Lab No.: 1 Date: 2082-06-07

TITLE: PROJECT PLANNING AND SCHEDULING USING MICROSOFT PROJECT

OBJECTIVE

- To learn how to create a Work Breakdown Structure (WBS) and Gantt Chart for project management using MS Project.

THEORY

1. Project Planning and Scheduling

Planning is one of the first stages of project development, in which companies and businesses determine how to achieve a specific goal or set of goals. During the planning phase, project managers create an action plan that helps them determine what steps are necessary to complete before they can achieve the project's goal.

Scheduling is the act of determining who might accomplish certain steps of a project's action plan and when they might do so. They create a time-based sequence to complete the project. During the scheduling phase, companies examine their resources and determine which ones are necessary to complete the project's goal.

2. WBS

A work breakdown structure (WBS) is a visual, hierarchical and deliverable-oriented deconstruction of a project. It is a helpful diagram for project managers because it allows them to break down their project scope and visualize all the tasks required to complete their projects.

3. Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale.

Steps to follow:

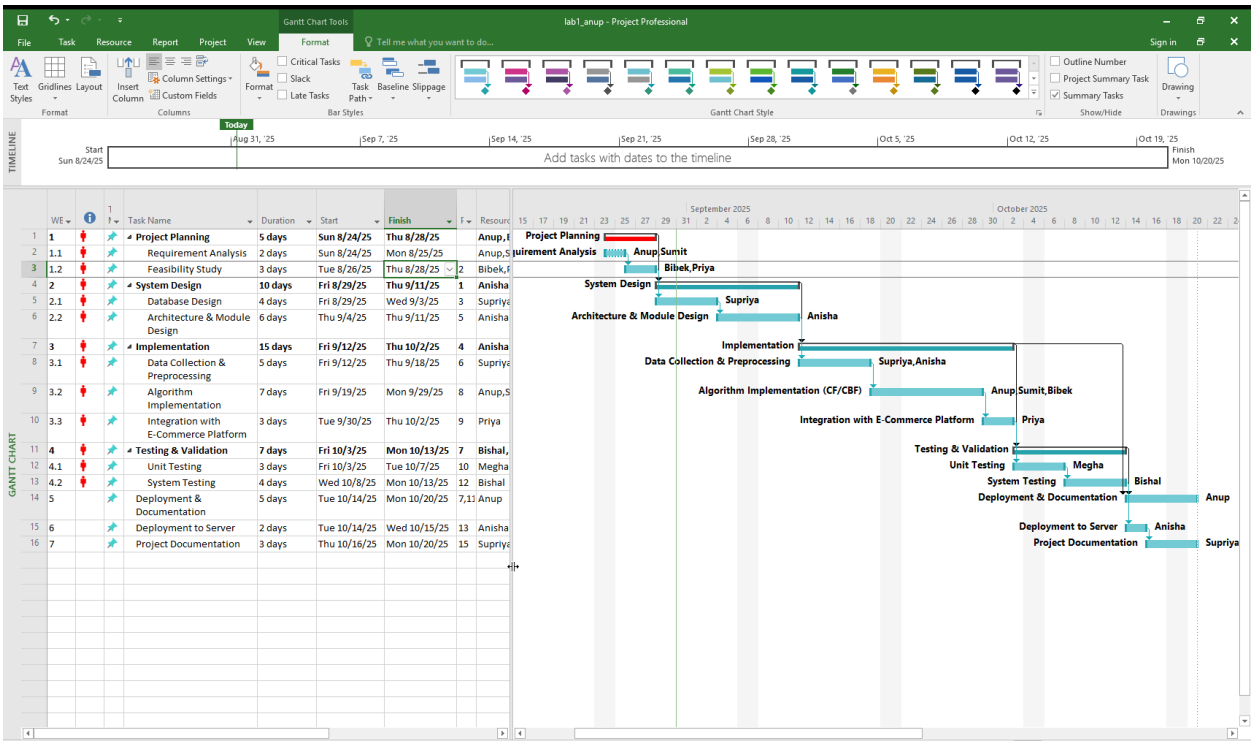
Step 1: Define Project Phases and Tasks (WBS)

Step 2: Create Project in MS Project

Step 3: Generate Gantt Chart

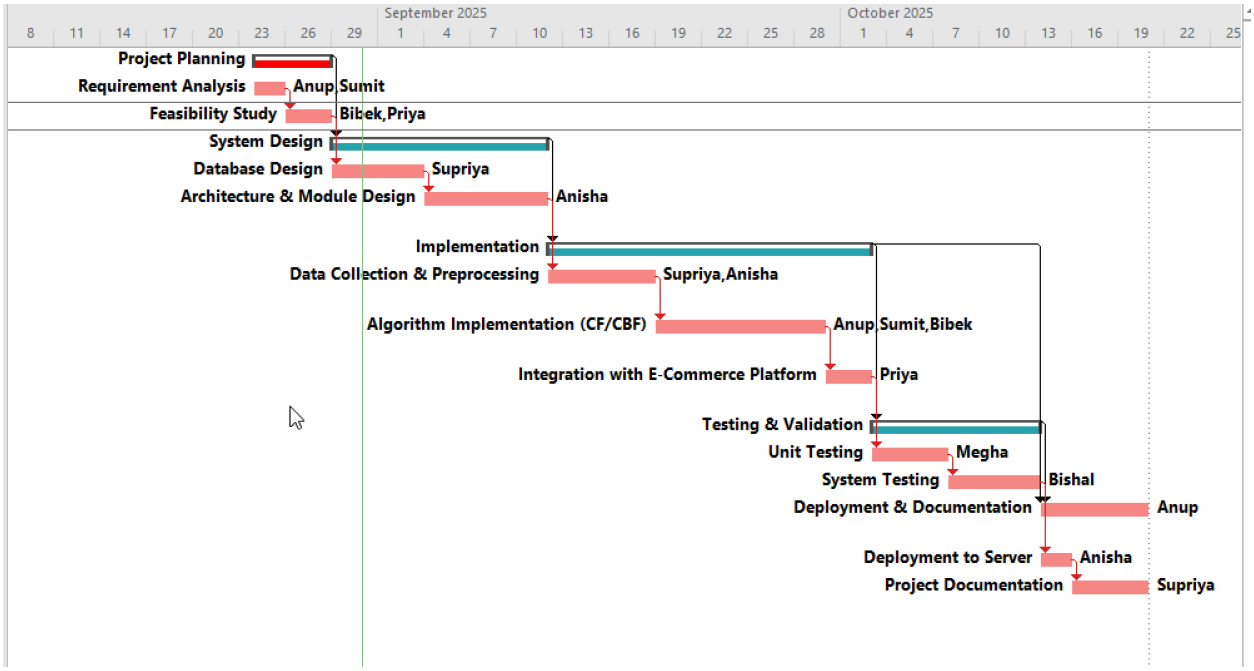
Step 4: OUTPUT

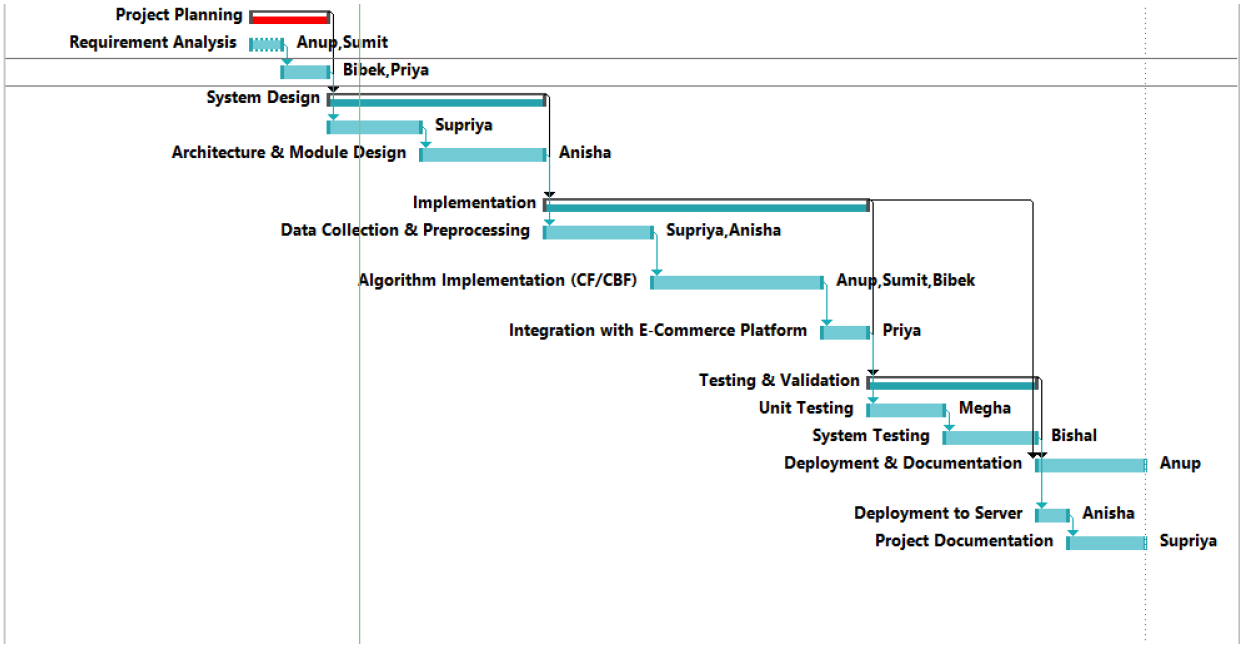
WBS



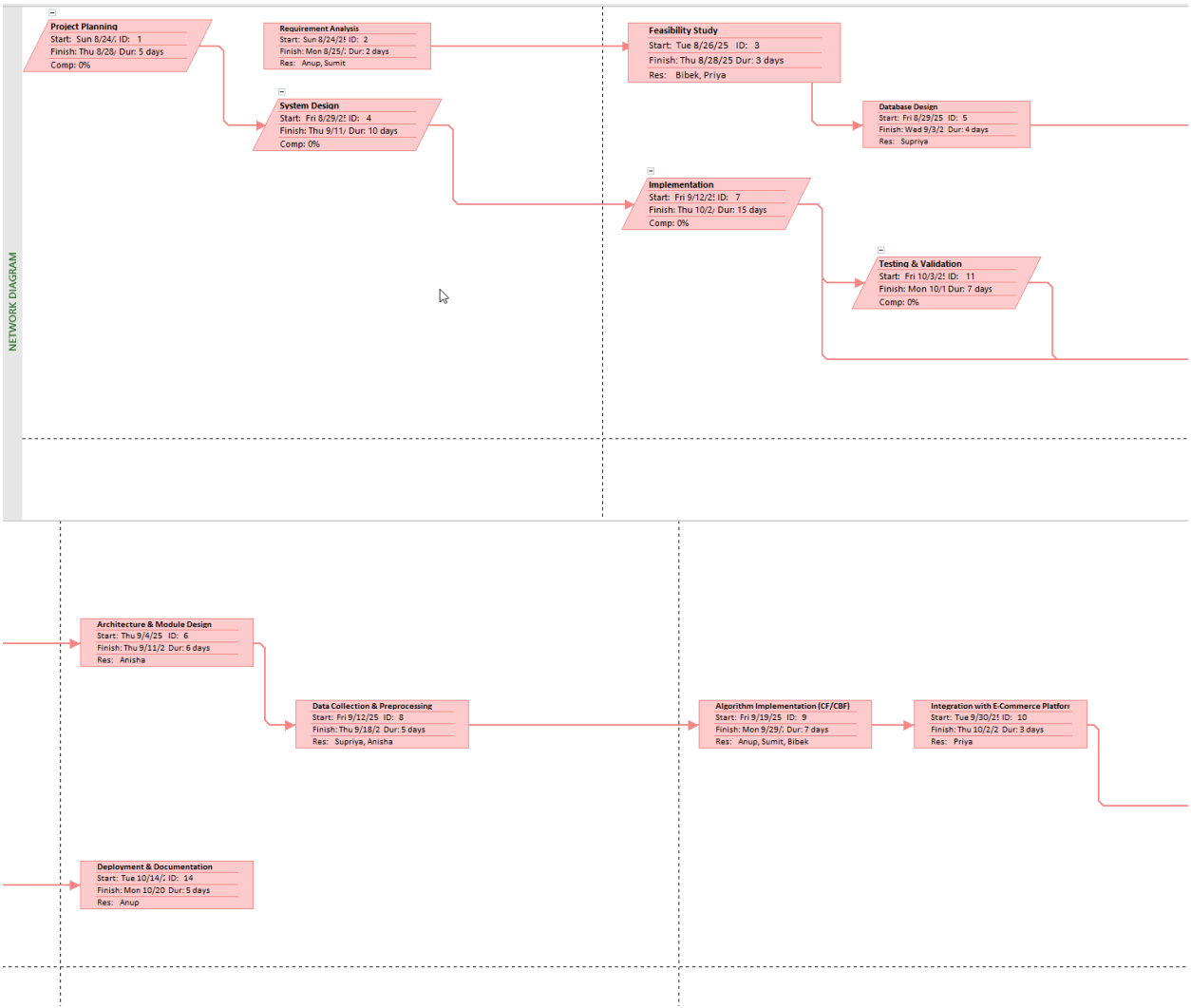
GANTT CHART

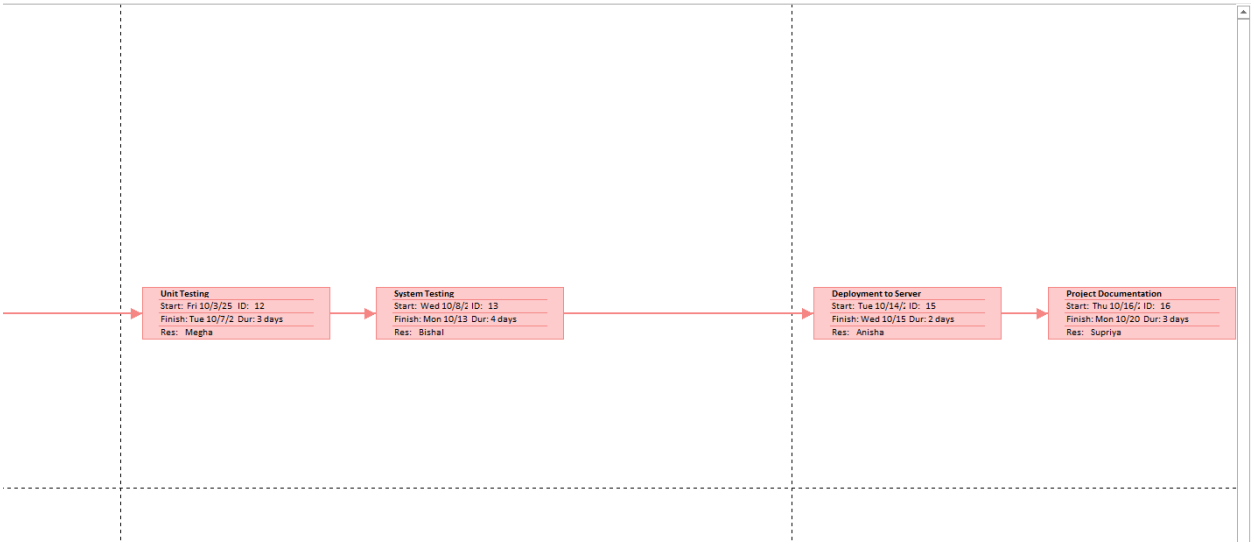
	WE			Task Name	Duration	Start	Finish	F	Resource Names	Ad
1	**1**			**Project Planning**	**5 days**	**Sun 8/24/25**	**Thu 8/28/25**		**Anup,Bibek,Priya,Sumit**	
2	1.1			Requirement Analysis	2 days	Sun 8/24/25	Mon 8/25/25		Anup,Sumit	
3	1.2			Feasibility Study	3 days	Tue 8/26/25	Thu 8/28/25	2	Bibek,Priya	
4	**2**			**System Design**	**10 days**	**Fri 8/29/25**	**Thu 9/11/25**	**1**	**Anisha,Anup,Bibek,Bishal,Me**	
5	2.1			Database Design	4 days	Fri 8/29/25	Wed 9/3/25	3	Supriya	
6	2.2			Architecture & Module Design	6 days	Thu 9/4/25	Thu 9/11/25	5	Anisha	
7	**3**			**Implementation**	**15 days**	**Fri 9/12/25**	**Thu 10/2/25**	**4**	**Anisha,Anup,Bibek,Priya,Sum**	
8	3.1			Data Collection & Preprocessing	5 days	Fri 9/12/25	Thu 9/18/25	6	Supriya,Anisha	
9	3.2			Algorithm Implementation	7 days	Fri 9/19/25	Mon 9/29/25	8	Anup,Sumit,Bibek	
10	3.3			Integration with E-Commerce Platform	3 days	Tue 9/30/25	Thu 10/2/25	9	Priya	
11	**4**			**Testing & Validation**	**7 days**	**Fri 10/3/25**	**Mon 10/13/25**	**7**	**Bishal,Megha**	
12	4.1			Unit Testing	3 days	Fri 10/3/25	Tue 10/7/25	10	Megha	
13	4.2			System Testing	4 days	Wed 10/8/25	Mon 10/13/25	12	Bishal	
14	5			Deployment & Documentation	5 days	Tue 10/14/25	Mon 10/20/25	7,11	Anup	
15	6			Deployment to Server	2 days	Tue 10/14/25	Wed 10/15/25	13	Anisha	
16	7			Project Documentation	3 days	Thu 10/16/25	Mon 10/20/25	15	Supriya	





NETWORK DIAGRAM





CONCLUSION

Using Microsoft Project helped in effectively creating a WBS and Gantt Chart, which simplified task management, clarified timelines, and improved overall project planning and control.

Lab No.: 2 Date: 2082-06-07

TITLE: PERT (Program Evaluation Review Technique) Evaluation

OBJECTIVE

- To draw a Project Network Diagram (PND) and implement PERT evaluation techniques to calculate expected project duration under uncertainty.

THEORY

Project Scheduling is the process of defining start and end times for tasks, allocating resources, and setting dependencies.

Common tools and techniques:

Gantt Chart – A bar chart showing tasks, duration, and dependencies.

PERT (Program Evaluation Review Technique) – A probabilistic network-based method that uses three-time estimates (Optimistic, Most Likely, Pessimistic) for each task.

CPM (Critical Path Method) – A deterministic technique that identifies the longest path of tasks, defining the minimum project duration.

MS Project – A project management tool that automates scheduling with WBS, Gantt, dependencies, and resource allocation.

PERT Formula for Expected Time:

Where:

O = Optimistic Time

M = Most Likely Time

P = Pessimistic Time

Standard deviation of an activity:

$$\sigma = \frac{P - O}{6}$$

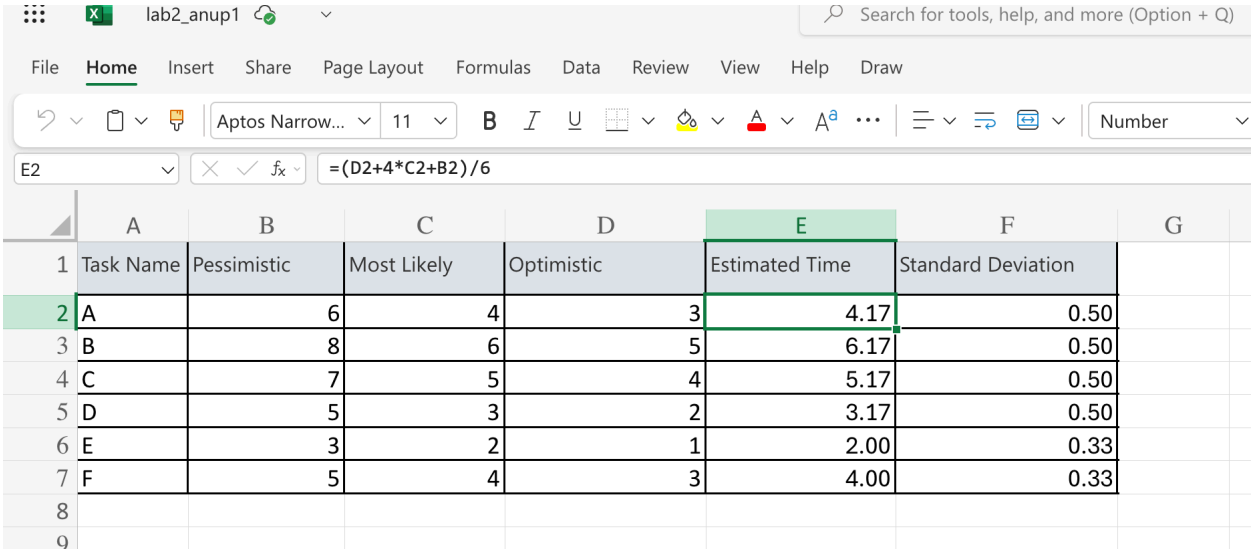
Variance:

$$\sigma^2 = \left(\frac{P - O}{6}\right)^2$$

Steps

- Identify project tasks (list deliverables and break them into activities).
- Estimate three durations for each task (O, M, P).
- Define dependencies (which task must finish before another can start).
- Draw a Network Diagram (PERT/CPM chart).
- Calculate Expected Duration (TE) for each task using the formula.
- Identify project paths and compute the total expected time.
- Find the Critical Path (path with the longest TE).
- Prepare a Gantt Chart using MS Project or Excel.
- Allocate resources and check feasibility.

OUTPUT



lab2_anup1

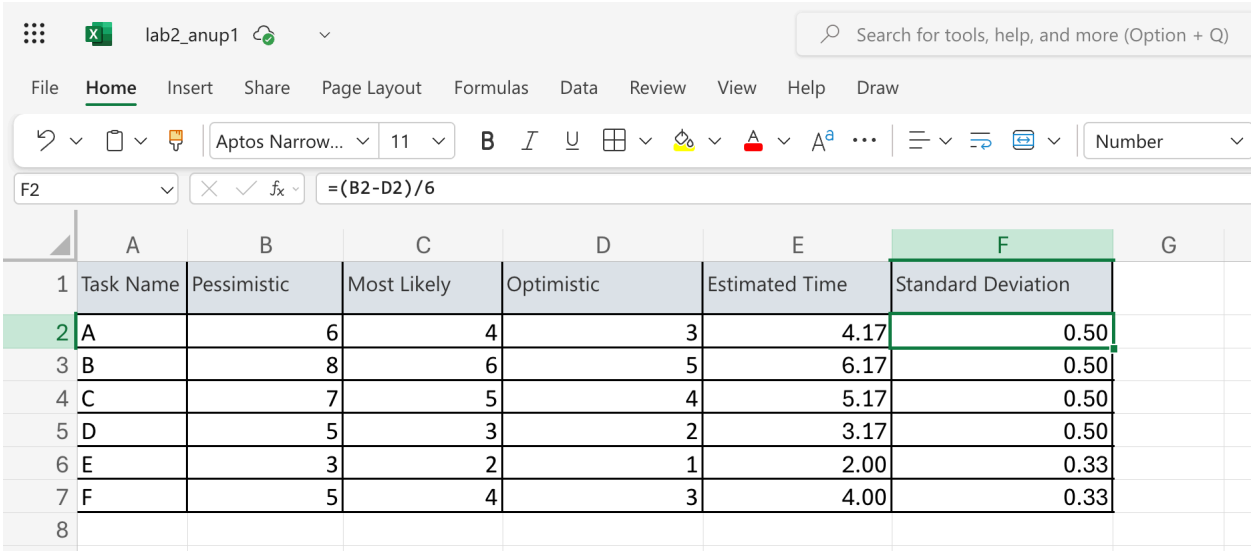
Search for tools, help, and more (Option + Q)

File Home Insert Share Page Layout Formulas Data Review View Help Draw

Aptos Narrow... 11 Number

E2 =(D2+4*C2+B2)/6

	A	B	C	D	E	F	G
1	Task Name	Pessimistic	Most Likely	Optimistic	Estimated Time	Standard Deviation	
2	A	6	4	3	4.17	0.50	
3	B	8	6	5	6.17	0.50	
4	C	7	5	4	5.17	0.50	
5	D	5	3	2	3.17	0.50	
6	E	3	2	1	2.00	0.33	
7	F	5	4	3	4.00	0.33	
8							



lab2_anup1

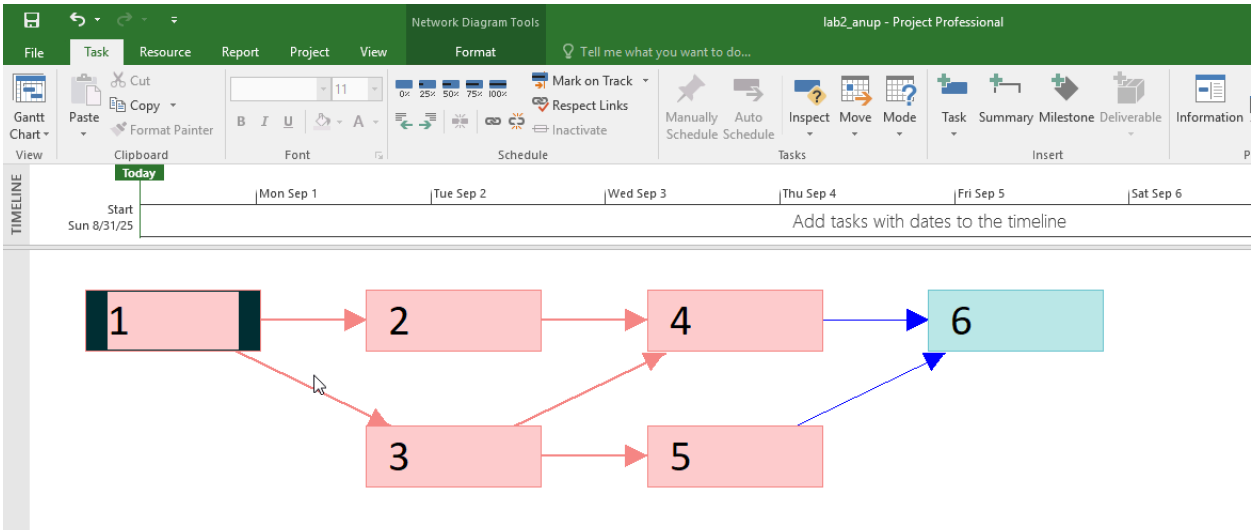
Search for tools, help, and more (Option + Q)

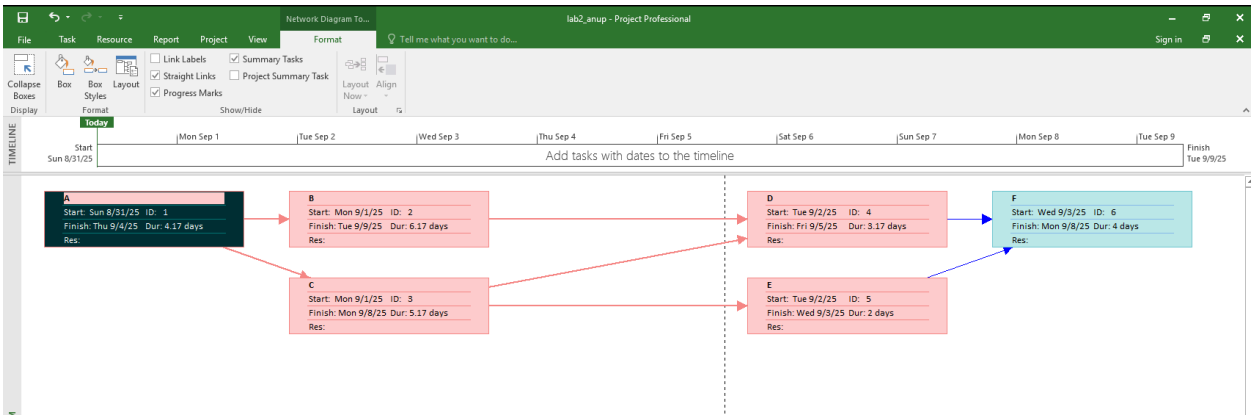
File Home Insert Share Page Layout Formulas Data Review View Help Draw

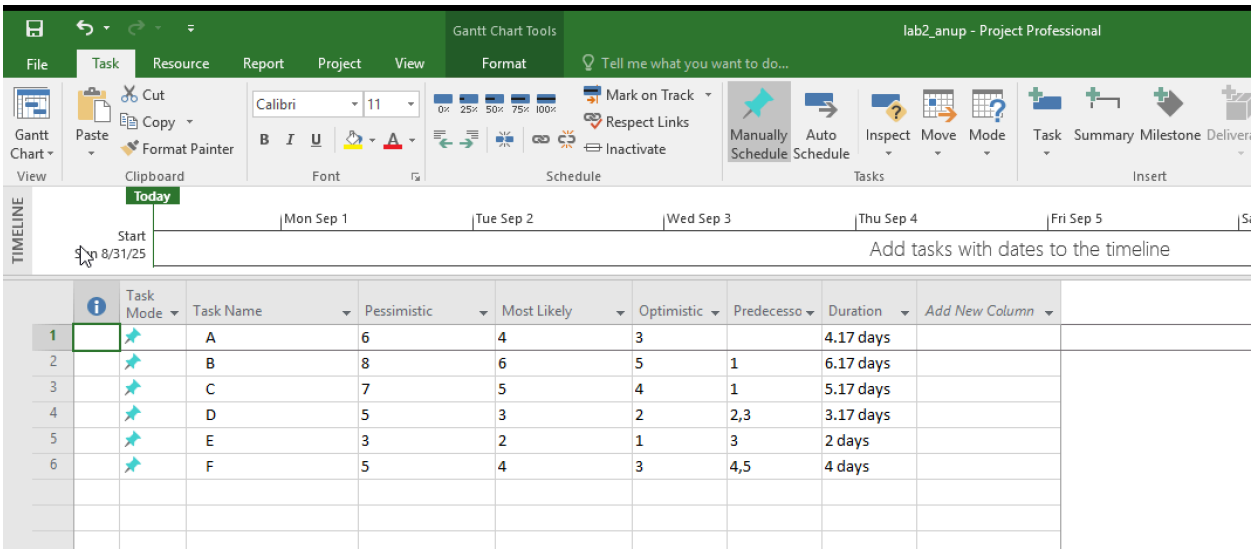
Aptos Narrow... 11 Number

F2 =(B2-D2)/6

	A	B	C	D	E	F	G
1	Task Name	Pessimistic	Most Likely	Optimistic	Estimated Time	Standard Deviation	
2	A	6	4	3	4.17	0.50	
3	B	8	6	5	6.17	0.50	
4	C	7	5	4	5.17	0.50	
5	D	5	3	2	3.17	0.50	
6	E	3	2	1	2.00	0.33	
7	F	5	4	3	4.00	0.33	
8							







	Task Mode	Task Name	Pessimistic	Most Likely	Optimistic	Predecesso	Duration	Add New Column
1		A	6	4	3		4.17 days	
2		B	8	6	5	1	6.17 days	
3		C	7	5	4	1	5.17 days	
4		D	5	3	2	2,3	3.17 days	
5		E	3	2	1	3	2 days	
6		F	5	4	3	4,5	4 days	

CONCLUSION

The use of PERT evaluation along with the Project Network Diagram helped in clearly identifying task dependencies, the critical path, and possible delays. By applying optimistic, most likely, and pessimistic time estimates, the expected project duration under uncertainty was successfully calculated.

Lab No.: 3 Date: 2082-06-07

TITLE: IDENTIFYING EARNED VALUE ANALYSIS (EVA)

OBJECTIVE

- To understand and calculate Earned Value Analysis (EVA) metrics for monitoring project performance.
- To analyze whether a project is on schedule, ahead, or behind and under or over budget.

THEORY

Earned Value Analysis (EVA) is a project management technique that integrates scope, schedule, and cost to evaluate project performance.

Key Terms:

- Planned Value (PV): Budgeted cost for work planned up to a certain date.
- Earned Value (EV): Budgeted cost of actual work completed up to that date.
- Actual Cost (AC): Actual cost incurred for completed work.

Important Formulas:

1. Cost Variance (CV) = EV – AC

   CV > 0 → under budget

   CV < 0 → over budget
2. Schedule Variance (SV) = EV – PV

   SV > 0 → ahead of schedule

   SV < 0 → behind schedule
3. Cost Performance Index (CPI) = EV / AC

   CPI > 1 → cost efficient

   CPI < 1 → cost overrun

4. Schedule Performance Index (SPI) = EV / PV

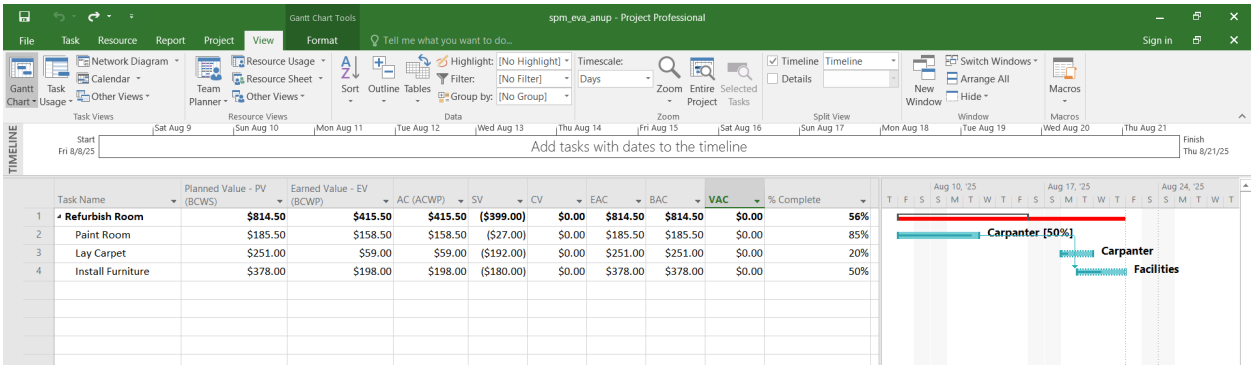
   SPI > 1 → ahead of schedule

   SPI < 1 → behind schedule

5. Estimate at Completion (EAC) = BAC / CPI

   BAC = Budget at Completion (total planned budget).

OUTPUT



	Task Name	Planned Value - PV (BCWS)	Earned Value - EV (BCWP)	AC (ACWP)	SV	CV	EAC	BAC	VAC	% Complete
1	**Refurbish Room**	**$814.50**	**$415.50**	**$415.50**	**($399.00)**	**$0.00**	**$814.50**	**$814.50**	**$0.00**	**56%**
2	Paint Room	$185.50	$158.50	$158.50	($27.00)	$0.00	$185.50	$185.50	$0.00	85%
3	Lay Carpet	$251.00	$59.00	$59.00	($192.00)	$0.00	$251.00	$251.00	$0.00	20%
4	Install Furniture	$378.00	$198.00	$198.00	($180.00)	$0.00	$378.00	$378.00	$0.00	50%

CONCLUSION

Earned Value Analysis (EVA) enables effective monitoring of project performance by integrating cost, schedule, and scope. By using metrics like CV, SV, CPI, SPI, and EAC, we can determine whether the project is on track, under or over budget, and ahead or behind schedule, making it a useful tool for project control and forecasting.

Lab No.: 4 Date: 2082-06-07

TITLE: SOFTWARE COST ESTIMATION – COCOMO MODEL USING SPREADSHEET

OBJECTIVE

- To calculate the effort, development time, and cost of a software project using the COCOMO model.

THEORY

COCOMO Model:

COCOMO stands for the constructive cost model, a cost estimation model for software projects based on LOC. The COCOMO model calculates a proposed software project's time, effort, cost, and quality.

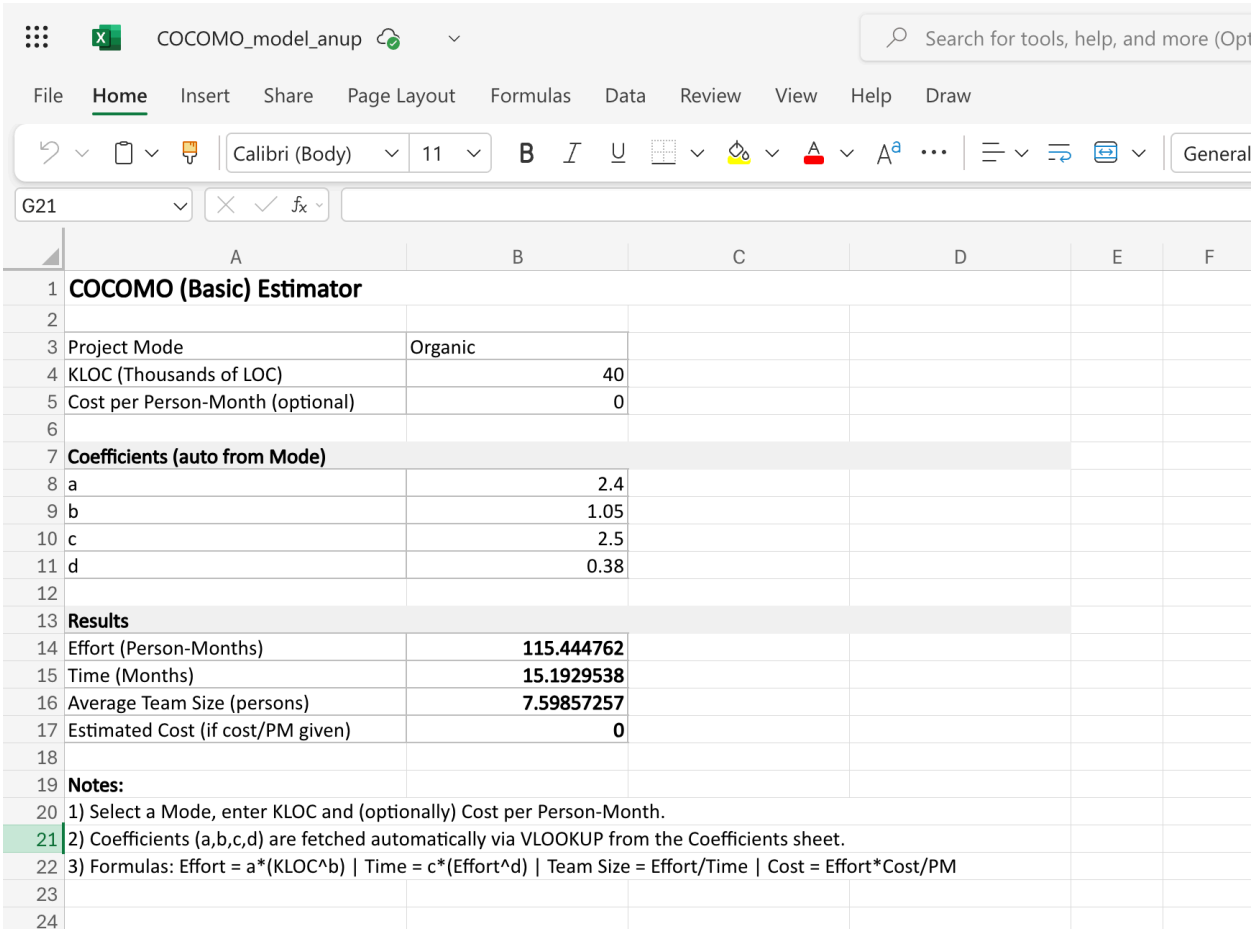
- **Basic COCOMO Formula:**
  - Effort (PM) = a × (KLOC)^b
  - Development Time (TDEV) = c × (Effort)^d
  - Where **a, b, c, d** depend on project type (Organic, Semi-Detached, Embedded).

Project Type	**a**	**b**	**c**	**d**
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

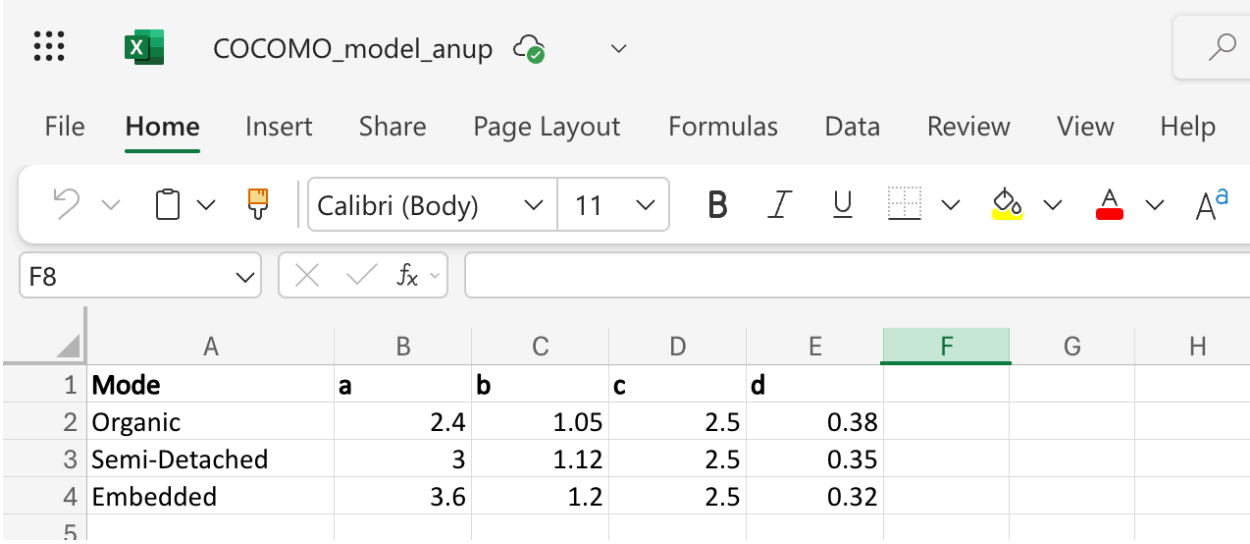
Steps:

1. Open **MS Excel/Google Sheets**.
2. Input **KLOC value** and **project type**.
3. Apply formulas for Effort (PM) and Development Time (TDEV).
4. Create additional columns for **Average Staffing = Effort / TDEV** and **Productivity = KLOC / Effort**.

OUTPUT



	A	B	C	D	E	F
1	**COCOMO (Basic) Estimator**					
2						
3	Project Mode	Organic				
4	KLOC (Thousands of LOC)	40				
5	Cost per Person-Month (optional)	0				
6						
7	**Coefficients (auto from Mode)**					
8	a	2.4				
9	b	1.05				
10	c	2.5				
11	d	0.38				
12						
13	**Results**					
14	Effort (Person-Months)	**115.444762**				
15	Time (Months)	**15.1929538**				
16	Average Team Size (persons)	**7.59857257**				
17	Estimated Cost (if cost/PM given)	**0**				
18						
19	**Notes:**					
20	1) Select a Mode, enter KLOC and (optionally) Cost per Person-Month.					
21	2) Coefficients (a,b,c,d) are fetched automatically via VLOOKUP from the Coefficients sheet.					
22	3) Formulas: Effort = a*(KLOC^b) | Time = c*(Effort^d) | Team Size = Effort/Time | Cost = Effort*Cost/PM					
23						
24						



	A	B	C	D	E	F	G	H
1	**Mode**	**a**	**b**	**c**	**d**			
2	Organic	2.4	1.05	2.5	0.38			
3	Semi-Detached	3	1.12	2.5	0.35			
4	Embedded	3.6	1.2	2.5	0.32			
5								

CONCLUSION

Using the COCOMO model in a spreadsheet, we can estimate the effort, development time, staffing, and productivity of a software project based on its size (KLOC) and type. This exercise shows how COCOMO provides a systematic and quantitative way to predict project cost and schedule, helping in better planning and resource allocation.

Lab No.: 5 Date: 2082-06-07

TITLE: AUTOMATED TESTING USING SELENIUM

OBJECTIVE

- To perform automated functional testing of a web application using Selenium WebDriver.

THEORY

- Selenium is an open-source tool for automating web browsers.
- It supports multiple languages (Java, Python, C#) and browsers (Chrome, Firefox, Edge).
- Used for functional testing, regression testing, and cross-browser testing.

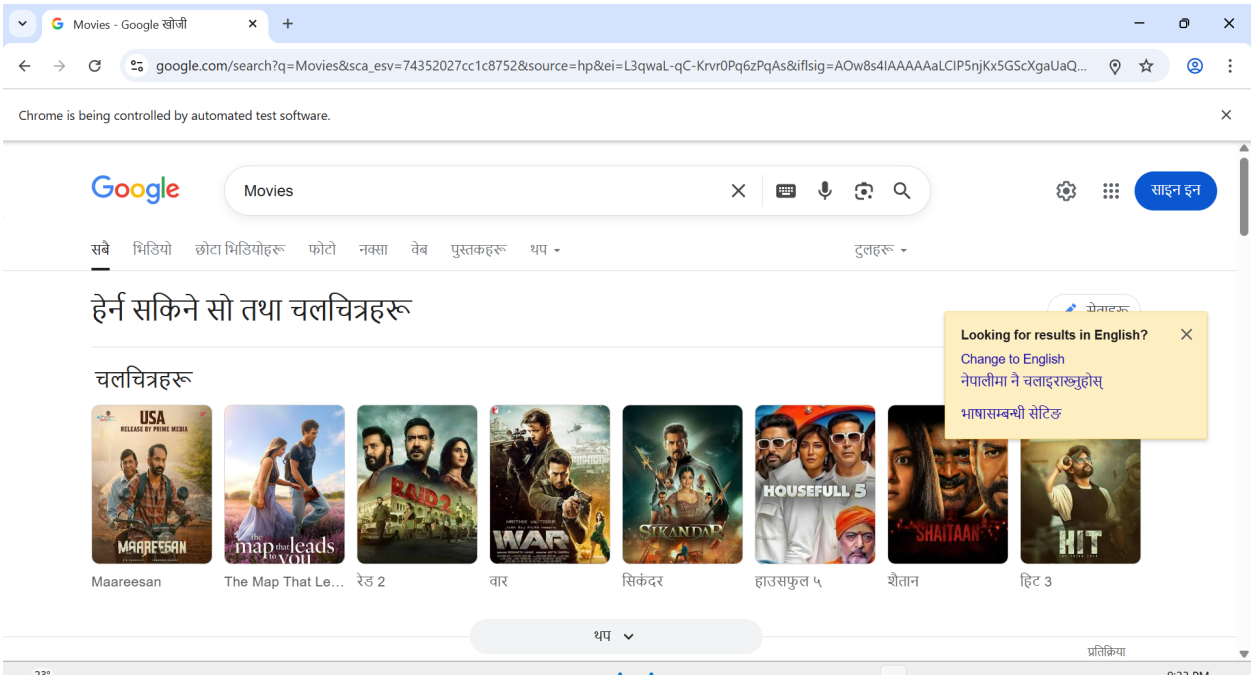
Steps:

1. Install Selenium WebDriver (e.g., in Python via pip install selenium).
2. Download the browser driver (e.g., ChromeDriver).
3. Write a test script to:
    - Open a browser.
    - Navigate to a website.
    - Perform actions (click, input text, submit form).
    - Validate results (assert page title, content).
4. Run the script and observe automated execution.

CODE

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.support import expected_conditions as EC

#test setup

options = Options()
options.add_argument("--start-maximized")

driver=webdriver.Chrome(options= options)

try:
    driver.get("https://www.google.com")
    print("[Info] Opened google homepage")

    wait= WebDriverWait(driver,30)
    # search_box= wait.until(EC.presence_of_all_elements_located((By.ID,"APjFqb")))
    search_box = wait.until(EC.presence_of_element_located((By.ID,"APjFqb")))

    print("Search Box Located")
    #type movies in search box
    search_box.send_keys("Movies")
    search_box.send_keys(Keys.RETURN)
    print("[Info] search submitted")

    wait.until(EC.title_contains("Movies"))
    print("[Info] Result page loaded")

    #Verification
    assert "Movies" in driver.title, "Test failed: 'movies' not found in page title"
    print("test passed: Google search is working correctly")

except Exception as e:
    print(f"[Test Failed] Error occured {e}")

finally:
    # driver.quit()
    print("[Info] browser closed”)
```

OUTPUT



CONCLUSION

Automated testing with Selenium WebDriver allows functional and regression tests to be executed quickly and accurately across browsers. By writing test scripts, we can simulate user actions, validate results, and ensure application correctness, making testing more efficient and reliable compared to manual testing.

Lab No.: 6 Date: 2082-06-07

TITLE: COLLABORATION USING JIRA TOOL

OBJECTIVE

- To learn software project collaboration, issue tracking, and agile workflow management using JIRA.

THEORY

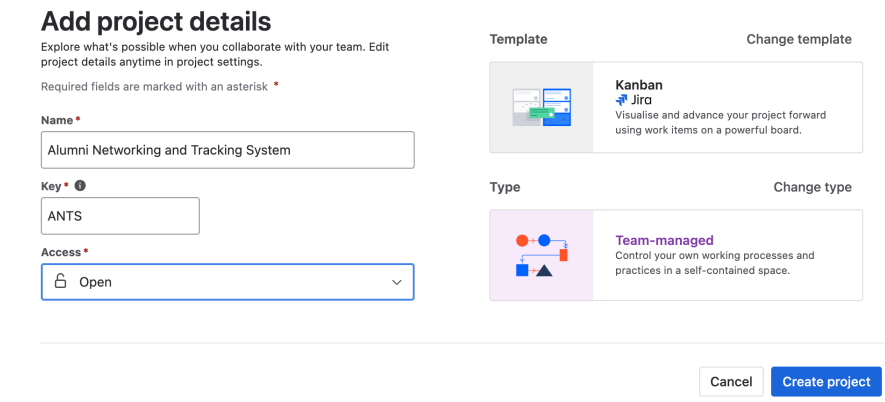
- JIRA is an agile project management and issue tracking tool used for Scrum, Kanban, and bug tracking.
- Teams can create user stories, sprints, epics, and tasks, assign them to members, and track progress with dashboards.

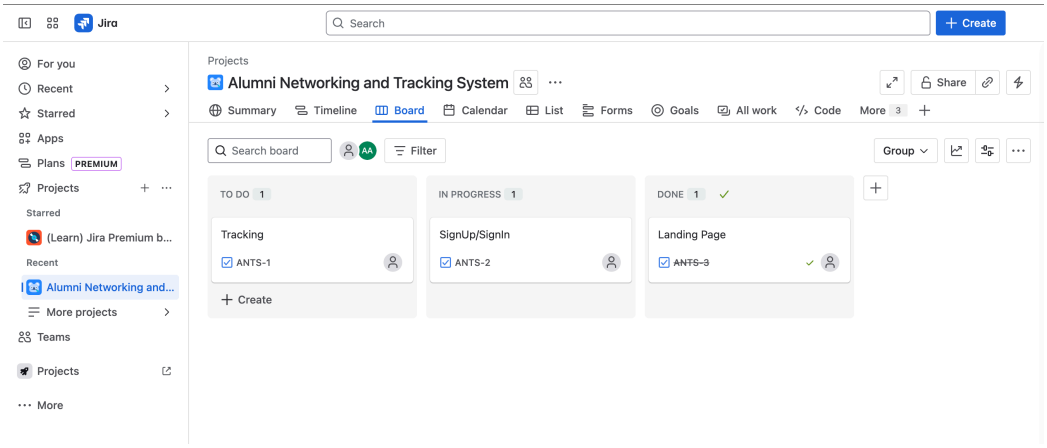
Steps

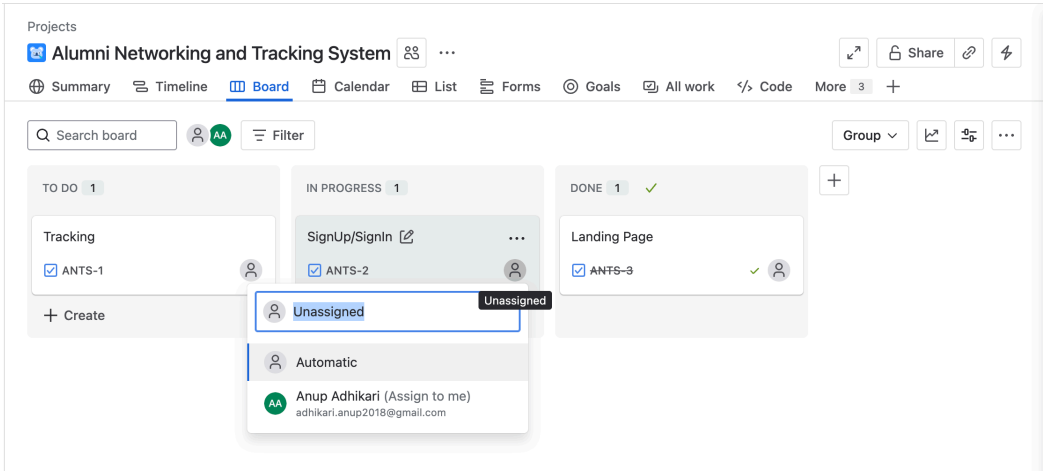
1. Create an account on Atlassian JIRA.
2. Create a new project (Scrum or Kanban).
3. Add issues (tasks, bugs, user stories, epics).
4. Assign tasks to team members.
5. Configure workflow (To Do → In Progress → Done).
6. Use the board view to track sprint progress.

Generate reports (burndown chart, velocity report).

OUTPUT


Add project details
Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.
Required fields are marked with an asterisk *
Name *
Alumni Networking and Tracking System
Key *
ANTS
Access *
Open
Template
Change template
Kanban
Jira
Visualise and advance your project forward using work items on a powerful board.
Type
Change type
Team-managed
Control your own working processes and practices in a self-contained space.
Cancel
Create project


Jira
Search
Create
For you
Recent
Starred
Apps
Plans PREMIUM
Projects
Starred
(Learn) Jira Premium b...
Recent
Alumni Networking and...
More projects
Teams
Projects
More
Projects
Alumni Networking and Tracking System
Share
Summary
Timeline
Board
Calendar
List
Forms
Goals
All work
Code
More 3
Search board
Filter
Group
TO DO 1
IN PROGRESS 1
DONE 1
Tracking
ANTS-1
SignUp/SignIn
ANTS-2
Landing Page
ANTS-3
Create


Projects
Alumni Networking and Tracking System
Share
Summary
Timeline
Board
Calendar
List
Forms
Goals
All work
Code
More 3
Search board
Filter
Group
TO DO 1
IN PROGRESS 1
DONE 1
Tracking
ANTS-1
SignUp/SignIn
ANTS-2
Landing Page
ANTS-3
Create
Unassigned
Unassigned
Automatic
Anup Adhikari (Assign to me)
adhikari.anup2018@gmail.com

Lab No.: 7 Date: 2082-06-07

TITLE: GIT & GITHUB FOR VERSION CONTROL AND COLLABORATION

OBJECTIVE

1. To understand version control and the purpose of Git.
2. To install and configure Git on their system.
3. To perform basic Git operations: init, add, commit, status, log.
4. To work with GitHub: create repositories, push, pull, clone, and manage branches.

THEORY

Git is a distributed version control system that records changes made to files and allows multiple people to work on a project at the same time. It was developed by Linus Torvalds in 2005 and is widely used for software development. Git stores data as a series of commits, where each commit is like a snapshot of the project at a specific point in time. Developers usually work on different branches to add new features or fix issues without disturbing the main code, and later they can merge their work into the main branch. Git has three key areas: the working directory where you edit files, the staging area where changes are prepared, and the repository where the history of commits is stored.

COMMANDS

- git init → Initialize a new repository

```
anupadhikari@Anups-MacBook-Air-2 git % git init
Initialized empty Git repository in /Users/anupadhikari/Desktop/git/.git/
```

- git status → Check the status of changes

```
anupadhikari@Anups-MacBook-Air-2 git % git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.pages

nothing added to commit but untracked files present (use "git add" to track)
```

- git add <file> → Add files to the staging area

```
anupadhikari@Anups-MacBook-Air-2 git % git add index.pages
```

- git commit -m "message" → Save changes to the repository

```
anupadhikari@Anups-MacBook-Air-2 git % git commit -m "index added"
[main (root-commit) 6cbeaa9] index added
 Committer: Anup Adhikari <anupadhikari@Anups-MacBook-Air-2.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.pages
```

- git clone <url> → Copy an existing remote repository

```
anupadhikari@Anups-MacBook-Air-2 git % git clone https://github.com/Anup0118/portfolio.git
Cloning into 'portfolio'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 71 (delta 16), reused 63 (delta 8), pack-reused 0 (from 0)
Receiving objects: 100% (71/71), 296.38 KiB | 1.03 MiB/s, done.
Resolving deltas: 100% (16/16), done.
```

- git branch “branchName” – Create a new branch

```
anupadhikari@Anups-MacBook-Air-2 git % git branch "gitpractice"
```

- git checkout “branchName” – Switch to that branch

```
anupadhikari@Anups-MacBook-Air-2 git % git checkout "gitpractice"
Switched to branch 'gitpractice'
```

- git branch → List all branches

```
anupadhikari@Anups-MacBook-Air-2 git % git branch
* gitpractice
  main
```