

# Online Gantt Chart Software

## Summary

This report describes the process and findings from my attempt to do full end-to-end testing of [Online Gantt Chart](#). The reason for this test project was to practice my skills as a tester following the Rapid Software Testing methodology.

I performed both experiential and instrumented testing. I performed primary and regression testing. I tested every significant aspect of the product as I understood it. I found many bugs, including several that I consider critical.

I would say that this product is not fit for use except as a demo or prototype of a possible design for project management software. It cannot handle realistically complex project data with reasonable performance and reliability.

## Product Information

The **Online Gantt Chart** (<https://www.onlinegantt.com>) is a web-based project management tool that helps visualize tasks and timelines in a clear, interactive format. It uses a horizontal bar chart to display project schedules, showing task durations, dependencies, and milestones. It also has features like resource allocation, progress tracking, multiple color schemes, project view, resource view, resource and settings editing.

The Online Gantt Chart has a cloud version and a local version. I tested only the local, free version, that does not require creating an account.

## General Test Process

My test process started with an open exploration phase. I conducted survey testing to gather information about the Online Gantt application, as I did not have any documentation, requirements, nor access to the development team. I created a draft of the test strategy based on what I learned about the product during the exploration. After several sessions of survey testing, I wrote down the test strategy. I designed the strategy with the help of the [Heuristic Test Strategy Model](#), which is part of the RST methodology.

The test strategy is attached at the end of this report.

## Test Results

### Task Data Consistency Testing

Data consistency means that any different views of the same data communicate the same information. For instance, a task with a Start and End date representing a five-day task should report a duration of five days. The simplest form of data consistency testing is to verify that the output of the program matches the data that I entered.

For my primary testing<sup>1</sup> I opened the application, generated a new Gantt file, and added new tasks. I tested that all fields (Task Name, Start/End Date, Duration, Progress, Color, Dependencies, Resources, and Notes) can accept a variety of data and observed whether any constraints were violated. Since there was no documentation about input constraints, I just applied my background knowledge. To test the input fields, I used a variety of data types: empty fields, all types of single-byte characters (windows-1252 code page), huge numbers of characters (e.g. 10000), and image data (I pasted a picture from Excel). I applied CRUD operations to check the accuracy and consistency between input and output data. I also checked that the data in the task area is consistent with the data in the timeline area and resource view.

I checked that for a set of tasks, consisting of a parent task and subtasks, the parent task's duration spans from the earliest Start date among its subtasks to the latest End date, containing the full period of its associated subtasks. I verified that the percentage in the Progress field of the parent task reflects the weighted average of the percentages of its subtasks.

I also made various configuration settings to check for consistency in the task area. Most of the settings are accurately reflected in the task area, however I did find some bugs.

I assessed the functionality of the Expand All, and Collapse All buttons, scroll bars, search box, and drop-down lists. I checked the functions such as autofit, sorting, enabling/disabling columns, and filtering.

I exported the Gantt file in Excel, PDF, and Image files and compared the data displayed in the application with the data exported in the files.

**Some of the most interesting bugs I found related to the task data consistency are listed below:**

- When using dates from the distant past and future, the application slows down and almost crashes. Although the Notes tab is part of the Task Information, it is not displayed as a column in the Task area.
- The default color is set to light blue. When no color type is selected in the task information, it is displayed in the timeline area, but not in the task area. IDs in Resource View are not consistent with the IDs in Project View.
- When all the days of the working week are disabled in settings, the timeline shows that only weekends are disabled
- Any settings made in the project View such as column width, column visibility, sorter, and zoom level get reset
- The table header in Project View is not consistent with the header in the Excel file (e.g. Task Name vs Name, End vs Finish, Progress vs Complete). When sorting and filtering functions are used, the exported files do not accurately reflect the data displayed on the online Gantt chart.

## Timeline Accuracy Testing

To test the timeline section, I applied CRUD operations to individual and group tasks, with a dependency relationship between them. I interacted with the chart bars to adjust task duration by dragging them left or right. I set various durations (e.g. 1 day, 15 days, 1 month, 1 year...) and verified that the adjusted duration in the timeline is consistent with the start and end dates in the task area. I verified that the application is interactive, dynamically updated, and maintains consistency with the task area. Then, I assessed the accuracy of the progress shown on the chart bars by settling on a specific progress level for

---

<sup>1</sup> Primary testing is the form of testing that we do approaching a product for the first time (as opposed to regression testing).

each task, including the parent-child relationship, and compared the progress percentages in both the timeline and task area for consistency.

During the testing of the “Zoom in”, “Zoom out”, and “Zoom to fit” buttons, I discovered that when zooming in to the maximum level:

- the application crashes
- the timeline showing the hours disappears
- the task dates are displayed on different dates on the timeline in accordance with the dates in the task area
- workdays are highlighted instead of weekends

When zooming out to the maximum level dependencies cannot be properly visualized.

I also tested the timeline accuracy with various configuration settings. I found some inconsistencies between the configuration settings and the timeline.

### Task Dependency Testing

To test the dependencies, I started a new project and created individual tasks and parent-child relationship tasks with different durations (1 day, 1 week, 1 year, 2 years). I tested the dependency setup and how it can be adjusted using CRUD operations. I visually confirmed that the dependencies reflected the established relationships and observed the system’s response to changes. I also set up dependencies with positive and negative offsets, as well as tasks with multiple dependencies. I’ve tested whether dependent tasks automatically adjusted their start and end dates when one of the tasks was modified.

I used the drag-and-drop functionality to check that dependencies are dynamically updated after taskbar reordering, to observe how the application responds to the changes, and to check the data consistency in the timeline and task area.

Further, I examined how the system responded to invalid dependency setups (e.g., circular dependencies).

I configured all the settings and checked how they affected the application’s behavior.

Finally, I exported the project to Excel, PDF, and Image formats and compared the data to verify that it remained consistent across all exports.

### Resource Testing

To test the resource section, I added several resources and assigned to each task a resource and multiple resources. I evaluated input validation with various data types (special characters, numbers) and tested the length limits during the resource addition. I applied CRUD operation in the Edit Resources, Project View, and Resource View sections, to check the behavior of the system. I checked that business rules are applied correctly when assigning resources: 1 resource to a task, 2 or more resources to a task, 1 resource to many tasks, and that the assigned resources are consistent in the project view, resource view, and exported files. I uploaded a Gantt file with 750 tasks, compared the data for specific tasks displayed in both the project view and the resource view, and concluded that the data in the resource view does not match the data in the project view.

### GUI Charisma Testing

GUI Charisma Testing is an evaluation of how comfortable, pleasant, and productive it is to learn and use this product.

Although I found it hard to use it for the first time, maybe I am not a representative user of this product. This is the first time I have interacted with a project management application.

After I got over my confusion about how project management software works, I found the application is easy to use. However, I found that the application can be annoying in some cases such as refreshing the application and losing all the data, and each time I configure some task table settings and then change the view, all the settings return to the default state. Also, when the name of a resource is changed, it is disconnected from its assigned task. The application does not have any documentation about its features.

Functions such as add, edit, and delete are easy to use, providing a smooth and efficient user experience. The fonts, colors, icons, taskbars, dependency lines, and spacing are consistent throughout the interface. However, some inconsistencies in the application are noted in the bug section.

The Online Gantt application has 5\* ratings in the Reviews section. However, when accessing all the apps where reviews about the Gantt chart are written, 4-star reviews can also be found.

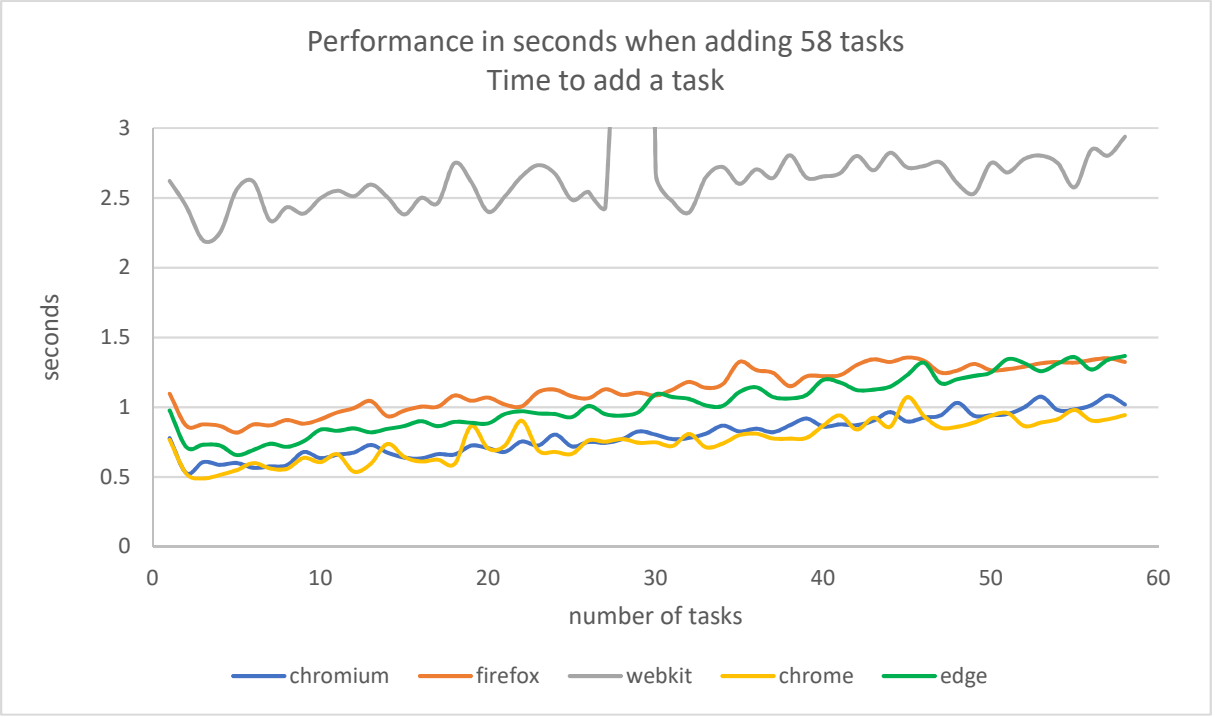
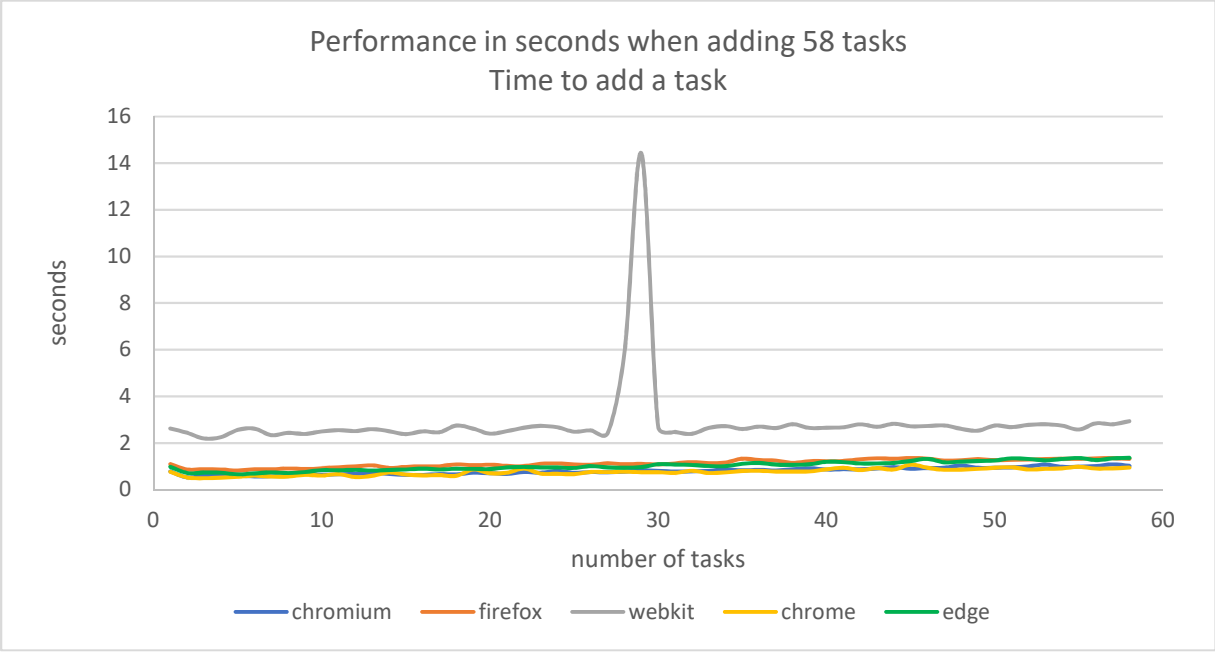
### Stress and Performance Testing

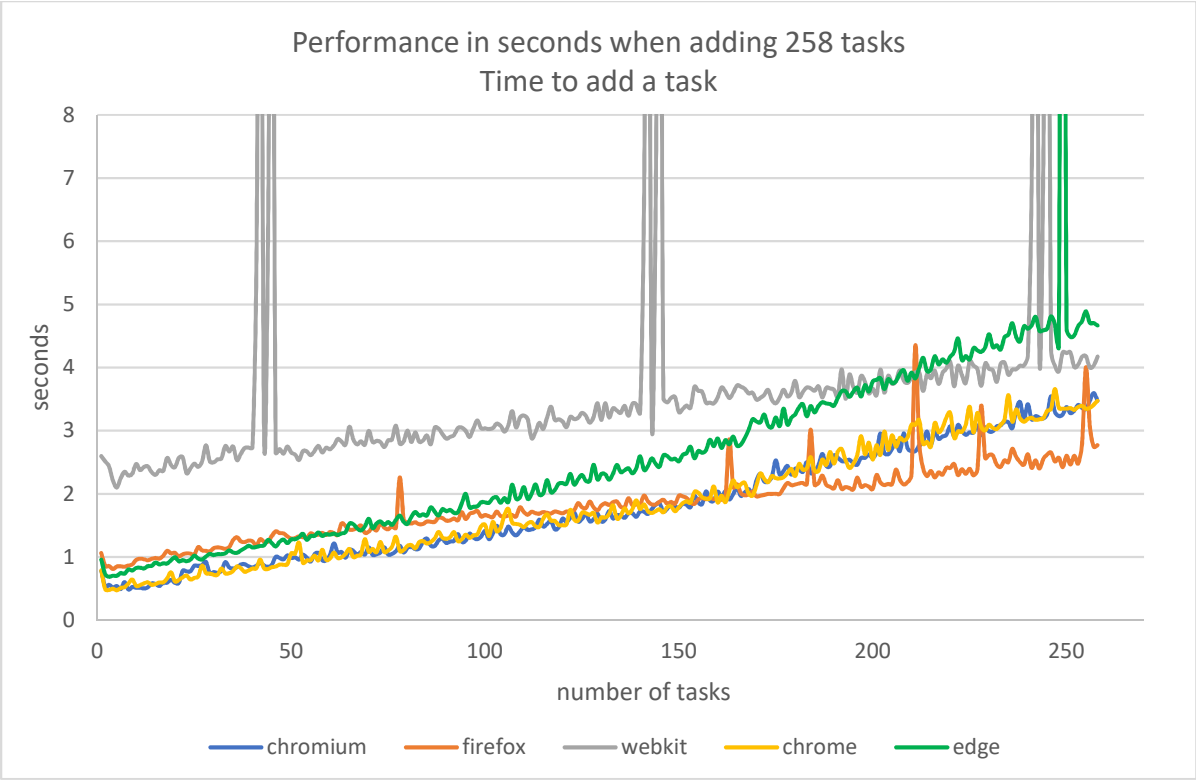
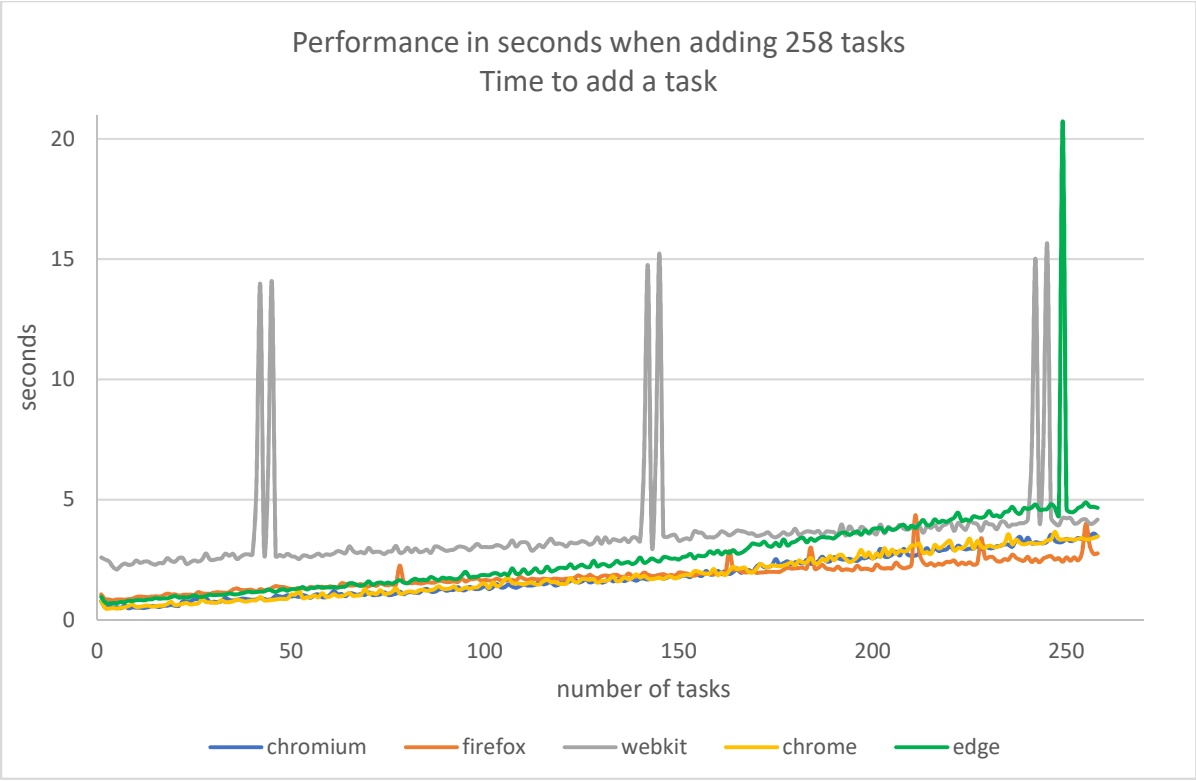
To test the performance of the application, I wrote a Playwright script ([https://github.com/SiposCristina/Onlinegantt.com\\_automation\\_code/tree/main](https://github.com/SiposCristina/Onlinegantt.com_automation_code/tree/main)). The script reads a JSON file that contains 58 tasks, another with 258 tasks, and a third with 508 tasks. Each file is read by the script and then it adds each task to the project. At the end of that process, the script performs a set of characteristic operations on six tasks, including indenting them, setting dependencies, assigning resources, and exporting project data.

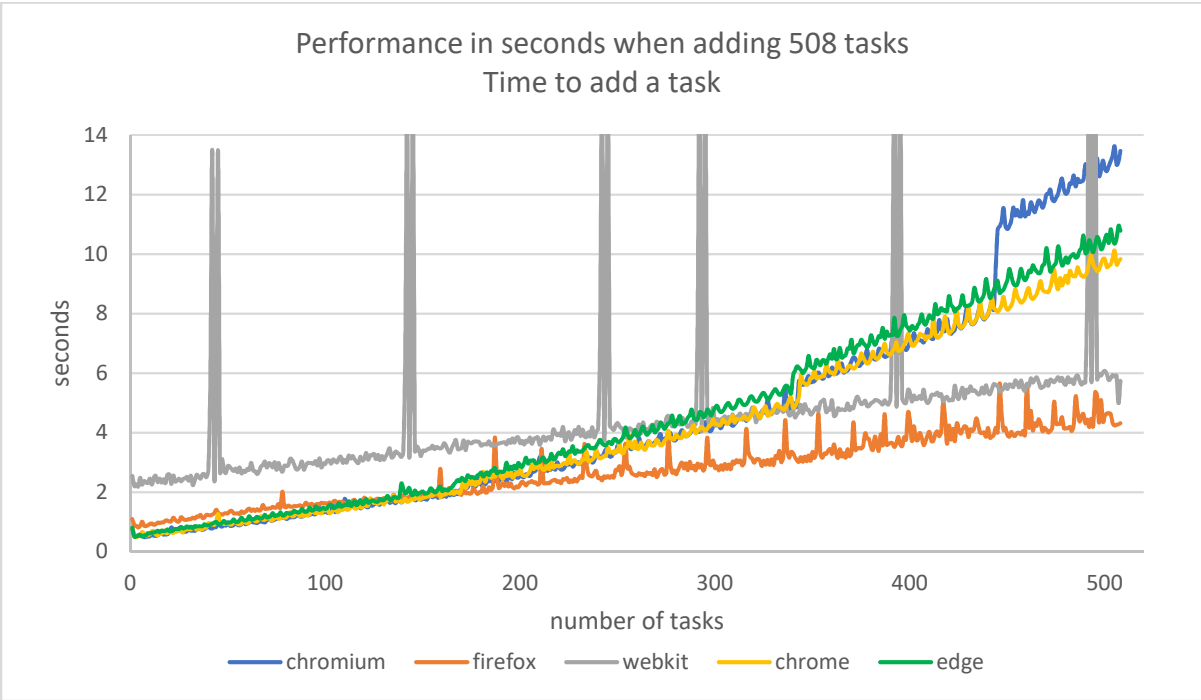
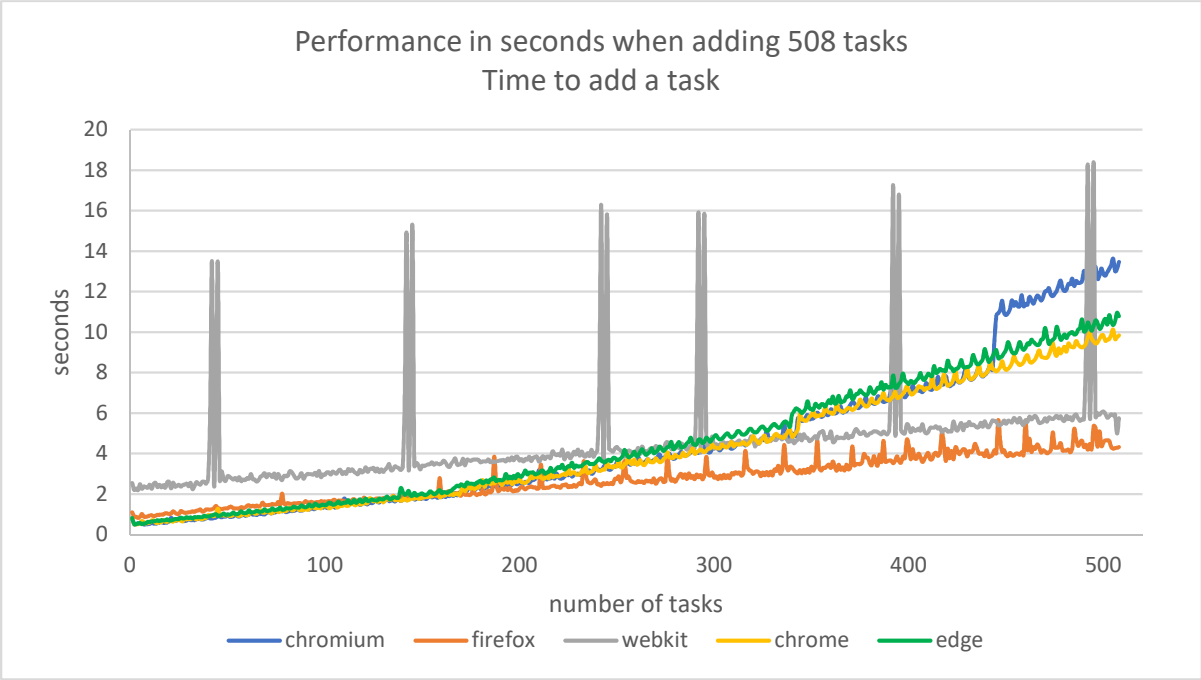
The purpose is to collect data and analyze it to observe how the application responds as the number of tasks increases.

The script was run on Chromium, Chrome, Firefox, Microsoft Edge, and WebKit browsers.

As the below charts illustrate, as more tasks are added, the application performance gradually slows down. The chart representing the data collected from running the automation for 58 tasks shows that the WebKit browser is inconsistent regarding the time it takes to add a task, with a dramatic spike in the time when a task date falls on a weekend. When running the automation for 258 tasks, the chart shows the same inconsistency for the WebKit browser, and at the end of the script, time to add a task spikes for Microsoft Edge as well. Analyzing the automation for 508 tasks, it can be observed that the Chromium browser has the same behavior. This pattern, with spikes in task addition times, points to slower performance, which leads to an experience of a gradual, but annoying slowdown in the application, which eventually makes it very hard to use. It is interesting that the performance dynamics are substantially different between browsers.

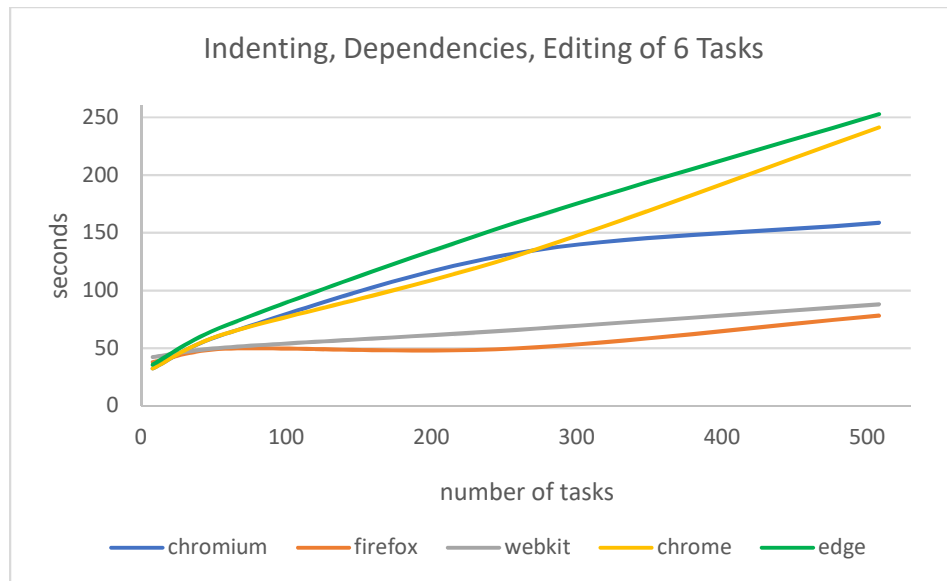






Also, the script provides information about how much time is required to indent six tasks, assign resources to six tasks, and add dependencies and notes to four tasks. This data is collected after all tasks (58, 258, and 508) have been added.

As illustrated in the chart below, the time required increases as more tasks are added, even when the number of indented tasks, assigned resources, added dependencies and notes remains constant. This indicates that the product becomes difficult to use as the size of the project reaches a couple hundred tasks.



## Compatibility Testing

I conducted the testing according to the test strategy.

When I ran the performance test script for 258 and 508 tasks under Firefox and WebKit, I observed a strange behavior: the application failed to export the PDF and PNG files.

## Negative Testing

Negative testing is testing the application by providing inputs or scenarios that violate required conditions. The purpose of negative testing is to provoke error handling.

I tested the following scenarios:

- Field length limits (excessive character lengths e.g. 10000, all types of characters, including special characters)
- Invalid dates (distant past and future dates, incorrect data combinations)
- Negative, zero, and large task duration (e.g. 1 year, 2 years, 5 years)
- Missing information (saving a task with empty start and end date fields)
- Dragging the Gantt Bar to Invalid Dates (dragging tasks to set and end date on a non-working day)
- Adding duplicate resource names in the Edit Resource sections
- Invalid dependencies
- File import errors (imported incompatible file types e.g. PDF, PNG, added huge Gantt file e.g. a file with 750 tasks)



In several scenarios, error messages were not explicitly displayed; however, the application rejected invalid inputs or actions and prevented them from being applied.

### Data Loss Testing

To test the application behavior for data loss I imported the 750 tasks file and left the application open overnight and over several days.

I opened a new project, added a few tasks, and then disconnected the laptop from the internet. I checked the behavior of the application by applying the major operations in the project view (e.g. CRUD operations, opening a new Gantt file, saving as a Gantt file, import/export file), as well as in edit resources (adding resources) and edit settings section. The application was able to perform most of the operations.

The application returned to its default state when I clicked on the refresh button, and when I clicked on the back button, it took me to the Google page.

The application crashes when zoomed to the maximum level, resulting in data loss. When changing a resource's name, the updated name is not displayed in the resource's column. Undo / Redo operations are non-functional.

### Regression Testing

The strategy for regression testing was designed to assess how the application handles the major activities. To achieve this, I wrote a Playwright script that reads a JSON file containing eight tasks. The file is read by the script and adds each task to the project. Two tasks are designed as parent tasks, each with three child tasks, resulting in six out of eight tasks being indented. Dependencies and notes are added to four tasks, and resources are assigned six tasks. Once complete, the project file is exported as an Excel, PDF, and Image file. The file names include the day and month of their export (e.g. downloads-21DEC\before).

Once this cycle is complete, the script proceeds by renaming the tasks, changing the dependency types, adding "Sunday" as a workday, and setting specific holiday dates for tasks scheduled on those days. Screenshots are taken to provide better visualization of the project, then the project file is exported as an Excel, PDF, and Image file. The file names include the day and month of their export (e.g. downloads-21DEC\after).

As shown in the screenshots below, at the time they were captured, the application was handling the activities as written in the script. The screenshots below are from the Chromium run.



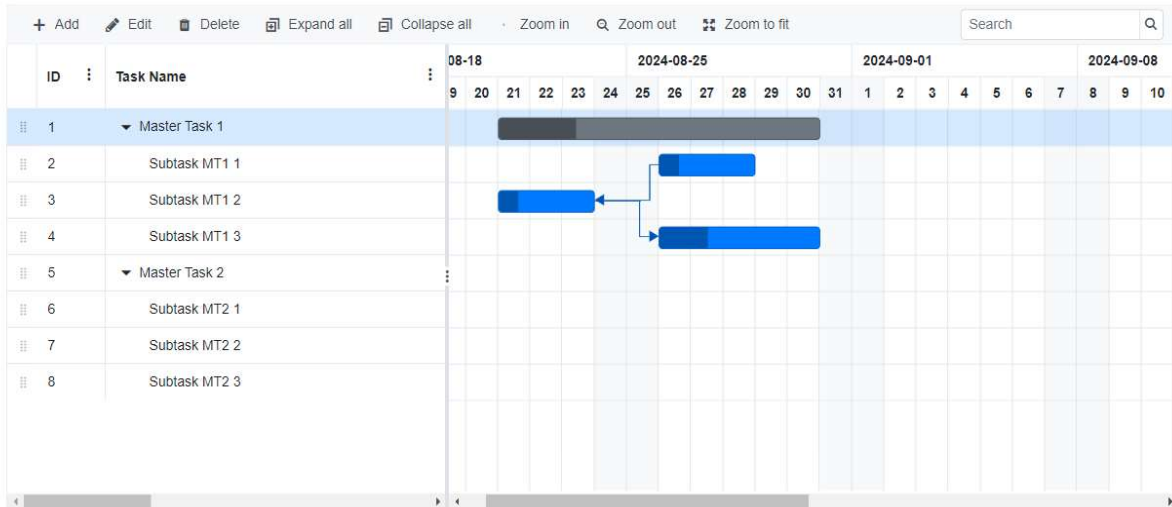
Free Online Gantt Chart Software. No Signup and No Download Required. 100% Free Forever.

Try Cloud

\* This 100% free tool allows you to save your gantt charts on your computer as .gantt files. Try our [cloud tool](#) to save your gantt charts on our cloud and share with your team. [★★★★★ Reviews](#)

New Open (.gantt file) Save As (.gantt file) Import / Export

Today All Tasks Project View



[Sponsored] Need a project management tool beyond gantt charts? Check out [this tool](#) for FREE.



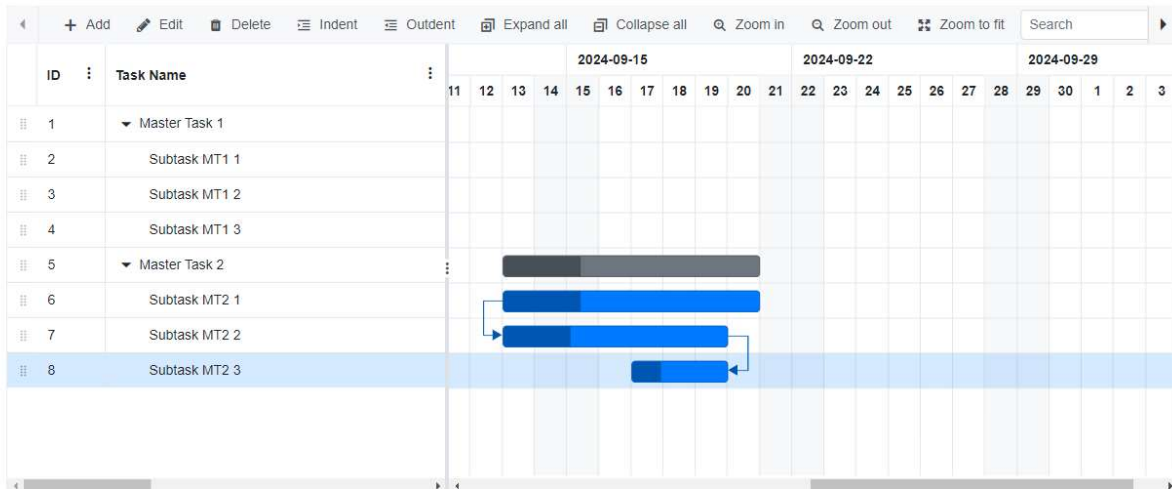
Free Online Gantt Chart Software. No Signup and No Download Required. 100% Free Forever.

Try Cloud

\* This 100% free tool allows you to save your gantt charts on your computer as .gantt files. Try our [cloud tool](#) to save your gantt charts on our cloud and share with your team. [★★★★★ Reviews](#)

New Open (.gantt file) Save As (.gantt file) Import / Export

Today All Tasks Project View



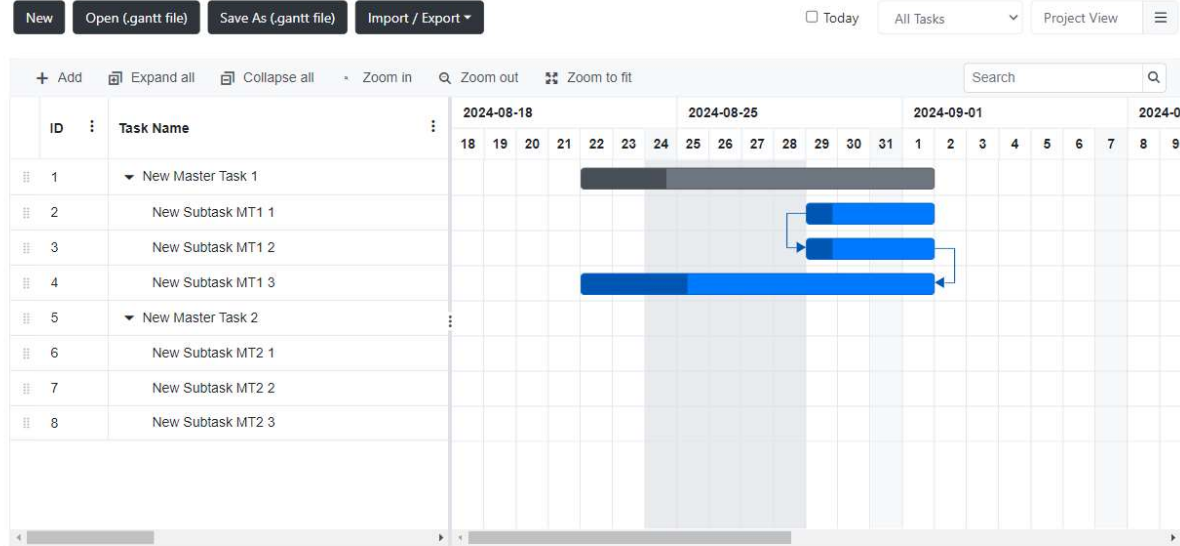
[Sponsored] Need a project management tool beyond gantt charts? Check out [this tool](#) for FREE.



Free Online Gantt Chart Software. No Signup and No Download Required. 100% Free Forever.

Try Cloud

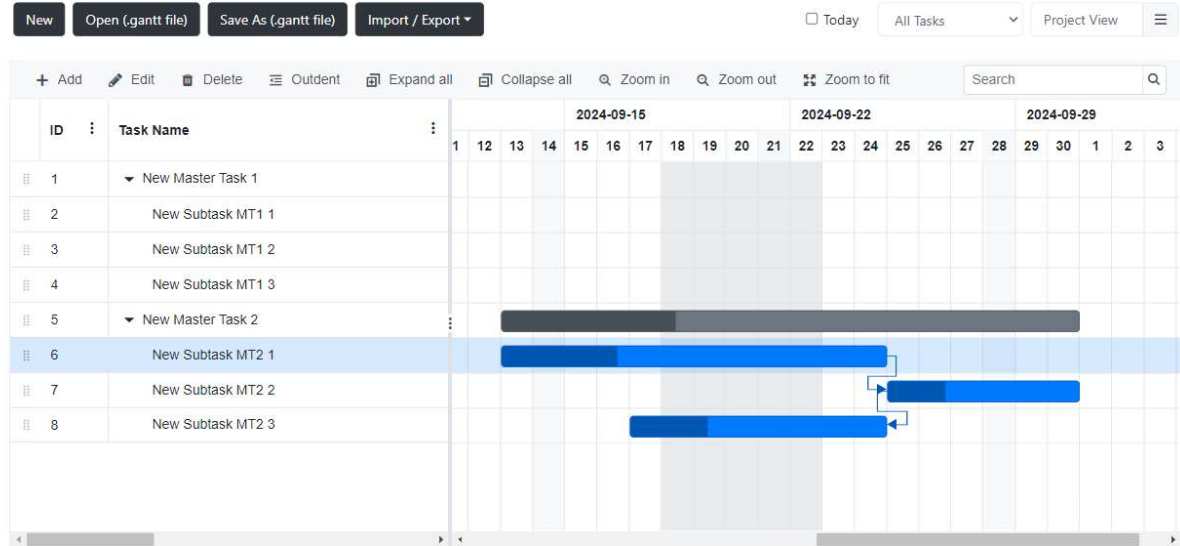
\* This 100% free tool allows you to save your gantt charts on your computer as .gantt files. Try our [cloud tool](#) to save your gantt charts on our cloud and share with your team. [★★★★★ Reviews](#)



Free Online Gantt Chart Software. No Signup and No Download Required. 100% Free Forever.

Try Cloud

\* This 100% free tool allows you to save your gantt charts on your computer as .gantt files. Try our [cloud tool](#) to save your gantt charts on our cloud and share with your team. [★★★★★ Reviews](#)



I used the “comp” command in the regression testing process. *comp* is the Windows command used to compare two files. I created a batch file called *check.bat* and I wrote the following code with the guidance of my mentor, James Bach:

```
comp /M Masterfiles\before\Export_to_Excel_File.csv downloads-  
%1\before\Export_to_Excel_File.csv  
  
comp /M Masterfiles\before\Export_to_Image_File.png downloads-  
%1\before\Export_to_Image_File.png  
  
comp /M Masterfiles\before\Export_to_PDF_File.pdf downloads-  
%1\before\Export_to_PDF_File.pdf  
  
comp /M Masterfiles\after\Export_to_Excel_File.csv downloads-  
%1\after\Export_to_Excel_File.csv  
  
comp /M Masterfiles\after\Export_to_Image_File.png downloads-  
%1\after\Export_to_Image_File.png  
  
comp /M Masterfiles\after\Export_to_PDF_File.pdf downloads-  
%1\after\Export_to_PDF_File.pdf
```

I compared the CSV, PDF, and PNG files. I created a folder called Masterfiles, with “before” and “after” directories, where I saved the files from different test runs. These were then compared with the dynamic folder path (downloads-1%).

I ran the test with this command: `tests\check 21DEC`, where `tests` is the folder containing the `check.bat` batch file, and `21DEC` is the folder containing the CSV file from the test run performed on the 21<sup>st</sup> of December.

Also, there are no differences between the two CSV and PNG files, but there are differences between the PDF files. This isn't due to a bug. Visually, the PDFs appear the same, but their binary data differs every time they are generated. Since these differences don't affect how the files look, a binary comparison doesn't provide meaningful insights.

Below are the test results displayed in the terminal:

```

C:\code\Online_Gantt>comp /M Masterfiles\after\Export_to_Excel_File.csv downloads-21DEC\after\Export_to_Excel_File.csv
Comparing Masterfiles\after\Export_to_Excel_File.csv and downloads-21DEC\after\Export_to_Excel_File.csv...
Files compare OK

C:\code\Online_Gantt>comp /M Masterfiles\after\Export_to_Image_File.png downloads-21DEC\after\Export_to_Image_File.png
Comparing Masterfiles\after\Export_to_Image_File.png and downloads-21DEC\after\Export_to_Image_File.png...
Files compare OK

C:\code\Online_Gantt>comp /M Masterfiles\after\Export_to_PDF_File.pdf downloads-21DEC\after\Export_to_PDF_File.pdf
Comparing Masterfiles\after\Export_to_PDF_File.pdf and downloads-21DEC\after\Export_to_PDF_File.pdf...
Compare error at OFFSET 5C28
file1 = B9
file2 = 85
Compare error at OFFSET 5C29
file1 = B6
file2 = 32
Compare error at OFFSET 5C2A
file1 = 26
file2 = B6
Compare error at OFFSET 5C2B
file1 = 7B
file1 = 7B
file2 = 78
Compare error at OFFSET 5C9A
file1 = C3
file2 = CA
Compare error at OFFSET 5C9B
file1 = 42
file2 = 32
Compare error at OFFSET 5C9D
file1 = 87
file2 = 8D

```

## Bugs

### Critical bugs

- When refreshing the page, the application returns to its default page, and all the data is lost
- When zooming in too much the application crashes
- When changing the name of a resource, and returning to the project view, the updated resource name is no longer displayed in the resource column
- Application crashes when a huge number is entered in the Duration field
- When you use a date from the distant past or future, the application crashes

### Normal bugs

- Duration field accepts a negative number
- Autofit is disabled when going to resource view or when deleting a task
- Resources can be deleted without any warning message
- Although none of the work week check box is selected in the settings area, the enabled days in the calendar (in the project view) are from Monday to Friday
- When using filters and/or sorting in the task area, and the file is exported in all three options (Excel, PDF, PNG), the filters and the sorting are not applied to the exported files

- The Autofit feature from the Task Name field is disabled each time resource view or settings are selected
- Task Name field accepts a huge number of characters (e.g. more than 400)
- Duration field accepts decimals
- When using the sort or filter function, the exported files do not accurately reflect the data displayed on the online Gantt chart
- Occasionally, when a task is set for 1 day, it becomes invisible on the timeline
- When dragging a chart bar toward the left or the right, and releasing it abruptly, the bar doesn't settle at the position where it was clicked
- The chart bar cannot be dragged beyond the limits of the scroll bar
- The Value field in the Progress filter section accepts negative values
- There were no warning messages when adding a resource with the same name or adding a huge number of characters in the resource field (e.g. 10000)
- When zooming in, the period in the chart bar does not align consistently with the task area
- When zoomed in to the maximum level, the timeline showing the hours disappears, the task is displayed at a different date (inconsistency with the task area), workdays are highlighted instead of weekends
- When zoomed out the bottom scroll bar disappears, and the dependencies between cannot be seen on the timeline
- The dependency line is not visible on the timeline when it is set at the start of the week for the entire project
- When working days are disabled from settings, and returning to project view, days from Monday to Friday appear as working days
- Functions such as autofit, sorting, enabling/disabling columns options, filtering, zoom in/out, and zoom to fit may be annoying due to their inconsistent behavior and illogical filtering options.
- Whenever a column is disabled and switched to resource view or settings, the column is re-enabled when I return to the project view
- When adjusting the middle bar to a specific position in the project view, it shifts when switching to the resource view and then returning to the project view
- Offset field accepts a huge number of special characters
- Dependencies cannot be set by dragging from one task to another when both tasks are starting on Monday, and Monday is set as the first day of the week
- Filtering by Start/End Date does not always work correctly

### Minor bugs

- When adding a new task and the color is not selected, the blue color is added by default, and in the task area the default color is not displayed, only in the timeline area
- ID's number from the project view is different from the resource view
- Sort Ascending / Descending does not function properly
- The color column has features such as Sort Ascending / Descending and Filter
- Inconsistent field naming: *Start / End* fields in task area vs *Start / End date* in settings
- Filter options in the Duration section are not relevant and are not working correctly
- In Resource View New, Open, Save As, and Import/Export buttons are clickable
- In Edit Resources and Edit Settings, New, Open, and Save As buttons are clickable
- The default field in the Filter option under the Dependency and Resources column is not consistent with the Task Name, Duration, and Color column

- If no color type is selected in the task information, it is displayed in the timeline area, but not in the task area. Also, the non-working days are not well highlighted in the timeline area.
- The header text (e.g. Start, End, Progress %) in the task area is inconsistent with the text in the pop-up box on the timeline (e.g. Start Date, End Date, Progress).

## Discoveries

- Notes are not displayed in the task area
- Only the 5\* ratings are displayed in the Reviews section. However, when accessing all the apps where reviews about the Gantt chart are written, 4-star reviews can also be found
- Task area columns can be dragged to the left or right side of the screen
- The transition from one year to the next is neither highlighted nor marked
- Once a parent task is deleted, its child tasks are also deleted

## Appendix: Test Strategy For OnlineGantt.com

The [Online Gantt](#) application is a basic web-based tool for project management that visually represents a project's timeline, tasks, and dependencies.

The purpose of this test strategy is to guide the testing of the Online Gantt application.

### Risks and Activities

The following table lists the risks that the application might have. The risks are listed on the left side of the table, and on the right side of the table are testing activities that help us understand the product's status with respect to that risk area.

<b>GUI Charisma Testing</b> <i>The product may be awkward or annoying to use in some ways.</i>	<ul style="list-style-type: none"><li>• Check the consistency of colors, fonts, icons, and buttons</li><li>• Do beta testing to evaluate the application</li></ul>
<b>Task Data Consistency Testing</b> <i>Task data may have inconsistencies, such as incorrect start/end dates, calculations, durations, progress, or dependencies.</i>	<ul style="list-style-type: none"><li>• Check the accuracy and consistency<ul style="list-style-type: none"><li>◦ of the data, dependencies, and resources in the task and timeline area</li><li>◦ of the data displayed in different views</li><li>◦ between the data displayed in the application and the exported files</li></ul></li><li>• Use certain configurations of settings to check they are applied correctly</li></ul>
<b>Timeline Accuracy Testing</b> <i>The correctness of the display may not be reflected in the task area.</i>	<ul style="list-style-type: none"><li>• Interact dynamically with the display and check whether the changes are reflected in the task area</li><li>• Use certain configurations of settings to check whether they are applied in the task area</li></ul>
<b>Task Dependency Testing</b> <i>Dependencies may be applied wrong.</i>	<ul style="list-style-type: none"><li>• Check all combinations of dependencies</li><li>• Verify functions such as creating, modifying, and deleting dependencies</li><li>• Check the visual representation on the bar chart</li></ul>
<b>Resource Testing</b> <i>Operations that associate resources with tasks may fail.</i>	<ul style="list-style-type: none"><li>• Add, delete, and edit the resources</li><li>• Check if the changes made in the resources are reflected in the project and resource view</li></ul>
<b>Stress and Performance Testing</b> <i>Adding a large number of tasks may slow down or crash the application.</i>	<ul style="list-style-type: none"><li>• Add a large number of tasks, dependencies, and resources simultaneously and progressively</li><li>• Create a project that has a big timeline</li><li>• Import/export large files</li></ul>
<b>Compatibility Testing</b> <i>The application may not be compatible with all browsers, and devices.</i>	<ul style="list-style-type: none"><li>• Test on:<ul style="list-style-type: none"><li>◦ Chrome is my default browser, therefore everything in this strategy is applied to Chrome</li><li>◦ Testing on Chrome tells me about Chromium; however, I also will test using the Playwright Chromium browser project</li><li>◦ For the Firefox browser I will apply the test strategy</li><li>◦ Regression and performance automation only for WebKit and Microsoft Edge</li></ul></li><li>• Testing will be performed on Windows 10</li></ul>
<b>Negative Testing</b> <i>Erroneous conditions may not be handled properly.</i>	<ul style="list-style-type: none"><li>• Use bad data for the error/warning displays</li><li>• Import incompatible file types</li><li>• Delete tasks, dependencies, and resources</li><li>• Configure incorrect dependency types</li><li>• Simulate network disconnection</li></ul>
<b>Data Loss Testing</b> <i>The entire project or parts of the project may be lost.</i>	<ul style="list-style-type: none"><li>• Do not save or export and leave the application open over a long time</li><li>• Hit the refresh and the back button</li><li>• Test for the possibility of accidental deletion</li></ul>
<b>Regression Testing</b> <i>Changes may cause bugs in the product.</i>	<ul style="list-style-type: none"><li>• Test using automation for basic functionality, such as adding and deleting tasks, updating tasks, adding and deleting dependencies and resources</li></ul>



## Product Coverage Outline

This is an outline of testable elements of the application which we use to help us plan and report the testing while performing the test activities above.

### Task

#### *Task Management Operations*

##### Add

##### New Task

##### General

ID

Name

Start Date

End Date

Duration

Progress

Color

Save

Cancel

Insert button from keyboard

##### Edit

##### Edit Button

##### Right Click on Existing Task

Task Information

Delete Task

##### Add

Above

Below

Child

Milestone

Delete Dependency

Convert

To Milestone

To Task

Updating timeline after editing

Delete

Indent

Outdent

Expand all

Collapse all

Zoom in

Zoom out

Zoom to fit

#### *Header*

ID

Name

Start Date

End Date

Duration

Progress %

Dependency

Resources

Color

- Drop-down menu
  - Autofit all columns
  - Autofit this column
  - Sort Ascending
  - Sort Descending
  - Columns
    - Check Box
      - On
      - Off
  - Filter
    - Drop-down List
      - Starts with
      - Ends with
      - Contains
      - Equal
      - Not Equal
    - Enter Value
    - Buttons
      - Filter
      - Clear

### **Timeline**

- Today Check Box
- Task Bar
  - Mouse drag and drop
  - Mouse Hover
- Sub Task Bar
  - Mouse drag and drop
    - Start Date
    - End Date
    - Progress arrow
  - Mouse Hover
- Task Link

### **Resources**

- Task / Sub Task / Independent Task
  - Add Resources
  - Edit Resources
    - Add
    - Delete
  - Resource View

### **Dependencies**

- Task / Sub Task
  - Add
    - ID
    - Name
    - Type
      - Start-Start
      - Start-Finish
      - Finish-Start
      - Finish-Finish
  - Offset
    - Minus Day
    - Plus Day

- Delete
- Buttons
  - Save
  - Cancel
- Independent Task
- File Handling Operations**
  - New
  - Online (.gantt file)
  - Save As (.gantt file)
  - Import / Export
    - Import from Excel File
    - Export to Excel File
    - Export to PDF File
    - Export to Image File
- Configurations / Settings**
  - Default Column Settings
  - Reset Task IDs
  - First Day of Week
    - Drop-down list
  - Work Week
    - Check boxes
  - Holidays
    - From
    - To
- Search box**
- Browser window**
  - Refresh
  - Resize
  - Zoom
  - Back button
- Application Tool Bar**
  - Show Entire Gantt Chart
  - Show Remaining Tasks
  - Show Overdue Tasks

