**Purpose**:  
The purpose of these tests is to verify the functionality of the [TaskManager](vscode-file://vscode-app/c:/Users/rento/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [TaskVisualizer](vscode-file://vscode-app/c:/Users/rento/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), and Sorter components, specifically focusing on task tagging, task completion visualization, and task sorting. The tests ensure that the system behaves as expected under various scenarios, including adding, saving, loading, sorting, and visualizing tasks.

Test Cases for test\_task\_tag.py

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Result |
| test\_add\_task | Verifies that tasks can be added to the TaskManager | Tasks are successfully added, and the task count matches the expected value. |
| test\_view\_tasks | Ensures that tasks are displayed correctly including their tags. | Task titles and tags are displayed in the correct format. |
| test\_save\_and\_load\_tasks | Tests saving tasks to a file and reloading them into the TaskManager | Tasks are saved and loaded correctly, with all attributes (e.g., tags) intact. |
| test\_tags\_are\_sorted | Verifies that tags are displayed in alphabetical order. | Tags are sorted alphabetically in the output. |

Test Cases for test\_task\_visualizer.py

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Result |
| test\_get\_completion\_data | Verifies that the number of completed and incomplete tasks is calculated correctly | The counts for completed and incomplete tasks match the expected values |
| test\_empty\_task\_manager | Ensures that the visualizer handles an empty TaskManager correctly | The visualizer returns zero for both completed and incomplete tasks |
| test\_pie\_chart\_data | Tests that the pie chart data is generated correctly | The pie chart reflects the correct proportions of completed and incomplete tasks |
| test\_tag\_changes\_affect\_completion | Verifies that changes to task tags update the completion status. | The completion data updates correctly when task tags are modified. |

Test Cases for test\_sorter.py

|  |  |  |
| --- | --- | --- |
| Test Case | Description | Expected Result |
| test\_title\_sorter | Verifies that tasks are sorted correctly by title | Tasks are sorted alphabetically by title |
| test\_date\_sorter | Verifies that tasks are sorted correctly by due date | Tasks are sorted chronologically by due date |
| test\_priority\_sorter | Verifies that tasks are sorted correctly by priority (High > Medium > Low) | Tasks are sorted by priority in the correct order. |
| test\_sorter\_integrity | Ensures that sorting does not modify the original task data | Task data remains intact after sorting |

Test Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test File | Total Tests | Passed | Failed | Remarks |
| test\_task\_tag.py | 4 | 4 | 0 | All tests passed. Task tagging and saving/loading functionality verified. |
| test\_task\_visualizer.py | 4 | 4 | 0 | All tests passed. Task visualization and completion tracking verified. |
| test\_sorter.py | 4 | 4 | 0 | All tests passed. Sorting functionality verified for title, date, and priority |

Observations:

* The TaskManager correctly handles task addition, saving, and loading, ensuring data integrity.
* Tags are displayed in alphabetical order, improving readability.
* The TaskVisualizer accurately calculates and displays task completion data, even when tasks are modified dynamically.
* The Sorter classes correctly sort tasks based on the specified criteria
* No performance issues

Recommendations:

* Add tests for edge cases
* Sorting tasks with identical titles, due dates, or priorities.