**Table of Contents**

[Lab 1: Getting Familiar with Microsoft Project 3](#_bookmark0)

* 1. [Creating a Project file 3](#_bookmark1)
  2. [Task 4](#_bookmark2)
  3. [Resource 4](#_bookmark3)
  4. [Report 4](#_bookmark4)
  5. [Project 5](#_bookmark5)
  6. [View 5](#_bookmark6)
  7. [Format 6](#_bookmark7)

[Lab 2: Creating Gantt chart in Microsoft Project 8](#_bookmark8)

* 1. [Inserting Task in MS Project 8](#_bookmark9)
  2. [Creating task hierarchy 9](#_bookmark10)
  3. [Defining duration, start & end dates 10](#_bookmark11)

[Lab 3: Adding Predecessors, Priority, Lag and Managing Dependencies 11](#_bookmark12)

* 1. [Adding Predecessors 11](#_bookmark13)
  2. [Adding Priorities 11](#_bookmark14)
  3. [Managing Dependencies and defining Lag 12](#_bookmark15)

[Lab 4: Allocating Resources to Task 14](#_bookmark16)

* 1. [Creating Resource Pool 14](#_bookmark17)

[Lab 5: Defining and Adjusting Work Time 16](#_bookmark18)

* 1. [Creating a new calendar 16](#_bookmark19)
  2. [Changing Work Time defaults 17](#_bookmark20)
  3. [Adding exceptions and holidays to the calendar 17](#_bookmark21)

[Lab 6: Inserting Progress of each Tasks 19](#_bookmark22)

[Lab 7: Critical Task, Slack and Late Tasks 20](#_bookmark23)

[Lab 8: Generating Report 20](#_bookmark24)

[Lab 9: Critical Path Numerical 22](#_bookmark25)

[Numerical 1 22](#_bookmark26)

[Numerical 2 23](#_bookmark27)

[Lab 10: Git and GitHub 25](#_bookmark28)

1. [git status 25](#_bookmark29)
2. [git init 25](#_bookmark30)
3. [git remote add origin <…> 26](#_bookmark31)
4. [git remote 27](#_bookmark32)
5. [git add <…> 27](#_bookmark33)
6. [git commit 28](#_bookmark34)
7. [git push –u origin <…> 28](#_bookmark35)
8. [git reset 29](#_bookmark36)

[8. git fetch 29](#_bookmark37)

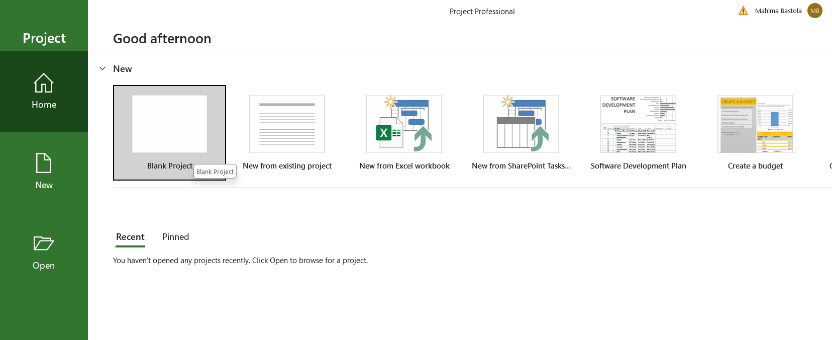
[10. git pull 31](#_bookmark38)

# Lab 1: Getting Familiar with Microsoft Project

This section covers the getting started section where we look at the different features of Microsoft Project.

## Creating a Project file

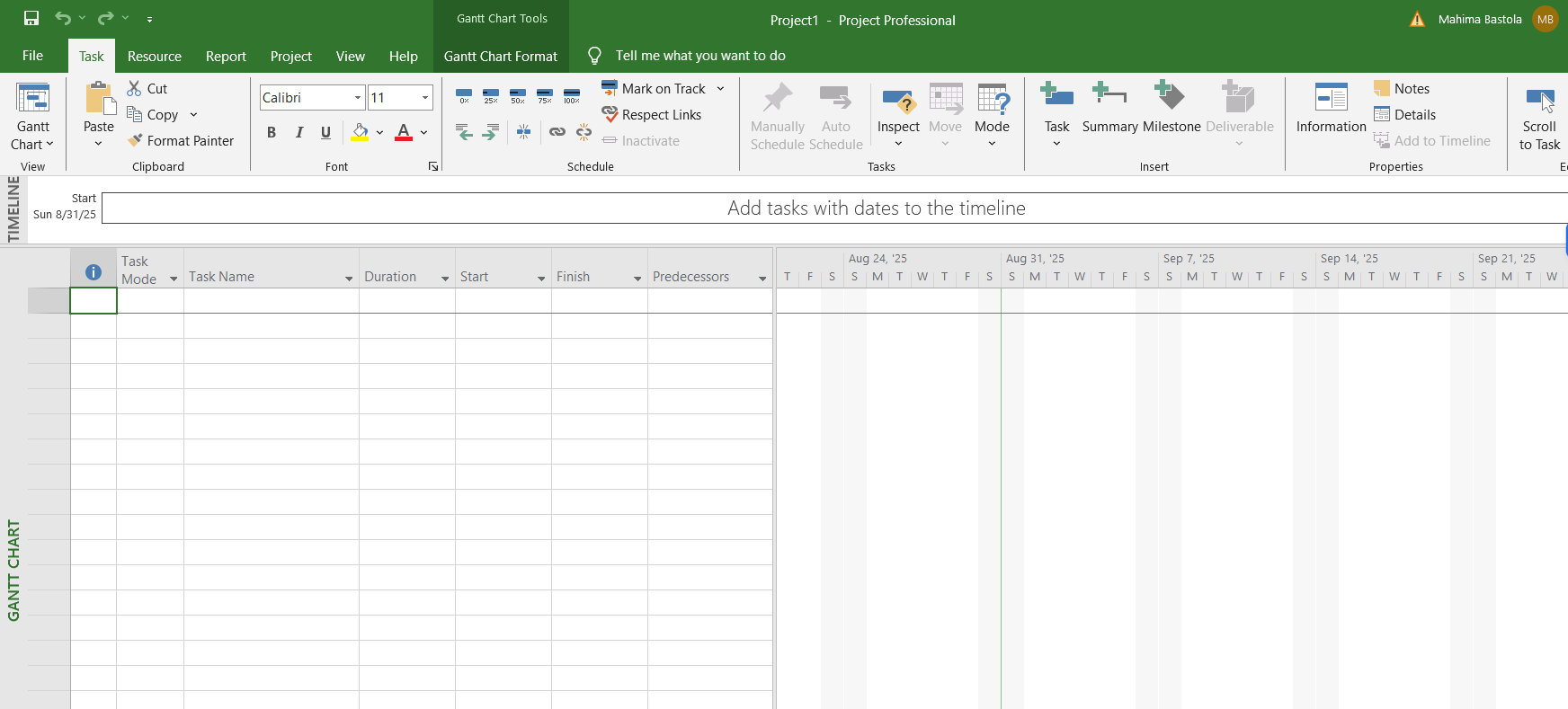
To create a project, we first install and open the Microsoft Project then we land on the start page:



From this page, we can start a project of our preference among the following options:

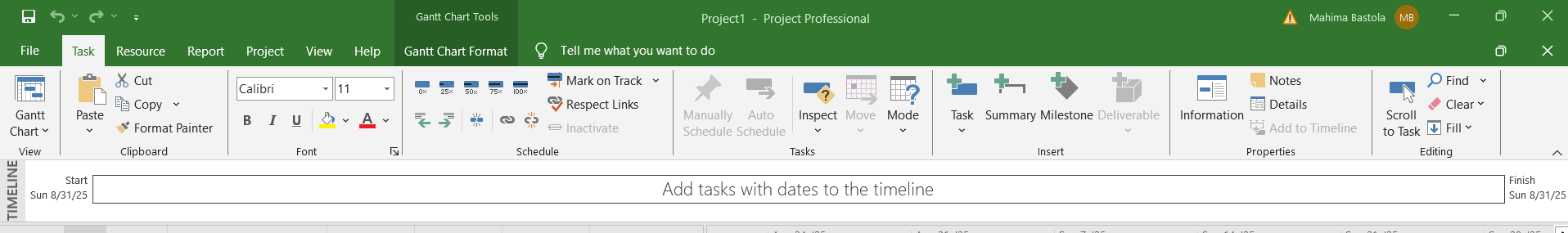
* + - Start a new *blank project*
    - Start a new project *from existing project*
    - Start a new project from *Excel workbook*
    - Start a new project using *templates*
    - Continue on the *recent* or *other* projects.

If we start a new *blank project*, we get to the following screen:



After we create a project file, we see multiple interfaces in menu bar namely, *Task, Resource, Report, Project, View and Format.*

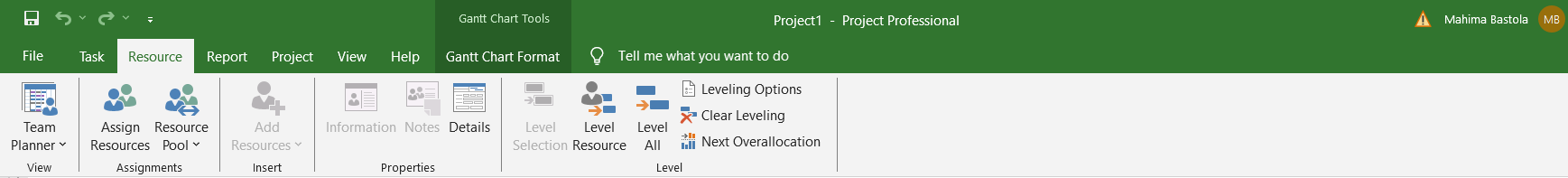
## Task

****

The first menu is Task. It contains all the functions that allows us to manage the tasks to be performed in a project. These include:

* + - Task Completion
    - Task Indentation
    - Task Linkage
    - Task Schedule and so on.

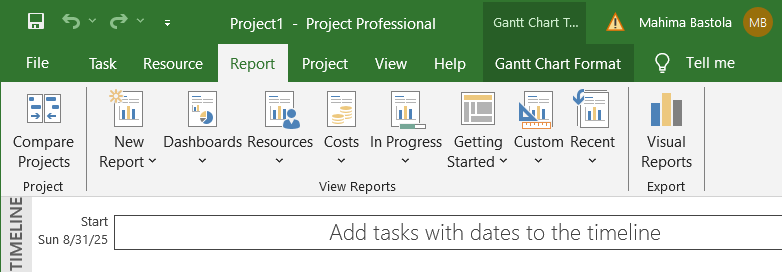
## Resource

****

The second menu is Resource. It allows us to manage the resources available and resources required for the project. These functions include:

* + - Adding Resources
    - Managing Resource Pool
    - Assigning Resources
    - Managing Resource Level

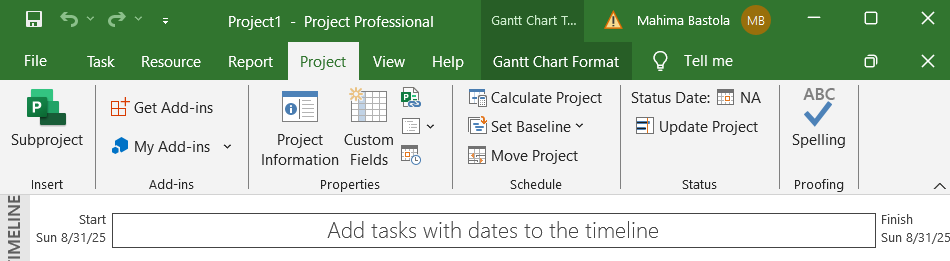
## Report

****

This menu is used when a report needs to be generated for managerial purpose. We can prepare all kinds of reports from the Microsoft Project. Some features related to report include:

* + - Dashboards: overview of selected category; Cost, Project, Task, etc.
    - Resources
    - Costs: cash flows, earned value, task cost, resource cost
    - In Progress: critical task, late task, slipping task, milestones
    - Visual Reports: visual representation of the project
    - Custom: report with user defined format

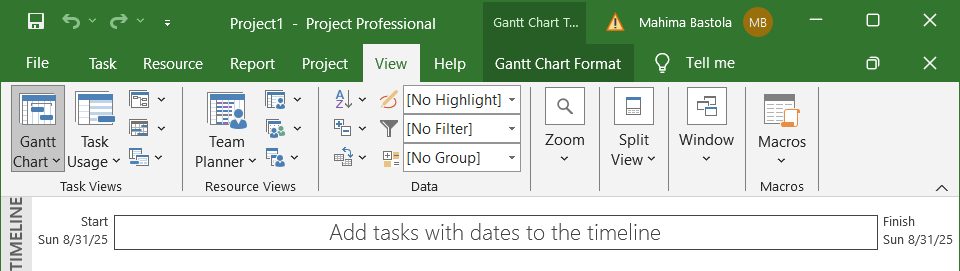
## Project

****

This menu mainly allows additional customization to the project such as addressing extra requirement or adding constraints or variables to the task and resources. Similarly, it also facilitates work breakdown and managing work time. Some of its most used options are:

* + - Change working time
    - Set Baseline
    - Links Between Projects
    - Custom Fields

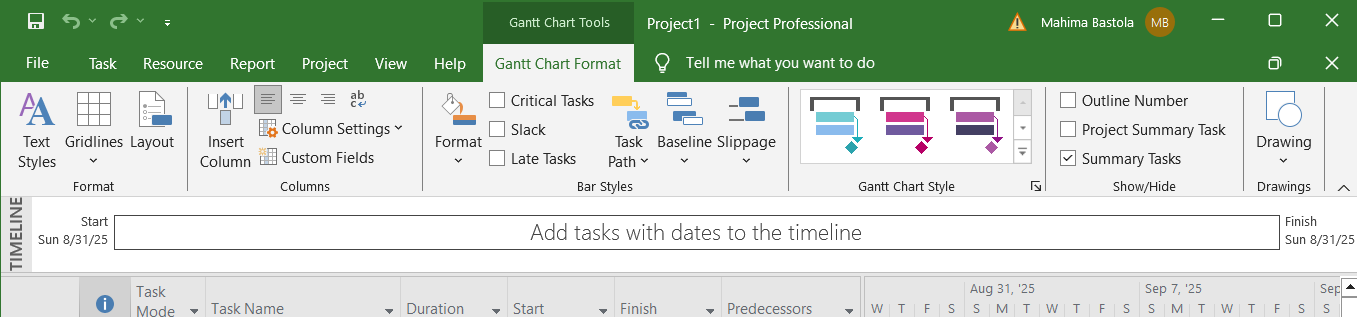
## View

****

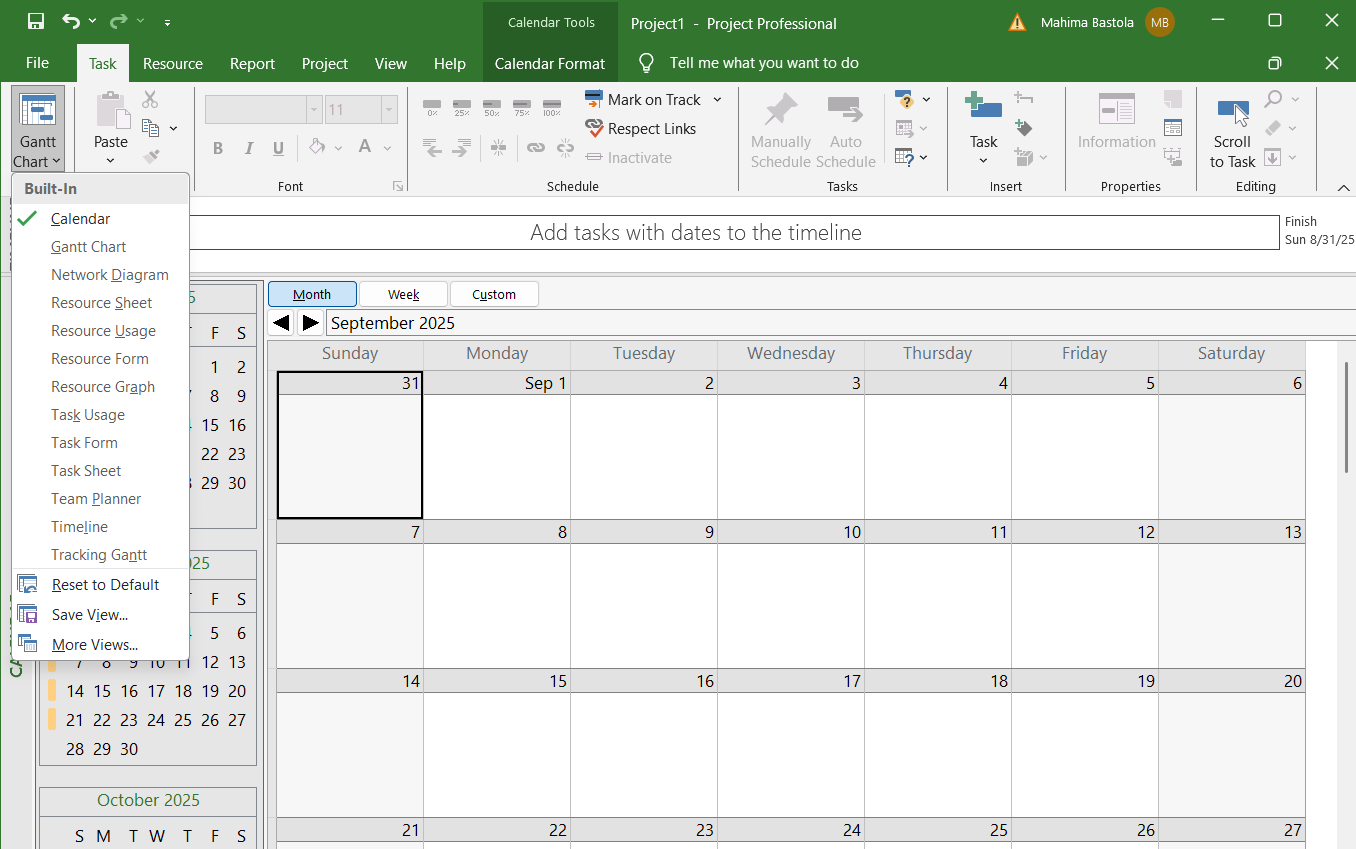
View menu lets us choose how we want to view the different elements of a projects such as task, resource, data, etc. It also lets us customize the view format of the software. Some functions on this menu are:

* + - Task Views
    - Resource Views
    - Data Filter
    - Split View

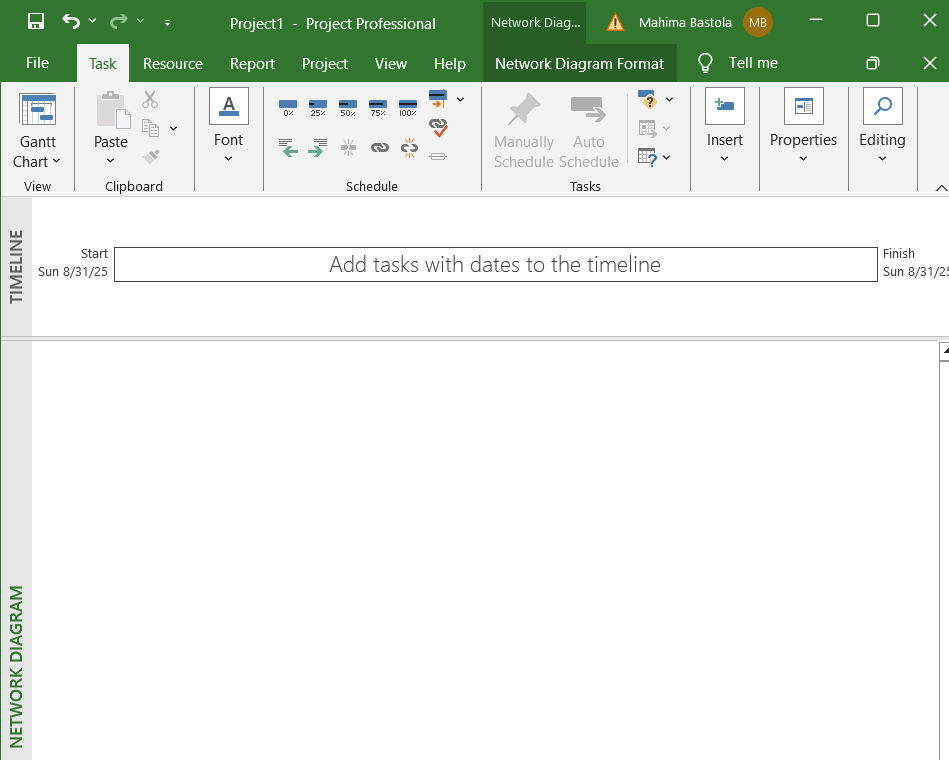
## Format

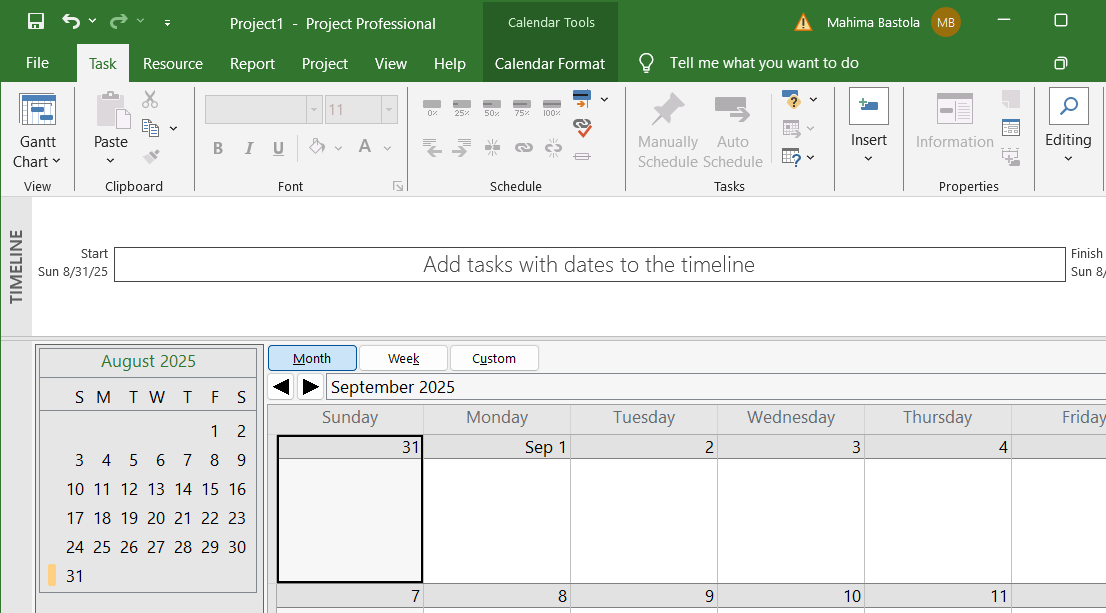
****

Format mew lets us edit the formatting details of the sheet we are working on. As the sheet we work on could change as per our requirement, the format menu also changes along with our choice of view. In the above figure the format is based on Gantt Chart Tools. It changes if we choose any other view as following:



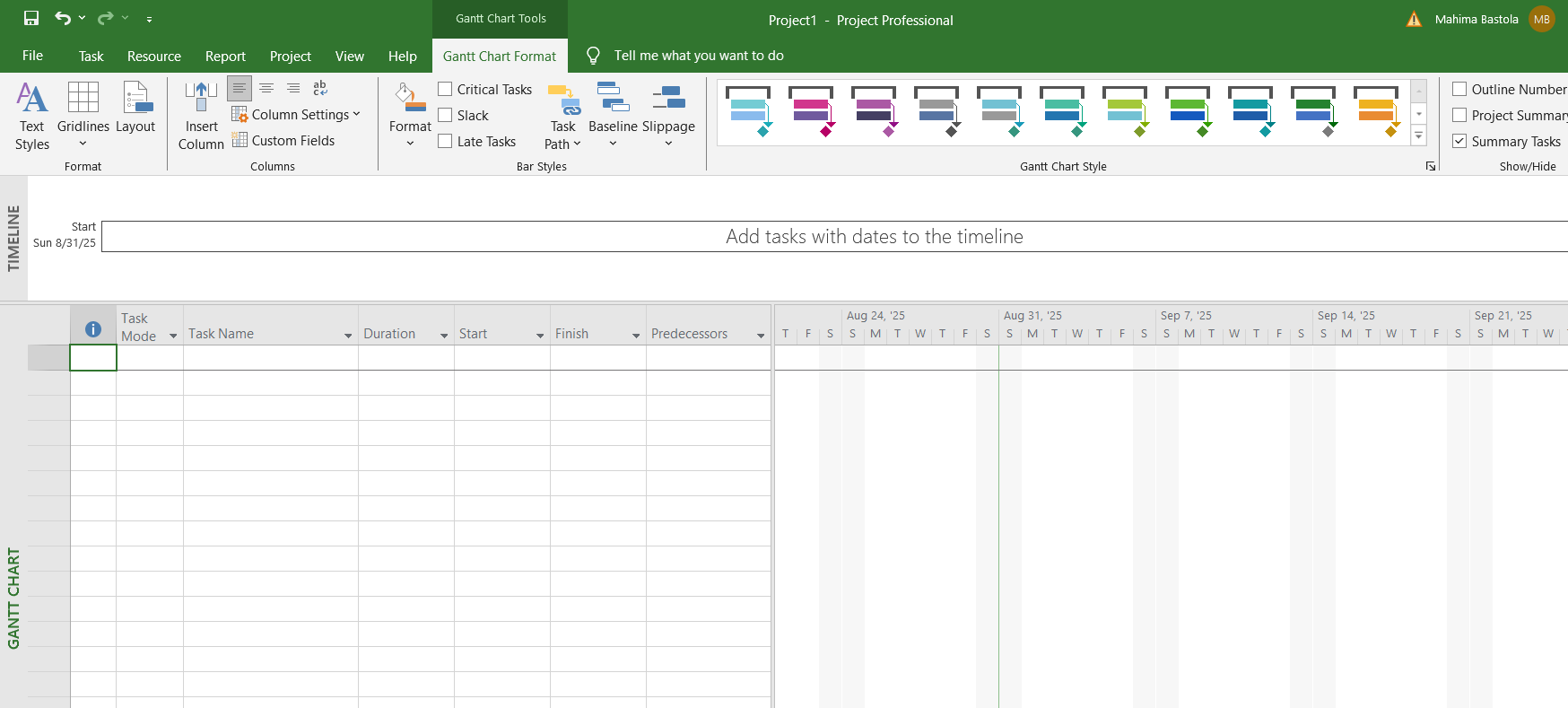
All the different views options available on the drop down menu on the left have their own different format menu options. Few of them are shown in the following pictures

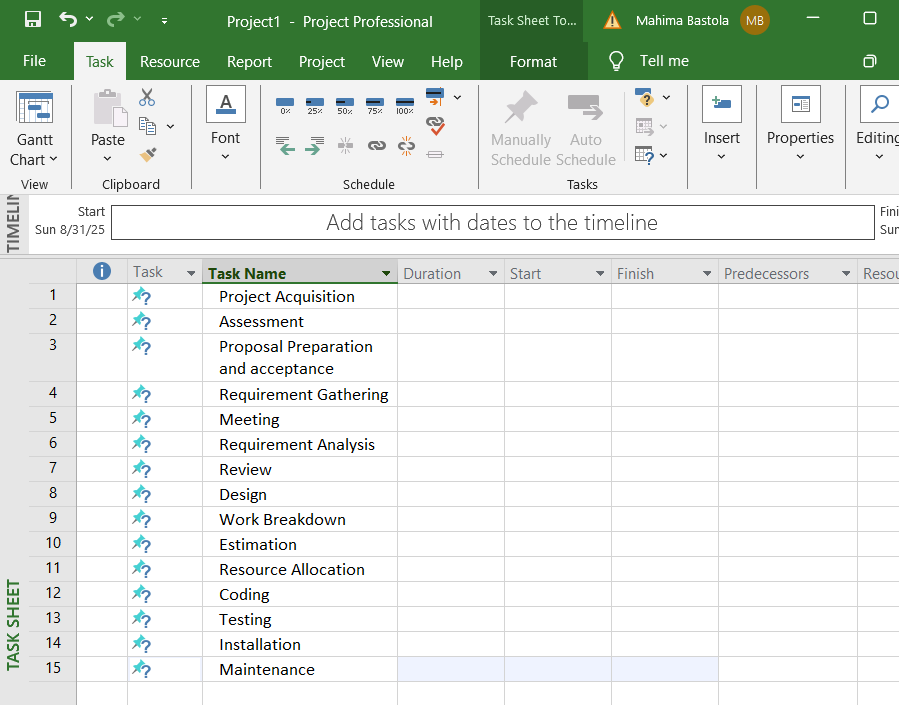




# Lab 2: Creating Gantt chart in Microsoft Project

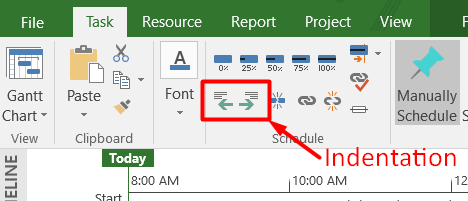
## Inserting Task in MS Project

1. Open MS Project and create a *new blank project*.
2. From the view options, choose *Gantt Chart* option as shown below:
3. Add the tasks in the Task Name column of the sheet. You can either copy & paste the tasks from another list or click in the text field and type the name of each task. It is suggest to list the task in progressive order to keep the project organized.

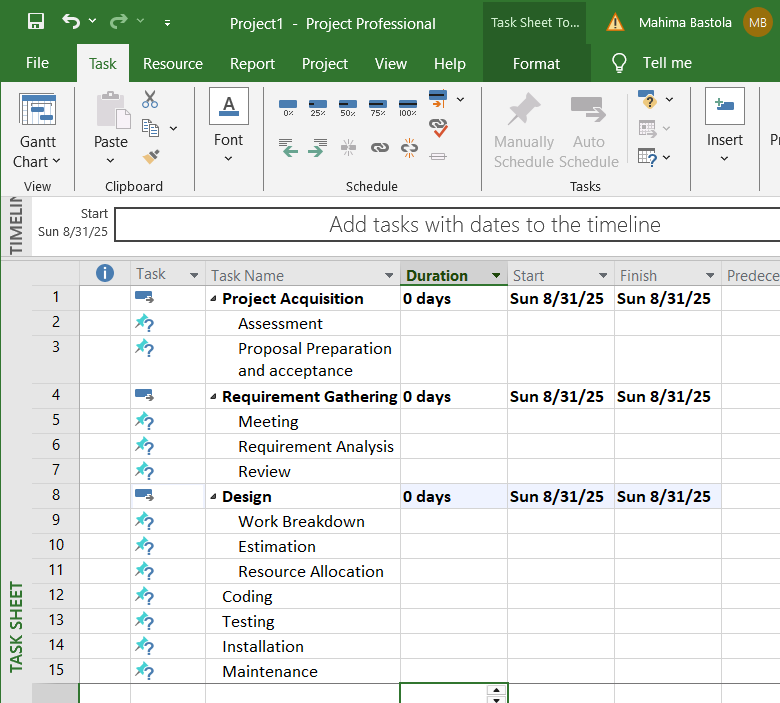
Currently, Gantt chart will not appear as the time parameter is yet to be added to the task list.

## Creating task hierarchy

After the tasks have been input in order, we can put hierarchy in the task using indentation. We can either indent a task or outdent a task using the icons in *Schedule* section of the Task menu bar.



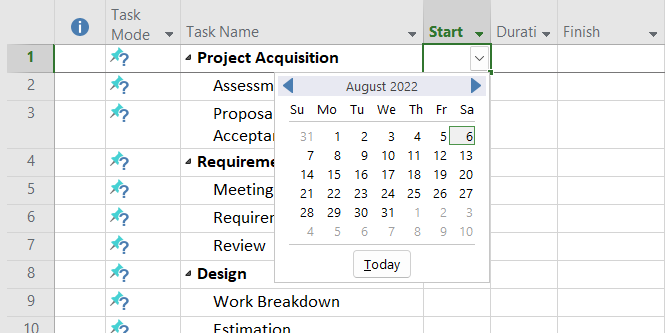
As our task list is in its normal state, no tasks can be outdented. We can create subtasks by indenting the desired tasks. When we indent a task, it becomes the subtask of the task above it. The more a task is indented the more its hierarchy decreases.



## Defining duration, start & end dates

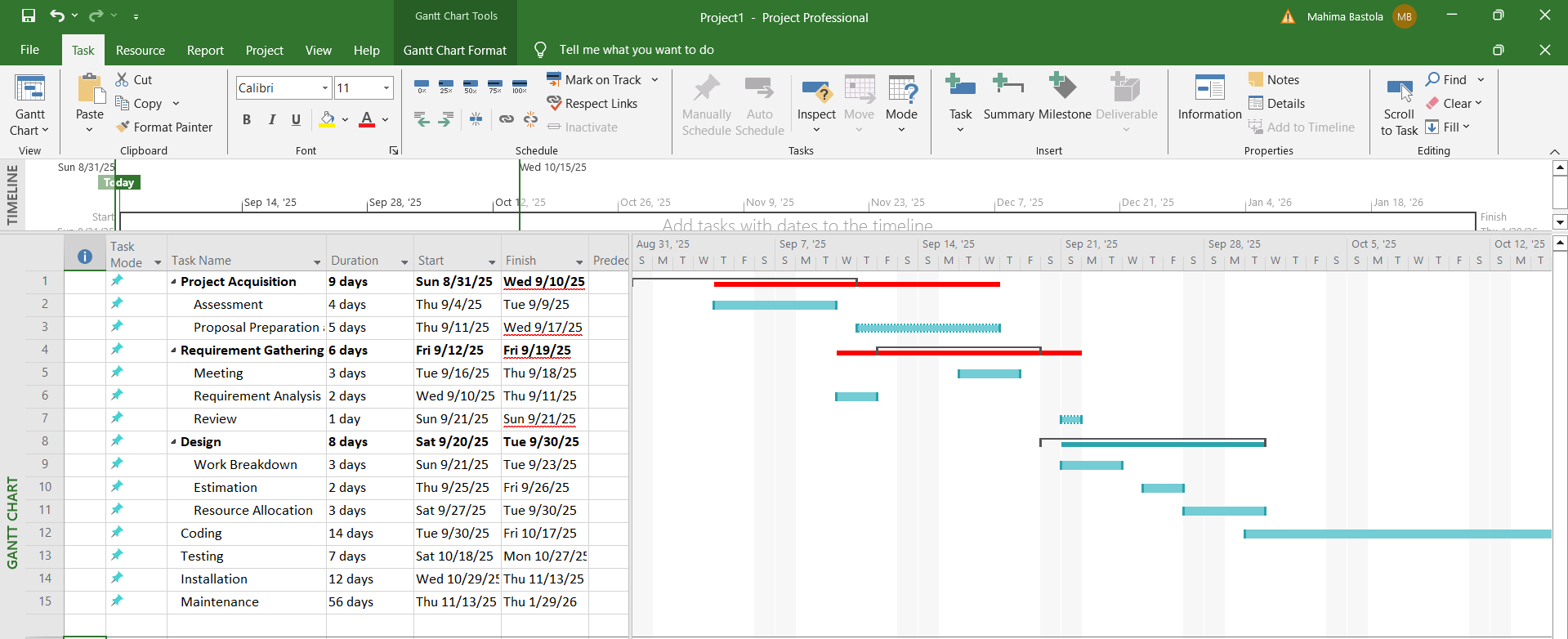
Defining start date and finish date are similar processes with a.

* + - Click on the start or finish date, then you can either enter the date manually or select one from the date picker.



* + - After, selecting the start and end date, the duration will be automatically calculated based on the work-time provided.
    - An alternative way of entering start and finish is to enter one of them and enter the estimated duration required to complete the task and the corresponding date will be automatically updated.

After assigning start and finish date for all of the tasks, the gantt chart will be created on the right hand side as shown in the figure below.

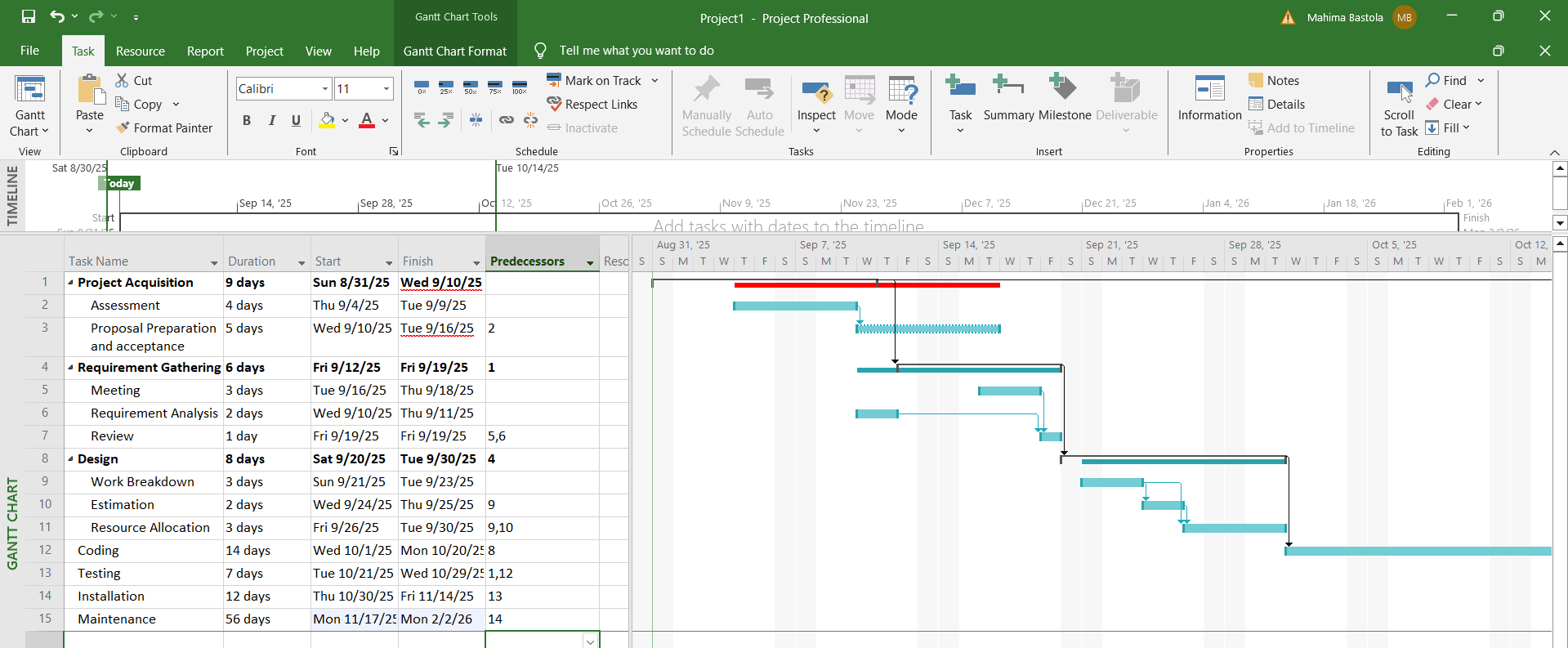


However, it is not an accurate gantt chart as it does not include linkages.

# Lab 3: Adding Predecessors, Priority, Lag and Managing Dependencies

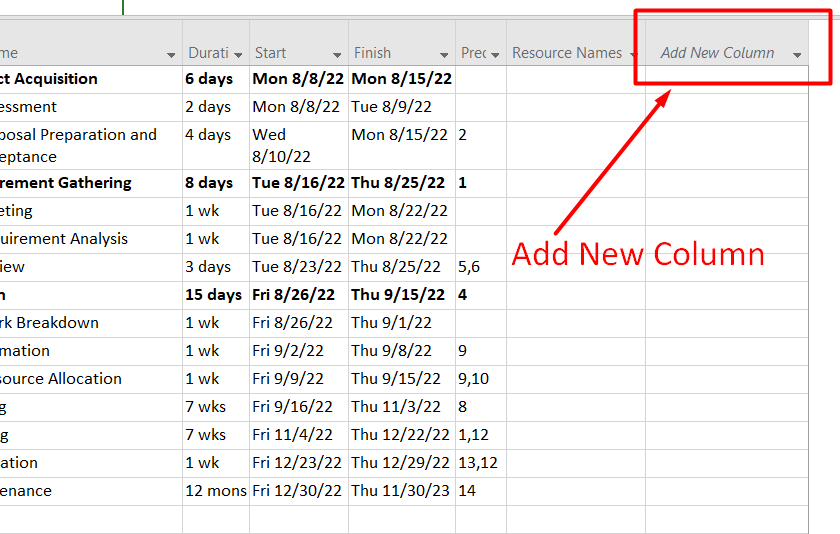
## Adding Predecessors

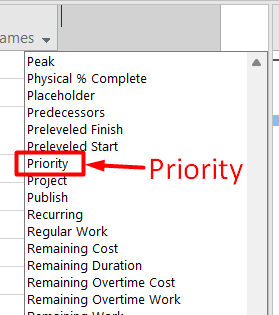
We can add predecessors to tasks, simply by adding the task number in the predecessor’s column. Task number of the predecessor task or tasks is typed into the predecessor column corresponding to the dependent task. This adds the simple finish-to-start dependency between the tasks. This linkage can also be seen in the Gantt chart.



## Adding Priorities

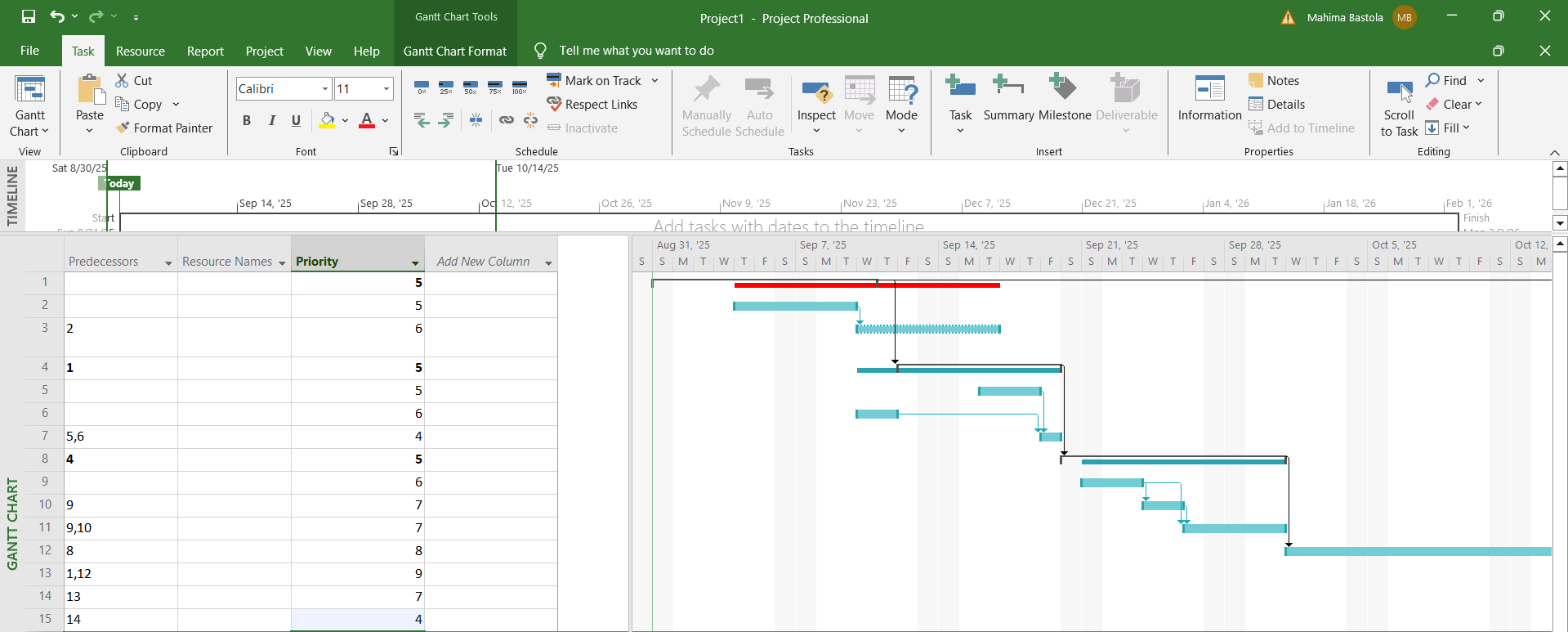
To add priority we have to add new column in the sheet. The *Add New Column* can be found at the end of the column in the sheet.





After clicking on *Add New Column*, we will get a list of options to choose from; among them we choose the *Priority* option to create a new column for Priority.

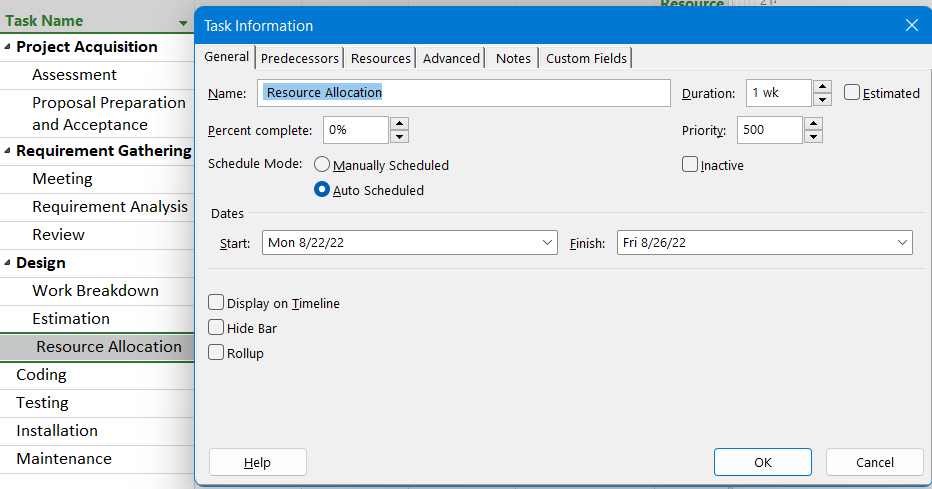
After the priority column is created, we can provide desired priority points to each task as required.



Adding priority does not bring any changes to the Gantt chart.

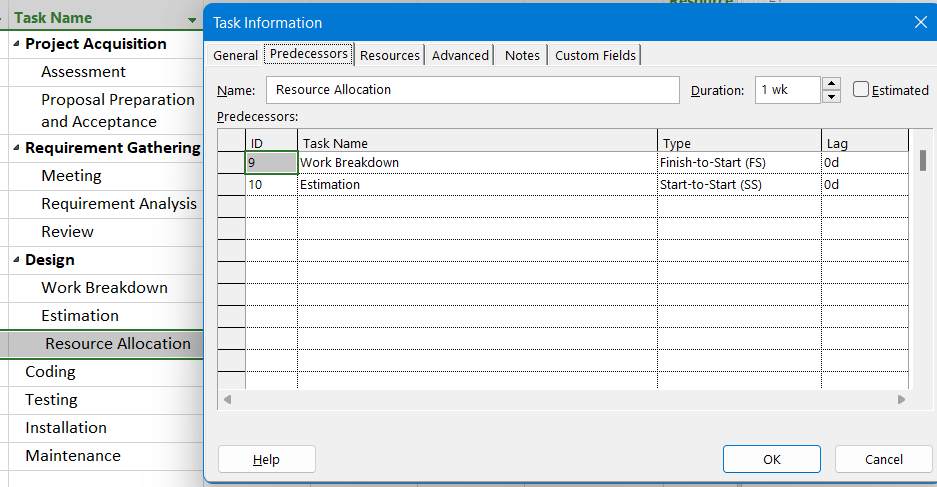
## Managing Dependencies and defining Lag

We added priority and predecessors using the columns in the sheet. However, we can double click on the task for more detailed control on the dependencies and other aspects of the task. It will open a Task menu containing tabs like *General, Predecessors, Resources,* and so on.



To manage the types of dependencies and define lag, click on the *Predecessors* tab. It will provide us with a sheet containing the list of predecessors along with its type and lag.

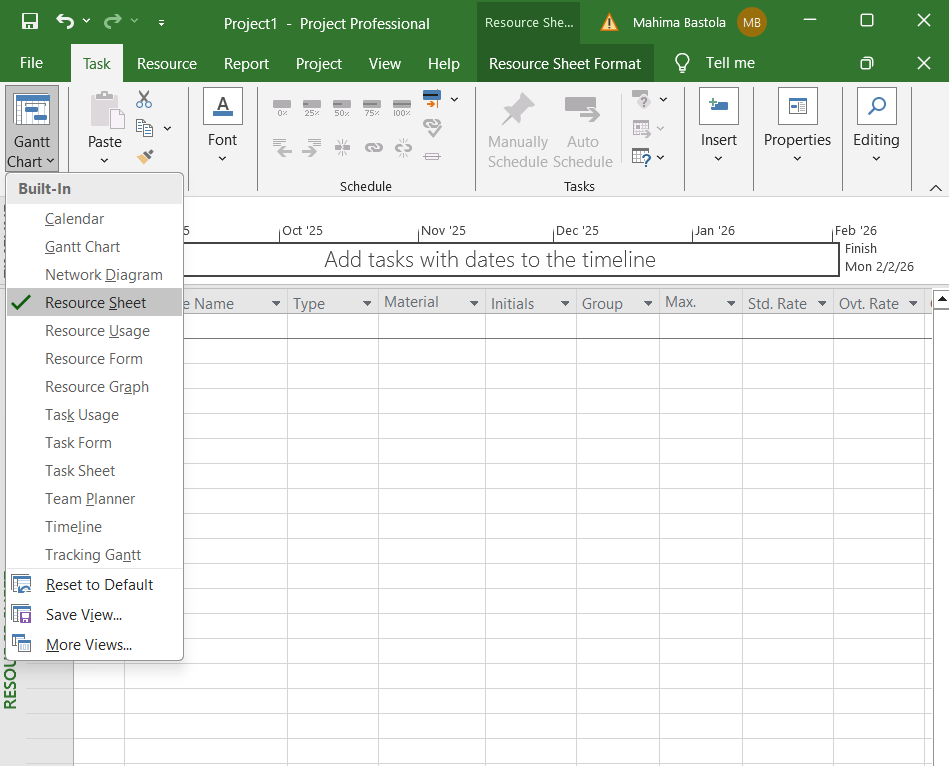
We can define the type of dependency and add desired amount of lag from this window.



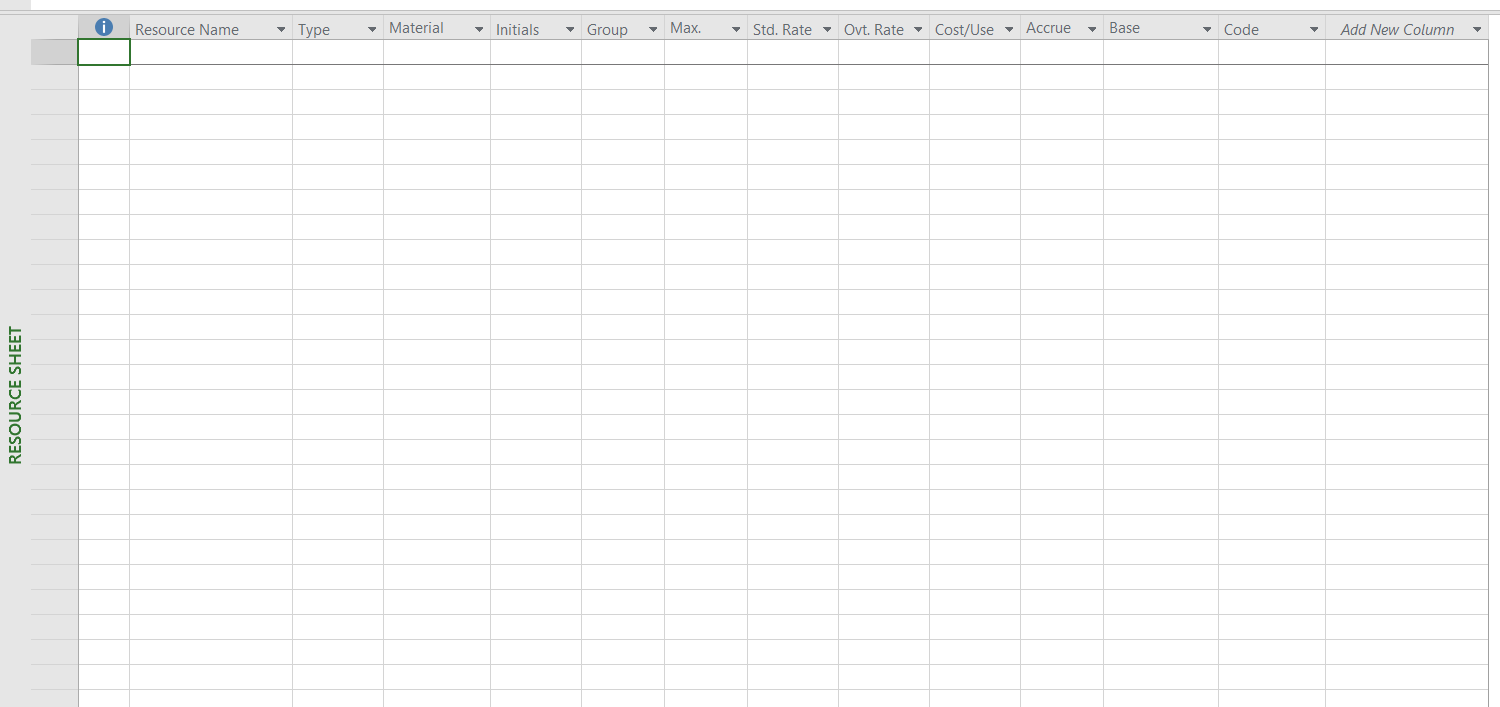
# Lab 4: Allocating Resources to Task

## Creating Resource Pool

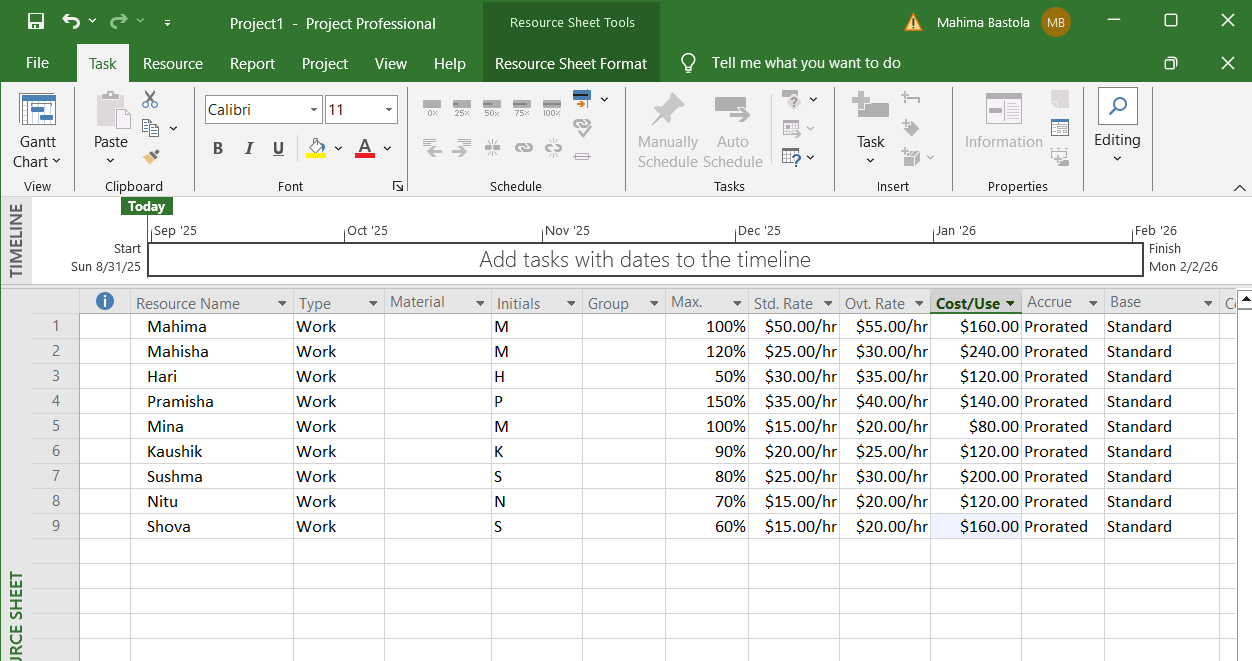
To create a resource pool, we first need to access the resource sheet. To access resource sheet, click on the view options or *Gantt chart* on top left corner of the menu ribbon. This will bring a drop down menu with different view options choose *Resource Sheet*.



After selecting *Resource Sheet*, we can now add resources through the sheet as follows:

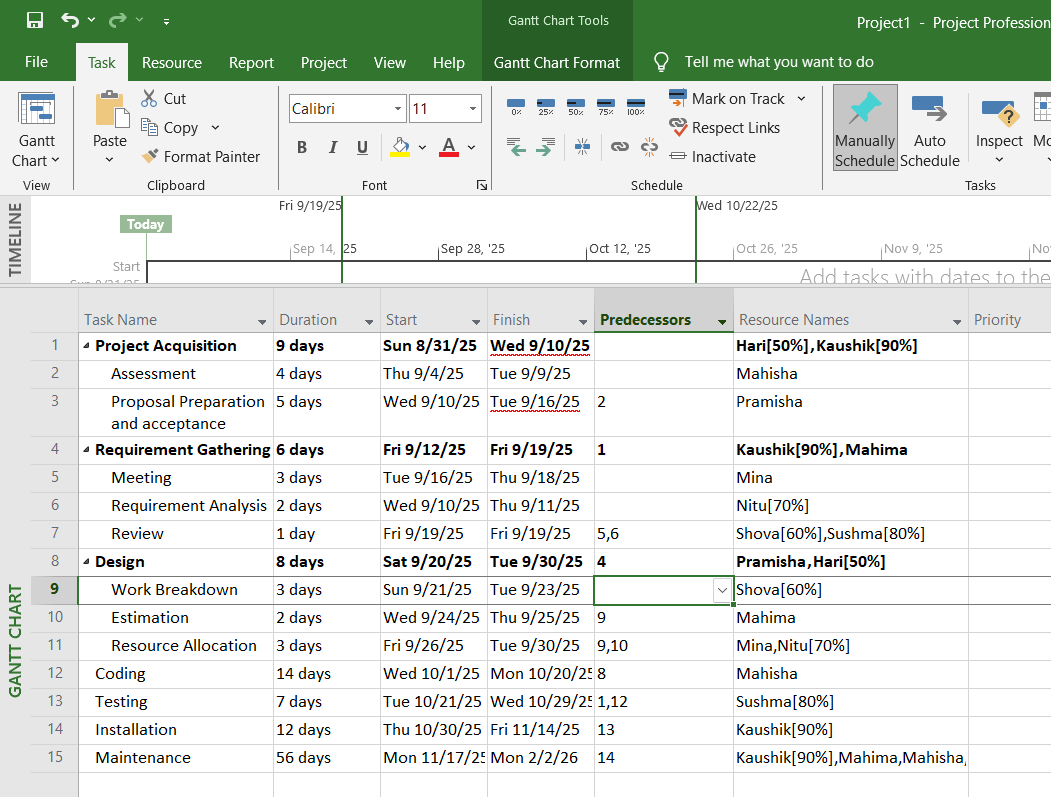


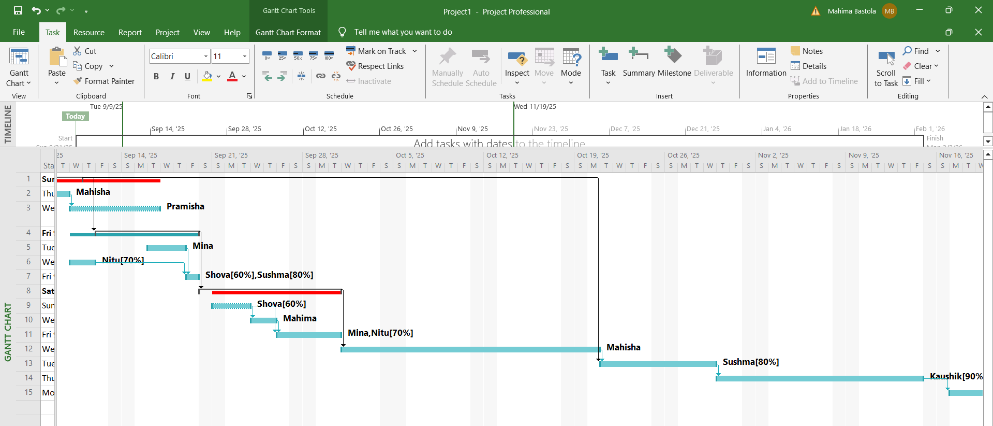
Now, we add the resources by filling in the necessary details related to the resource. For this lab work, we used employees as resources and after entering required data, it looks like the following:



The sheet shows the list of available resources during the project along with their respective details.

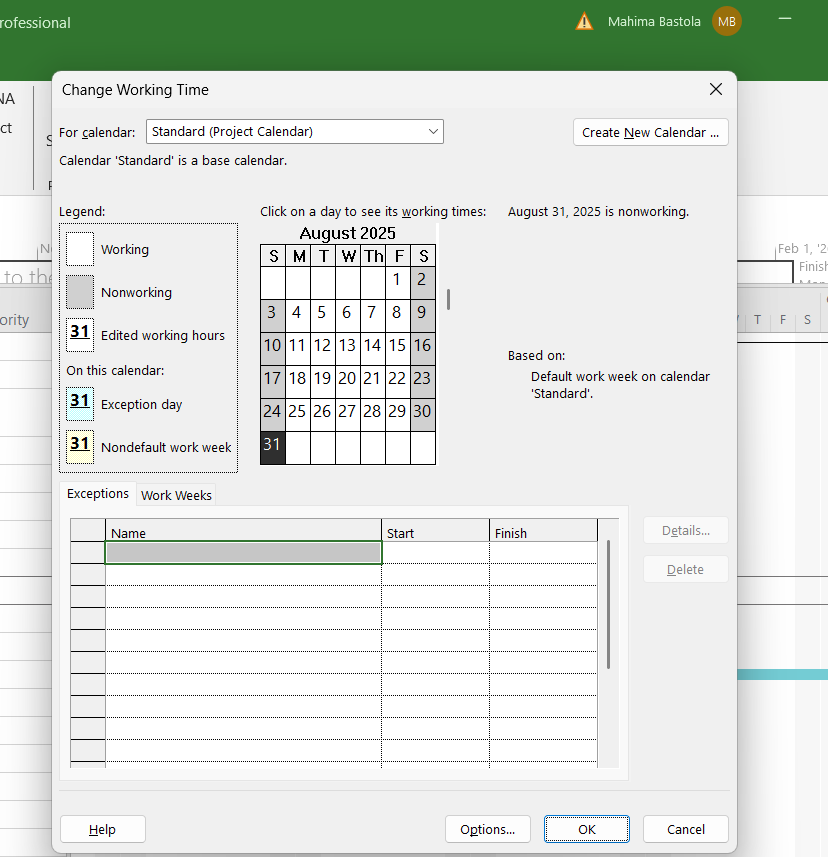
Now, we again choose *Gantt chart* from the view menu options and input desired resource names for each task in the *Resource Names* column. We can also assign multiple resources to a single task. After the resource allocation, our sheets look as following:





# Lab 5: Defining and Adjusting Work Time

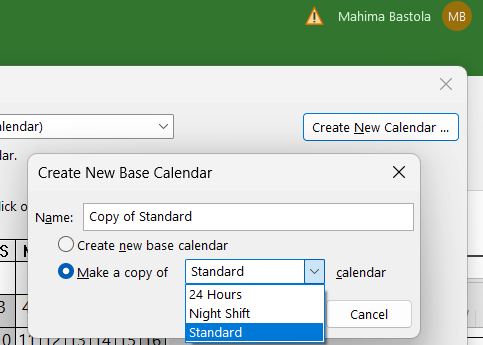
To make any changes related to work time, we go to the *Project* tab on the menu bar. Then we click on *Change Working Time.* This opens up a dialog box named ‘Change Working Time,’ where we can manage, edit or adjust anything related to the work time of the project.



## Creating a new calendar

From the *Change Working Time* dialog box, we can adjust time on the standard calendar or create a new calendar.

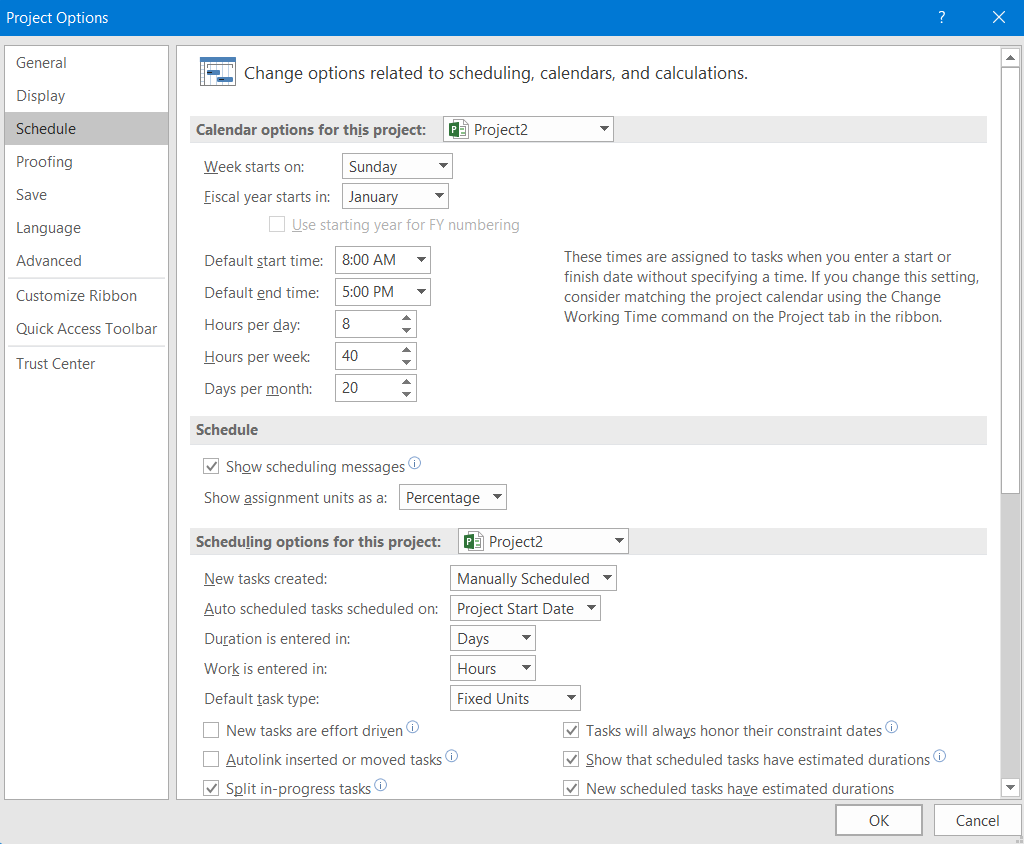
To create a new calendar click on *Create New Calendar* on the top right corner. This will open a dialog box asking how we want to create the new calendar. This will open a new dialog box which will allow us to change calendar defaults.



We can either create a completely new base calendar or create a copy of already available *Standard, Night Shift, or 24 Hours* calendar.

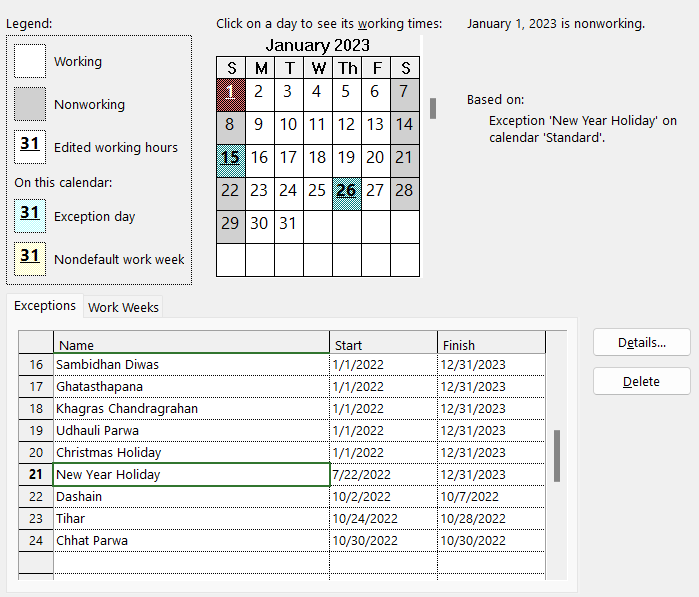
## Changing Work Time defaults

Certain schedule elements such as start of the day, work start time, work end time, hours per day, etc. are set at a default value. The default values can be adjusted as per our preferences. To change defaults, click on *Options* button at the bottom right for the *Change Working Time* dialog box.

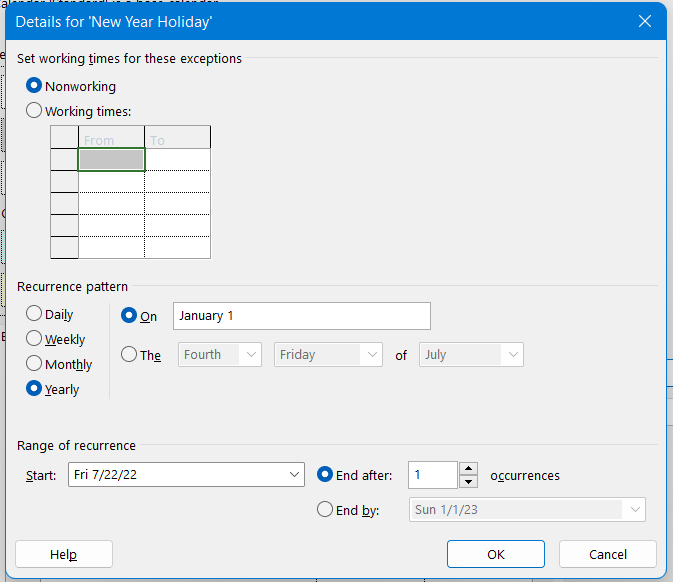


## Adding exceptions and holidays to the calendar

To add any exceptions or holidays to the calendar, we simply list down the exceptions in the exception sheet of *Change Working Time* dialog box. We have to input the name of the exception along with its start date and finish date.



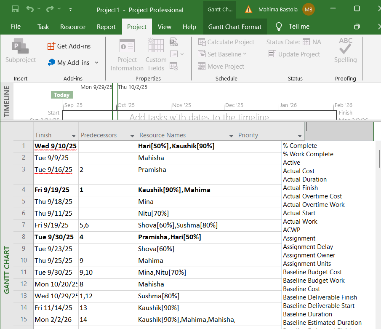
We can also insert additional details to the exceptions such as half-day work, recurrence pattern or range of recurrence. This can be done by simply double clicking on the exception’s name or clicking on the *Details* button present on the right side.

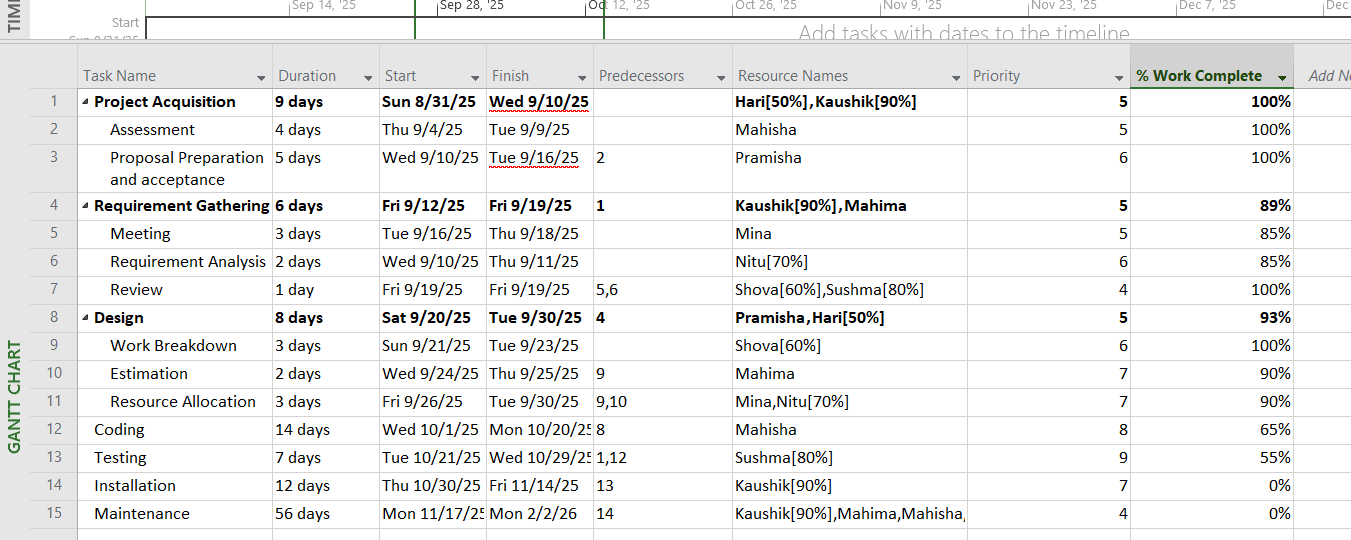
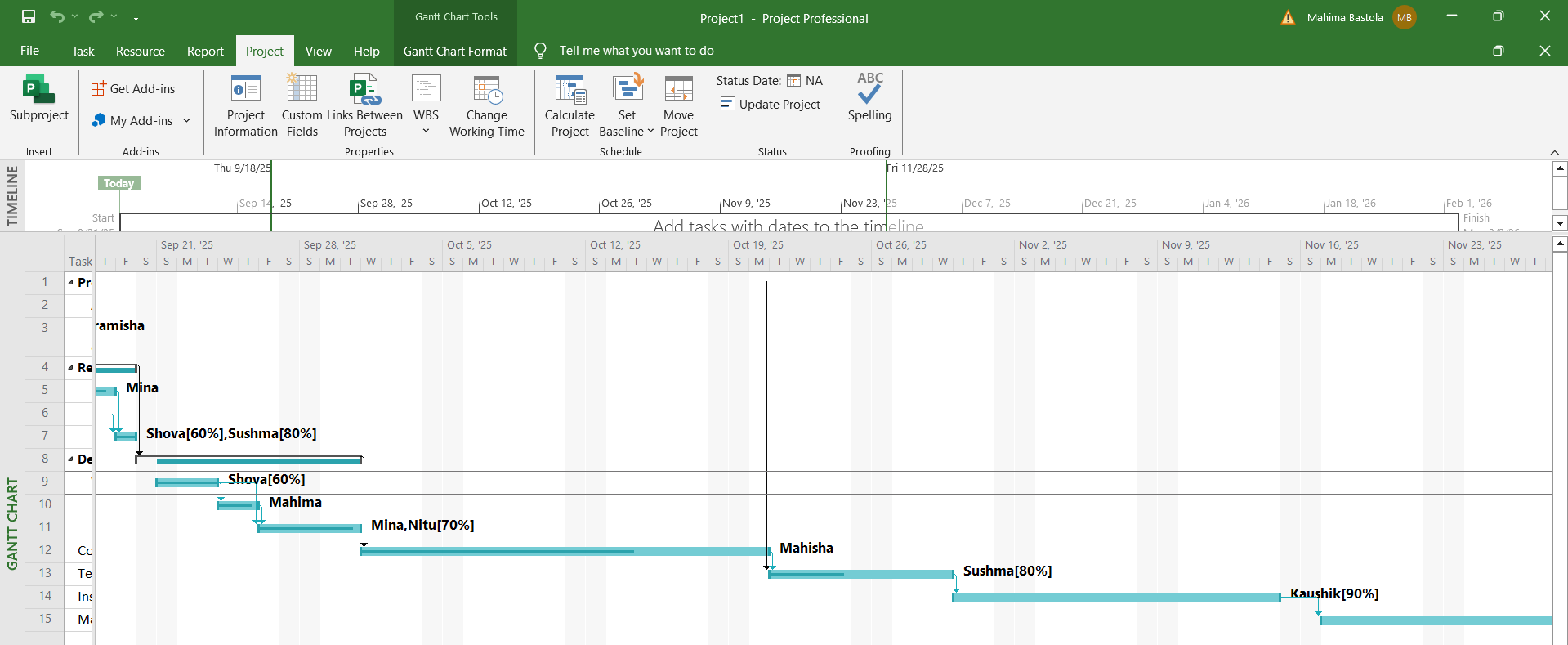


# Lab 6: Inserting Progress of each Tasks

To add progress details for tasks we add a new column. This process is similar to adding priority.

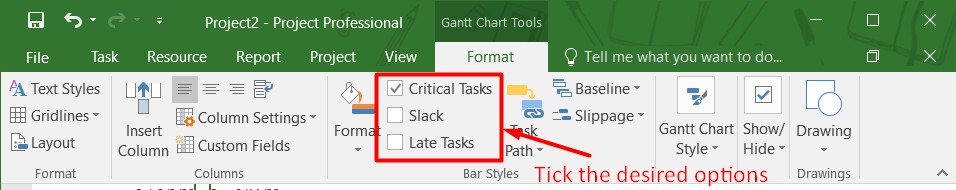
Click on add new column in the Gantt chart view then select ‘% Work Complete.’ In this field we can add the percentage of work completed of each task.



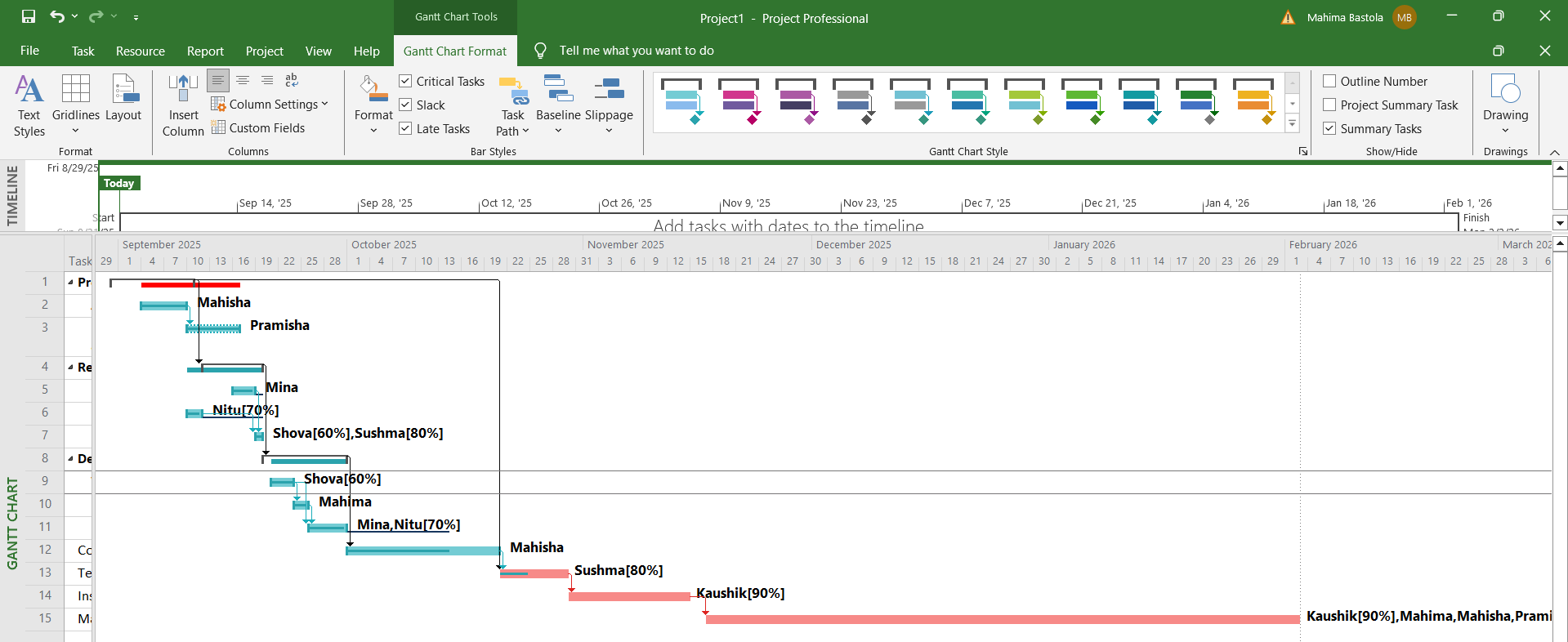
In the chart, the dark line in the task bar represent task progress.

# Lab 7: Critical Task, Slack and Late Tasks

Discovering critical task, slack and late task is a child’s play in Microsoft Project. For this, we go to *Format* menu on *Gantt chart* view then on bar styles tick the desired option among cirtical task, slack and late tasks.



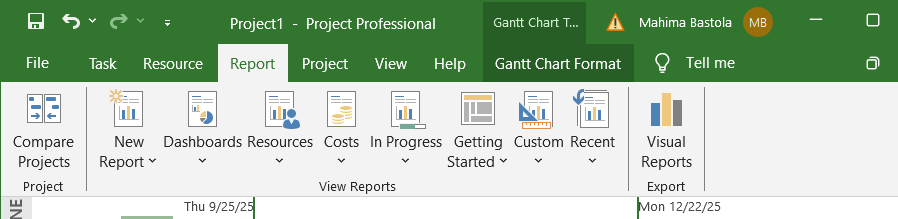
After we tick the desired options, the output shows in Gantt chart.



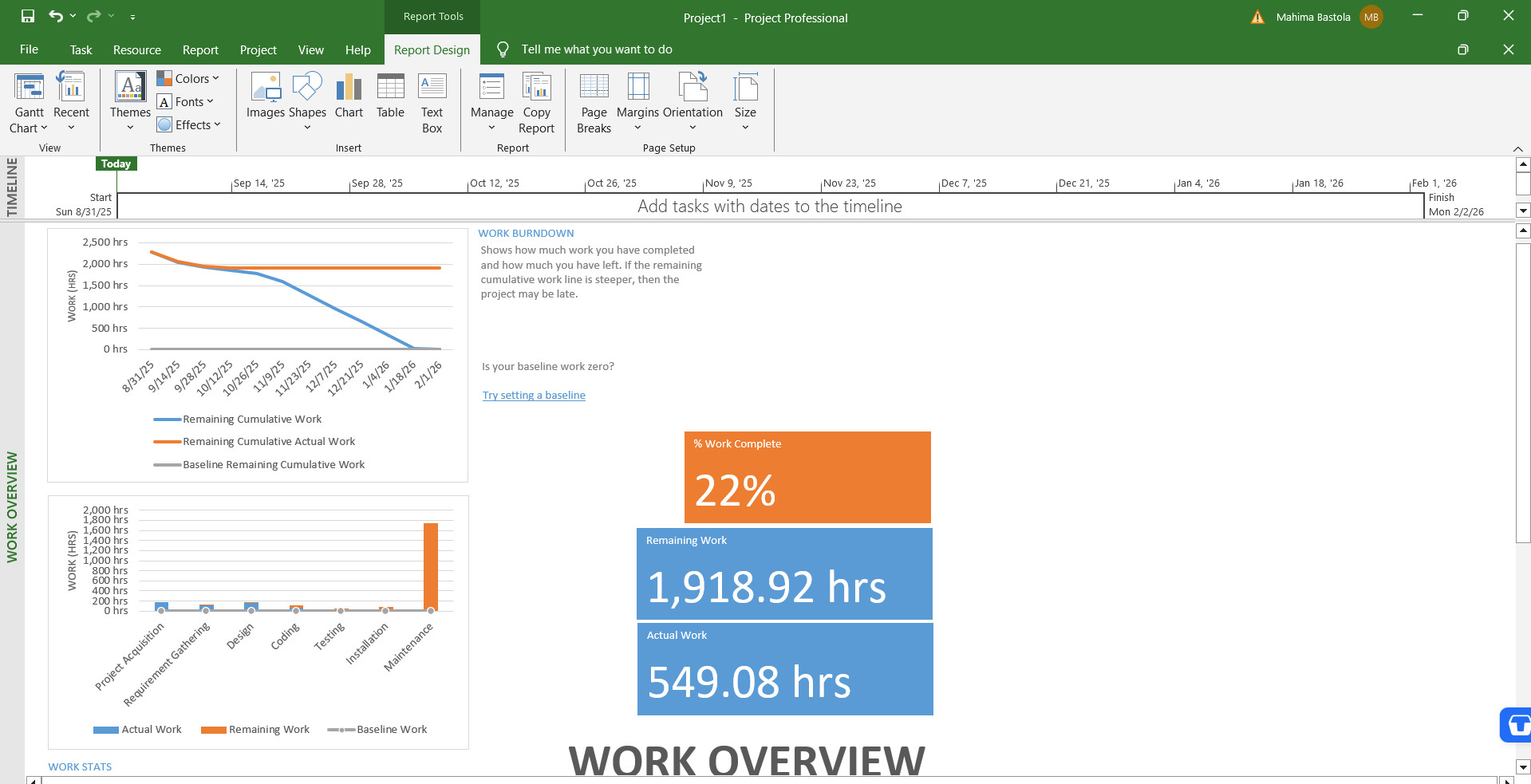
In above Gantt chart, the task bar with stars at its start and end are the critical tasks and the task bar with circled down symbol are late tasks. Similarly, the lines that extend for some taskbars are slack.

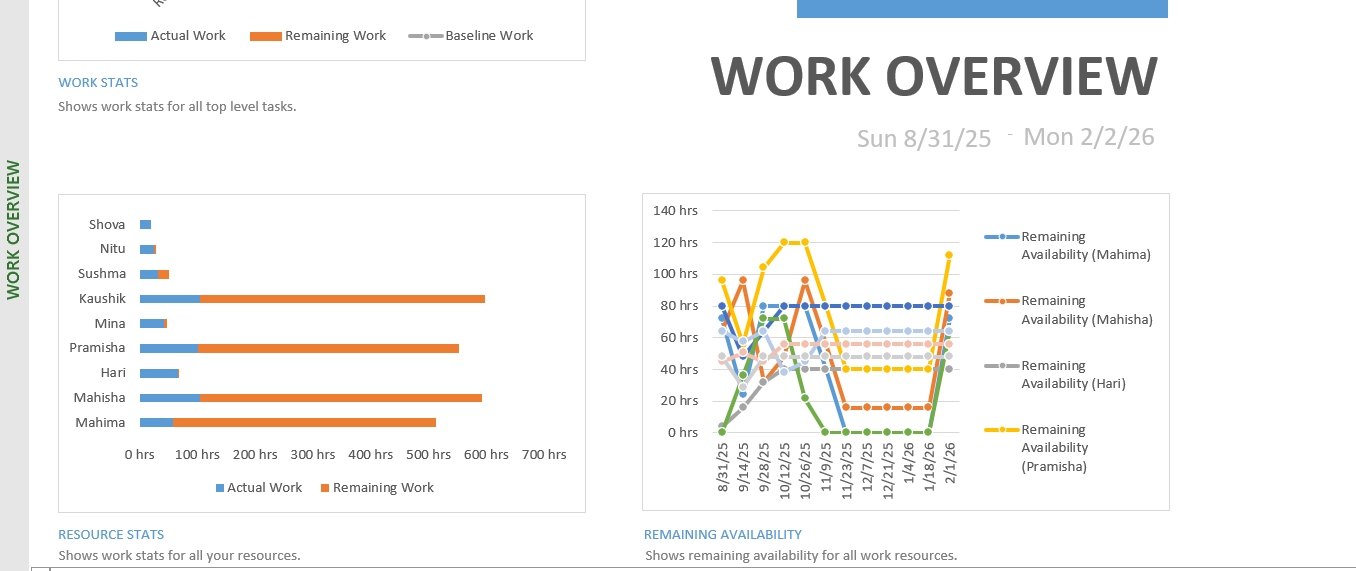
# Lab 8: Generating Report

Preparing reports is also an easy tasks with the feature of Microsoft Project. To prepare report, we go to *Report* menu then select the desired report. The report can be an overview report or related to resources, costs or progress.



An example of report (work overview) is as following:





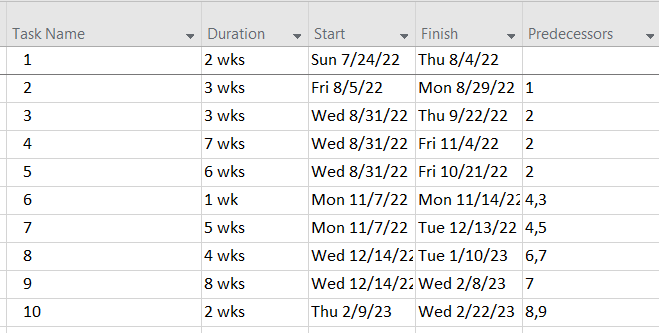
# Lab 9: Critical Path Numerical

For this lab we solved two critical path numerical using Microsoft Project.

## Numerical 1:

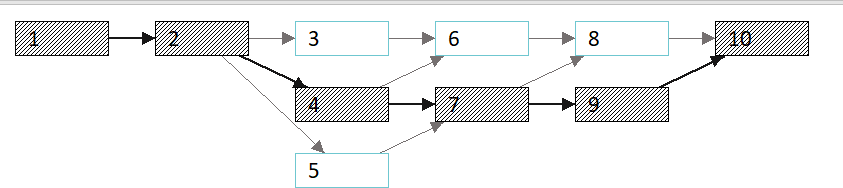
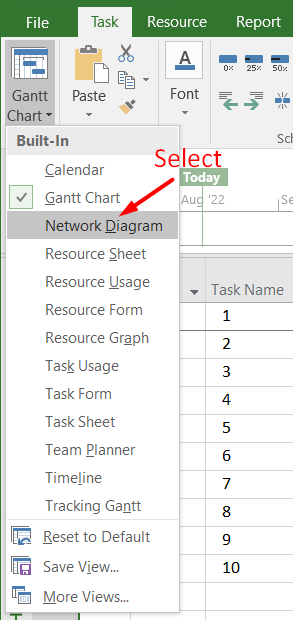
|  |  |  |
| --- | --- | --- |
| **Activity** | **Preceding Activity** | **Duration (weeks)** |
| 1 | - | 2 |
| 2 | 1 | 3 |
| 3 | 2 | 3 |
| 4 | 2 | 7 |
| 5 | 2 | 6 |
| 6 | 4,3 | 1 |
| 7 | 4,5 | 5 |
| 8 | 6,7 | 4 |
| 9 | 7 | 8 |
| 10 | 8,9 | 2 |

For finding critical path, we just enter the given details on Project sheet as follows:



For start and finish, we can select a random start date and let project auto schedule the remaining ones.

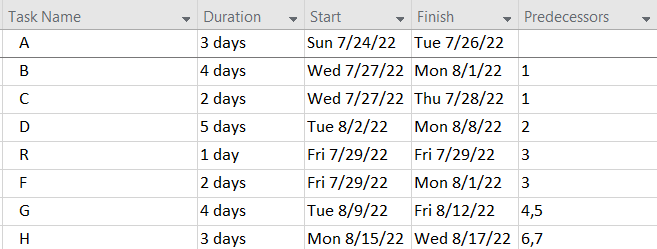
After the input of given date, we change the view from *Gantt chart* to *Network Diagram.*

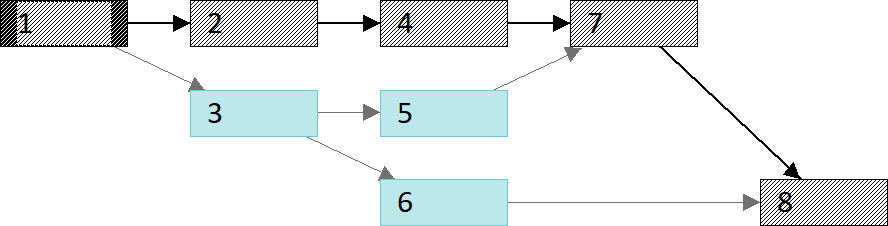


Here, the darker shades represent critical path. Hence, the critical path is 1247910.

## Numerical 2:

|  |  |  |
| --- | --- | --- |
| **Activity** | **Preceding Activity** | **Duration (days)** |
| A | - | 3 |
| B | A | 4 |
| C | A | 2 |
| D | B | 5 |
| E | C | 1 |
| F | C | 2 |
| G | D,E | 4 |
| H | F,G | 3 |



****

Hence, the crital path is 12478 or ABDGH.

# Lab 10: Git and GitHub

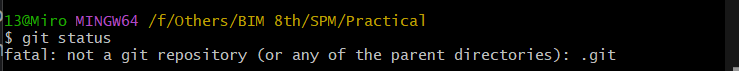
Git and GitHub are version control program that allows us to track and manage changes to software code or a project. Its main objective is source control and its importance can be seen over time when the project gets more and more complex.

## git status

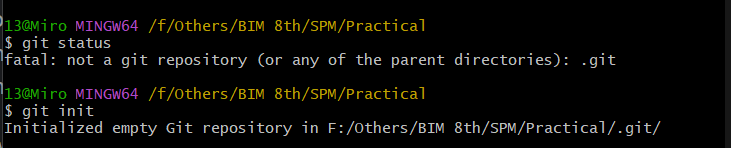
While working on git, we need to know what the state of our folders, files and repositories. This details are provided by ‘git status’

## git init

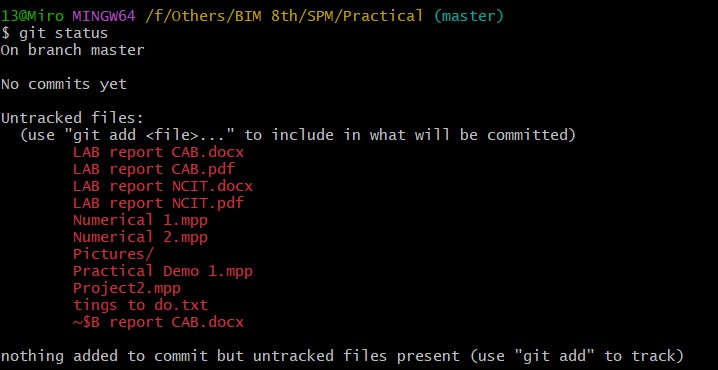
Before doing anything on git, we first have to create a repository and git init allows us to do so.



When we first check git status on a random folder on our PC, we get the message that it is not a repository. Hence any work done here is not stored on the GitHub. So, the first step is to make the folder a repository using ‘git init.’



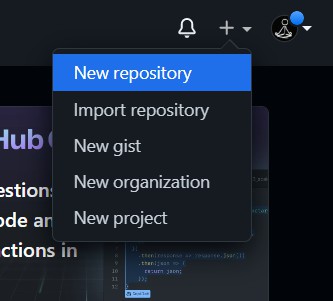
Here, ‘git init’ has create a local repository on our PC. Now if we use *git init,* it gives us details on the files present on our repositories.



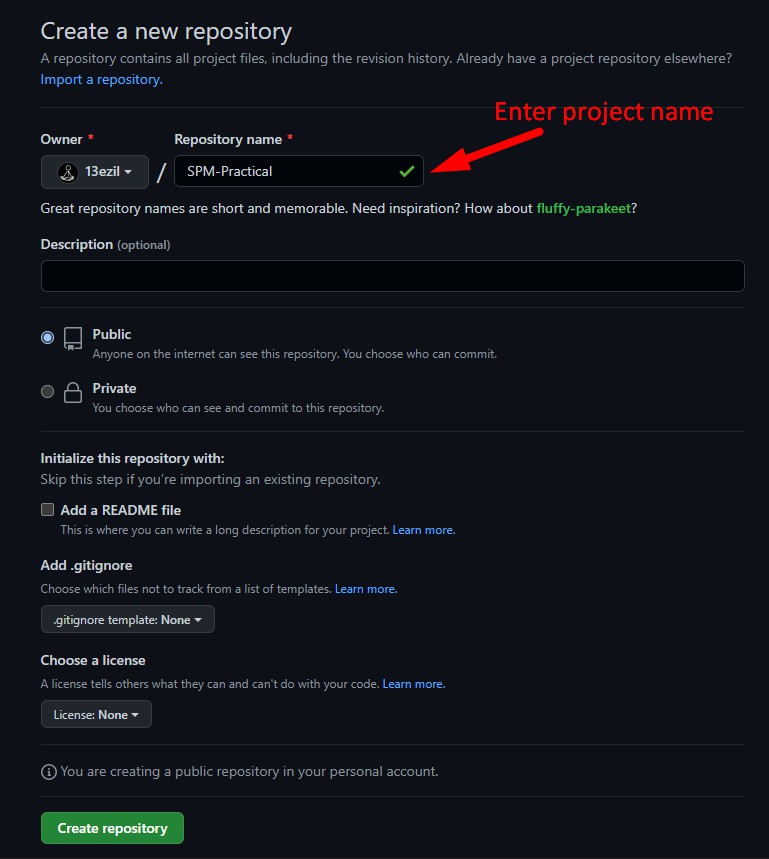
However, it is still stored only on the local storage.

## git remote add origin <…>

To place the project or files on github, we have to create a remote origin or remote repository to store the files from local repository. To do so, we register and login on *github.com.* After login, we click on the ‘+’ sign near *profile* to open a drag down menu, from which we select *new repository.*

**

This opens up a form asking for details regarding the new repository like the following:



In the ‘Repository name’ it is suggested to provide the project name for convenience and effectiveness. After filling the form, click the ‘Create repository’ button and our remote repository is ready.

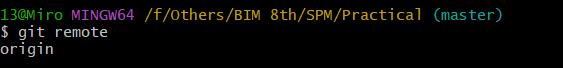
Now, we add the local repository to the remote repository using *git remote add origin <…>.*

**

Here, ‘origin’ is the name of the remote repository and ‘git@github.com:13ezil/SPM-Practical.git’ is the link to the repository.

## git remote

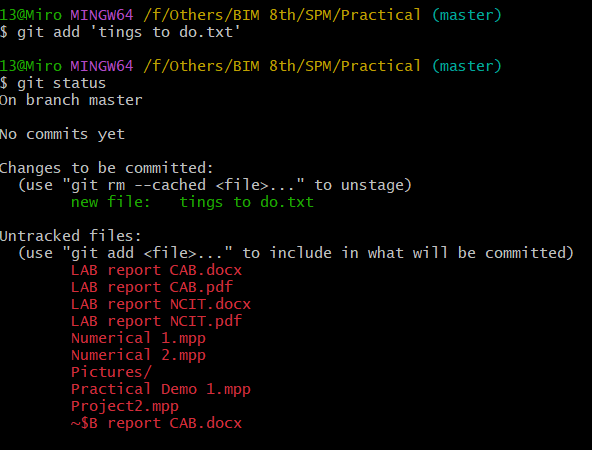
We use git remote, to check the available or connected remote repositories.



Here, we can see we are connected to a remote repository named ‘origin.’

## git add <…>

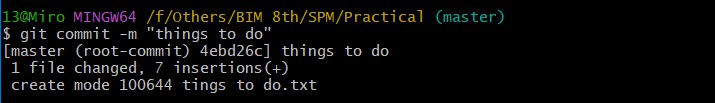
We have created both local and remote repository but our repository is still empty. To add files to be tracked to our repository, we first use *git add <…>* to get them ready or staged*.*

**

We have now added or staged the file ‘tings to do.txt’ for the repository. The results can be seen after using *git status.* However, the file is still not tracked or present on the repository.

## git commit

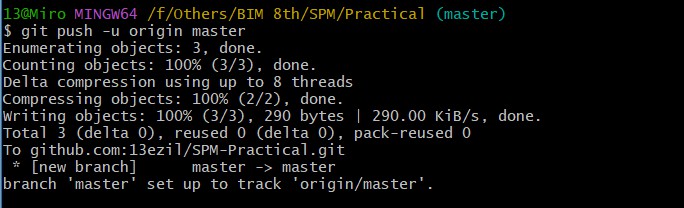
After files or changes are finalized and are ready to be added to the main source or remote repository. They are committed using *git commit.* This means the files are now added to the repository. But it is still only on the local repository.



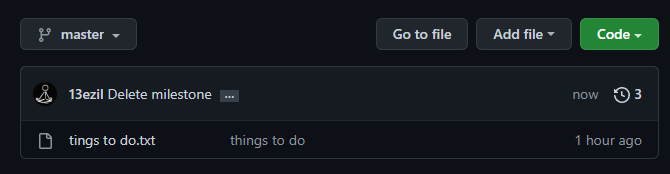
Here, -m is an argument used to pass message related to the changes we are saving or committing. This this case the message or remarks is “things to do.”

## git push –u origin <…>

Whenever we want to upload the changes or files to the main source or remote repository on github, we use *git push*.

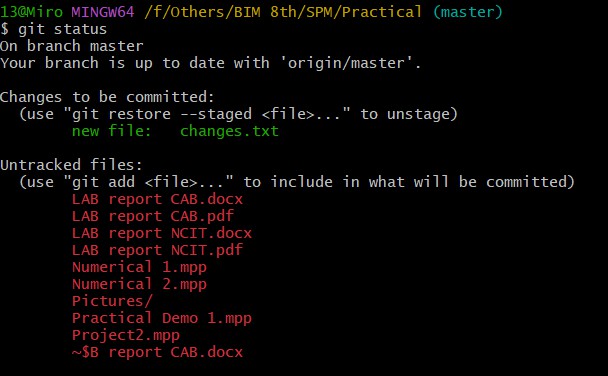


Here, ‘origin’ is the name of the remote repository and ‘master’ is the name of local repository. Hence, the command *git push –u origin master* means pushing the files and changes to the repository named ‘origin’ from the local repository named ‘master’

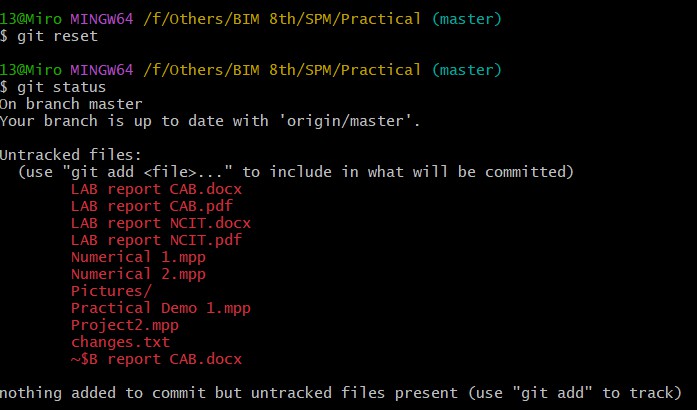


## git reset

When we add or stage files and later decide to remove them, we can use *git reset* to undo the staging. It is shown as follows:

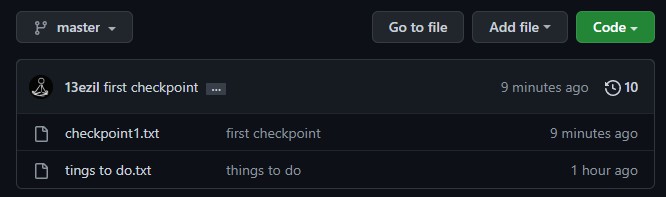


We added ‘changes.txt’ but now we do not want to commit it so we use *git reset* to unstage the changes file.

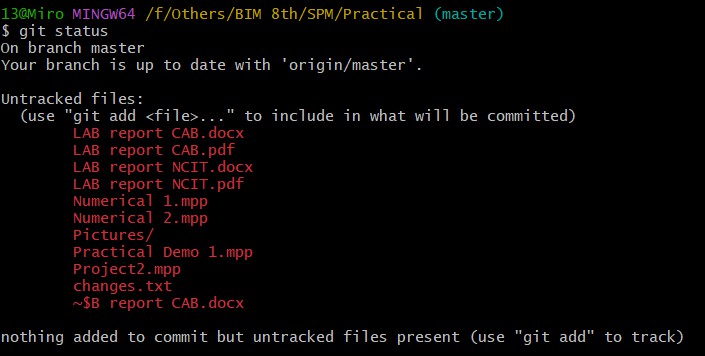


## 8. git fetch

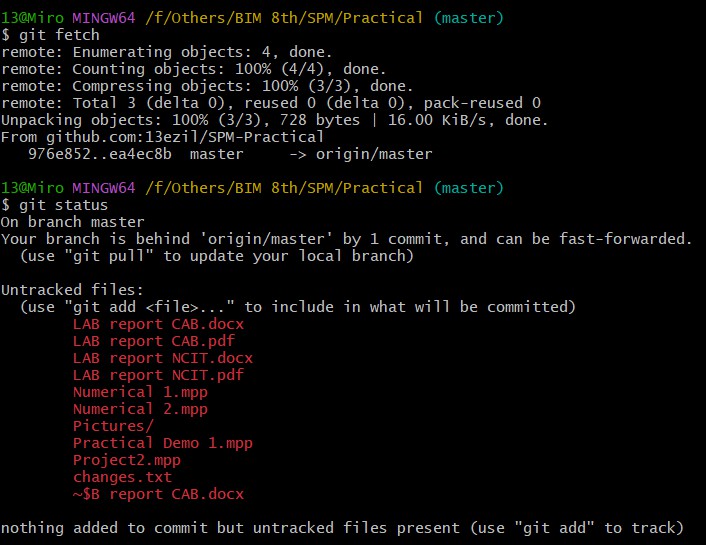
Whenever changes are made in the source or remote repository, we are not automatically notified. We use *get fetch* to check if any changes have been made in the repository. For this we first added a file to the remote repository.



After adding a file to the remote repository, we checked status from local repository.



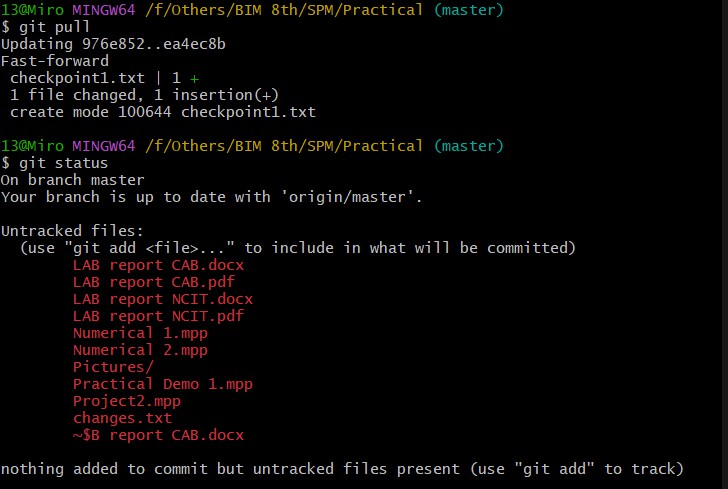
Here, we are not shown any changes although, they have been already made. Now, we use *git fetch* to get information about the remote repository and use *git status* again.



After using fetch, we can see that our local repository is one commit behind the remote repository or the source. However, this is only a notice and the changes have not been brought to the local repository.

## 10. git pull

To bring changes into our local repository we use, *git pull.* It brings all the commits as well as downloads the files that are present only on the source or remote repository.



As we can see *git pull* has added one commit and changed 1 file. Moreover, it has also created a new file named ‘checkpoint1.txt’ on the local directory or repository. When we run *git status* after *git pull*, we can see both local and remote repository are updated and syncronized.