implement functionality for users to create custom categories for tasks (e.g., Work, Personal).	3	6			
15	Add filters to the to-do list section so users can sort tasks by category.	2	7		19	28.5
16	Set up a background process for handling notifications and reminders for both events and tasks.	5	8			
17	Create a notification system that displays pop-ups or sends emails for upcoming events and tasks.	5	9			
18	Allow users to configure notification settings (e.g., reminder times and mediums like pop-ups or emails).	3	10			
19	Implement functionality for setting recurring calendar events (e.g., daily, weekly, monthly).	5	8			
20	Implement recurring to-do list tasks (e.g., daily habits).	5	8			
21	Ensure that recurring events and tasks are auto-generated when the app is launched.	3	8			
24	Implement theme settings so users can choose between different color schemes and layouts.	3	11			
25	Allow users to configure notification preferences (e.g., when to receive notifications and how).	2	11			
26	Allow users to personalize font style and size for better accessibility.	2	11			
27	Organize the code into modular components for calendar management, task management, notifications, and UI.	5	12			
28	Integrate with Google Calendar API for syncing external events (if time permits).	8	13			