Day 5:-

5.write 2 script for daily stand up meeting

Script 1:

Meeting Start Time: 9 AM

Scrum Master: (Coach)

Good Morning team, let's start our daily stand-up meeting.

Keep your updates short and clear.

Share your tasks for today.

Highlight any blockers you have.

Let's start with Pranathi for the updates.

Pranathi:

Yesterday, I worked on improving the search functionality.

Today, I will focus on optimizing the database queries.

No blockers from my side.

Kiran:

Yesterday, I worked on the notification system.

Today, I will be working on the email templates.

No blockers currently

Vishal:

Yesterday, I worked on the API for the new feature.

Today, I will continue with the API development.

I am having trouble with the API response time.

Scrum Master:

Thank you for the updates. If anyone has issues or needs further discussion, please stay back for clarification.

Meeting End Time: 9:15 AM

Script 2:

Meeting Start Time: 9 AM

Scrum Master: (Coach)

Good Morning team, let's start the daily stand-up meeting.

Keep your updates brief and to the point.

Discuss your plans for today.

Mention any blockers you're encountering.

Let's begin with Pranathi for the updates.

Pranathi:

Yesterday, I worked on the user profile page.

Today, I will focus on the settings page.

I don't have any blockers.

Raj:

Yesterday, I completed the integration of the payment gateway.

Today, I will start working on the refund process.

No blockers at the moment.

Amit:

Yesterday, I was testing the new feature release.

Today, I will be preparing the deployment.

I am facing an issue with the deployment script.

Scrum Master:

Thank you all for the updates. For any issues or challenges, please stay back after the meeting for further discussion.

Meeting End Time: 9:15 AM

6.explain the responsibility of scrum roles

scrum master

product owner

development team

**1.Scrum Master Responsibilities**:

-> Facilitates Scrum events (Daily Stand-ups, Sprint Planning, Sprint Review, Sprint Retrospective).

-> Removes impediments to the team’s progress.

-> Ensures the Scrum process is followed and understood by the team.

-> Coaches the team on Agile principles and practices.

-> Protects the team from outside interruptions and distractions.

-> Helps the team improve and work more efficiently.

-> Serves as a mediator between the team and external stakeholders.

**2.Product Owner Responsibilities**:

-> Defines and manages the product backlog.

-> Prioritizes backlog items based on business value and stakeholder needs.

-> Clearly communicates the product vision and goals to the team.

-> Ensures that the team understands the requirements and user stories.

-> Acts as the primary point of contact for stakeholders and manages their expectations.

-> Makes decisions about product features and release dates.

-> Accepts or rejects completed work during the Sprint Review.

**3.Development Team Responsibilities**:

-> Delivers potentially shippable increments of the product at the end of each sprint.

-> Collaborates on defining tasks and estimating effort during Sprint Planning.

-> Self-organizes and manages its own work to achieve the sprint goals.

-> Maintains high quality in development practices and adheres to the Definition of Done.

->Participates in all Scrum events and contributes to continuous improvement.

day 5: - (Linux)

--------

3.write some 30 Linux commands.

Here’s the list of common **Linux commands**

1. [Cd command](https://bytescout.com/blog/most-used-linux-commands.html#1)
2. [ls command](https://bytescout.com/blog/most-used-linux-commands.html#2)
3. [Man command](https://bytescout.com/blog/most-used-linux-commands.html#3)
4. [Cat command](https://bytescout.com/blog/most-used-linux-commands.html#4)
5. [Mkdir command](https://bytescout.com/blog/most-used-linux-commands.html#5)
6. [Chmod command](https://bytescout.com/blog/most-used-linux-commands.html#6)
7. [Rmdir command](https://bytescout.com/blog/most-used-linux-commands.html#7)
8. [Touch command](https://bytescout.com/blog/most-used-linux-commands.html#8)
9. [Locate command](https://bytescout.com/blog/most-used-linux-commands.html#9)
10. [Clear command](https://bytescout.com/blog/most-used-linux-commands.html#10)
11. [Rm command](https://bytescout.com/blog/most-used-linux-commands.html#11)
12. [MV command](https://bytescout.com/blog/most-used-linux-commands.html#12)
13. [Curl command](https://bytescout.com/blog/most-used-linux-commands.html#13)
14. [Echo command](https://bytescout.com/blog/most-used-linux-commands.html#14)
15. [Free command](https://bytescout.com/blog/most-used-linux-commands.html#15)
16. [Groups command](https://bytescout.com/blog/most-used-linux-commands.html#16)
17. [Head command](https://bytescout.com/blog/most-used-linux-commands.html#17)
18. [History command](https://bytescout.com/blog/most-used-linux-commands.html#18)
19. [Passwd command](https://bytescout.com/blog/most-used-linux-commands.html#19)
20. [Ping command](https://bytescout.com/blog/most-used-linux-commands.html#20)
21. [alias command](https://bytescout.com/blog/most-used-linux-commands.html#21)
22. [ZIP command](https://bytescout.com/blog/most-used-linux-commands.html#22)
23. [dd command](https://bytescout.com/blog/most-used-linux-commands.html#23)
24. [chown command](https://bytescout.com/blog/most-used-linux-commands.html#24)
25. [sudo command](https://bytescout.com/blog/most-used-linux-commands.html#25)
26. [cal command](https://bytescout.com/blog/most-used-linux-commands.html#26)
27. [bc command](https://bytescout.com/blog/most-used-linux-commands.html#27)
28. [change command](https://bytescout.com/blog/most-used-linux-commands.html#28)
29. [df command](https://bytescout.com/blog/most-used-linux-commands.html#29)
30. [name command](https://bytescout.com/blog/most-used-linux-commands.html#30)
    * cd - Changes the current directory.
    * ls - Lists directory contents.
    * man - Displays the manual page for a command.
    * cat - Concatenates and displays file contents.
    * mkdir - Creates a new directory.
    * chmod - Changes the permissions of a file or directory.
    * rmdir - Removes an empty directory.
    * touch - Creates an empty file or updates the timestamp of an existing file.
    * locate - Finds files by name using a database.
    * clear - Clears the terminal screen.
    * rm - Removes files or directories.
    * mv - Moves or renames files or directories.
    * curl - Transfers data from or to a server using various protocols.
    * echo - Displays a line of text or variables to the terminal.
    * free - Displays memory usage information.
    * groups - Shows the groups a user is a member of.
    * head - Outputs the first part of files.
    * history - Displays the command history.
    * passwd - Changes a user's password.
    * ping - Sends ICMP ECHO\_REQUEST packets to network hosts.
    * alias - Creates an alias for a command.
    * zip - Compresses files into a zip archive.
    * dd - Converts and copies files.
    * chown - Changes the ownership of files or directories.
    * sudo - Executes a command as another user, typically the superuser.
    * cal - Displays a calendar.
    * bc - An arbitrary precision calculator language.
    * chage - Changes the user password expiry information.
    * df - Reports file system disk space usage.
    * uname - Prints system information.

2. basic task

a. writes a command to create a file named demo.txt inside Movies directory

b. copy hello.txt files from Desktop to Downloads directory

c. Write a command to display all the files from Movies Directory

d. display first 15 lines of demo.txt file

e. using cat command, create a new file and write the data to the file.

f. read the file content of demo.txt

1. **writes a command to create a file named demo.txt inside Movies directory**

* touch Movies/demo.txt

This command creates a file named demo.txt inside the Movies directory using the touch command, which is typically used to create empty files.

**b. copy hello.txt files from Desktop to Downloads directory**

🡪 cp Desktop/hello.txt Downloads/

This command copies the file hello.txt from the Desktop directory to the Downloads directory using the cp (copy) command.

1. **Write a command to display all the files from Movies Directory**

ls Movies/

This command lists all the files and directories in the Movies directory using the ls (list) command.

**d. display first 15 lines of demo.txt file**

head -n 15 Movies/demo.txt

This command displays the first 15 lines of the demo.txt file located in the Movies directory using the head command with the -n option to specify the number of lines to display.

**e. using cat command, create a new file and write the data to the file.**

cat > newfile.txt

After running this command, you can start typing the content you want to write to the file. Once you are done, press Ctrl+D to save the content and exit from the input mode.

Here's a breakdown of how this command works:

* cat: Invokes the cat command, which is normally used to concatenate and display files.
* >: Redirects the output to a new file (newfile.txt in this case).
* newfile.txt: Specifies the name of the new file where the content will be written.

When you run cat > newfile.txt, the terminal will wait for your input. You can type multiple lines of text. Press Enter after each line. To finish inputting text and save the file, press Ctrl+D.

**f. read the file content of demo.txt**

cat Movies/demo.txt

This command will output the entire contents of the demo.txt file located in the Movies directory to the terminal. The cat command is commonly used to concatenate and display files, and in this case, it simply displays the content of demo.txt.

3. explain absolute and relative path

### **Absolute Path**

An absolute path specifies the exact location of a file or directory from the root of the file system. It provides the full path starting from the root directory (/ on Unix-like systems or C:\ on Windows). Here’s a simple way to understand it:

An example of an absolute path in [Linux](https://phoenixnap.com/kb/what-is-linux) is:

/home/user/documents/file.txt

## Relative Path

 A relative path is the location of a file or directory relative to your current working directory. Unlike an absolute path, the relative path doesn't start with a slash or drive letter since it doesn't start from the root directory.

When a current working directory is */home/user/documents*, and you want to reference a file named *file.txt* in the same directory, use the following relative path:

file.txt