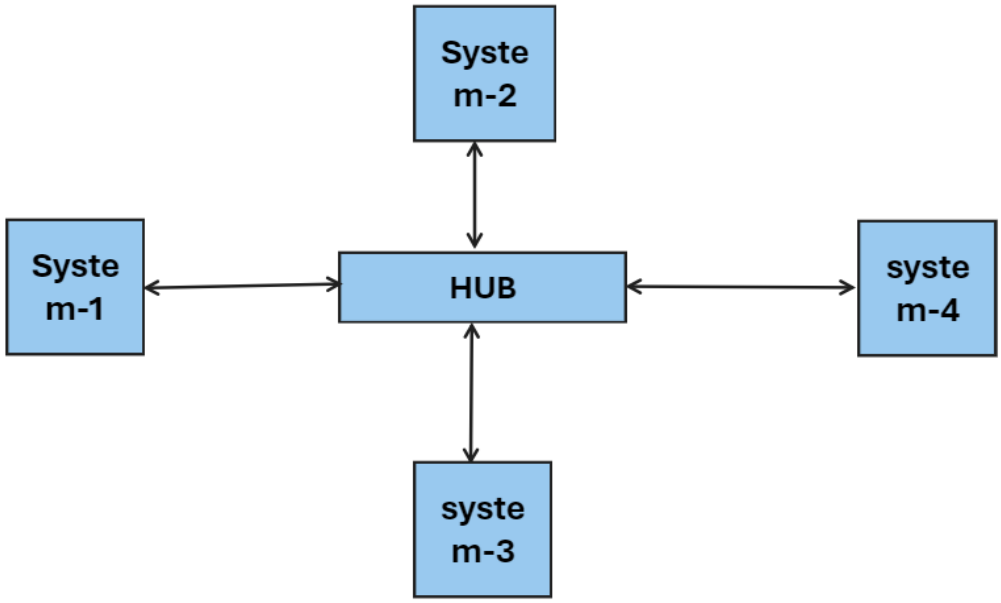
## Tasks

# DAY-1

**Assignment-1:-**

Draw Your Home Network Topology and explain how you are accessing the RPS Lab Environment

- **Home Network Topology(Star Topology):**




- **How we are accessing the RPS Lab Environment:**

  Step-1:Go to the URL which they provide.

  Step-2:Enter your username and password(Make sure entered username and password is correct).

  Step-3:After successfully Login you entered into VMWARE(UBUNTU).

  Step-4:After Entering into cloud-lab you have to enter your RPS lab password.

  Step-5:Then You entered into a UBUNTU machine that is cloud lab.

  Step-6:Use The resources.

**Assignment 2:** Identify a real-world application for both parallel computing and networked systems. Explain how these technologies are used and why they are important in that context.

**Real-Time Example of Parallel Computing:**

Real-Time Example of parallel computing is in the field of movie rendering. When creating visual effects for films, rendering plays a crucial role in generating the final film that viewers see on the screen.

**Usage:**

In movie rendering, parallel computing is used to divide the rendering process into smaller tasks that can be processed simultaneously by multiple processors or cores. Each processor works on a different part of the scene , allowing for faster rendering times compared to sequential processing.

**Importance:**

Parallel computing enables movie studios to render high-quality images and visual effects efficiently, reducing production time and costs.

**Real-Time Example of Networked Systems:**

Real-Time Example of networked systems is in online multiplayer gaming. In online multiplayer games, players from around the world connect to a main server interact with each other in real-time.

**Usage:**

Networked systems enable seamless communication between players, allowing them to collaborate and interact with the gaming environment. Players send and receive data packets over the network and game states to the server and other players.

**Importance:**

The importance of networked systems in online gaming lies in providing a smooth and responsive gaming experience for players, regardless

of their location. By maintaining low-latency and reliable connections, networked systems ensure that players can enjoy fast action.

# DAY-2

**Assignment 1:**Design Pattern Explanation - Prepare a one-page summary explaining the MVC (Model-View-Controller) design pattern and its two variants. Use diagrams to illustrate their structures and briefly discuss when each variant might be more appropriate to use than the others.

**MVC (MODEL-VIEW-CONTROLLER):**

The MVC design pattern is a software architecture pattern that separates an application into three main components Model, View and Controller. This separation allows to reuse the components.

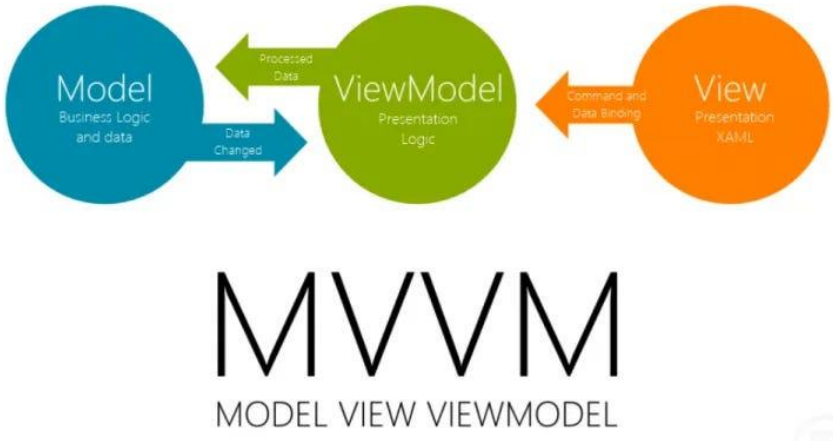
**MODEL:** The component contain all the data and business-related logic, i.e the database of an Application.

**VIEW:** This Component is used for the UI Logic and how to visually present the data, i.e the front-end of a web Application or Mobile Application.

**CONTROLLER: T**his Component act as interface between the Model and view, and handles all the incoming Requests, By using the data from the Model and send it to a View to render the Desired output.

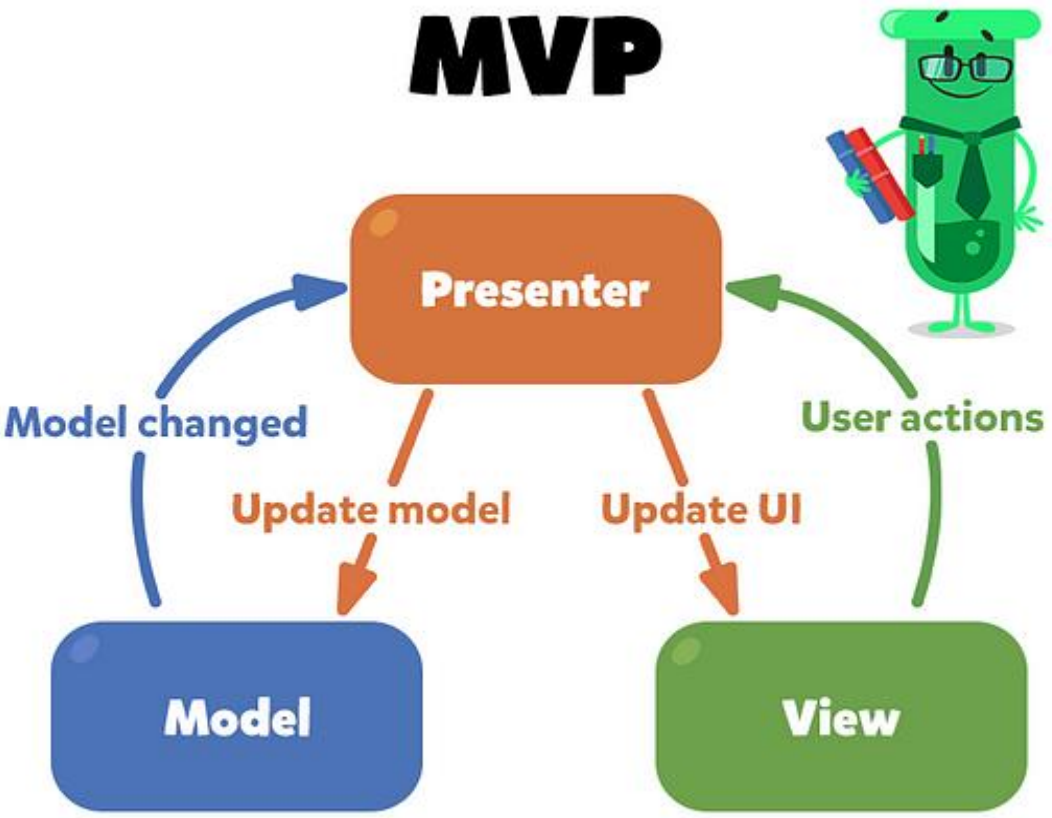
**TWO VARIANTS:**

**1.MODEL-VIEW-VIEWMODEL(MVVM):**



- It is a variation of MVC primarily used in client-side UI frameworks like WPF and AngularJS.
- MVVM is particularly well-suited for applications with complex UI logic and data binding requirements.
- View Model acts as an intermediary between the View and the Model, abstracting the View's state and behaviour.
- View Model exposes data and commands from the Model to the View, often using data binding techniques.

**2.MODEL-VIEW-PRESENTER(MVP):**




- Model represents the data and the rules that govern access to and updates of this data. It is responsible for managing the data and providing it to the Presenter.
- View is responsible for rendering the data provided by the Presenter in a form suitable for interaction, typically a user interface element.
- Presenter acts as an intermediary between the Model and the View. It retrieves data from the Model, processes it, and passes it to the View for rendering.

**Assignment 2:** Principles in Practice - Draft a one-page scenario where you apply Microservices Architecture and Event-Driven Architecture to a hypothetical e-commerce platform. Outline how SOLID principles could enhance the design. Use bullet points to indicate how DRY and KISS principles can be observed in this context.

**E-Commerce : Amazon**

Amazon-like E-Commerce platform, it aims to provide a seamless online shopping experience to customers. To achieve this, we'll design a scalable and maintainable system using Microservices Architecture and Event-Driven Architecture.

**MICROSERVICES ARCHITECTURE:**

- **Product Catalog Service**: responsible for managing products, including creation, update, and deletion.
- **Order Service:** handles order processing, including payment and fulfillment.
- **Payment Gateway Service:** It integrates with third-party payment gateways for secure transactions.
- **Search Service:** It enables users to search for products using various criteria.
- **Recommended Service:** It provides personalized product recommendations based on user Search.

**EVENT-DRIVEN ARCHITECTURE:**

Each microservice will communicate with others using events, which are triggered by specific actions.

For example:

- When a new product is created, the Product Catalog Service publishes a **ProductCreated** event. The Inventory Service listens to the **ProductCreated** event and updates the product's inventory.
- When an order is placed, the Order Service publishes an **OrderPlaced** event.
- The Payment Gateway Service listens to the **OrderPlaced** event and initiates payment processing.
- The Recommendation Service listens to the **OrderPlaced** event and updates the user's purchase history for future recommendations.

**SOLID PRINCIPLES:**

To Enhance the design we'll apply Solid Principles

- **Single Responsibility Principle (SRP):** Each microservice has a single responsibility, making it easier to maintain and update.

- **Open/Closed Principle (OCP):** Microservices are designed to be open for extension but closed for modification, allowing for easy integration of new features.
- **Liskov Substitution Principle (LSP):** Microservices can be substituted with alternative implementations without affecting the overall system.
- **Interface Segregation Principle (ISP)**: Each microservice exposes a specific interface, making it easier to understand and use.
- **Dependency Inversion Principle (DIP):** Microservices depend on abstractions rather than concrete implementations, reducing coupling and increasing flexibility.

**DRY (Don't Repeat Yourself):**

- Reusable components, such as authentication middleware or logging services, are implemented once and shared across multiple microservices.
- Common business logic, such as calculating shipping costs or handling discounts, is encapsulated in shared libraries or microservices to avoid duplication.

**KISS (Keep it Simple, Stupid):**

- Microservices are designed to be simple, focused, and easy to understand.
- Each Service solve a specific problem without necessary Complexity.
- Complexity is managed by breaking down the system into smaller, manageable components.

**Assignment 3:**

Trends and Cloud Services Overview - Write a three-paragraph report covering: 1) the benefits of server-less architecture, 2) the concept of Progressive Web Apps (PWAs), and 3) the role of AI and Machine Learning in software architecture. Then, in one paragraph, describe the cloud computing service models (SaaS, PaaS, IaaS) and their use cases.

**Server-less Architecture Benefits:**

Server-less architecture is a cloud computing execution model where the cloud provider dynamically manages the allocation of machine resources. The leading benefit of server-less architecture is the reduction of operational overhead, as the cloud provider handles infrastructure management, patching, and scaling. This model allows developers to focus on writing code and creating features that bring value to end-users. Additionally, server-less architecture offers a pay-per-use cost structure, contributing to significant cost savings compared to traditional server-based architectures.

**Progressive Web apps (PWAs):**

Progressive Web Apps (PWAs) represent a collection of technologies and design concepts that aim to deliver a native app-like experience in a web browser. PWAs are web applications that can work offline, receive push notifications, and be installed on a user's device, just like native apps. Key benefits of PWAs include improved performance, increased engagement, and the ability to reach a broader audience without the need for app store

submissions. PWAs are also cost-effective, as they leverage a single codebase for multiple platforms.

**AI and Machine Learning (ML) in Software Architecture:**

Artificial Intelligence (AI) and Machine Learning (ML) are increasingly being integrated into software architecture to enhance user experiences, streamline processes, and drive data-driven decision-making. AI/ML models can analyze vast amounts of data, identify patterns, and make predictions or recommendations. Incorporating AI/ML into software architecture enables more personalized user experiences, improved automation, and better insights. Edge computing, which involves processing data closer to the source, is an emerging trend in AI/ML-driven software architecture, reducing latency and bandwidth requirements.

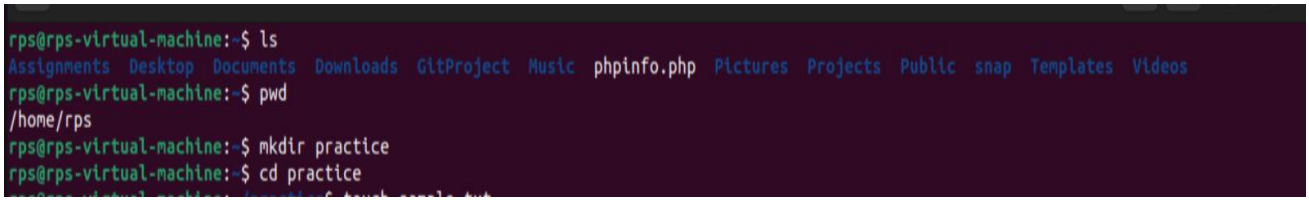
**Cloud Computing Service Models:**

- **Software as a Service (SaaS):** SaaS is a delivery model where a third-party provider hosts and manages applications, making them available to customers over the internet. Use cases include productivity tools (e.g., Microsoft Office 365).
- **Platform as a Service (PaaS):** PaaS provides a complete development and deployment environment in the cloud, enabling developers to build, test, and deploy applications without managing infrastructure.
- **Infrastructure as a Service (IaaS):** IaaS offers virtualized computing resources, such as servers, storage, and networking, on-demand in the cloud. IaaS is ideal for organizations looking to maintain control over their infrastructure while benefiting from cloud scalability and cost savings.

# LINUX [Tasks]

## Day-1

**Task 1:** File Navigation and Directory Structure
**Objective:** Demonstrate basic file navigation and directory structure understanding.
**Action Items:**
List all files in the home directory using a single command.
Display the absolute path of the current working directory.
Create a new directory called Practice in the home directory and navigate into it.




**Task 2:** File Management Commands

**Objective:** Use file management commands to organize files and directories.
**Action Items:**
In the Practice directory, create a new file called sample.txt.
Copy sample.txt to a new file called duplicate.txt.
Delete duplicate.txt using a command-line command.

```
rps@rps-virtual-machine:~/practice$ touch sample.txt
rps@rps-virtual-machine:~/practice$ cp sample.txt duplicate.txt
rps@rps-virtual-machine:~/practice$ rm duplicate.txt
```

**Task 3:** Using grep to Search within Files
**Objective:** Utilize grep to search text within files.
**Action Items:**
Use grep to find all instances of the word "Linux" in the LinuxHistory.txt file.
Redirect the output to a new file called LinuxInstances.txt.

```
rps@rps-virtual-machine:~/practice$ vi Linuxhistory.txt
rps@rps-virtual-machine:~/practice$ grep "Linux" Linuxhistory.txt>LinuxInstance.txt
rps@rps-virtual-machine:~/practice$ ls
Linuxhistory.txt  LinuxInstance.txt  sample.txt
```

**Task 4:** Permissions and Ownership
**Objective:** Understand and modify file permissions and ownership.
**Action Items:**
View the current permissions for sample.txt.
Change the permissions to read-only for the owner and no permissions for others.

```
rps@rps-virtual-machine:~/practice$ ls -l sample.txt
-rw-rw-r-- 1 rps rps 0 May  9 14:36 sample.txt
rps@rps-virtual-machine:~/practice$ chmod 400 sample.txt
rps@rps-virtual-machine:~/practice$ ls -l sample.txt
-r-------- 1 rps rps 20 May  9 14:47 sample.txt
```

**Task 5:** Write all commands history.

1 sudo apt update
2 sudo nano /etc/resolv.conf
3 sudo apt update
4 sudo apt upgrade
5 sudo apt install gcc++
6 sudo apt install gcc
7 sudo apt install build-essentials

```
 8  sudo apt install build-essential
 9  sudo apt install gdb
10  init 0
11  gcc --version
12  cd "/home/rps/Desktop/C Demo/" && gcc first.c -o first &&
"/home/rps/Desktop/C Demo/"first
13  sudo su
14  tar xvfz node_exporter-1.7.0.linux-amd64.tar.gz
15  cd node_exporter-1.7.0.linux-amd64/
16  sudo mv node_exporter /usr/local/bin/
17  sudo tee /etc/systemd/system/node_exporter.service<<EOF
18  [Unit]
19  Description=Node Exporter
20  After=network.target
21
22  [Service]
23  User=rps
24  Group=rps
25  Type=simple
26  ExecStart=/usr/local/bin/node_exporter
27
28  [Install]
29  WantedBy=multi-user.target
30  EOF
31  sudo systemctl daemon-reload
32  sudo systemctl start node_exporter
33  sudo systemctl enable node_exporter
34  sudo systemctl status node_exporter
```

```
35  sudo apt install gcc
36  sudo apt install git
37  sudo apt install make
38  sudo apt install vim
39  sudo apt install g++
40  sudo apt install gedit
41  sudo apt install cmake
42  sudo apt install g++
43  cd
44  sudo apt update -y
45  sudo apt install mysql-server
46  sudo systemctl status mysql.service
47  sudo mysql
48  sudo mysql_secure_installation
49  sudo mysql
50  sudo mysql_secure_installation
51  sudo mysql -u root -p
52  sudo snap install mysql-workbench-community
53  java -version
54  mysql --version
55  python --version
56  man ls
57  man cp
58  gcc -man
59  clear
60  man
61  man man
```

```
  62  vi hello1.txt
  63  vi hello.txt
  64  ls
  65  ls Desktop
  66  ls Lang
  67  ls Programs
  68  vi hello2.txt
  69  cp ./Programs/hello.txt./Lang
  70  cp ./Programs/hello.txt ./Lang
  71  man cp
  72  cp hello1.txt hello2.txt
72  cp hello1.txt hello2.txt
  73  cat hello1.txt
  74  cat hello2.txt
  75  cp hello2.txt hello1.txt
  76  cat hello1.txt
  77  cp hello1.txt copy.txt
  78  cat copy.txt
  79  ls
  80  cp copy.txt hello1.txt hello2.txt Programs
  81  cp copy.txt hello1.txt hello2.txt ~/Desktop/Programs
  82  ls
  83  cd ~/Desktop/Programs
  84  ls
  85  cp -i hello.txt copy.txt
  86  cat copy.txt
  87  cat hello.txt
```

```
 88  cp -r Music Desktop
 89  cd ~/Desktop
 90  ls
 91  ./Home
 92  man cp
 93  ls
 94  cp -r Music Videos
 95  cd ~/Desktop
 96  ls
 97  cp -r Lang Programs
 98  cd ~/Lang
 99  cd Lang
100  clear
101  ls
102  cp -v hello.txt demo.txt
103  cp -v hello1.txt demo.txt
104  cat demo.txt
105  ls
106  cp demo.txt /Programs
107  cp demo.txt ./Programs
108  cd Programs
109  cd ~/Desktop
110  ls
111  cd Programs
112  ls
113  cd Lang
114  ls
```

```
115  clear
116  su
117  sudo
118  who
119  -h
120  sudo -h
121  ls -l copy.txt
122  ls -l hello.txt
123  ls -l hello1.txt
124  ls -l Programs
125  clear
126  cd ~
127  ls
128  cd Downloads
129  ls
130  cd ~
131  cd Documnets
132  cd Documents
133  ls
134  mv linux-commands-handbook.pdf ~/Desktop
135  clear
136  cd ~
137  ls
138  cd Desktop
139  ls
140  cd ~
141  ls
```

```
  142  cd Documnets
  143  cd Documents
  144  ls
  145  mv DS286.AUG2016.Lab2_.cpp_tutorial.pdf ~/Desktop
  146  cd ~
  147  cd Desktop
  148  ls
  149  clear
  150  mkdir f1
  151  ls
  152  cd f1
  153  touch first.txt
  154  touch second.txt
  155  touch third.txt
  156  clear
  157  cd ~
  158  cd Desktop
  159  ls
  160  mkdir F1
  161  ls
  162  cd F1
  163  cat>>first.txt
  164  cat>>second.txt
  165  cat>>third.txt
  166  ls
  167  cp -r F1 ~/Downloads
  168  cd .
```

```
  169  cd ..
  170  cp -r F1 ~/Downloads
  171  cd ~
  172  cd Downloads
  173  ls
  174  cd F1
  175  ls
  176  cat first.txt
  177  echo *.txt
  178  clear
  179  cd ~
  180  history
  181  who am i
  182  whoami
  183  su
  184  help cp
  185  info cp
  186  man -cp
  187  man -k cp
  188  cd ~
  189  ls
  190  cd Desktop
191  ls
  192  cp Programs /Desktop/snap
  193  cp -r  Programs /Desktop/snap
  194  cp -r  Programs snap
  195  cd snap
```

196 ls
197 cd Lang
198 ls
199 dpkg -S /bin/cp
200 clear
201 cd ~
202 cd desktop
203 ls
204 cd Desktop
205 mkdir test
206 ls
207 cd test
208 vi hello.txt
209 ls
210 cat hello.txt
211 cp hello.txt copy.txt
212 ls
213 cat copy.txt
214 gcc --version
215 cd c-Programs
216 ls
217 cd Documents/C-Programs
218 ls
219 clear
220 gcc hello.c
221 ./a.out
222 cat tex1.txt

```
  223  cat text.txt
 224  man sudo
  225  sudo nano text.txt
  226  nano text.txt
  227  cd ~
  228  sudo apt-get update
  229  sudo apt-get upgrade
  230  man unmask
  231  man tractroute
  232  ls
  233  cd Desktop
  234  pwd
  235  ls
  236  mv F1 Lang
  237  ls
  238  cd Lang
  239  ls
  240  pwd
  241  cd ..
  242  cd Desktop/Lang
  243  ls
  244  open demo.txt
  245  open Files
  246  cd ...
  247  cd ..
  248  open Files
  249  open Trash
```

```
250  open Desktop
251  open .
252  cd Desktop/Lang
253  ln copy.txt demo.txt
254  ln copy.txt demo1.txt
255  ls
256  cat copy.txt
257  cat demo1.txt
258  cat demo.txt
259  ln -s  demo.txt demo2.txt
260  ls
261  cat demo2.txt
262  rm demo.txt
263  ls
264  cat demo2.txt
265  gzip copy.txt
266  ls
267  gzip hello1.txt
268  ls
269  gzip -d hello1.txt
270  ls
271  gzip hello1.txt hello2.txt
272  ls
273  gzip -r F1
274  ls
275  cd F1
276  ls
```

```
277  cd ..
278  ls -al
279  alias ll='ls'
280  ll
281  less copy.txt
282  tail -f copy.txt
283  open copy.txt
284  tail -n 1 copy.txt
285  tail -n 0 copy.txt
286  tail -n 2 copy.txt
287  tail -n+ 2 copy.txt
288  tail -n +2 copy.txt
289  tail -n +1 copy.txt
290  ls -al copy.txt
291  echo copy>>copy.txt
292  ls -al copy.txt
293  cat copy.txt
294  wc -l copy.txt
295  wc -w copy.txt
296  wc -c copy.txt
297  grep welcome copy.txt
298  cat copy.txt
299  grep copy copy.txt
300  grep -nC 2 copy copy.txt
301  grep -n copy copy.txt
302  grep -f copy copy.txt
303  grep -i copy copy.txt
```

```
304  grep -i COPY copy.txt
305  sort copy.txt
306  cat copy.txt
307  sort -r  copy.txt
308  cat>>copy.txt
309  sort -u copy.txt
310  sort copy.txt
311  ls
312  gzip -d hello1.txt.gz
313  ls
314  gzip -d hello2.txt.gz
315  diff hello1.txt hello2.txt
316  cat hello1.txt
317  cat hhello2.txt
318  cat hello2.txt
319  cat>>hello2.txt
320  diff hello1.txt hello2.txt
321  diff hello2.txt hello1.txt
322  diff -y hello1.txt hello2.txt
323  diff -y  hello2.txt hello1.txt
324  echo "Gowtham"
325  who
326  chown rps hello1.txt
327  who
328  unmask
329  unmass
330  man unmask
```

```
331  ls-al
332  ls -al
333  chmod a+r demo1.txt
334  chmod 000 dem01.txt
335  chmod 000 demo1.txt
336  ls -al
337  du *
338  du
339  df
340  df copy.txt
341  ping google.com
342  vi demo3.txt
343  cat demo3.txt
344  less demo3.txt
345  tail demo3.txt
346  tail copy.txt
347  chmod 777 copy.txt
348  ls -al
349  tail copy.txt
350  grep Apple
351  grep Apple copy.txt
352  uniq copy.txt
353  ps
354  gcc --version
355  history
356  clear
357  history
```

```
  358  sudo adduser Gowtham
  359  who
 360  su Gowtham
  361  cat/etc/passwd
  362  cd ~
  363  sudo adduser Gowtham
  364  cut -d: -f1 /etc/passwd
  365  who
  366  sudo adduser gowtham
  367  who
  368  cut -d: -f1 /etc/passwd
  369  su gowtham
  370  ip addr show
  371  ifconfig
  372  sudo apt install net-tools
  373  route -n
  374  trackroute www.google.com
  375  sudo install <deb name>
  376  sudo apt install <deb name>
  377  sudo apt install trackroute
  378  cd ~
  379  history
  380  ls
  381  cd Desktop
  382  ls
  383  cd Lang
  384  ls
```

```
  385  ls -l copy.txt
  386  ls
  387  ln copy.txt hello.txt
  388  ls -l copy.txt
  389  cat hello.txt
  390  ls
  391  cat>>hello.txt
  392  cat copy.txt
  393  ln -s copy.txt hello.txt
  394  ln -s copy.txt hell.txt
  395  ls
  396  rm copy.txt
397  ls
  398  cat hello.txt
  399  ls -l hello.txt
  400  ls -al
  401  wc -c demo1.txt
  402  cat demo1.txt
  403  wc -l demo1.txt
  404  wc -w demo1.txt
  405  su gowtham
  406  ls
  407  cd Desktop
  408  ls
  409  cd Lang
  410  ls
  411  ls -l demo1.txt
```

```
  412  chmod 760 demo1.txt
  413  ls -l demo1.txt
  414  vi text.txt
  415  ls -l text.txt
  416  ln text.txt text1.txt
  417  ls -l text.txt
  418  ln text.txt tex1.txt
  419  ls -l text.txt
  420  ln text.txt tex1.txt
  421  ln text.txt tex2.txt
  422  ls -l text.txt
  423  man ln
  424  cat tex1.txt
  425  ls -l tex1.txt
  426  cat text1.txt
  427  cat>>tex1.txt
  428  sudo apt-get upgrade
  429  sudo apt-get update
  430  clear
  431  lsb_release -a
  432  git init
 433  cd /home
  434  ls
  435  cd rps
  436  ls
  437  git add .
  438  git commit
```

439 git config --global user.email "gowtham.poalmreddy143@gmail.com"

440 git config --global user.name "PolamReddy venkata Gowtham Reddy"

441 git remote add origin https://github.com/Gowtham9615/Test/GitProject.git

442 git push -u origin main

443 git commit

444 git push -u origin main

445 git commit

446 git push -u origin main

447 git commit -m

448 git commit -m "Comiting"

449 git commit -a

450 git branch

451 git branch -M main

452 git branch

453 git committtttttttttttt

454 git commit

455 git push -u origin main

456 git remote add origin https://github.com/Gowtham9615/Gowtham9615/GitProject.git

457 git remote add origin https://github.com/Gowtham9615/Demo/GitProject.git

458 git remote set-url origin https://github.com/Gowtham9615/Demo/GitProject.git

459 git push -u origin main

460 git add .

461 git commit

462 git push -u origin main

463 git branch

```
  464  git remote -v
  465  git remote remove origin
  466  git remote -v
  467  git remote set-url origin https://github.com/Gowtham9615/Test/GitProject.git
  468  git remote set-url origin https://github.com/Gowtham9615/Demo/GitProject.git
  469  git remote add origin https://github.com/Gowtham9615/Demo/GitProject.git
  470  git add hello.txt
  471  git add .
  472  git commit -m "commit"
  473  git push origin main
  474  remote -v
  475  git remote -v
  476  sudo apt-get install get
  477  clear
  478  git --version
  479  ls
  480  cd GitProject
  481  ls
  482  git init
  483  git add .
  484  git commit -m "Initial commit"
  485  git remote add origin https://github.com/Gowtham9615/Assignments.git
  486  git push -u origin master
  487  git status
```

```
  488  git remote set-url origin
https://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@github.com/Go
wtham9615/Assignments
  489  git push -u origin master
  490  git add .
  491  git commit -m "Commit"
  492  git push -u origin master
  493  mkdir GitProject
  494  ls
  495  cd GitProject
  496  vi hello.txt
  497  ls
  498  vi First.c
  499  gcc First.c
  500  ./a.out
  501  cd ~
  502  ls
503  mkdir Projects
  504  ls
  505  cd Projects
  506  vi hello.c
  507  ls
  508  git init
  509  git add .
  510  git commit -m "Commit"
  511  git remote add origin https://github.com/Gowtham9615/Demo.git
  512  git push -u origin master
  513  cd ~
```

```
514  mkdir Assignments
515  vi hello.c
516  ls
517  mv hello.c ~/Assignments
518  cd Assignments
519  ls
520  git init
521  git add .
522  git commit -m "commit"
523  git remote add origin https://github.com/Gowtham9615/Tasks.git
524  git push -u origin master
525  ls
526  vi text.txt
527  ls
528  git add .
529  git commit -m "Commit changes"
530  git add text.txt
531  git commit -m "Initial Commit"
532  git branch
533  ls
534  git push -u origin main
535  git push -u origin master
536  sudo mysql_secure_installation
537  sudo apt update
538  sudo apt install mariadb-server mariadb-client
539  sudo systemctl status mariadb
540  sudo mysql -u root -p
```

```
  541  create database myapp;
  542  sudo mysql -u root -p
  543  sudo apt install apache2 php
  544  sudo systemctl status apache2
  545  sudo apt search ^php-
  546  sudo apt install php-curl php-gd php-mbstring php-mysql php-zip php-
json php-xml
  547  sudo systemctl restart apache2
  548  ls
  549  ls /etc/apache2/mods-available
  550  sudo a2enmod rewrite
  551  sudo systemctl restart apache2
  552  sudo a2query -m
  553  sudo nano /etc/apache2/envvars
  554  sudo chown -Rf $(whoami):$(whoami) /var/www/html
  555  sudo systemctl restart apache2
  556  systemctl status apache2.service
  557  sudo systemctl start apache2
  558  sudo systemctl status apache2
  559  sudo systemctl start apache2
  560  sudo systemctl status apache2.service
  561  sudo nano /etc/apache2/envvars
  562  sudo systemctl status apache2.service
  563  sudo systemctl start apache2
  564  sudo systemctl status apache2
  565  who
  566  whoami
  567  sudo chown -Rf $(whoami):$(whoami) /var/www/html
```

```
568  sudo systemctl status apache2
569  sudo systemctl restart apache2
570  pwd
571  vi index.php
572  cat index.php
573  vi phpinfo.php
574  cat phpinfo.php
575  vi index.php
576  sudo systemctl status apache2
577  ls
578  vi index.php
579  sudo gnome-text-editor phpinfo.php
580  sudo nano /etc/apache2/envvars
581  who
582  whoami
583  sudo chown -Rf $(whoami):$(whoami) /var/www/html
584  sudo nano /etc/apache2/envvars
585  sudo gnome-text-editor /var/www/html/phpinfo.php
586  ls
587  sudo gnome-text-editor /var/www/html/phpinfo.php
588  sudo mysql -u root -p
589  sudo apt install apache2 php
590  sudo systemctl status apache2
591  sudo apt search ^php-
592  sudo a2query -m
593  sudo a2dismod moduleName
594  sudo nano /etc/apache2/envvars
```

```
595  sudo chown -Rf $(whoami):$(whoami) /var/www/html
596  sudo systemctl restart apache2
597  sudo systemctl status apache
598  sudo systemctl status apache2
599  sudo nano /etc/apache2/envvars
600  history
```