

Problem Statement:

Download and install Ubuntu in VirtualBox.

Aim:

To download install ubuntu in VirtualBox.

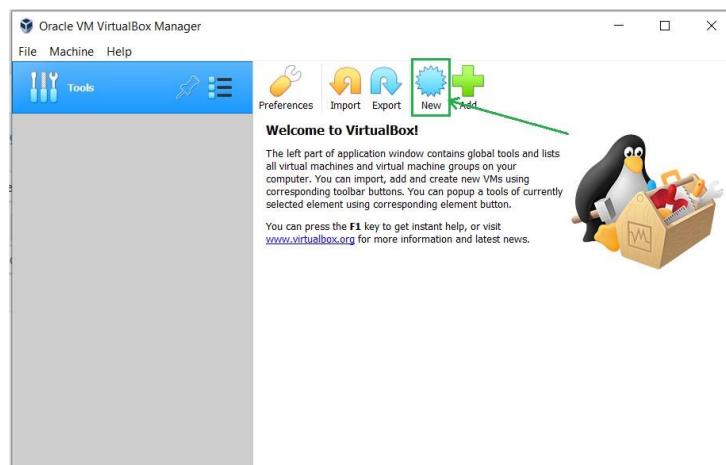
Procedure:

First visit the official website of Ubuntu from your favourite web browser. Once the page loads, click on Download.

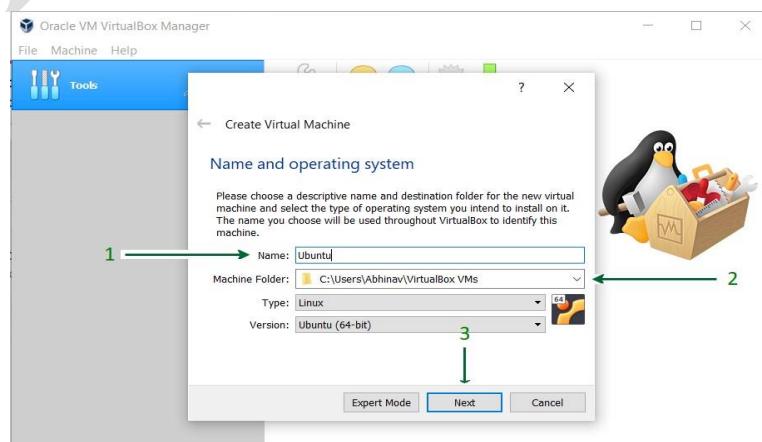


After the downloading is over, you can install Ubuntu on VirtualBox with the help of following instructions:

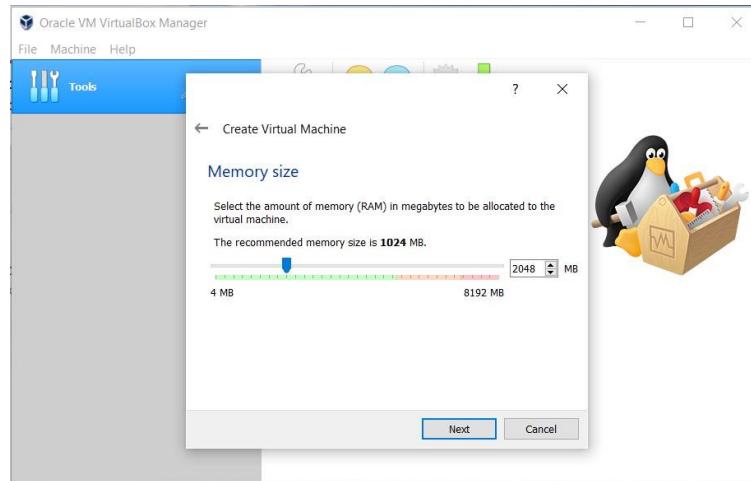
1. Open **VirtualBox** and click on the **New** button.



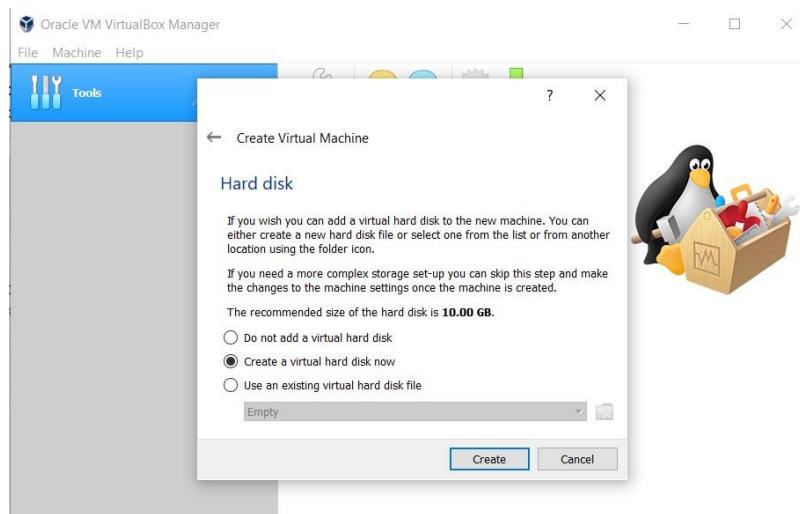
2. Give a name to your Virtual Machine and select the location for it to install.



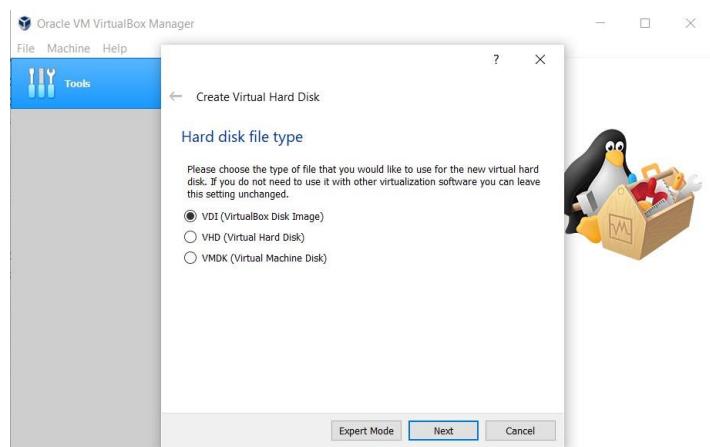
3. Assign RAM size to your Virtual Machine.



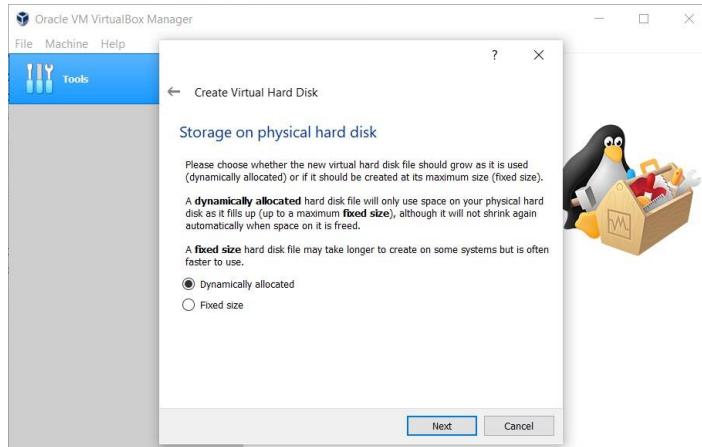
4. Create a Virtual Hard disk for the machine to store files.



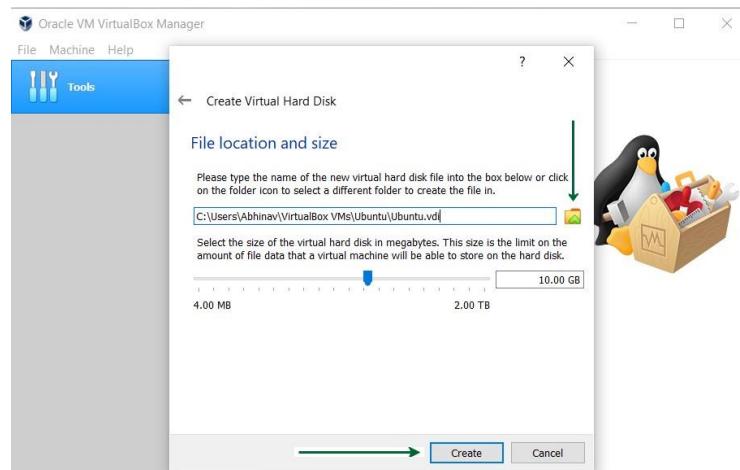
5. Select the type of Hard disk. Using **VDI** type is recommended.



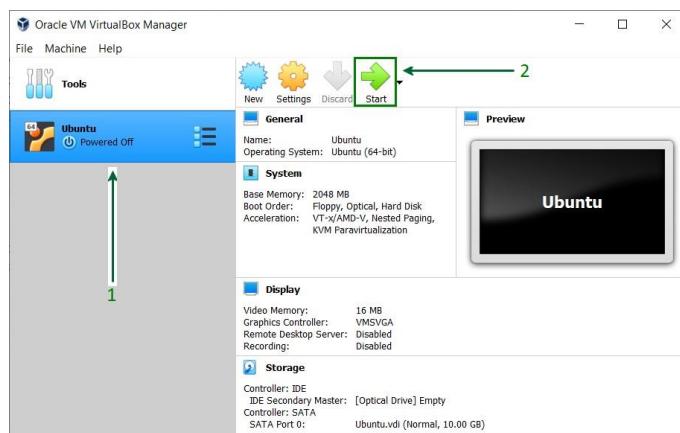
6. Either of the physical storage type can be selected. Using Dynamically allocated disk is by default recommended.



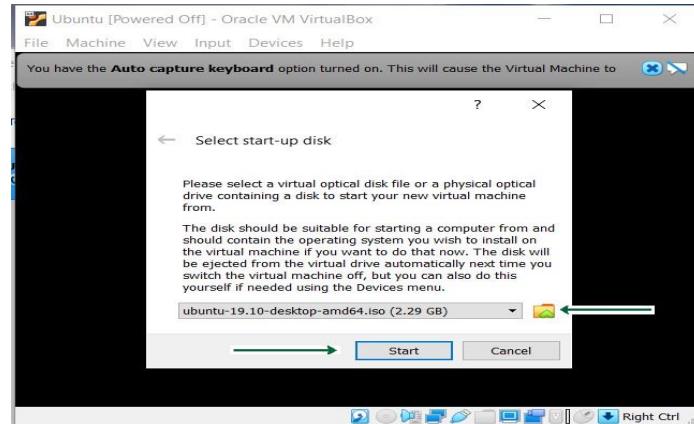
7. Select disk size and provide the destination folder to install.



8. After the Disk creation is done, boot the Virtual Machine and begin installing Ubuntu.



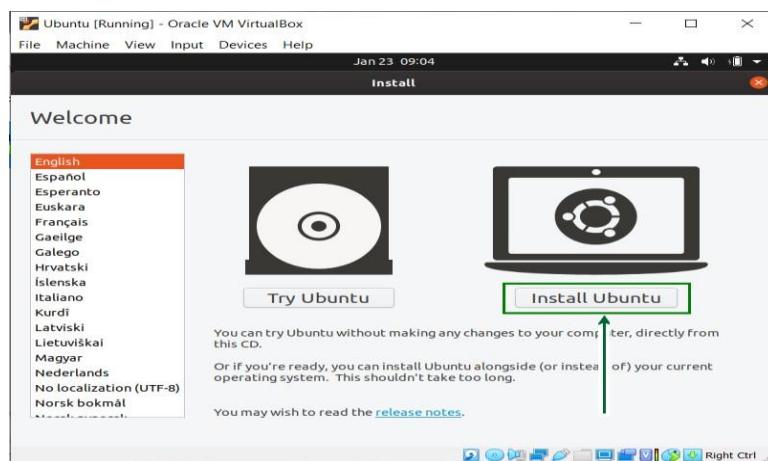
9. If the installation disk is not automatically detected. Browse the file location and select the ISO file for Ubuntu.



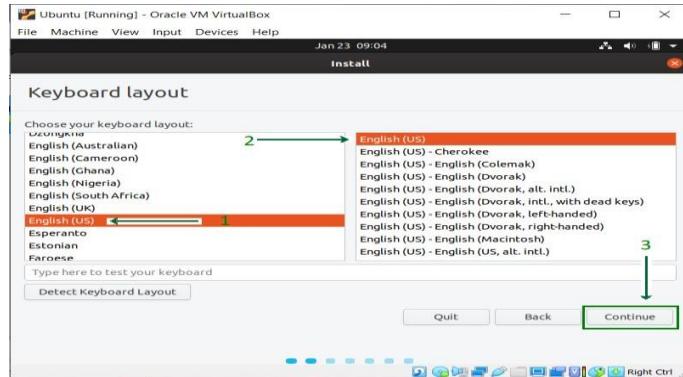
10. Proceed with the installation file and wait for further options.



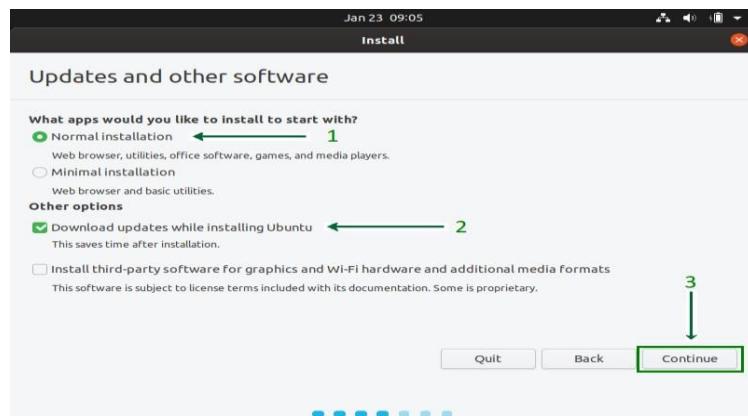
11. Click on the Install Ubuntu option, this might look different for other Ubuntu versions.



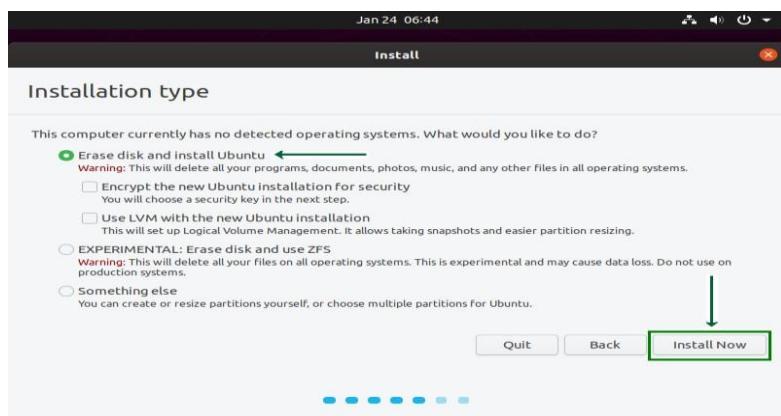
12. Select Keyboard layout, if the defaults are compatible, just click on the **continue** button and proceed.



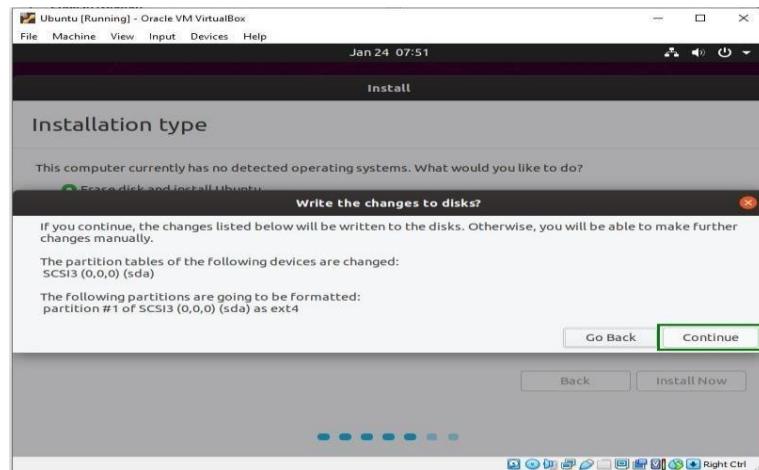
13. Select installation type. By default, it is set to Normal installation, which is recommended, but it can also be changed to Minimal installation if there is no need for all Ubuntu features.



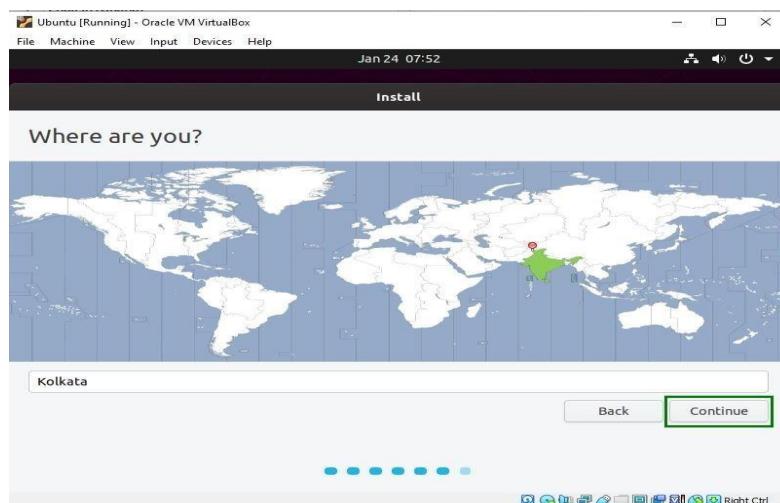
14. Click on the **Install Now** button and carry on with the installation. Do not get worried with the Erase disk option, it will only be effective inside the virtual machine, other system files outside the VirtualBox remain intact.



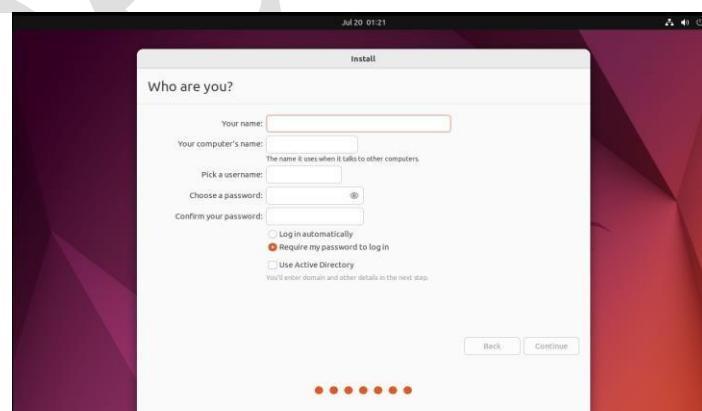
15. Click on the **continue** button, and proceed with writing changes on the disk.



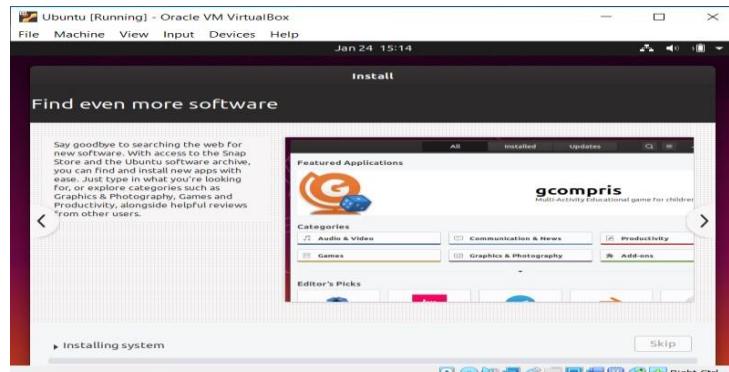
16. Select your location to set the Time Zone.



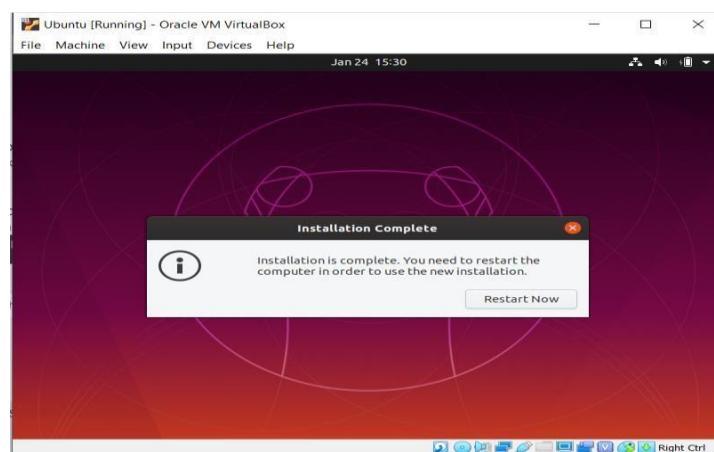
17. Choose a name for your computer and set a password to secure login info.



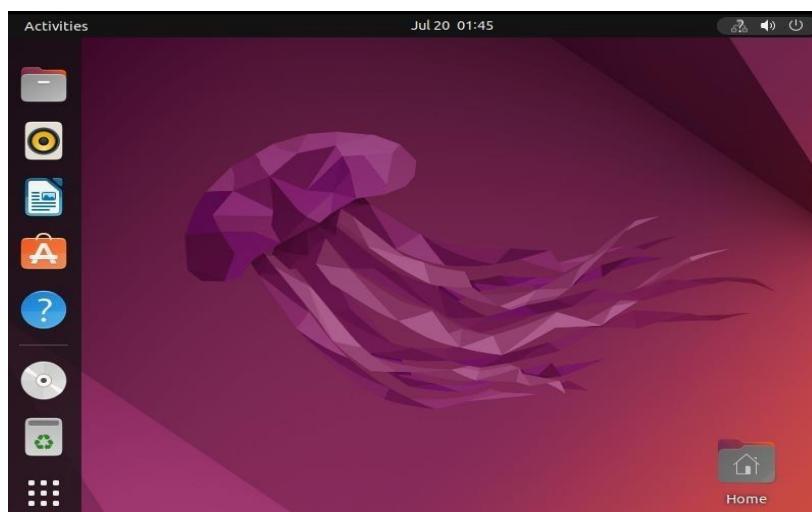
18. Wait for the installation process to complete.



19. Once the installation process is over, reboot your Virtual Machine.



20. Voila!! You're finished with the installation process. Now you can use Ubuntu along with the Windows, without creating a dual boot.



Result:

Downloading and Installation of Ubuntu on VirtualBox was completed successfully.

J.N.T.U.A College of Engineering (Autonomous), Pulivendula

Department of Computer Science and

Engineering

Problem Statement:

Execute Linux commands required for DevOps.

Aim:

To execute Linux Commands required for DevOps.

Linux Commands:

1.Man command:

man command in Linux is used to display the user manual of any command that we can run on the terminal

Syntax: \$ man [COMMAND NAME]

Output:

```
LS(1)                               User Commands                               LS(1)

NAME
ls - list directory contents

SYNOPSIS
ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
      do not ignore entries starting with .

-A, --almost-all
      do not list implied . and ..

--author
      with -l, print the author of each file

-b, --escape
      print C-style escapes for nongraphic characters

Manual page ls(1) line 1/246 9% (press h for help or q to quit)
```

2.pwd Command:

The **pwd** command is used to display the location of the current working directory.

Syntax: \$ pwd

Output:

```
vrupeh@vrupeh-virtual-machine:~$ pwd
/home/vrupeh
vrupeh@vrupeh-virtual-machine:~$ mkdir devops_lab
vrupeh@vrupeh-virtual-machine:~$ ls
```

3.mkdir Command:

The **mkdir** command is used to create a new directory under any directory.

Syntax: \$ mkdir <directory name>

Output:

```
vrupeh@vrupeh-virtual-machine:~$ mkdir devops_lab
vrupeh@vrupeh-virtual-machine:~$ ls
Desktop  Documents  Music  Public  Templates
devops_lab  Downloads  Pictures  snap  Videos
```

4.rmdir Command:

The **rmdir** command is used to delete a directory.

| | | | | | |
|------|--|----------|--|----------|--|
| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|

Syntax: -\$ rmdir <directory name>

Output:

```
vrupesh@vrupesh-virtual-machine: $ rmdir devops_lab
vrupesh@vrupesh-virtual-machine: $ ls
Desktop Downloads Pictures snap Videos
Documents Music Public Templates
```

5.ls Command:

The ls command is used to display a list of content of a directory.

Syntax: -\$ ls

Output:

```
vrupesh@vrupesh-virtual-machine: $ ls
Desktop Documents Music Public Templates
devops_lab Downloads Pictures snap Videos
vrupesh@vrupesh-virtual-machine: $ rm devops_lab
rm: cannot remove 'devops_lab': Is a directory
```

6.cd Command:

The cd command is used to change the current directory.

Syntax: -\$ cd <directory name>

Output:

```
vrupesh@vrupesh-virtual-machine: $ cd /home
vrupesh@vrupesh-virtual-machine: /home$ pwd
/home
vrupesh@vrupesh-virtual-machine: /home$
```

7.touch Command

The touch command is used to create empty files. We can create multiple empty files by executing it once.

Syntax: -\$ touch <file name>

-\$ touch <file 1><file 2>

-\$ touch <file 1><file 2>

Output:

```
vrupesh@vrupesh-virtual-machine: $ touch sample.txt
vrupesh@vrupesh-virtual-machine: $ ls
Desktop Downloads Pictures sample.txt Templates
Documents Music Public snap Videos
```

8.echo command:

It is used to display a line of text that is passed in as an argument.

Syntax: echo [option] [string]

Output:

```
vrupesh@vrupesh-virtual-machine: /home$ echo "hello world"
hello world
vrupesh@vrupesh-virtual-machine: /home$ whoami
vrupesh
vrupesh@vrupesh-virtual-machine: /home$ su
```

9.whoami command:

It displays the username of the current user when this command is invoked.

Syntax: whoami [OPTION]

| | | | |
|------|--|----------|----------|
| Date | | EXP .N0. | Page No. |
|------|--|----------|----------|

Output:

```
vrupesh@vrupesh-virtual-machine:/home$ whoami
vrupesh
vrupesh@vrupesh-virtual-machine:/home$ su
Password:
```

10.su Command:

The [su](#) command provides administrative access to another user. In other words, it allows access of the Linux shell to another user.

Syntax: su <user name>

Output:-

```
vrupesh@vrupesh-virtual-machine:/home$ whoami
vrupesh
vrupesh@vrupesh-virtual-machine:/home$ su
Password:
```

11.sudo bash command:

It allows users to run programs with the security privileges of another user

Syntax:\$ sudo bash

Output:-

```
vrupesh@vrupesh-virtual-machine:/home$ sudo bash
[sudo] password for vrupesh:
```

12.cat Command:

The [cat](#) command is a multiple-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file and more.

Syntax: cat <file name>

Output:-

```
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ cat sample.txt
Welcome this is ubuntu.
team leader jaswanth
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ rm sample.txt
```

13.rm command:

It helps to delete files and directories

Syntax: rm [OPTION]... FILE...

Output:-

```
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ rm sample.txt
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ ls
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$
```

14.cp command:

To copy files or directories from one location to another

Syntax:cp [OPTION]

Output:-

J.N.T.U.A College of Engineering (Autonomous), Pulivendula

Department of Computer Science and

```
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ cp sample.txt team10.txt
vrupesh@vrupesh-virtual-machine:~/Desktop/devops lab$ ls
sample.txt team10.txt team11.txt
```

15.mv command:

mv stands for **move**. **mv** is used to move one or more files or directories from one place to another in a file system like UNIX.

Syntax: mv [Option] source destination

Output:

```
vrupeh@vrupeh-virtual-machine:~/Desktop/devops_lab$ mv sample.txt team2.txt
vrupeh@vrupeh-virtual-machine:~/Desktop/devops_lab$ ls
team10.txt  team11.txt  team2.txt
```

16.head command:

To print first N lines of a given file content.

Syntax: head [OPTION]... [FILE]...

Output:

```
vrupeh@vrupeh-virtual-machine:~/Desktop/devops_lab$ head -n 5 team2.txt
welcome to devops_lab1
welcome to devops_lab2
welcome to devops_lab3
welcome to devops_lab4
welcome to devops_lab5
```

17.Tail command:

To print last N lines of a given file content

Syntax: tail [OPTION]... [FILE]...

Output:

```
vrupeh@vrupeh-virtual-machine:~/Desktop/devops_lab$ tail -n 5 team2.txt
welcome to devops_lab1
welcome to devops_lab2
welcome to devops_lab3
welcome to devops_lab4
welcome to devops_lab5
```

18.History command:

Used to view a history of all commands previous executed inside bash terminal

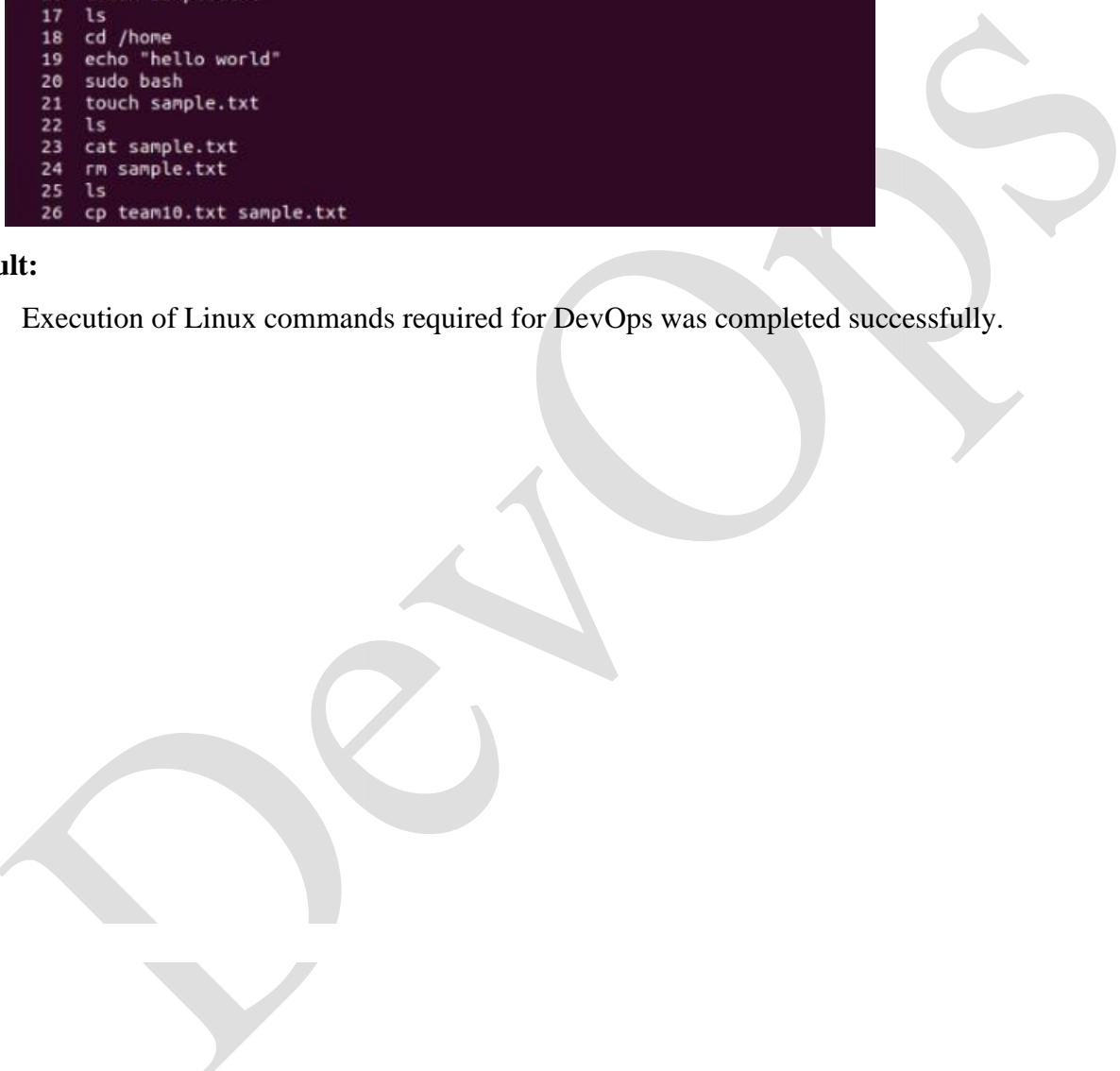
Syntax:\$history

Output:

```
welcome to devops_lab4
welcome to devops_lab5
vrupesh@vrupeh-virtual-machine:~/Desktop/devops_lab$ history
1 man ls
2 ls
3 man ls
4 pwd
5 mkdir devops_lab
6 ls
7 rm devops_lab
8 ls
9 rmdir devops_lab
10 ls
11 cd/home
12 cd /home
13 pwd
14 touch sample.txt
15 touch sampl.txt
16 touch sample.txt
17 ls
18 cd /home
19 echo "hello world"
20 sudo bash
21 touch sample.txt
22 ls
23 cat sample.txt
24 rm sample.txt
25 ls
26 cp team10.txt sample.txt
```

Result:

Execution of Linux commands required for DevOps was completed successfully.



Git Basic Commands

Problem Statement:

Execute Git basic commands for DevOps.

Aim:

To execute Git commands for DevOps.

Procedure:

Git Commands

git config

Usage: git config --global user.name “[name]”

Usage: git config --global user.email “[email address]”

This command sets the author name and email address respectively to be used with your commits.

```
spk@spk-virtual-machine:~/Desktop$ git config --global user.name "PavanSPK"
spk@spk-virtual-machine:~/Desktop$ git config --global user.email "sadupavan2000@gmail.com"
```

1.git init

Usage: git init [repository name]

This command is used to start a new repository.

```
spk@spk-virtual-machine:~/Desktop$ git init /home/spk/Documents/Demo
hint: Using 'master' as the name for the initial branch. This default branch name
      is subject to change. To configure the initial branch name to use in all
      of your new repositories, which will suppress this warning, call:
      hint:
      hint:   git config --global init.defaultBranch <name>
      hint:
      hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
      hint: 'development'. The just-created branch can be renamed via this command:
      hint:
      hint:   git branch -m <name>
Initialized empty Git repository in /home/spk/Documents/Demo/.git/
```

2.git clone

Usage: git clone [url]

This command is used to obtain a repository from an existing URL.

```
spk@spk-virtual-machine:~/Desktop$ git clone https://github.com/PavanSPK/samplegit
Cloning into 'samplegit'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

3.git add

Usage: git add [file]

This command adds a file to the staging area.

Usage: git add *

This command adds one or more to the staging area.

```
spk@spk-virtual-machine:~/Desktop$ git add file3.txt
spk@spk-virtual-machine:~/Desktop$ git add new
spk@spk-virtual-machine:~/Desktop$ git add new.txt
spk@spk-virtual-machine:~/Desktop$ git add new1.txt
```

4.git commit

Usage: git commit -m “[Type in the commit message]”

| Date | | EXP .NO. | Page No. |
|------|--|----------|----------|
|------|--|----------|----------|

This command records or snapshots the file permanently in the version history.

Usage: git commit -a

This command commits any files you've added with the git add command and also commits any files you've changed since then.

```
spk@spk-virtual-machine:~/sample$ git commit -m "first commit"
[main (root-commit) bfb326d] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello
```

5.git diff

Usage: git diff

This command shows the file differences which are not yet staged.

```
spk@spk-virtual-machine:~/sample$ git diff
```

Usage: git diff --staged

This command shows the differences between the files in the staging area and the latest version present.

```
spk@spk-virtual-machine:~/sample$ git diff --staged
diff --git a/hello b/hello
new file mode 100644
index 0000000..e69de29
diff --git a/hi b/hi
new file mode 100644
index 0000000..e69de29
```

6.git reset

Usage: git reset [file]

This command unstages the file, but it preserves the file contents.

Usage: git reset [commit]

This command undoes all the commits after the specified commit and preserves the changes locally.

Usage: git reset --hard [commit] This command discards all history and goes back to the specified commit.

```
A spk@spk-virtual-machine:~/sample$ git reset hi
```

7.git status

Usage: git status

This command lists all the files that have to be committed.

```
spk@spk-virtual-machine:~/Desktop$ git status
On branch master
? No commits yet
Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  file3.txt
  new file:  new
  new file:  new.txt
  new file:  new1.txt
Untracked files:
(use "git add <file>..." to include in what will be committed)
  Demo/
  samplegit/
```

| Date | | EXP .NO. | Page No. |
|------|--|----------|----------|
|------|--|----------|----------|

8.git rm

Usage: git rm [file]

This command deletes the file from your working directory and stages the deletion.

```
spk@spk-virtual-machine:~/sample$ git rm hi
```

9.git log

Usage: git log

This command is used to list the version history for the current branch.

Usage: git log --follow[file]

This command lists version history for a file, including the renaming of files also.

```
spk@spk-virtual-machine:~/sample$ git log
commit bfb326d66a12aeed9b698c588a49780aaa5dd495 (HEAD -> main, origin/main)
Author: PavanSPK <sandupavan2000@gmail.com>
Date:   Wed Aug 17 23:57:33 2022 +0530

first commit
```

10.git show

Usage: git show [commit]

This command shows the metadata and content changes of the specified commit.

```
spk@spk-virtual-machine:~/sample$ git show
commit bfb326d66a12aeed9b698c588a49780aaa5dd495 (HEAD -> main, origin/main)
Author: PavanSPK <sandupavan2000@gmail.com>
Date:   Wed Aug 17 23:57:33 2022 +0530

first commit

diff --git a/hello b/hello
new file mode 100644
index 000000..e69de29
```

11.git tag

Usage: git tag [commitID]

This command is used to give tags to the specified commit.

```
spk@spk-virtual-machine:~/sample$ git tag hi
```

12.git branch

Usage: git branch

This command lists all the local branches in the current repository.

```
spk@spk-virtual-machine:~/sample$ git branch
```

Usage: git branch [branch name]

This command creates a new branch.

```
spk@spk-virtual-machine:~/sample$ git branch hlo
```

13.git checkout

Usage: git checkout [branch name]

This command is used to switch from one branch to another.

Usage: git checkout -b [branch name]

This command creates a new branch and also switches to it.

```
spk@spk-virtual-machine:~/sample$ git checkout -b hi
Switched to a new branch 'hi'
```

14.git remote

Usage: git remote add [variable name] [Remote Server Link]

This command is used to connect your local repository to the remote server.

```
spk@spk-virtual-machine:~/sample$ git remote rm origin
spk@spk-virtual-machine:~/sample$ 
git remote add origin https://ghp_5DGqykaxxTG3inWAVwjLEQpG7yAJYz0mGFFK@github.com/PavanSPK/sample.git
```

15.git push

Usage: git push [variable name] master

This command sends the committed changes of master branch to your remote repository.

Usage: git push [variable name] [branch]

This command sends the branch commits to your remote repository.

Usage: git push –all [variable name]

This command pushes all branches to your remote repository.

Usage: git push [variable name] :[branch name]

This command deletes a branch on your remote repository.

```
A spk@spk-virtual-machine:~/sample$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 212 bytes | 212.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PavanSPK/sample.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

16.git pull

Usage: git pull [Repository Link]

This command fetches and merges changes on the remote server to your working directory.

```
spk@spk-virtual-machine:~/sample$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

git branch --set-upstream-to=origin/<branch> hi
```

17.git stash

Usage: git stash save

This command temporarily stores all the modified tracked files.

Usage: git stash pop

This command restores the most recently stashed files.

```
spk@spk-virtual-machine:~/sample$ git stash pop
No stash entries found.
```

Usage: git stash list

This command lists all stashed changesets.

Usage: git stash drop

This command discards the most recently stashed changeset.

```
spk@spk-virtual-machine:~/sample$ git stash
No local changes to save
```

Results:

Git basic commands for devOps was completed successfully.

| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|
|------|--|----------|--|----------|--|

Problem Statement: Push the total Project from your local system into the git by creating a Remote Repository.

Aim: To Push the total Project from our local system into the git by creating a Remote Repository.

Procedure:

1. Locate to the project folder in your system.
2. Right click and open the terminal.
3. Open the browser and login into your github.
4. Create the new repository for your project.
5. Now follow the commands to push the entire project into your repository.

Commands,

1. git config --global user.name "PavanSPK"
2. git config --global user.email "sandupavan2000@gmail.com"
3. git init
4. git add .
5. git commit -m "RPA project"
6. git branch -M main
7. git remote add origin <https://github.com/PavanSPK/RPA-Project.git>

After the above command the terminal ask your to login into your git account using your credentials. Now refresh the github page..

You will see the entire project has been pushed into the repository.

Output:

Create a new repository

Repository name *

PavanSPK / RPA Project

Description (optional)

Public

Private

Initialize this repository with:

Quick setup — if you've done this kind of thing before

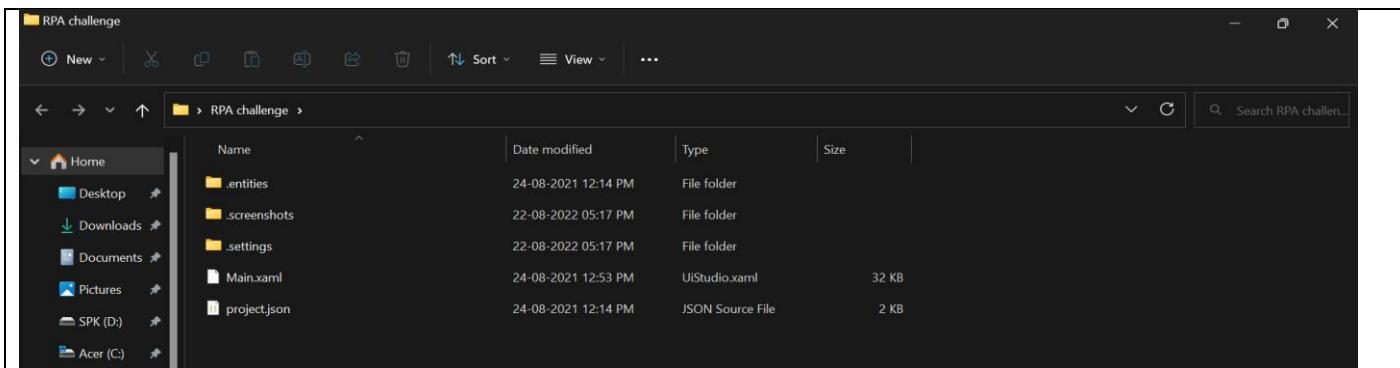
Set up in Desktop or HTTPS SSH <https://github.com/PavanSPK/RPA-Project.git>

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# RPA-Project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/PavanSPK/RPA-Project.git
git push -u origin main
```

| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|
|------|--|----------|--|----------|--|



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sandu\Desktop\RPA challenge> git init
Initialized empty Git repository in C:/Users/sandu/Desktop/RPA challenge/.git/
PS C:\Users\sandu\Desktop\RPA challenge> git add .
PS C:\Users\sandu\Desktop\RPA challenge> git commit -m "RPA project"
[master (root-commit) 154c1c8] RPA project
 41 files changed, 7521 insertions(+)
create mode 100644 .local\AllDependencies.json
create mode 100644 .local\PackageCache.json
create mode 100644 .local\ProjectSettings.json
create mode 100644 .local\db\references.db
create mode 100644 .local\nuget.cache
create mode 100644 .screenshots\1d8e3da16d51f24d3a2b70c482f851c.png
create mode 100644 .screenshots\14bce44e39c5154962c1cd115db3aa4a.png
create mode 100644 .screenshots\1d8e05797da24b7edba0f7b35aac1d24.png
create mode 100644 .screenshots\2c58130302c338bcd64e2b4944b912a8.png
create mode 100644 .screenshots\31fc4451b1847592837ec1ed374c68df.png
create mode 100644 .screenshots\5532a6a16e74b92d5e8b34965aaeb849.png
create mode 100644 .screenshots\557addcb4e6013edde56509407c8683d.png
create mode 100644 .screenshots\60604ae2cc1806245e26260dbc611b20.png
create mode 100644 .screenshots\7165d13f39bf0be6578f0736620aad68.png
create mode 100644 .screenshots\77e8b39f2f94f75f2ac122d119414157.png
create mode 100644 .screenshots\7ace31ced2ba322195eea02c220c9ed.png
create mode 100644 .screenshots\8cd74d98be042004188aa7b1e56c95ae.png
create mode 100644 .screenshots\ae29ee17c0d2f0c1b3b663f5b9e689b.png
create mode 100644 .screenshots\af5f4f079c0d171fe80f28aa0ef00732c.png
create mode 100644 .screenshots\b1797f6d9dfc77ed063226977c66c4a0.png
create mode 100644 .screenshots\c2452e3867c73195e2f327099294afa8.png

```

```

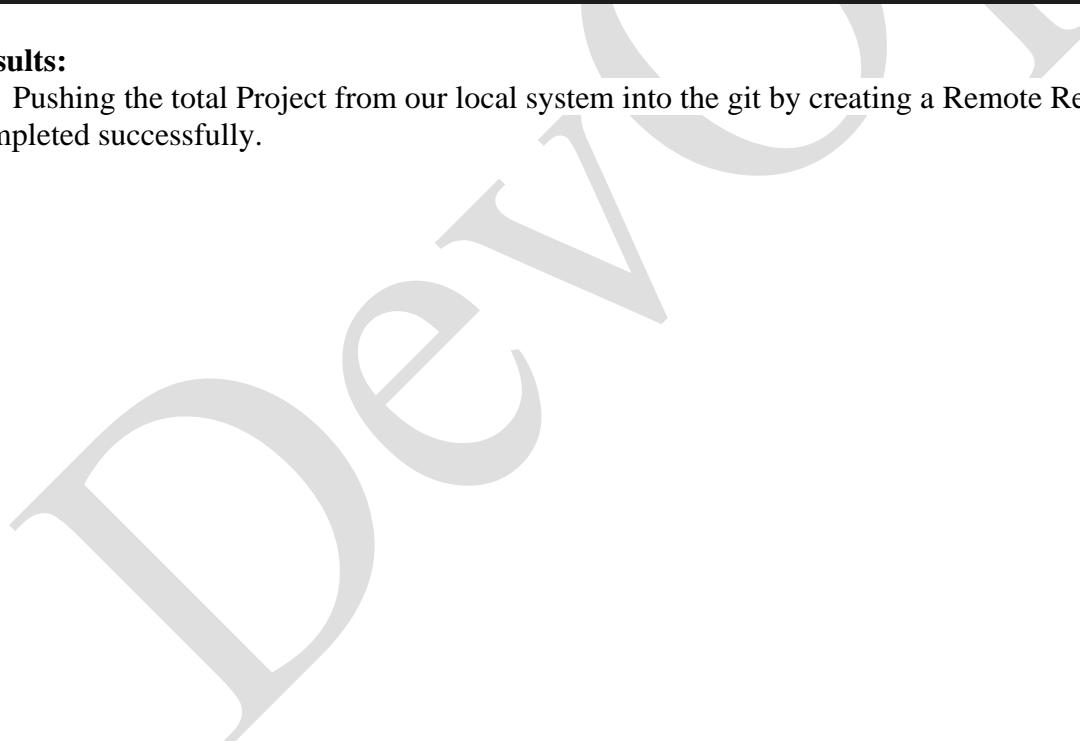
Windows PowerShell
x + v

create mode 100644 .settings/Debug/settings-9e9290da.json
create mode 100644 .settings/Debug/settings-a1b2dac1.json
create mode 100644 .settings/Debug/settings-afad0842.json
create mode 100644 .settings/Release/settings-1a5b880a.json
create mode 100644 .settings/Release/settings-5190454b.json
create mode 100644 .settings/Release/settings-590bb27b.json
create mode 100644 .settings/Release/settings-9e9290da.json
create mode 100644 .settings/Release/settings-a1b2dac1.json
create mode 100644 .settings/Release/settings-afad0842.json
create mode 100644 Main.xaml
create mode 100644 project.json
PS C:\Users\sandu\Desktop\RPA challenge> git branch -M main
PS C:\Users\sandu\Desktop\RPA challenge> git push -u origin main
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\sandu\Desktop\RPA challenge> git remote add origin https://github.com/PavanSPK/RPA-Project.git
PS C:\Users\sandu\Desktop\RPA challenge> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 42, done.
Counting objects: 100% (42/42), done.
Delta compression using up to 8 threads
Compressing objects: 100% (41/41), done.
Writing objects: 100% (42/42), 119.86 KiB | 5.21 MiB/s, done.
Total 42 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/PavanSPK/RPA-Project.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\sandu\Desktop\RPA challenge>

```

| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|
|------|--|----------|--|----------|--|



A screenshot of a GitHub repository page for 'PavanSPK/RPA-Project'. The repository is public and contains one branch ('main') and no tags. The commit history shows five commits from '154c1cb' made 2 minutes ago. The files listed are '.local', '.screenshots', '.settings', 'Main.xaml', and 'project.json', all categorized as 'RPA project'. The 'About' section indicates no description, website, or topics provided. It shows 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published and a link to 'Create a new release'. The 'Packages' section shows no packages published and a link to 'Publish your first package'. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating the date as 22-08-2022 at 05:22 PM.

Results:

Pushing the total Project from our local system into the git by creating a Remote Repository was completed successfully.

| | | | | | |
|------|--|----------|--|----------|--|
| Date | | EXP .NO. | | Page No. | |
|------|--|----------|--|----------|--|

Problem Statement: Create the file in your github globally and pull the file to your local system.

Aim: To Create the file in our github globally and pull the file to our local system.

Procedure:

1. Locate to the project folder in your system.
2. Right click and open the terminal.
3. Open the browser and login into your github.
4. Open the project repository.
5. Now create the file you want in the github itself using create file option.
6. name the file and click on commit.
7. The file is created globally.
8. Now follow the commands to pull the created file into your local project folder.

Commands

1. git init
2. git branch -M main
3. git remote add origin <https://github.com/PavanSPK/RPA-Project.git>

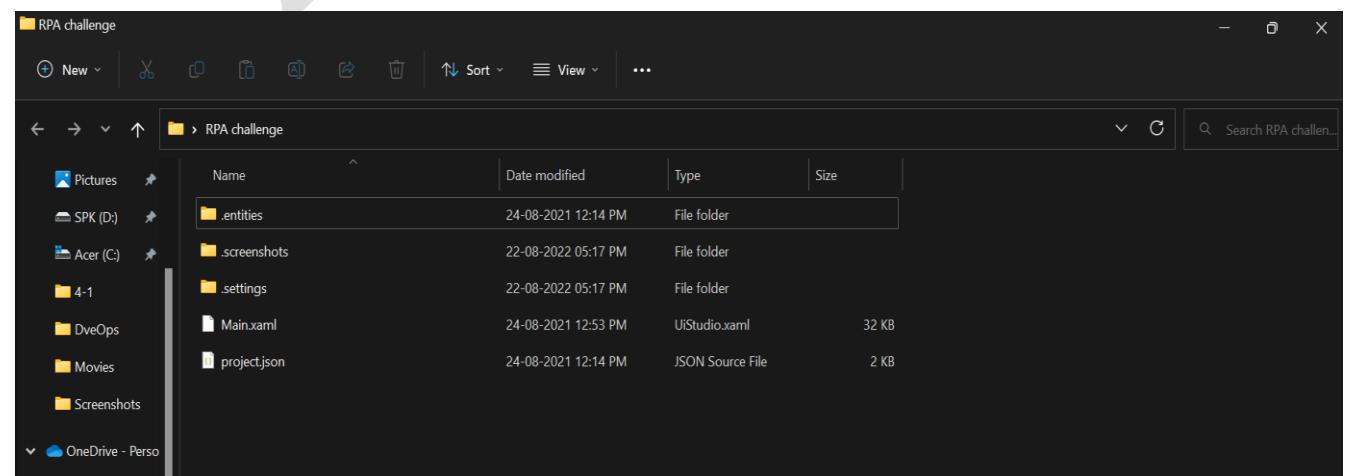
After the above command the terminal ask your to login into your git account using your credentials.

4. git pull

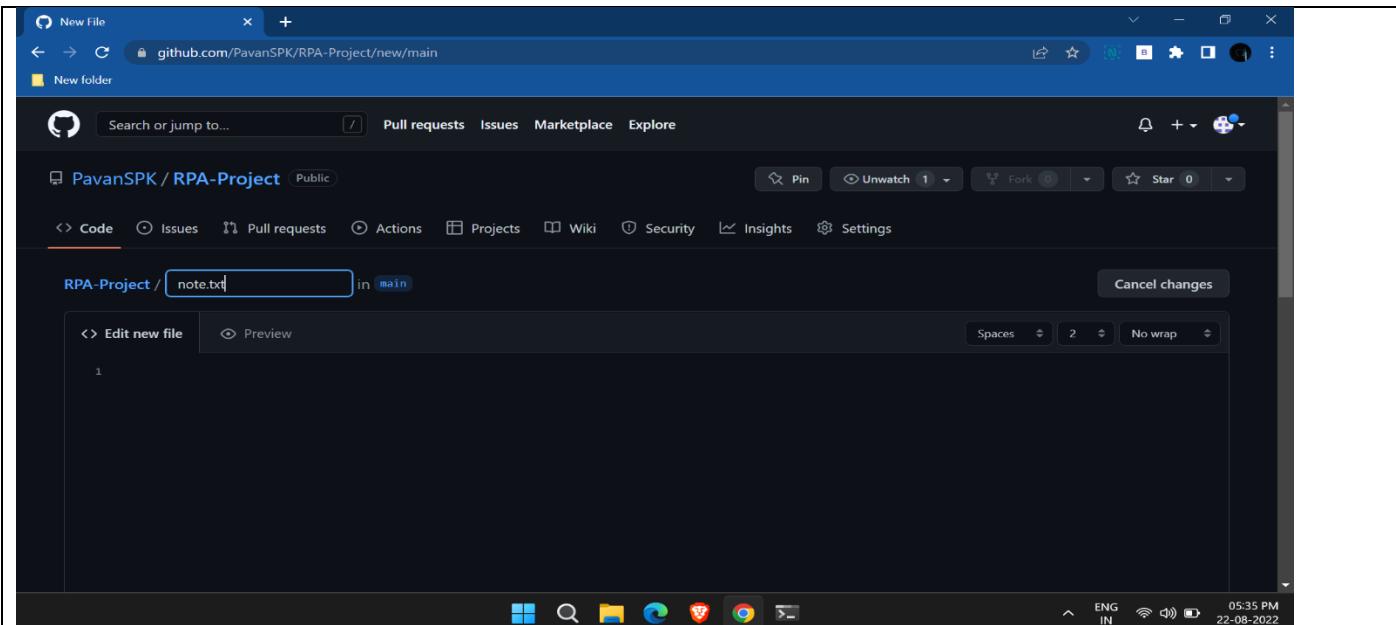
Now check your local project folder..

You can see the file which is created globally.

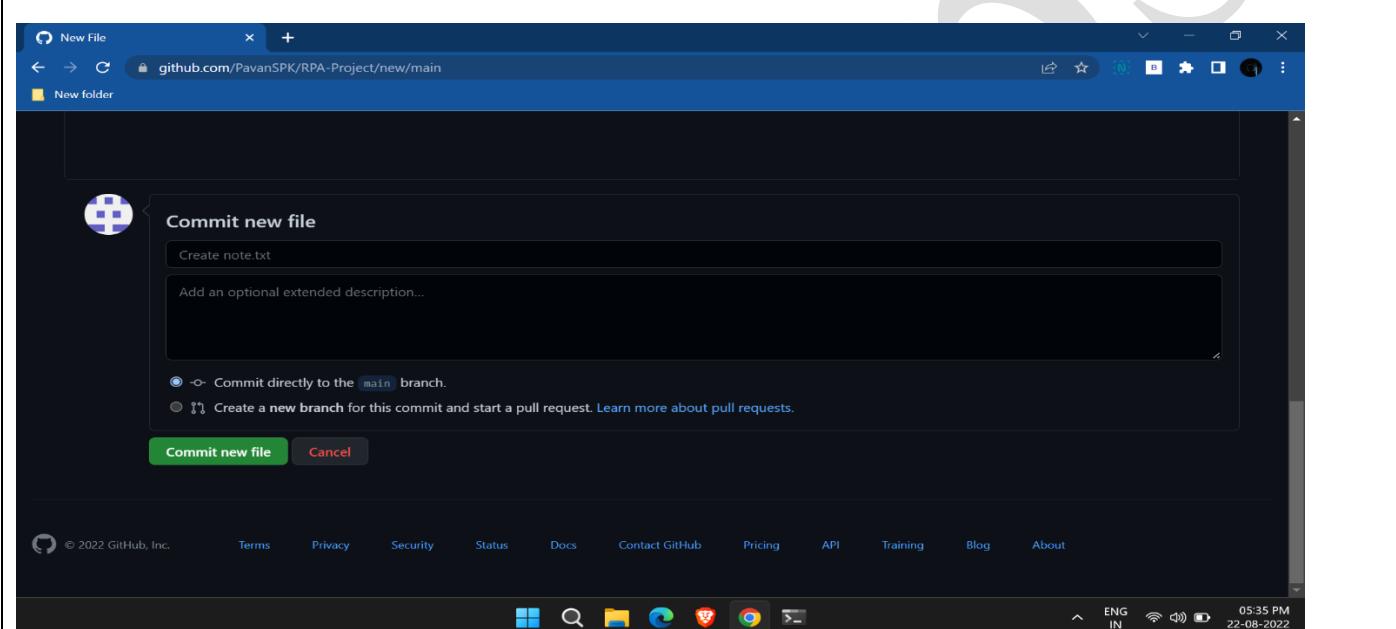
Output:



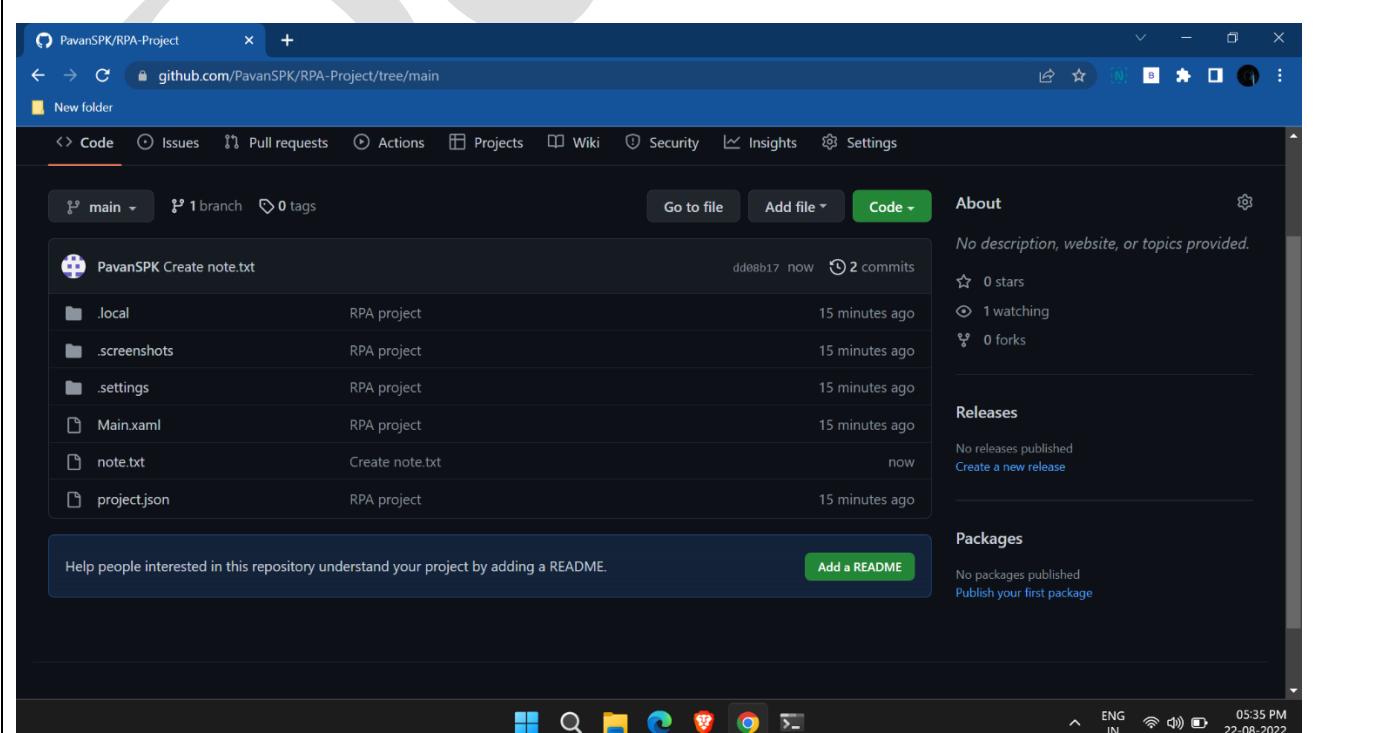
| | | | | | |
|------|--|----------|--|----------|--|
| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|



The screenshot shows a GitHub repository named 'PavanSPK / RPA-Project'. A new file, 'note.txt', is being created in the 'main' branch. The file content is currently empty, containing only a single space character. The GitHub interface includes standard navigation tabs like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The bottom status bar indicates the date and time as 22-08-2022 at 05:35 PM.

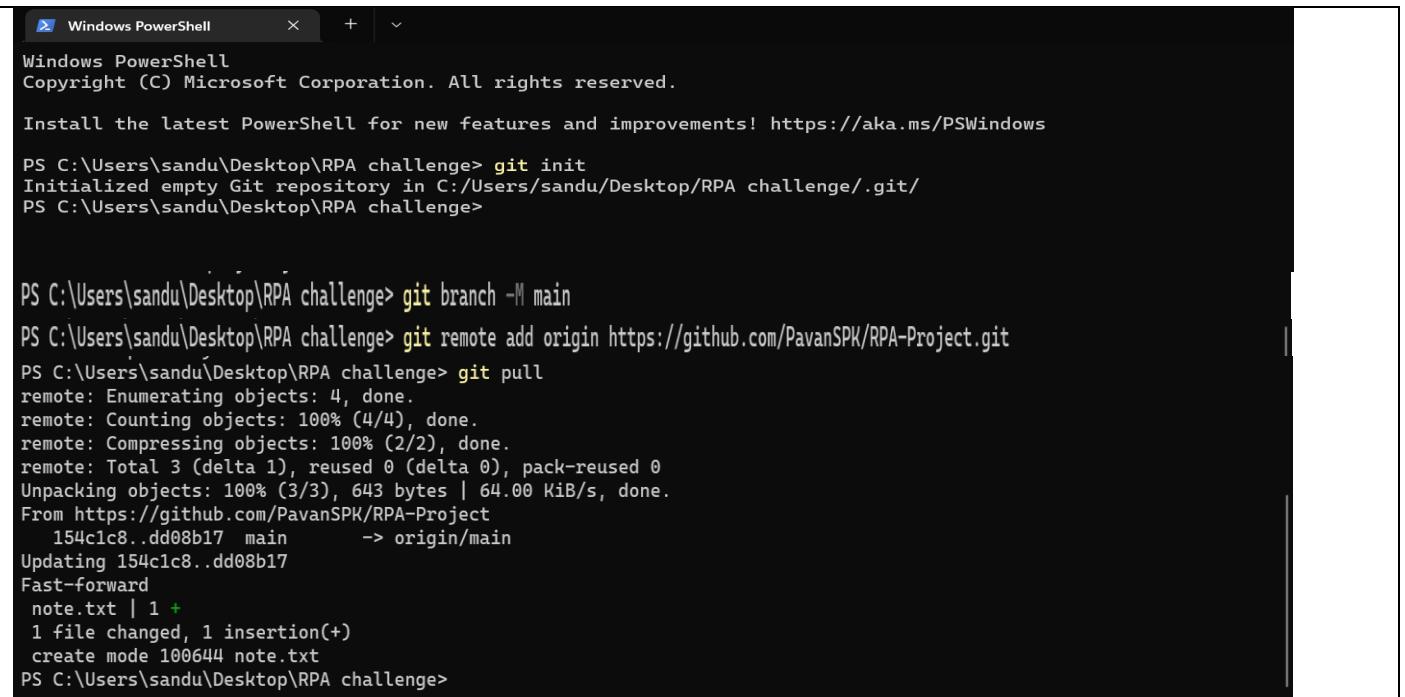


The screenshot shows the 'Commit new file' dialog for the 'note.txt' file. The file is titled 'Create note.txt'. There is a text area for an optional extended description. Two radio button options are available: 'Commit directly to the main branch.' (selected) and 'Create a new branch for this commit and start a pull request.' Below the dialog are 'Commit new file' and 'Cancel' buttons. The GitHub footer includes links for Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About. The bottom status bar shows the same date and time as the previous screenshot.



The screenshot shows the repository tree for 'PavanSPK/RPA-Project'. It contains one branch ('main') and no tags. The files listed are '.local', '.screenshots', 'settings', 'Main.xaml', 'note.txt', and 'project.json', all of which were created 15 minutes ago. On the right side of the repository page, there are sections for 'About' (no description, website, or topics provided), 'Releases' (no releases published), and 'Packages' (no packages published). A button to 'Add a README' is visible at the bottom left of the repository tree area. The bottom status bar shows the date and time as 22-08-2022 at 05:35 PM.

| | | | | |
|------|--|----------|--|----------|
| Date | | EXP .NO. | | Page No. |
|------|--|----------|--|----------|



```

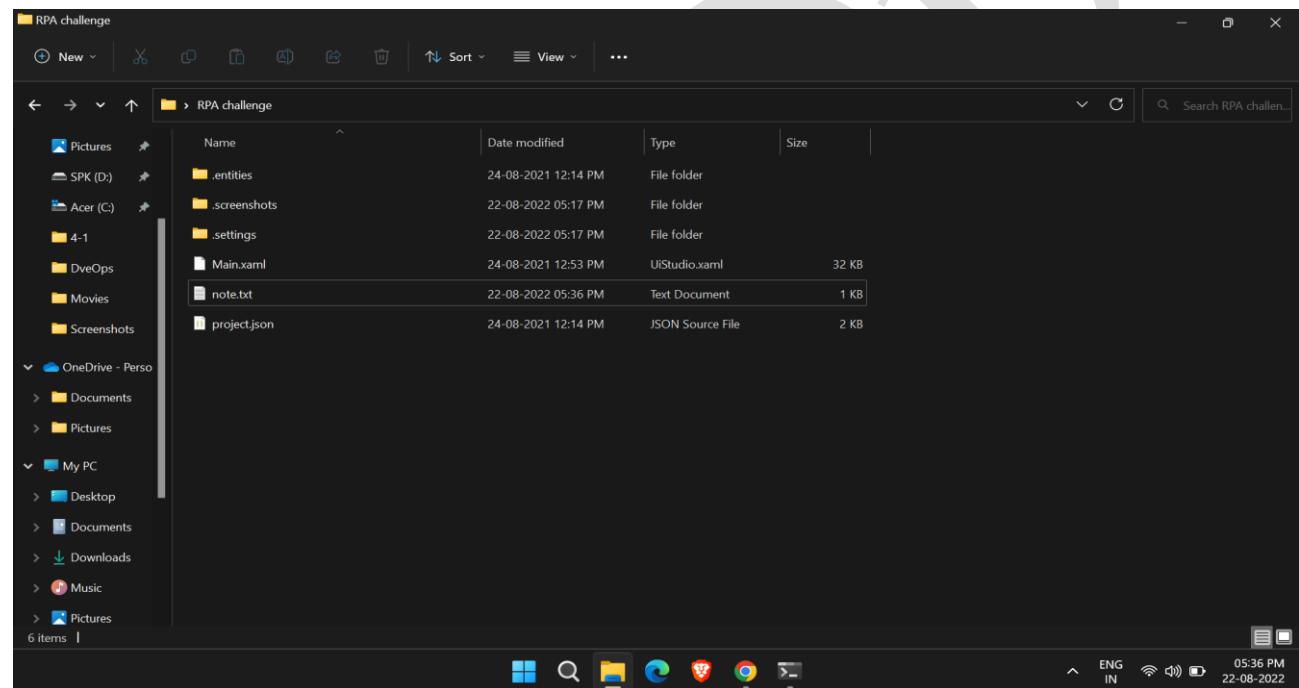
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sandu\Desktop\RPA challenge> git init
Initialized empty Git repository in C:/Users/sandu/Desktop/RPA challenge/.git/
PS C:\Users\sandu\Desktop\RPA challenge>

PS C:\Users\sandu\Desktop\RPA challenge> git branch -M main
PS C:\Users\sandu\Desktop\RPA challenge> git remote add origin https://github.com/PavanSPK/RPA-Project.git
PS C:\Users\sandu\Desktop\RPA challenge> git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 643 bytes | 64.00 KiB/s, done.
From https://github.com/PavanSPK/RPA-Project
  154c1c8..dd08b17  main      -> origin/main
Updating 154c1c8..dd08b17
Fast-forward
 note.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 note.txt
PS C:\Users\sandu\Desktop\RPA challenge>

```



Results:

Creating the file in our github globally and pulling the file to our local system was completed successfully.

Problem Statement: Download and install selenium and setup the environment using the pycharm.

Aim: To Download and install selenium and setup the environment using the pycharm.

Description:

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others.

Selenium WebDriver is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly. Selenium WebDriver allows you to choose a programming language to create test scripts.

Procedure:

1. Install Python in your system.
2. Install selenium.
3. Install web drivers.
4. Install pycharm.

Install selenium:

1. Open terminal (cmd)
2. type the following command
3. pip install selenium

Install web drivers:

Selenium requires a driver to interface with the chosen browser.

supported browsers will have their own drivers available. Links to some of the more popular browser drivers follow.

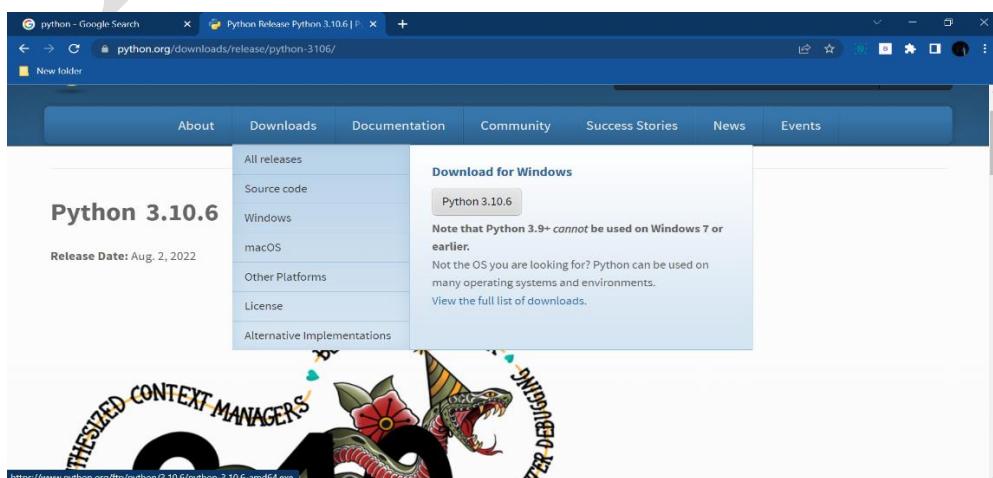
| | |
|-----------------|---|
| Chrome: | https://sites.google.com/chromium.org/driver/ |
| Edge: | https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/ |
| Firefox: | https://github.com/mozilla/geckodriver/releases |
| Safari: | https://webkit.org/blog/6900/webdriver-support-in-safari-10/ |

Install pycharm:

1. Download pycharm community application from [jetbrains.com/pycharm/download/windows](https://www.jetbrains.com/pycharm/download/windows)
2. Install the pycharm.exe and complete the installation process.
3. open the pycharm application setup the project.
4. In pycharm open terminal and run the command: pip install selenium
5. Now create project and perform the selenium tests.

Output:

Python install



| Date | | EXP .N0. | Page No. |
|------|--|----------|----------|
|------|--|----------|----------|

Install Selenium

```
Command Prompt
Microsoft Windows [Version 10.0.22610.1]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sandu>pip install selenium
Requirement already satisfied: selenium in c:\programdata\anaconda3\lib\site-packages (4.3.0)
Requirement already satisfied: trio<=0.17 in c:\programdata\anaconda3\lib\site-packages (from selenium) (0.21.0)
Requirement already satisfied: urllib3[secure,socks]<~1.26 in c:\programdata\anaconda3\lib\site-packages (from selenium) (2.1.1)
Requirement already satisfied: trio-websocket<~0.9 in c:\programdata\anaconda3\lib\site-packages (from selenium) (0.9.1)
Requirement already satisfied: sortedcontainers in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (2.1.0)
Requirement already satisfied: attrs>=19.2.0 in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (2.1.0)
Requirement already satisfied: async-generator>=1.9 in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.2.0)
Requirement already satisfied: idna in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (3.2)
Requirement already satisfied: outcome in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.2.1)
Requirement already satisfied: sniffio in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.2.1)
Requirement already satisfied: cffi>=1.14 in c:\programdata\anaconda3\lib\site-packages (from trio<=0.17->selenium) (1.1.3)
Requirement already satisfied: pycparser in c:\programdata\anaconda3\lib\site-packages (from cffi>=1.14->trio<=0.17->selenium) (2.20.0)
Requirement already satisfied: wsproto>=0.14 in c:\programdata\anaconda3\lib\site-packages (from trio-websocket<~0.9->selenium) (0.14.0)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]<~1.26->selenium) (1.7.1)
Requirement already satisfied: pyOpenSSL>=0.14 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]<~1.26->selenium) (0.14.0)
Requirement already satisfied: cryptography>=1.3.4 in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]<~1.26->selenium) (1.3.4)
Requirement already satisfied: certifi in c:\programdata\anaconda3\lib\site-packages (from urllib3[secure,socks]<~1.26->selenium) (2022.01.18)
Requirement already satisfied: six>=1.5.2 in c:\users\sandu\appdata\roaming\python\python39\site-packages (from pyOpenSSL>=0.14->selenium) (1.15.0)
Requirement already satisfied: h11<1,>=0.9.0 in c:\programdata\anaconda3\lib\site-packages (from wsproto>=0.14->trio-websocket<~0.9->selenium) (0.11.0)

C:\Users\sandu>
```

Install Webdrivers

The screenshot shows a web browser window with the URL chromedriver.storage.googleapis.com/index.html?path=105.0.5195.19/. The page displays a list of files under the '105.0.5195.19' directory, including various platform-specific zip files and a notes.txt file.

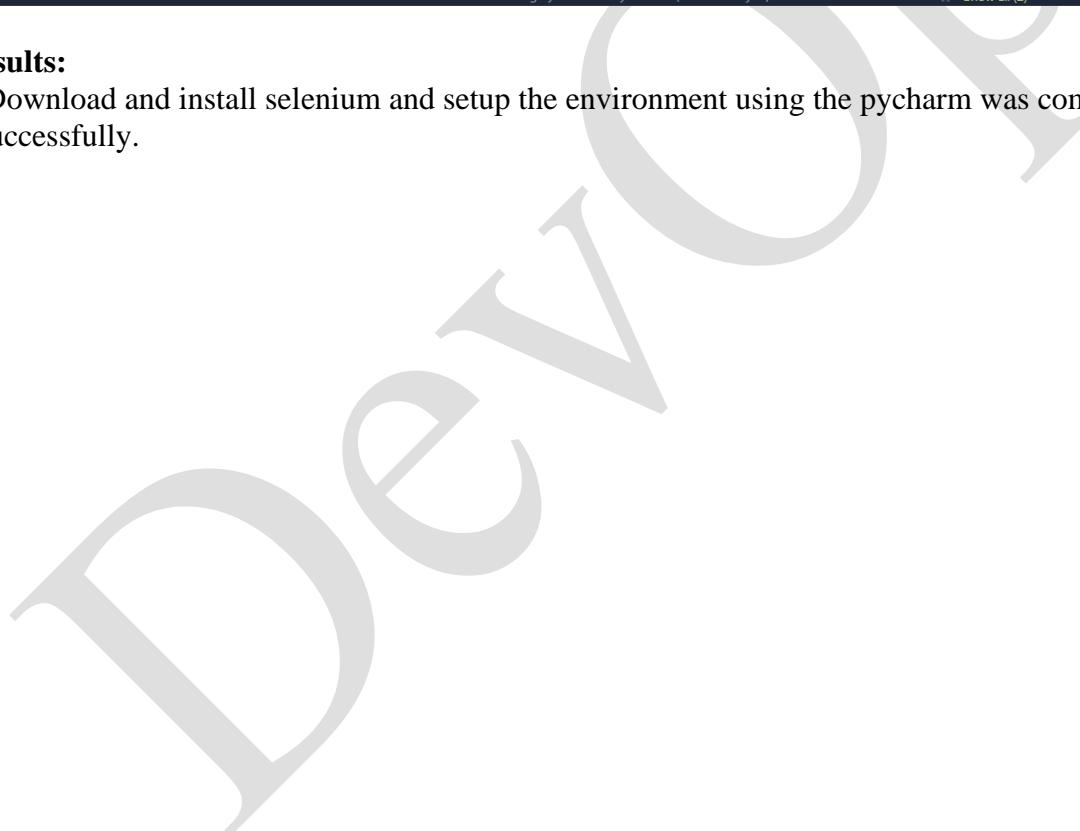
Index of /105.0.5195.19/

| Name | Last modified | Size | ETag |
|---|---------------------|--------|----------------------------------|
| Parent Directory | | - | |
| chromedriver_linux64.zip | 2022-08-08 09:16:09 | 6.15MB | b3944a67d3dd6f9369cec74c1cc69c41 |
| chromedriver_mac64.zip | 2022-08-08 09:16:11 | 8.07MB | 43d4460f63ef5bc8a5e711aff5ca00e3 |
| chromedriver_mac64_m1.zip | 2022-08-08 09:16:14 | 7.46MB | 70f5243170812fe88f53e4502e03fb10 |
| chromedriver_win32.zip | 2022-08-08 09:16:16 | 6.24MB | eff85de8df53272a1e18ab65fdcbebf5 |
| notes.txt | 2022-08-08 09:16:22 | 0.00MB | 5c39d0751819d261ae8b2072976494dd |

Install Pycharm

The screenshot shows a web browser window with the URL jetbrains.com/pycharm/download/#section=windows. The page is titled 'Download PyCharm' and offers two main download options: 'Professional' and 'Community'. Both versions are available for Windows, macOS, and Linux. The 'Professional' version is described as being for both Scientific and Web Python development, while the 'Community' version is for pure Python development. Both versions include a 30-day trial period.

| Date | | EXP .NO. | | Page No. | |
|------|--|----------|--|----------|--|
|------|--|----------|--|----------|--|



The screenshot shows the PyCharm IDE interface. The title bar says "seleniumProject – main.py". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The toolbar has icons for file operations like Open, Save, and Run. The Project tool window on the left shows a "seleniumProject" folder containing "venv", "library root", "Lib", "Scripts", ".gitignore", and "pyvenv.cfg". A "main.py" file is selected. The code editor window displays the following Python script:

```

1  # This is a sample Python script.
2
3  # Press Shift+F10 to execute it or replace it with your code.
4  # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7  def print_hi(name):
8      # Use a breakpoint in the code line below to debug your script.
9      print(f'Hi, {name}')  # Press Ctrl+F8 to toggle the breakpoint.
10
11
12  # Press the green button in the gutter to run the script.
13  if __name__ == '__main__':
14      print_hi('PyCharm')
15
16  # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

The status bar at the bottom shows "Indexing Python SDK 'Python 3.9 (seleniumProject)'".

Results:

Download and install selenium and setup the environment using the pycharm was completed successfully.

| | | | | | |
|------|--|----------|--|----------|--|
| Date | | EXP .N0. | | Page No. | |
|------|--|----------|--|----------|--|

Problem Statement: Automate a web browser to capture title of the page, URL of the page using selenium.

Aim: To Automate a web browser

1.to capture title of the page

2.to get URL of the page using selenium.

Procedure:

1.open pycharm application

2.Import the required libraries.

3.write the code for the problem statement.

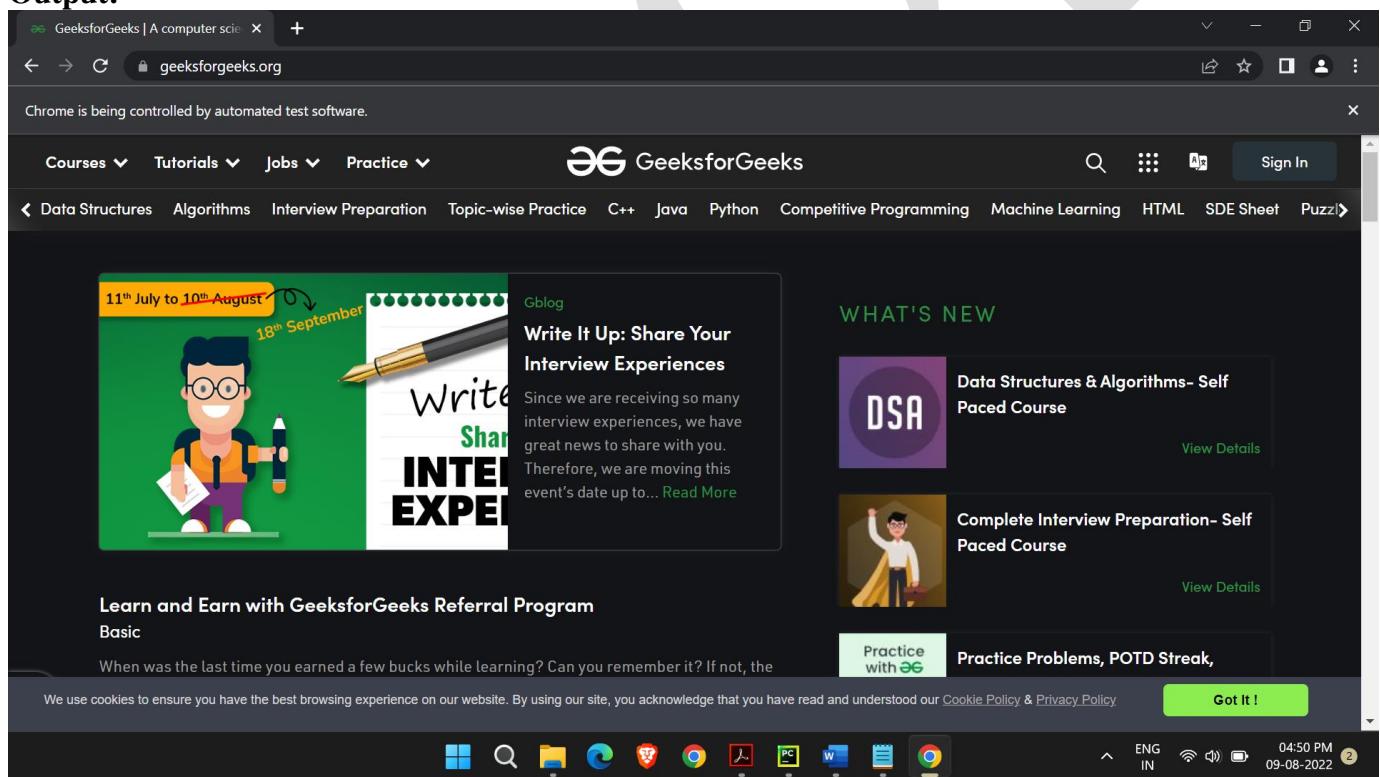
4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common import keys
import time
```

```
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://www.geeksforgeeks.org/")
driver.maximize_window()
print(driver.title)
print(driver.current_url)
time.sleep(3)
driver.close()
```

Output:



GeeksforGeeks | A computer science portal for geeks

<https://www.geeksforgeeks.org/>

Results:

Automation of a web browser to capture title of the page, URL of the page using selenium was completed successfully.

Problem Statement: Automate a web browser to test for Title matching with selenium tool.

Aim: To Automate a web browser to test for Title matching with selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import time

serv_obj = Service("C:\Drivers\chromedriver_win32\chromedriver.exe")
driver = webdriver.Chrome(service=serv_obj)
driver.get("https://jntuacep.ac.in/")
driver.maximize_window()
actualTitle = driver.title
expectedTitle="JNTUA College of Engineering Pulivendula"
time.sleep(3)

if (actualTitle == expectedTitle):
    print("title matched")
else:
    print("title not matched")
```

driver.quit()

Output:

Chrome is being controlled by automated test software.

JNTUA College of Engineering (Autonomous) Pulivendula योग्य कर्मसु कीशलम्

RTI Anti Ragging Contact Us

Anti-Ragging Toll Free No: 1800 180 5522
EAMCET Code: JNTP
NEW NIRF-2022 NEW NIRF-2021

Home About Administration Academics Departments Facilities Training & Placements Examination Portal Results Online Classes Tenders More

Latest News

- II Semester (R15).
- IV B.Tech II Semester (R15) Regular and Supplementary Examinations, May 2022.
- Time Tables for II B.Tech (R20) I-Sem Supplementary Examinations.
- Time Tables for I.M.Tech (R21&R17) I-Sem Regular & II-Sem Supplementary Examinations.
- Time Tables for I, II, III, IV B.Tech (R19, R15 & R13) I & II Sem Supplementary Examinations.
- BTech I Year I Semester (R20) Regular & Supplementary Examination April 2022 Results
- GOOGLE FORM LINK FOR OPTING THE PROFESSIONAL AND OPEN ELECTIVE SUBJECTS FOR IV.B.TECH I SEMESTER

05:01 PM 09-08-2022

title matched

Results:

Automation of a web browser to test for Title matching with selenium tool was completed successfully.

Problem Statement: Automate a web browser for filling the web forms with selenium tool.

Aim: To Automate a web browser for filling the web forms with selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")

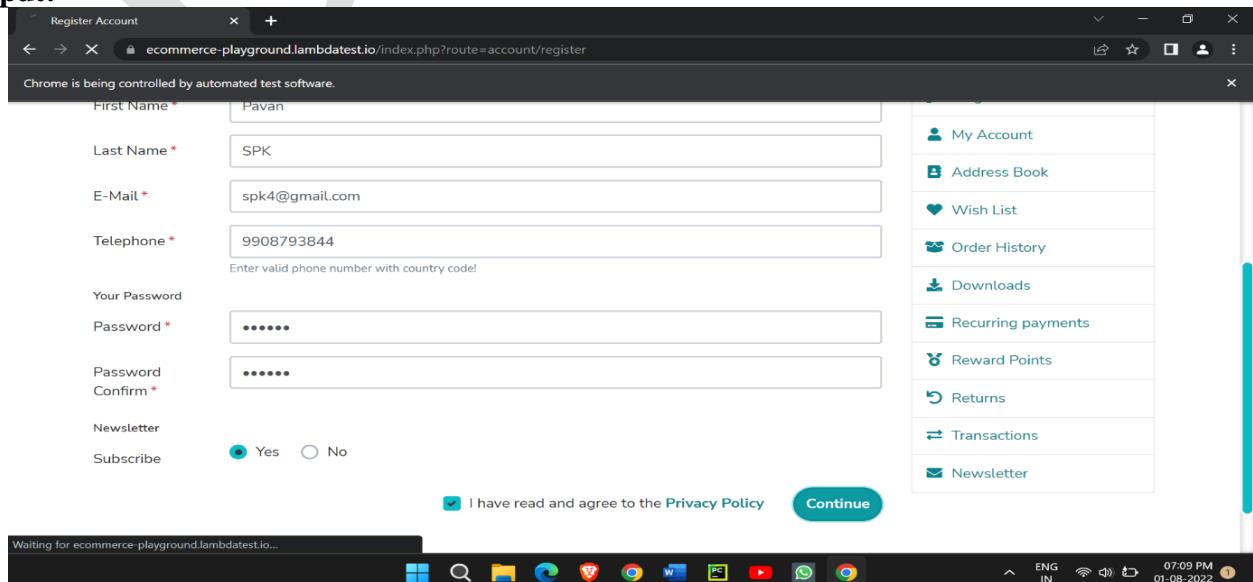
driver.get("https://ecommerce-playground.lambdatest.io/index.php?route=account/register")

inputboxes=driver.find_element(By.CLASS_NAME,'form-control')
driver.find_element(By.ID,'input-firstname').send_keys("Pavan")
driver.find_element(By.ID,'input-lastname').send_keys("SPK")
driver.find_element(By.ID,'input-email').send_keys("spk4@gmail.com")
driver.find_element(By.ID,'input-telephone').send_keys("9908793844")
driver.find_element(By.ID,'input-password').send_keys("123321")
driver.find_element(By.ID,'input-confirm').send_keys("123321")
newsletter = driver.find_element(By.XPATH, value="//label[@for='input-newsletter-yes']")
newsletter.click()
terms = driver.find_element(By.XPATH, value="//label[@for='input-agree']")
terms.click()

continue_button = driver.find_element(By.XPATH, value="//input[@value='Continue']")
continue_button.click()
continue_button = driver.find_element(By.XPATH, value="//input[@value='Continue']")
continue_button.click()

assert browser.title == "Your Account Has Been Created!"
time.sleep(10)
driver.close()
```

Output:



Your Account Has Been Created! X

ecommerce-playground.lambdatest.io/index.php?route=account/success

Chrome is being controlled by automated test software.

LAMBDATEST Playground All Categories Search For Products SEARCH

Shop by Category Home Special Hot Blog Mega Menu AddOns Featured My account

This is a dummy website for Web Automation Testing

Account Success

Your Account Has Been Created!

Congratulations! Your new account has been successfully created!

You can now take advantage of member privileges to enhance your online shopping experience with us.

If you have ANY questions about the operation of this online shop, please e-mail the store owner.

A confirmation has been sent to the provided e-mail address. If you have not received it within the hour, please contact us.

[Continue](#)

Waiting for ecommerce-playground.lambdatest.io...

My Account Edit Account Password Address Book Wish List Notification Order History

ENG IN 07:08 PM 01-08-2022

Your Account Has Been Created!

Results:

Automation of a web browser for filling the web forms with selenium tool was completed successfully.

Problem Statement: Automate a web browser to upload a file in web forms using selenium tool.

Aim: To Automate a web browser to upload a file in web forms using selenium tool.

Procedure:

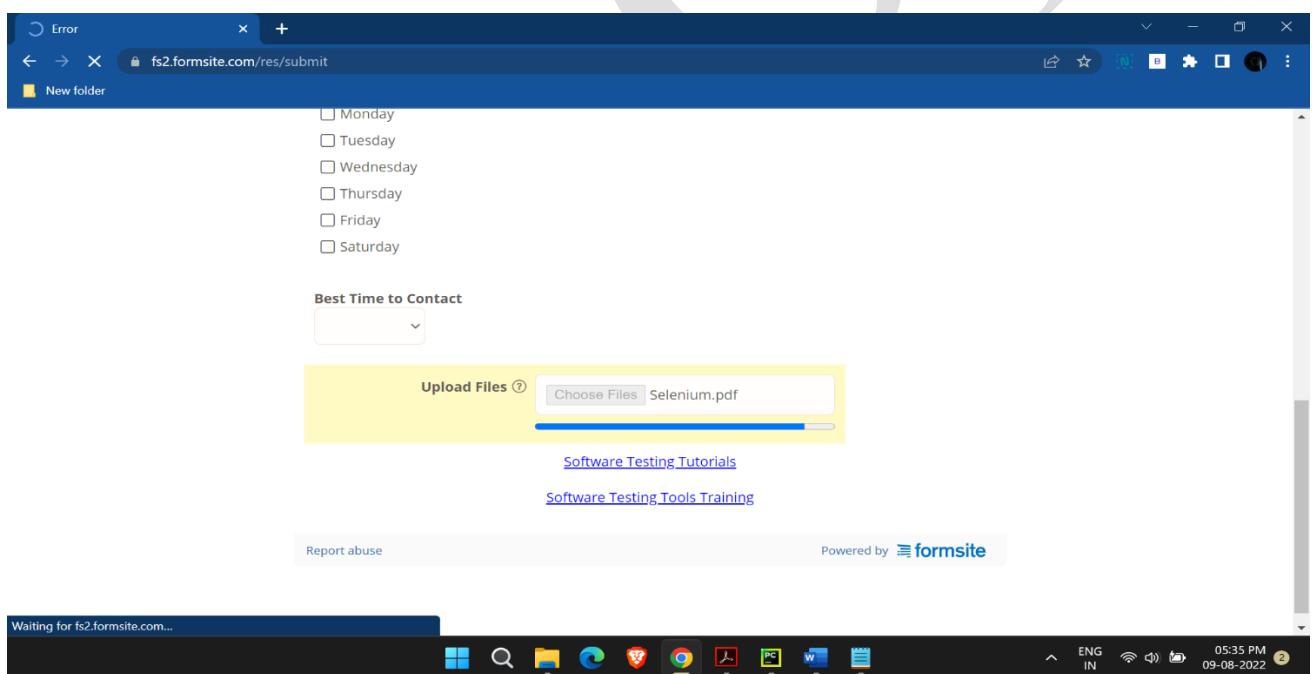
- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()
driver.find_element(By.ID,"RESULT_FileUpload-10").send_keys("C://Users/sandu/Pictures/Selenium.pdf")
time.sleep(5)
driver.close()
```

Output:



Results:

Automation of web browser to upload a file in web forms using selenium tool was completed successfully.

Problem Statement: Automate a web browser for working on a drop-down list in web forms using selenium tool.

Aim: To Automate a web browser for working on a drop-down list in web forms using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
import time

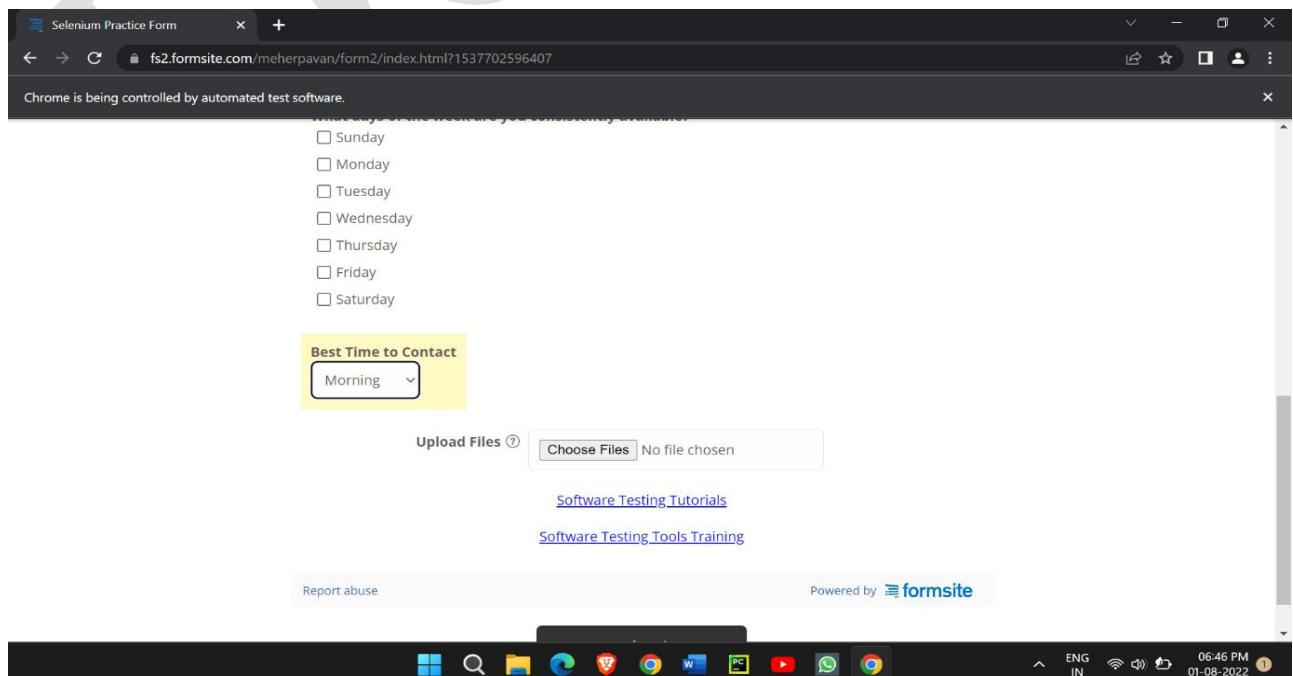
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()

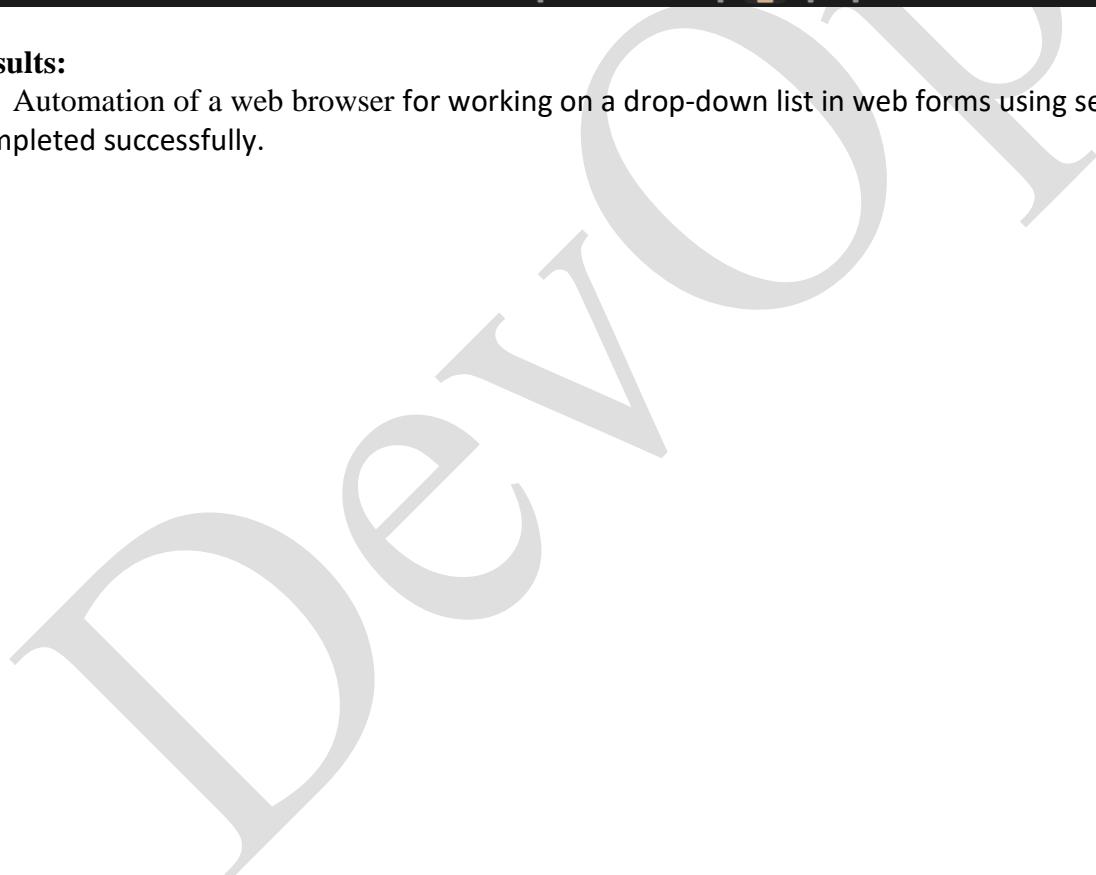
element=driver.find_element(By.NAME,"RESULT_RadioButton-9")
drp=Select(element)
print(len(drp.options))

all_options=drp.options
for option in all_options:
    print(option.text)

drp.select_by_value("Radio-0")
time.sleep(5)
print("Morning selected")
driver.close()
```

Output:





Selenium – D:\Programs\selenium\venv\dropdown.py

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
selenium > venv > dropdown.py
Project: selenium D:\Programs\se
  > venv library root
    main.py
  > External Libraries
  > Scratches and Consoles
11 drp=Select(element)
12 print(len(drp.options))
13
14 all_options=drp.options
15 for option in all_options:
16     print(option.text)
17
18 drp.select_by_value("Radio-0")
for option in all_options

Run: dropdown (1) ×
  D:\Programs\Selenium\venv\Scripts\python.exe D:/Programs/selenium/venv/dropdown.py
D:\Programs\selenium\venv\dropdown.py:6: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
4
Morning
Afternoon
Evening
Morning selected

Process finished with exit code 0

```

Version Control Run Python Packages TODO Python Console Problems Terminal Services

15:27 CRLF UTF-8 4 spaces Python 3.9 (Selenium) ENG IN 06:46 PM 01-08-2022

Results:

Automation of a web browser for working on a drop-down list in web forms using selenium tool was completed successfully.

Problem Statement: Automate a web browser for working on a Radio and check boxes in formfilling using selenium tool.

Aim: To Automate a web browser for working on a Radio and check boxes in formfilling using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import time

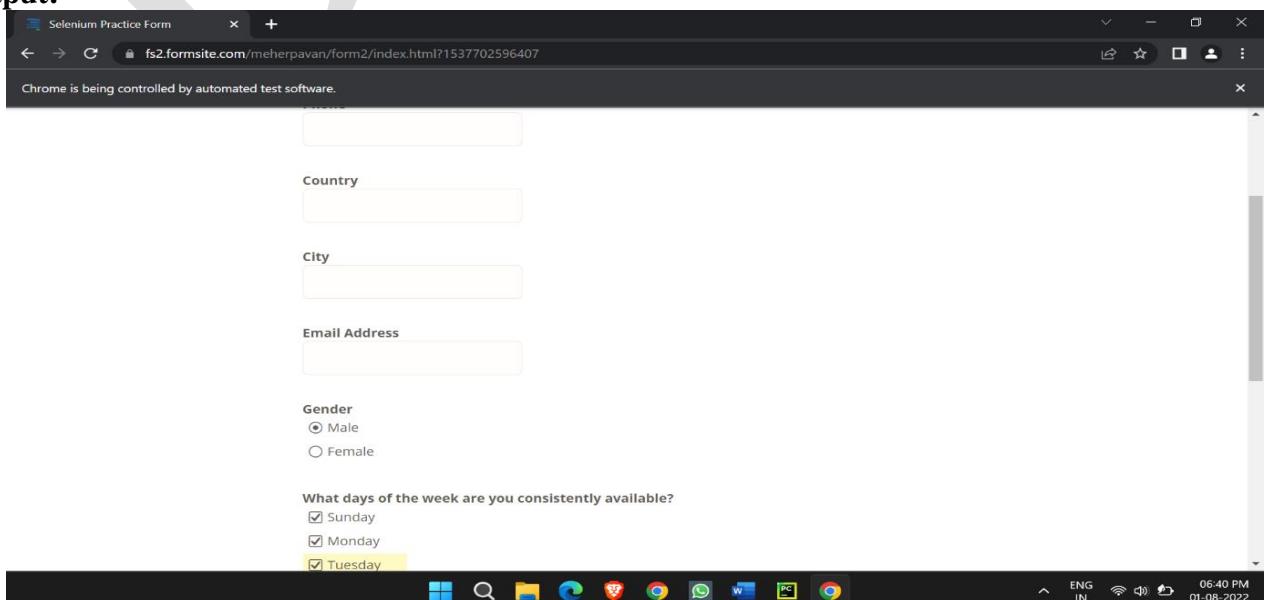
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
driver.get("https://fs2.formsite.com/meherpavan/form2/index.html?1537702596407")
driver.maximize_window()

gender=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']")
gender.click()

status=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']").is_selected()
print(status)

driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_0']").click()
#sunday
status=driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_1']").click()
#monday
driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_2']").click()
#tuesday
print(status)
time.sleep(5)
driver.quit()
```

Output:



| | | | | | |
|------|--|---------|--|----------|--|
| Date | | EXP.N0. | | Page No. | |
|------|--|---------|--|----------|--|

Selenium - D:\Programs\selenium\venv\radio_checkboxes.py

File Edit View Navigate Code Refactor Run Tools VCS Window Help

selenium venv radio_checkboxes.py

Project

selelium D:\Programs\selelium\venv\radio_checkboxes.py

main.py firsttest.py radio_checkboxes.py dropdown.py links.py textbox.py upload_files.py unittestframework.py

Reader Mode

status=driver.find_element(By.ID,'RESULT_RadioButton-7_0').is_selected()
print(status)"'"

gender=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']").click()

status=driver.find_element(By.XPATH, value="//label[@for='RESULT_RadioButton-7_0']").is_selected()
print(status)

driver.find_element(By.XPATH, value="//label[@for='RESULT_CheckBox-8_0']").click() #uncheck

Run: radio_checkboxes (1) ×

D:\Programs\Selenium\venv\Scripts\python.exe D:/Programs/selenium/venv/radio_checkboxes.py
D:\Programs\selenium\venv\radio_checkboxes.py:5: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
False
None
Process finished with exit code 0

Bookmarks

Structure

Version Control Run Python Packages TODO Python Console Problems Terminal Services

8:1 CRLF UTF-8 4 spaces Python 3.9 (Selenium) ENG IN 06:41 PM 01-08-2022

Results:

Automation of a web browser for working on a Radio and check boxes in formfilling using selenium tool.

| Date | | EXP .N0. | Page No. | |
|------|--|----------|----------|--|
|------|--|----------|----------|--|

Problem Statement: Automate a web browser for unit-test framework on a simple project using selenium tool.

Aim: To Automate a web browser for unit-test framework on a simple project using selenium tool.

Procedure:

- 1.open pycharm application
- 2.Import the required libraries.
- 3.write the code for the problem statement.
- 4.test the code.

Program code:

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
chrome_executable = Service(executable_path='C:\Drivers\chromedriver_win32\chromedriver.exe',
log_path='NULL')
from selenium.webdriver.common.by import By
import unittest
import time

class SearchEnginesTest(unittest.TestCase):
    @classmethod
    def testGoogle(self):

        self.driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
        self.driver.get("https://www.google.com/")
        print("Title of the page is:",self.driver.title)
        self.driver.close()

    @classmethod
    def test_Bing(self):

        self.driver=webdriver.Chrome(executable_path="C:\Drivers\chromedriver_win32\chromedriver.exe")
        self.driver.get("https://bing.com")
        print("Title of the page is:", self.driver.title)
        self.driver.close()

if __name__ == "__main__":
    unittest.main()
driver.quit()
```

| Date | | EXP .NO. | | Page No. |
|------|--|----------|--|----------|
|------|--|----------|--|----------|

Output:

The screenshot shows a Windows desktop environment with three main windows:

- Top Window (Google Chrome):** Displays the Google homepage in English. A tooltip indicates "Chrome is being controlled by automated test software." A "Sign in to Google" dialog box is overlaid on the page. Below the search bar, it says "Google offered in: हिन्दी वर्गा लेख मराठी तमिळ गुजराती कन्नಡ मलयാളം ఫోన్మీ".
- Middle Window (Microsoft Edge):** Displays the Microsoft Bing homepage in English. It shows trending topics like "cuet admit card 2022", "Laal Singh Chaddha 2022 Film", etc. A tooltip indicates "We're updating our Terms of Use. Learn more".
- Bottom Window (Code Editor):** Shows Python code for a Selenium test framework. The code includes imports for `unittest` and `time`, and defines a class `SearchEnginesTest` with methods for Google and Bing searches. The code uses `chromedriver.exe` as the executable path for the WebDriver.

Title of page is: Google

Title of page is: Bing

Closes the browser

Ran 2 tests in 14.065s

Status: OK

All Test classes finished

| Date | | EXP .N0. | Page No. | |
|------|--|----------|----------|--|
|------|--|----------|----------|--|

Results:

Automate a web browser for unit-test framework on a simple project using selenium tool.

DevOps