A

REPORT

OF

# “Open Source Software laboratory” Code: 5IT452

## BIE: Beginner (B)

Submitted by

**MR. Pratik Umesh Pawar (2019BTEIT00006)**

**Email:** pawarpratik80009@gmail.com

**Mobile:** 9022871939

DEPARTMENT OF INFORMATION TECHNOLOGY

**WALCHAND COLLEGE OF ENGINEERING,**

**SANGLI**

**(An Autonomous Institute) 2022-2023**

Mr.Pawar P.U. and Dr. A. J. Umbarkar.

## CERTIFICATE



This is to certify that the report entitled “*Open Source Software laboratory (OSS Lab) 5IT452*” submitted by *MR. Pratik Umesh Pawar(2019BTEIT00006)* is a record of student's own work carried out by them during the academic year 2022-2023, as per the curriculum/syllabus laid down for OSS Lab at Final Year B.Tech IT Sem-I. They have carried out experiments/FOSS tools hands-on on *18* assignments under BIE: Beginner (B) category successfully.

Dr. A. J. Umbarkar

(Course Teacher)

### Declaration

We, the undersigned, hereby declare that the B.TECH Report entitled **“***Open Source*

*Software laboratory (OSS Lab) (5IT452)* **“**submitted by us to OSS Lab report at Final Year B.Tech IT Sem-I, is our original/experimented/experience work. We further declare that, to the best of our knowledge and belief, this report has not been previously submitted or copied by us.

We declare that this report reflects my thoughts about the subject in our own words. We have sufficiently cited and referenced the original sources, referred or considered in this work. We have not misrepresented, fabricated, or falsified any idea/data/fact/source in this our submission. We understand that any violation of the above will be cause for disciplinary action by the Course Teacher/Institute.

#### (Sign)

Date: **Mr.Pratik Pawar**

Place:

### Acknowledgement

We feel immense pleasure in submitting the report entitled “*Open Source Software laboratory (OSS Lab) (5IT452)***"**. We are thankful to our guide **Dr. A. J. Umbarkar** for their valuable guidance and kind help during implementing the OSS Lab.

Acknowledged By,

Sign

**Mr. Pratik Pawar**

#### 5IT452: Open Source Software Lab (B: Beginner) Final Assignment List

**1. Demonstrationof Linux Distributions OS’s and their purpose with comparisons.**

**(Fedora/CentOS/any other/etc.: Any One)**

**(Submission by Individual [I])**

***Objective:*** *To install and demonstrate Various Linux Distributions and their Purpose/comparison/differences.*

***Outcome:*** *Self learning/lifelong learning (PO: b, k, l)*

Student asks to study at least two Linux Distros, with their comparisons and installation on Virtual Box OR Installation Linux on Live USB pen drive. [https://fedoraproject.org/wiki/How\_to\_create\_and\_use\_Live\_USB]

In Journal, they have to write information of that distros, such as:-

i. Various versions of that distros with code name ii. Default desktop GUI iii. Main purpose of that iv. Package management of that distros v. List of Default Packages vi. Screenshots of that distros vii. Compare '/etc' hierarchy

viii. Compare package managers ix. Pros/cons of both distros

x. Which one is better for development and why? xi. Which one is easy to use (for beginner) and why? xii. Explorer any top 10 commands of that distro on command prompt.

xiii. Make the Official Repositories of **Fedora/CentOS** on docker store

(https://hub.docker.com/) and experiment for above.

**Reference:-**

i. List of Linux Distros:- http://distrowatch.com/ ii. For installation on Virtual Box:- https://help.ubuntu.com/community/ListOfOpenSourcePrograms

iii. http://www.psychocats.net/ubuntu/virtualbox iv. https://help.ubuntu.com/

**2. Use of Open Source tools for Project Management.**

 **(Sonar, Targetprocess, Redbooth, Pivotal Tracker, OrangeScrum etc.: Any**

**One)**

**(Submission by Individual [I])**

***Objective:*** *To install and demonstrate the use of various open source software’s that used in day to day life of software Engg.*

***Outcome:*** *(PO: k, l)*

Students have to experiment at last two project management tools / software’s and they have to use for their project/FOSS project/mini project.

In journal, they have to write information about that tool such as:-

1. Purpose behind that tool.
2. Various versions of those tools.
3. Installation and Configuration of that tool.
4. How to use that tool.

Reference:-

1. https://bitnami.com/stacks
2. Sphinx for all lab/college documentation by the students. iii. http://www.sonarqube.org/ iv. Wikipedia List Of Software’s:- http://en.wikipedia.org/wiki/List\_of\_free\_and\_open- source\_software\_packages
3. Open Source Software Mega List :- <http://www.datamation.com/open-source/open-> source-software-the-megalist.html
4. https://fedoraproject.org/wiki/Education\_Spin This has lots of relevant packages
5. <http://www.methodsandtools.com/tools/targetprocess.php>
6. <https://blog.capterra.com/free-open-source-project-management-software/>
7. <http://www.targetprocess.com/>
8. <https://www.pivotaltracker.com/features/>
9. https://redbooth.com/features

**3. Use of Bug Tracking**

**(Phabricator, Youtrack, Mantis, Futuramo, etc.: Any One )**

**(Submission by Individual [I])**

Tools:(Trac, Redmine, Phabricator, Youtrack, Mantis, Bugzilla, Donedone (new) etc. : Any One ) Students have to experiment at last 1 tools / software’s that they uses and their day to day life/industry , with their installation, multiuser configuration on their system and compariosn all other tools (free/paid).

In journal, they have to write information about that tool such as:-

1. Purpose behind that tool/Features
2. Various versions of that tools
3. Installation and Configuration of that tool
4. How to use that tool.
5. Add sample or choosen FOSS project in tool and add three bugs and allot to users of the tool.
6. Compare

Reference:-

1. <https://bitnami.com/stacks> (<https://bitnami.com/stack/trac>)
2. Wikipedia List Of Software’s:- [http://en.wikipedia.org/wiki/List\_of\_free\_and\_open -](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages)  [source\_software\_package](http://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages) s
3. Open Source Software Mega List :-  [http://www.datamation.com/open-source/open -](http://www.datamation.com/open-source/open-source-software-the-mega-list-1.html)  [source- software-the-mega-list-1.htm](http://www.datamation.com/open-source/open-source-software-the-mega-list-1.html) l
4. [donedone.com](https://bitnami.com/stacks)
5. **Use of Version Control System.**

***Objective****: To use/experiment the online and offline Version Control System for your project (foss project) work.*

***Outcome****: lifelong learning (PO: b, c, k, l)*

*Recommded : Git (offline)and github (online). Experiment in indivial and*

Tools: (GitLab, Subversion (svn), gitorious, Gogs, Microsoft Visual Source Safe, Mercurial (hg), Ba Students have to study experiment any one vcs with example on both

Windows and Linux Platform.

In Journal, They have to write Basic Information about GIT, Perforce Helix, SVN and their working with Commands

Reference:-

* + 1. https://try.github.io/levels/1/challenges/1
    2. <https://github.com/princeton-8/princeton-8.github.io>
    3. http://wiki.openhatch.org/Open\_Source\_Comes\_to\_Campus/Practicing\_Git/Students
    4. GIT Official Documentation:- [http://git-scm.com/documentatio](http://git-scm.com/documentation)n
    5. SVN Official Documentation:-<http://svnbook.red-bean.com/en/1.7/index.html>
    6. Perforce Helix is a commercial, proprietary revision control system developed by Perforce

Software vii.  [https://www.smashingmagazine.com/2008/09/the-top-7-open-source-version-control- systems/](https://www.smashingmagazine.com/2008/09/the-top-7-open-source-version-control-systems/)

* 1. http://wiki.bazaar.canonical.com/WindowsDownloads
  2. <https://gitexplorer.com/> (good for online practice )

1. **Installation and Use of CMS software’s.**

***Objective:*** *To comprehend the use of Content Management System and their Use for personal website/dept CMS.*

***Outcome:*** *Self learning (PO: b, I, j, k, l)*

Recommaded : Wordpress under B

Tools: Word-press. Moodle, Drupal, Joomla, MediaWiki, etc

Students have to study at least experiment one **CMS** and one **Wiki**. Use **Wiki** for giving the information to class student to perform FOSS assignments. Use **CMS** for giving the creating your personal website/blog or FOSS course website/blog.

In Journal, They have to write,

i. Installation on Linux Platform. ii. Administration of CMS/wiki iii. How to Use. iv. Screenshots.

v. Host the CMS on websever/free webspace.

References:-

* 1. Drupal Tutorials:-  [http://drupal.org/documentation/customization/tutorial](http://drupal.org/documentation/customization/tutorials) s
  2. Moodle Tutorials:- <http://docs.moodle.org/22/en/Moodle_video_tutorials>
  3. [www.wordpress.com](http://www.wordpress.com/) iv. https://bitnami.com

1. **Comprehend the Open Source Software Development for any one Linux distro.**

**(Topic 3rd in Syllabus) (Fedora/CentOS, etc :Any One) (Submission by Individual or Group [I or G])**

***Objective:*** *To comprehend the open source software development.*

***Outcome:*** *Self learning/lifelong learning (PO: b, k, l)*

In this student have to study open source software development process of **any one above Linux distro.**

Get the details following information like –Name of community, website, Mailing List, wiki, version control, bug tracking and documentation of the particular distro to comprehend.

Sample of Ubuntu Development:-

1. Development Communities:- http://www.ubuntu.com/community/
2. Mailing List:- https://lists.ubuntu.com/
3. IRC channels:- https://wiki.ubuntu.com/IRC/ChannelList iv. Ubuntu Wiki:- https://wiki.ubuntu.com/

v. Ubuntu Version Control:- https://code.launchpad.net/ubuntu vi. Ubuntu Bug Tracking:- https://bugs.launchpad.net/ubuntu

vii. Ubuntu Localization :- <https://translations.launchpad.net/ubuntu>viii. Ubuntu Documentation:- https://help.ubuntu.com/community

Sample of Fedora

I. <https://getfedora.org/>

Ii. <https://fedoraproject.org/wiki/IRC>

Iii. <https://fedoraproject.org/wiki/>

Iv.<https://fedoraproject.org/wiki/Packaging:Versioning> /

<https://fedoraproject.org/wiki/Infrastructure/VersionControl>

V. <https://fedoraproject.org/wiki/Bugzilla>

Vi. <https://fedoraproject.org/wiki/Category:Localization>

Vii. <https://docs.fedoraproject.org/en-US/index.html>

Sample of Mint

i. mint Version Control http://community.linuxmint.com/software/view/subversion ii. Mint Bug Tracking:- http://forums.linuxmint.com/viewtopic.php?f=60&t=157099 iii. Mint Localization :-http://www.linuxmint.com/communities.php iv. Mint Documentation:-http://www.linuxmint.com/documentation.php

**7. Compilation of Linux Kernel selected above. (Fedora/CentOS, etc: Any One)**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To demonstrate how to compile Linux Kernel.* ***Outcome:*** *Self learning (PO: k, l)*

In this student have to do the compilation any one mentioned above Linux distro Linux Kernel on their system/VMware/Virtual box or pen drive or docker container.

Reference:-

i. Installation Of Linux Kernel on Fedora:- http://www.howopensource.com/2011/08/how-to-install-compile-linux-kernel-3-0- in-fedora-15-and-14/

Or/and ii. https://fedoraproject.org/wiki/Building\_a\_custom\_kernel

iii. Latest kernel installation on Fedora and Cent OS:- http://www.tecmint.com/kernel-

3-5- released-install-compile-in-redhat-centos-and-fedora/ iv. http://tldp.org/guides.html

v. Installation Of Linux Kernel on Ubuntu:-

Latest kernel installation on :-

<http://www.backtracklinux.org/forums/showthread.php?t=49347>Installation of Linux Kernel on Suse:-https://en.opensuse.org/Kernel

In Journal you have to write the step by step process of compilation.

**8. Create of RPM or DEB packages (Any One)**

**(Submission by Individual [I])**

***Objective:*** *To Create package for any above Linux distros.* ***Outcome:*** *(PO: b, I, j, k, l)*

Students have to study RPM or DEB package building for their C, C++ or JAVA Codes(any one programming languages codes). They must build an rpm or debian package and install it through package manager such as YUM or APT-GET

Reference:-

1. Build Simple rpm package:- http://rhce.dposs.org/index.php?

title=Build\_a\_simple\_RPM\_that\_packages\_a\_single\_file

1. Fedora rpm doc:- http://fedoraproject.org/wiki/How\_to\_create\_an\_RPM\_package iii. Simple DEB package for your C code:- http://linuxconfig.org/easy-way-to-createa- debian-package-and-local-package-repository
2. Simple DEB build guide:- http://askubuntu.com/questions/90764/how-do-i-createa- deb-package-for-a-single-python-script
3. Deb Package Build YouTube:- http://www.youtube.com/watch?v=nhoRyd2CEVs In Journal you have to write the package building process.

**9. Install and demonstrateof Server based services and their Uses.**

**(**web **server apache or tomcat or IIS, NFS,NIS**: **Any One)**

**(Submission by Individual [I])**

***Objective:*** *To know server installations and Configurations on Linux Platform* ***Outcome:*** *(PO: b, I, j, k, l)*

Students are asked to install and configure at least 2 servers, such as FTP, HTTP server (web server), TELNET, NFS, NIS etc. All configurations must be done on Linux Platform

In Journal, they have to write installations, Configurations and Screenshots of server on which they worked.

Make the Official Repositories of any one above **server** on docker store (https://hub.docker.com/) and experiment.

References:-

i. Server World:- http://www.server-world.info/en/ ii. Yolinux :- http://www.yolinux.com/

iii. GUI based tool for server configuration Webmin <http://www.webmin.com/index.html>

**10. Development of new Open Source Software or contribution to existing Open Source Software.**

**(Any small application other than Music Player or Calculator or Text Editor in java/python/perl/c/cpp/etc: Any One or New open source development).**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To contribute/introduce the open source software by understanding the GPL Licensing.*

***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

a. Develop simple software for basic needs such as Calculator, editor etc.

Use following:-

i. Language:- C/C++, Python, Perl, PHP, Java, .net ii. Version Control :- GIT or SVN iii. Package Building:- debian or rpm iv. Translation:- Marathi or Hindi

v. Documentation:- Use Mallard for your Help

**References:-**

i. http://teachingopensource.org/start-contributing-using-open-source-software/ ii. https://www.fossology.org/get-started iii. http://foss2serve.org/index.php/Category:Projects iv. http://www.hfoss.org/index.php/project\_gallery

v. GIT version control Tutorial:- http://git-scm.com/documentation vi. SVN :- http://michael-zamir.blogspot.in/2012/01/svn-tutorial.html vii. Translation :

[-http://www.tuxamito.com/joomla/index.php/es/component/content/article/60gettext](http://www.tuxamito.com/joomla/index.php/es/component/content/article/60-gettext)- tutorial viii. Using Malarad:- <http://projectmallard.org/about/learn/tenminutes.html>

1. http://www.hfoss.org/index.php/project\_gallery
2. http://www.shlomifish.org/philosophy/computers/open-source/how-to-startcontributing/tos-document.html

In Journal you have to write the process in Brief.

**11. Docker container : An open source software development platform (any two)**

**(Submission by Individual or Group [I or G])**

***Objective:*** *To understand and use the docker virtualization as OSS.*

***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

1. With the help of Docker/Container show the any one above Linux distros selected.

(in assignment 1.)

1. 1. Create image/container of any FOSS tool and upload on Docker Hub.

2. Pull images/containers from docker-hub: https://hub.docker.com/

1. (FOSS tool bug tracking tool, Project management tool, Version control system, CMS, python, java language running/compilation support, etc. and follow respective tool assignment)
2. Contribute/Introduce the docker/container to make the resource management easy and lighter.
3. Create IPC between two OS container. \*\*\* for TY UOS\*\*\*
4. With the help of Docker-compose deploy the ‘Wordpress’ and ‘Mysql’ container and access the front end of ‘Wordpress’
5. Docker image:
   1. Create a simple Hello-world python flask application and create the docker imageof that Flask application.
   2. Run the docker container from recently created image and run that docker container to 5000 port of host system.
6. Create the ‘nginx’ container from ‘nginx’ image. And create the load balancing so that if we go to tha address of ‘nginx’ it can redirect it to the above created applications (Flask and Wordpress).

**Note.** Docker has to be installed first, to carry out docker based experiments. Prefer the Linux OS to do this assignment.

**References:-**

1. https://www.docker.com/
2. https://opensource.com/resources/what-docker

iii.https://mobyproject.org/ iv.https://labs.play-with-docker.com/

**Extra Resources docker/container learning:**

1. play with docker

<http://labs.play-with-docker.com/>

1. docker curriculum on github <https://docker-curriculum.com/>
2. awesome-docker on github <https://github.com/veggiemonk/awesome-docker>
3. docker cheatsheet <https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf><https://github.com/wsargent/docker-cheat-sheet>
4. basics of docker pdf used in workshop <ftp://10.10.13.13/Basics%20of%20Docker.pdf>
   1. **Find python kernel code and compile it or use any python library for any application.**

**(Submission by Individual or Group [I])**

***Objective:*9.10.**  *To use the python open source ready module for application development.*  ***Outcome:*** *Self learning/lifelong learning (PO: b, I, j, k, l)*

Example:

* + 1. A Python library to write a table in various formats: CSV / Elasticsearch / HTML /

JavaScript / JSON / Jupyter Notebo…

* + 1. Python Driver for ArangoDB, a NoSQL graph database
    2. Deep learning (ANN liabray)

*Libraries for concurrent and parallel execution.*

* + [eventlet](http://eventlet.net/)  - Asynchronous framework with WSGI support.
  + [gevent](http://www.gevent.org/)  - A coroutine-based Python networking library that uses [greenlet](https://github.com/python-greenlet/greenlet).
  + [multiprocessing](https://docs.python.org/2/library/multiprocessing.html)  - (Python standard library) Process-based "threading" interface.
  + [threading](https://docs.python.org/2/library/threading.html)  - (Python standard library) Higher-level threading interface.
  + [Tomorrow](https://github.com/madisonmay/Tomorrow)  - Magic decorator syntax for asynchronous code.
  + [uvloop](https://github.com/MagicStack/uvloop)  - Ultra fast implementation of asyncio event loop on top of libuv. **Profiler**
  + [line\_profiler](https://github.com/rkern/line_profiler)  - Line-by-line profiling.
  + [memory\_profiler](https://github.com/fabianp/memory_profiler)  - Monitor Memory usage of Python code.
  + [profiling](https://github.com/what-studio/profiling)  - An interactive Python profiler.   [vprof](https://github.com/nvdv/vprof)  - Visual Python profiler.

Game Development : *Awesome game development libraries.*

* + [Cocos2d](http://cocos2d.org/)  - cocos2d is a framework for building 2D games, demos, and other graphical/interactive applications. It is based on pyglet.
  + [Panda3D](https://www.panda3d.org/)  - 3D game engine developed by Disney and maintained by Carnegie Mellon's Entertainment Technology Center. Written in C++, completely wrapped in Python.
  + [Pygame](http://www.pygame.org/news.html)  - Pygame is a set of Python modules designed for writing games.
  + [PyOgre](http://www.ogre3d.org/tikiwiki/PyOgre)  - Python bindings for the Ogre 3D render engine, can be used for games, simulations, anything 3D.
  + [PyOpenGL](http://pyopengl.sourceforge.net/)  - Python ctypes bindings for OpenGL and it's related APIs.
  + [PySDL2](http://pysdl2.readthedocs.io/en/rel_0_9_5/)  - A ctypes based wrapper for the SDL2 library.   [RenPy](https://www.renpy.org/)  - A Visual Novel engine.

Ref: <https://github.com/vinta/awesome-python>

* 1. **Agile s/w engineering by using Tuleap, review board and gitea (tools of agile setup)** https://www.tuleap.org/1. Read and register for online use. (B and I)
  + Offline use, setup Tuleap, review board and gitea setup on docker container?(E)
  + Share you tool details setup, usage, videos link etc in assignment write-up. Take small final year project in this environment as case study for demonstration.  This assignment is introduced in year 19-20 first time.

**14. Learn Open source programming language GO (Compulsory for E)**

* Write 3 suitable programs using GO language.
* Compare GO language with functional and procedural languages.
* Enlist the features of GO language
* Commands and compiler, debuggers of GO.
* Applications of GO language
* Put programs (with statements) in write-up with compilation steps details and upload.

References:

* https://gobyexample.com/
* https://golang.org/doc/
* https://www.tutorialspoint.com/go/index.htm

1. **(extra ISE bonus )Sugarlab is an activity-focused, free/libre OSS learning platform for children. (optimal in B)**

<https://sugarlabs.org/>

Sugar is an activity-focused, free/libre open-source software learning platform for children.

Collaboration, reflection, and discovery are integrated directly into the user interface. Through Sugar's clarity of design, children and teachers have the opportunity to use computers on their own terms. Students can reshape, reinvent, and reapply both software and content into powerful learning activities. Sugar's focus on sharing, criticism, and exploration is grounded in the culture of free software (FLOSS)

1. **FOSS HacktoberFest Pull requests can be made in any GitHub-hosted repositories/projects (Optional to assignment 10)**

This is Optional assignemnt to assignemnt 10. Do any one at least.

Visit for this site before start and reads first page carefully.

Every year octomber is FOSS month.... hence this is FOSS contribution initiateves.

Year: 2022: 04 push/bugs to be done of foss project at github.

<https://hacktoberfest.com/>

<https://hacktoberfest.digitalocean.com/>

**Steps**.

* + 1. Create account on above link.
    2. Make 4 valid Pull Requests on Github between 1st -31st October.
    3. Post the screenshot of your Hacktoberfest profile and screenshot of Github Profileon Schoology. (In write-up of this assignment)
    4. Give details of all 4 Pull Requests in write-up.
    5. Also attach the screenshot of email in write-up.

1. **Emac editor experience for coding, documentation, indentation, foss std/google std coding in languages (Optional in B)**

Experiment Emac editor on linux/ windows for coding, documentation, indentation, foss std/google std coding in langauges like c,cpp, python, java, latex, etc.

In word file upload

* + 1. Give detail/steps of emac editor installation and pakages of coding, documentation, indentation, foss std/google std coding.
    2. Explain you experience of coding with Emac editor.
    3. Comapre Emac with ecllpise IDE.
    4. Give important link/ reference /videos of Emac editor

1. **Licensing (terms n conditions) compariosnons: 1. social media, 2. email 3. public cloud 4.**

**Propritory sw 5. FOSS**

* 1. social media (Youtube, facebook, tweeter, tiktok, linkedin)
  2. email (gmail, rediff, yahoo)
  3. public cloud (AWS, azure, GCP, Alibaba)
  4. Propritory softwares (any two well known: Ex: Win vs MS Office)
  5. FOSS softwares (Mozilla Firefox, LibreOffice,GIMP,VLC Media Player)

Any two comparisons to be submitted in table form (point wise) with respect to comment on security/use of personal data

**Course Objective:**

* + 1. To Configure the Open Source Software.
    2. To contribute/ develop software (system) for open source environment.
    3. To use FOSS for Software Engineering.

**Percentage of Objective achieved by students:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objective No. | Not  achieved | 40%  achieved | 70% achieved | 100% achieved |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

**Course Learning Outcomes:**

* 1. Use of FOSS in software development
  2. Understanding the economics of FOSS 3. Define free software, Open source FOSS Software.

**Percentage of Outcome achieved by students:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outcomes | Not achieved | 40% achieved | 70% achieved | 100% achieved |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

(Sign) (Sign)

**Mr. Sanatan Sanket Mr. Pawar Pratik**

**Program Outcomes:**

1. **Engineering Knowledge:** Apply the knowledge of mathematics, engineering run and computational science to the solution of engineering problems.
2. **Problem Analysis:** Identify, formulate. interpret and analyze the complex engineering problems leading to substantiated conclusions
3. **Design/Development of Solutions**: Design systems, components or processes to meet desired needs within realistic constraints such as economic, environmental, societal and ethical considerations.
4. **Conduct investigations of Complex Problems**: Use research based knowledge and methods including design of experiments, analysis, interpretation and synthesis of information to provide valid conclusions.
5. **Modern Tool Usage**: Select and apply appropriate techniques, engineering skills and modem IT tools to prototype the model of complex engineering activities.
6. **The Engineer and Society:** Apply contextual knowledge pertaining social, secure. Legal and cultural issues with consequent responsibilities relevant to IT.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in social, environmental and the global contexts, demonstrating the knowledge of and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to the professional ethics with responsibilities and norms of the engineering practices.
9. **Individual and Team Work:** Work effectively as an individual as well as a member or a leader in diverse teams for multidisciplinary settings.
10. **Communication:** Communicate effectively with the engineering community and with society at a large, such as, being able to comprehend and write reports and design documentation to make effective presentations.
11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply those to original work or contemporary issues, as a member or land a leader in a team or an entrepreneur to manage projects in multidisciplinary environments.
12. **Life-long Learning**: Recognize the need and prepare to engage independent and in lifelong learning.

**PO Mapping with Tutorial List**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment No *\*PO | (a) | (b) | (c) | (d) | (e) | (f) | (g) | (h) | (i) | (j) | (k) | (l) |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |

(Sign)  (Sign)  **Mr.Sanatan Sanket Mr.Pratik Pawar**

**Course Learning outcomes:**

|  |  |  |  |
| --- | --- | --- | --- |
| CO | After the completion of the course the student should be able to | Bloom’s Cognitive | |
| Level | Descriptor |
| CO1 | Exercise the FOSS tools in software development. | 3 | Applying |
| CO2 | Analyze the economics of FOSS. | 4 | Analyzing |
| CO3 | Create new FOSS or contribute to existing FOSS in FOSS environment. | 6 | Creating |

**CLO mapping with assignment list**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Assignment | BIE | % Copy | Mapping | category |
| 1 | Demonstration of Linux Distributions OS’s and their purpose with comparisons. | B | 5% | CO1 | Application |
| 2 | Use of Open Source tools for Project Management. | B | 1% | CO2 | Application |
| 3 | Use of Bug Tracking | B | 1% | CO1 | Knowledge |
| 4 | Use of version control system | B | 0% | CO2 | Application |
| 5 | Installation and Use of CMS software | B | 5% | CO1 | Application |
| 6 | Comprehend the open source software development for any one LinuxDistro | B | 1% | CO3 | Synthesis Design |
| 7 | Compilation of Linux kernel | B | 1% | CO3 | Synthesis |
| 8 | Creation of RPM or DEB packages | B | 0% | CO2 | Application |
| 9 | Install and demonstrate server based services and their uses | B | 5% | CO2 | Application |
| 10 | Development of new Open Source Software or contribution to existing Open Source Software. | B | 1% | CO2 | Application |
| 11 | Docker container: An open source software development platform | B | 1% | CO2 | Application |
| 12 | Find python kernel code and compile it or use any python library for any application. | B | 0% | CO3 | Application |
| 13 | Agile s/w engineering by using Tuleap, review board and gitea (tools of agile setup) | B | 5% | CO1 | Synthesis Design |
| 14 | Learn Open source programming language GO | B | 1% | CO2 | Knowledge |
| 15 | Sugarlab is an activity-focused, free/libre OSS learning platform for children. | B | 1% | CO2 | Creating |
| 16 | FOSS HacktoberFest Pull requests can be made in any GitHub-hosted repositories/projects | B | 10% | CO1 | Creating |
| 17 | Emac editor experience for coding,  documentation, indentation, foss std/google std coding in languages | B |  | CO2 |  |
| 18 | Licensing (terms n conditions) compariosnons:  1. social media, 2. email 3. public cloud 4.  Propritory sw 5. FOSS | B |  |  |  |

**Rubrics Used:**

1. Quiz Objective 2. ClassQuestioning.

* 1. Quiz Subjective
  2. Open Book Test
  3. Assignment
  4. Program
  5. Seminar
  6. Mini project
  7. PPT
  8. Demo Simulator
  9. ISE1/ISE2/ESE
  10. Videos
  11. Posters
  12. Presentations

**Rubrics mapping with assignment list:**

Assignment No./Rubric No.

1

2

3

4

5

6

7

8

9

10

11

12

13

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

**Self Evaluation by Student:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PRN | Mail ID | Sign |
| Mr. Sanatan Sanket | 2019BTEIT00071 | sanketsanatan970@gmail.com |  |
| Mr. Pawar Pratik | 2019BTEIT00006 | pawarpratik80009@gmail.com |  |

**Assignment 1**

Objective: To install, demonstrate and study Various Linux Distributions and their Purpose/comparison/differences.

Outcome: Self learning/lifelong learning

**i]Various versions of that distros with code name**

**a)Fedora Version:**Fedora Core 1 (Codename: Yarrow)

Fedora Core 2 (Codename: Tettnang

Fedora Core 3 (Codename: Heidelberg) Fedora Core 4 (Codename: Stentz)

Fedora Core 5 (Codename: Bordeaux)

Fedora Core 6 (Codename: Zod)

Fedora 7 (Codename: Moonshine)

Fedora 8 (Codename: Werewolf)

Fedora 9 (Codename: Sulphur)

Fedora 10 (Codename: Cambridge)

Fedora 11 (Codename: Leonidas)

Fedora 12 (Codename: Constantine)

Fedora 13 (Codename: Goddard)

Fedora 14 (Codename: Laughlin)

Fedora 15 (Codename: Lovelock)

Fedora 16 (Codename: Verne)

Fedora 17 (Codename: Beefy Miracle)

Fedora 18 (Codename: Spherical Cow)

Fedora 19 (Codename: Schrödinger's Cat)

Fedora 20 (Codename: Heisenbug)

**b)Debian:**Debian 1.1 (Buzz)

Debian 1.2 (Rex)

Debian 1.3 (Bo)

Debian 2.0 (Hamm) Debian 2.1 (Slink)

Debian 2.2 (Potato)

Debian 3.0 (Woody)

Debian 3.1 (Sarge

Debian 4.0 (Etch)

Debian 5.0 (Lenny)

Debian 6.0 (Squeeze)

Debian 7 (Wheezy)

Debian 8 (Jessie)

Debian 9 (Stretch)

Debian 10 (Buster)

Debian 11 (Bullseye) Debian 12 (Bookworm)

**ii) Default Desktop Environment-**

**What is Desktop Environment ?**

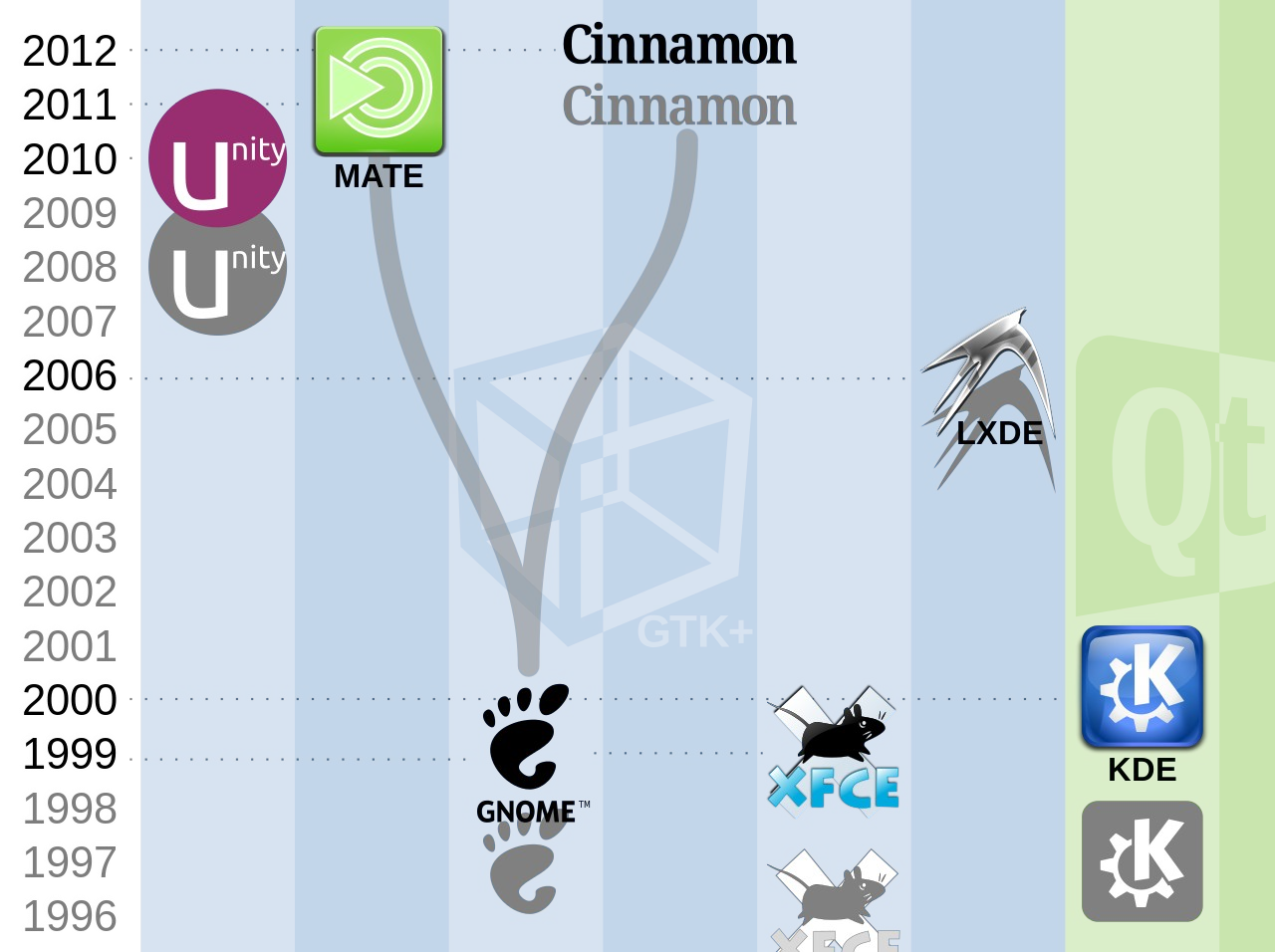
A desktop environment is the bundle of components that provide you common graphical user interface (GUI) elements such as icons, toolbars, wallpapers, and desktop widgets. Thanks to the desktop environment, you can use Linux graphically using your mouse and keyboard like you do in other operating systems like Windows and macOS.

There are several desktop environments and these desktop environments determines what your Linux system looks like and how you interact with it.

Most of the desktop environments have their own set of integrated applications and utilities so that users get a uniform feel while using the OS. So, you get a file explorer, desktop search, menu of applications, wallpaper and screensaver utilities, text editors and more.

Without a desktop environment, your Linux system will just have a terminal like utility and you’ll have to interact it using commands only.

Below is the A brief timeline of the most popular modern desktop environments for Unix-like operating systems (greyscale logos indicate when the project's development started, while colorized logos indicate the project's first release)



Desktop Enviroment usually includes everything from how the windows look and feel, to the style of the icons, files, folders, and the mouse pointers. A desktop environment also dictates what file manager to use, default text editor, image viewer, wallpapers, and the interface used to log in and log out of the local system.

# Different desktop environments in Linux

There are various desktop environment in Linux

Think of the desktop environments as clothes. The clothes determine what you look like. If you wear skinny jeans and flat shoes, you would look good but running or hiking in those clothes won’t be comfortable.

Some desktop environments such as GNOME focus on a modern look and user experience while desktop like Xfce focus more on using fewer computing resources than on fancy graphics.



Screenshot Xfce Desktop Environment

Your clothes depend on your need and determine your looks, the same is the case with the desktop environments. You have to decide whether you want something that looks good or something that lets your system run faster.

Some of the popular desktop environments are:

i. GNOME – Uses plenty of system resources but gives you a modern, polished system ii. Xfce – Vintage look but light on resources iii. KDE – Highly customizable desktop with moderate usage of system resources iv. LXDE – The entire focus is on using as few resources as possible

v. Budgie – Modern looks and moderate on system resources



Screenshot of GNOME Desktop Environment

# iii]Main Purpose

## a)Fedora:-

* Fedora OS offers many architectures.
* Fedora OS is a very reliable and stable operating system.
* It provides unique security features.
* Fedora OS provides a very powerful firewall.
* Fedora OS is very easy to use.
* It supports a large community.
* Fedora OS is actively developed.
* Fedora OS is an open-source OS.
* The interface of Fedora OS is very attractive.
* This operating system offers live mode tools.
* This operating system enhances internet speed.

**b)Debian:-**There are a lot of reasons to choose Debian as your operating system – as a user,

as a developer, and even in enterprise environments. Most users appreciate the stability, and the smooth upgrade processes of both packages and the entire distribution. Debian is also widely used by software and hardware developers because it runs on numerous architectures and devices, offers a public bug tracker and other tools for developers. If you plan to use Debian in a professional environment, there are additional benefits like LTS versions and cloud images.

**iv]. Package management of that distros -**

Most Linux distributions have a package management system with online repositories containing thousands of packages. This makes it very easy to install and remove applications, operating system components, documentation and much more.

We first discuss the Debian package format .deb and its tools dpkg, apt-get and aptitude. This should be similar on Debian, Ubuntu, Mint and all derived distributions.

Then we look at the Red Hat package format .rpm and its tools rpm and yum. This should be similar on Red Hat, Fedora, CentOS and all derived distributions.

## a] Fedora

Fedora is a distribution that uses a package management system. This system is based on rpm , the RPM Package Manager, with several higher level tools built on top of it, most notably PackageKit (default gui) and DNF. GNOME Software is another GUI package manager.

## Advantages of package management systems

Package management systems have many advantages:

vi. It’s easy to query what version of a package is installed or available. vii. It’s easy to remove a package entirely, making sure all its files are gone.

1. It’s easy to verify the integrity of the packages files, so you can see if it’s been corrupted or tampered with.
2. It’s easy to upgrade a package by installing the new version and removing all the old versions files. This will make sure not to leave any lingering files from the old package around to confuse or break things.
3. It’s easy to see what packages require or provide things that other packages provide or require, so you can be sure to have the needed items for the package to function correctly. xi. It’s easy to install or remove groups of packages.

xii. In many cases it’s possible to downgrade back to a previous version of a package, for example when a new version has a bug.

## Disadvantages of package management systems

• You are restricted to either using the versions of the package that are available or having to make your own package if you need a different version.

## Why mixing source installs and packages is a bad idea

Package management systems have no way to query or note when you bypass them and install something from source. You should avoid mixing source installs and packaged installs for (at least) the following reasons:

* You lose all the advantages above from a package managed system.
* Installing from source may overwrite, delete, or change existing files that are in a package, making that package not function correctly.
* The source install may override a package install causing undefined behavior in the package or source installed item.
* Installing from source makes it impossible or very difficult for anyone to help you debug issues, since versions can’t be easily queried and integrity checked.
* Fedora packages may include patches or configuration to work with other packages, but upstream source does not, leading to loss of functionality.
* Software installed from source will not upgrade with package managed packages, leading to breakage in the source install package on upgrades or os updates.

Strongly consider making your own package if you need a different version or a version of some package with changes

## Package Management tools

Here are some tools for managing packages:

* dnf-Dandified Yum
* PackageKit- PackageKit gui tool ('add/remove software' in your menu)
* GNOME - Graphical package manager for GNOME • KDE discover Graphical pacakge manager for KDE Plasma
* rpm- RPM package manager. • yumex- Yum Extender. '''

### b] Debian-

**Here are some key points for package configuration on the Debian system.**

* The manual configuration by the system administrator is respected. In other words, the package configuration system makes no intrusive configuration for the sake of convenience.
* Each package comes with its own configuration script with standardized user interface called **debconf**(7) to help initial installation process of the package.
* Debian Developers try their best to make your upgrade experience flawless with package configuration scripts.
* Full functionalities of packaged software are available to the system administrator. But ones with security risks are disabled in the default installation.
* If you manually activate a service with some security risks, you are responsible for the risk containment.
* Esoteric configuration may be manually enabled by the system administrator. This may create interference with popular generic helper programs for the system configuration.

### The Debian Package Manager Debian-

GNU/Linux provides several package management tools, primarily intended to facilitate the building, installation, and management of binary packages. Debian package names generally end in .deb. The Debian package management tools include:

**1] dpkg-** The original Debian packaging tool. Used to install or uninstall packages or as a frontend to dpkg-deb. Getting and installing packages is usually done with apt-get, but dpkg is still commonly used to install a package that is already on your system. In fact, apt-get calls dpkg to do the installation once it's gotten the package.

**2]dpkg-deb-** Lower-level packaging tool. Used to create and manage the Debian package archives.

Accepts and executes commands from dpkg or can be called directly. **3]dselect-** An interactive frontend to dpkg.

‘

### The Advanced Package Tool (APT) -

APT is a modern, user-friendly package management tool that consists of a number of commands. The most frequently used of these commands is apt-get, which is used to download and install a Debian package. apt-get can be run from the command line or selected as a method from dselect. One of the features of apt-get is that you can use it to get and install packages across the Internet by specifying an FTP or HTTP URL.we can also use it to upgrade all packages currently installed on your system in a single operation.

**v]List of packages(Fedora and Debian):-**

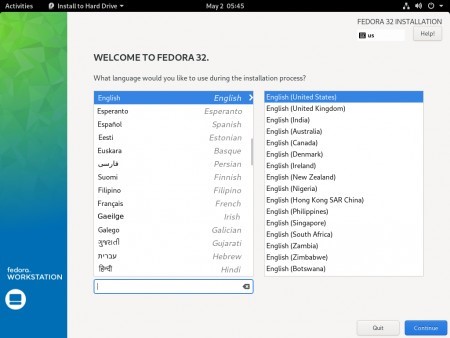
abiword (3.0.5) , alsa-lib (1.2.7.2 , amdgpu (22.20.1),bash (5.1.16),bind

(9.18.5),chromium ,cups (2.4.2),dhcp (4.4.3) ,e2fsprogs (1.46.5),firefox (103.0.1),freetype

(2.12.1),gcc (12.1.0),gimp (2.10.32),glibc (2.36),gnome-shell (42.3.1),gnucash (4.11),gnumeric

(1.12.52),grub (2.06),gtk (4.6.6),httpd (2.4.54),inkscape (1.2.1) ,k3b (22.04.3) ,kmod (30),krita (5.0.8),libreoffice (7.3.5),linux (5.19),lxpanel (0.10.1) ,mariadb (10.8.3),matedesktop (1.26.0),mesa (22.1.5),mysql (8.0.30),nautilus (42.2),NVIDIA (515.65.01),openbox (3.6.1),openjdk (18.0.1),openssh (9.0p1) ,openssl (3.0.5),perl (5.36.0),php (8.1.9),plasmadesktop ,postfix (3.7.2),postgresql (14.4) ,Python (3.10.6),qt (6.3.1),samba (4.16.4),systemd (251.3),thunderbird (102.1.1),vim (9.0),vlc (3.0.17.4),xfdesktop (4.16.0),xorg-server (21.1.4)  **vi] Screenshots a] Fedora**







### b]Debian





**vii]Compare '/etc' hierarchy**

**a)‘etc’ directory Fedora**

More than 65MB of system configuration files and directories reside under the /etc directory if you install all the software included with this book. Some major software packages, such as Apache, OpenSSH, and xinetd, have directories of configuration files under /etc. Other important system-related configuration files in /etc are fstab, inittab, passwd.

**b)‘etc’ directory Debian :**

This directory is for those types of configuration files and folders in which the system does not know where to put them. So, it is an “et Cetra” directory for the Linux Operating system.

This directory mostly contains the static program local files that affect all users. Since this directory mostly contains files related to the configuration, it is better to call it “Everything to Configure”.

# viii]Package Management:-

## a)Fedora:-

One of the advantages of Fedora is the huge amount of software available for it. Finding, installing, updating, and removing this software can be a daunting task, simply due to the amount of software available.

Fortunately, Fedora uses a software management system called *RPM Package Manager* or simply *RPM* (formerly RedHat Package Manager). RPM rolls all of the programs, scripts, documentation, configuration files, and data used by a piece of software into a single file called a *package*. The package also contains metadata describing the package, license, maintainers, and the packages upon which the package depends (for example, a KDE application will need other components of the KDE system to operate).

What RPM doesn’t provide is *dependency resolution*: the ability to *automatically* resolve dependency issues. However, the *yum* system builds on RPM to provide this capability, automatically searching external *repositories* to find needed packages and install them automatically.

## b)Debian:-

There are multiple tools that are used to manage Debian packages, from graphic or text-based interfaces to the low level tools used to install packages. All the available tools rely on the lower level tools to properly work and are presented here in decreasing complexity level.

It is important to understand that the higher level package management tools such as **aptitude** or **synaptic** rely on **apt** which, itself, relies on **dpkg** to manage the packages in the system.

**Ix]Pros/cons of both distros fedora**

**Pros :** Quite cutting-edge – sometimes bleeding-edge,lots of supported software with recent release

**Cons:** Can be less than stable due to fast-paced package and release updates ,generally thought to be Red Hat’s testing distribution

**Debian:**

**Pros:** Very stable and reliable ,wide variety of support packages, all-volunteer maintainers, supports large variety of architectures

**Cons**: Very conservative , release cycle can be slow

**x]Which one is better for development and why?**

Fedora is reliable, secure with overwhelming support, making it the best platform for development and production environments.

Fedora is a pioneer in creating a platform for the latest cutting-edge technologies and tools in IoT, containerization, and AI.

xiii.Fedora is a consistent operating system where developers can use an extensive list of software packages on development and target production environments.

1. Fedora offers freedom in both hardware and software support.
2. The extensive Fedora community “Fedora Friends” is always willing to help educate, troubleshoot, and discuss ideas.

xvi.Fedora enjoys enterprise-level support from Red Hat. xvii. Fedora always seeks to provide the future first

**xi]Which one is easy to use (for beginner) and why?**

Fedora is incredibly easy to install. They provide a graphical tool to make bootable USBs, and the installer is intuitive and easy to use. Enabling non-free packages is a simple checkbox. Once you’ve installed the OS, there’s a nice tour of the desktop. The software centre is super simple, and provides almost all the software you could want. If you want to install the NVIDIA drivers, just search «NVIDIA» and hit install. Same with Steam, Lutris etc.

Stock GNOME is a great experience out of the box. It’s very stable on Wayland, and has excellent

HiDPI and scaling support. (Granted, framebuffer scaling stilll needs a command line command. It’s

2022 GNOME. Cmon.)

**xii] Top 10 commands distro on command prompt.**

## a)Fedora:-

1. dnf check
2. dnf upgrade
3. dnf system
4. dnf install
5. dnf remove
6. type
7. trap
8. wait
9. unset
10. ownership

## b)Debian:-

1. awk - pattern-directed scanning and processing language
2. clear - clear the terminal screen
3. cmp - Compare two files
4. comm - Compare two sorted files line by line
5. crontab - Schedule a command to run at a later time
6. date - Display or change the date & time
7. dc - Desk Calculator
8. dd - Data Dump - Convert and copy a file
9. df - Display free disk space

10.diff - Display the differences between two files

**Assignment 2:** Use Of Open Source tools for project management.

**Objective:**To install, use, and demonstrate the project management tool that used in day to day life of software engineering.

**Slack**

**Theroy:**

**a)Installation on windows:** How to install the Slack app on Windows

1. Visit slack.com/downloads.
2. Click Download.
3. Once the download is complete, double-click the file (called SlackSetup.exe). Slack will launch automatically once installed.

**2)Purpose behind slack**:

**a)Project Friendly Structure** :

The first step in using slack for project management is to set up your team. You can create a single team channel for your entire company to communicate in. Or,if your company employs a large number of people ,create different teams for each segment of your group.

**b)Integreate your other tools** :

There are some standard project management features that slack just can’t replicate along. If you are busy project manager and need the heavy lifting power of more traditional app, slack can still help you out with integrations.

**c)Easy automated reminders:**

Reminders are one of our favorite slack tricks . Keeping track of your daily to-do list is a simple concept for project mangers but its lot easier said that done. Just type “/remind” in a channel, tell slackbot what you would like to be reminded of and it will ping you when the time comes.  **d)Easy Note taking**:

Want to jot some notes for a project you’re working on? Need to set a reminder? Maybe you just need a place where you can type some things where they won’t get lost. You can start a new Slack message chain with yourself. This way, your direct messages will be stored where you can easily find them later. Because they won’t get pushed down by new messages from your teammates, they’ll always be right where you left them.

**e)Audio and video calls:**

You can also start an audio or video call with up to 15 people. This means you won’t need to switch from Slack to Skype or Google Hangouts to make your work call. This feature is free, so you won’t need to pay for another service.

**Slack Versions**:

* Slack 4.27. 154. 14 June 2022. What's new. ...
* Slack 4.26. 31 May 2022. Bug fixes. ...
* Slack 4.26. 16 May 2022. Bug fixes. ...
* Slack 4.25. 11 April 2022. Bug fixes. ...
* Slack 4.25. 24 March 2022. What's new. ..

. ● Slack 4.24. 28 February 2022. Bug fixes. ...

* Slack 4.23. 7 December 2021. Bug fixes. .. . ● Slack 4.22. 8 November 2021. Security guidance.

**b)How to use Slack**

**1)Install slack**

**2) Inviting your teammates** : Other than inviting teammates one by one, you can also approve certain email domains so that your teammates can easily join your Slack team as long as they sign up with the approved email address.

**3)Set up your Slack channels:**

Your Slack team conversations will be grouped by channels that you created. You can create channels for different topics, functional teams and projects. This way, you can organize your team conversations in ways that best fits your team.

**4)Connect Slack with your existing tools :**

Add apps to your Slack workspace to connect tools or services that your team is already using. By connecting your existing tools with Slack, you can do your work without ever leaving your Slack workspace, helping you to be more productive. To see what apps are available for slack, check out their Slack App Directory.

**5)Slack commands:**

Slack has a lot of built-in slash commands that act as shortcuts for specific actions in Slack. You simply need to type the slash key (/) then followed by some keywords in any Slack channel or direct message to trigger specific actions in Slack (e.g. type /away to mark your status to “away” quickly).

**6)Mention your teammates:**

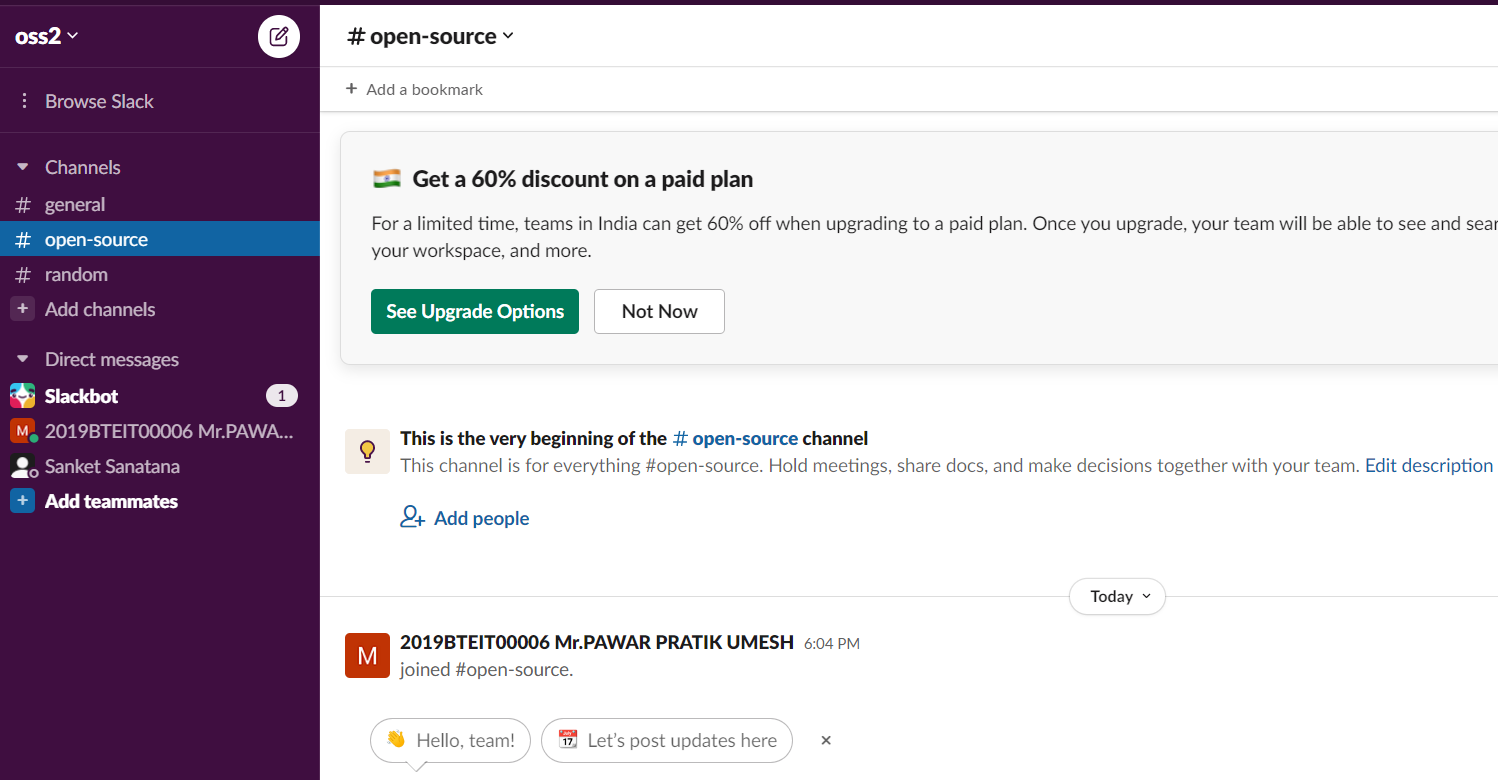
Mentions are a direct way to draw someone’s attention to a particular message even when you send that message in a channel that involves a lot of teammates. If you mention someone in a channel, that person will receive a notification (if that person is in that channel, otherwise Slack will ask if you want to invite that person to the channel) and they will be able to see the message easily by visiting the “Mentions & reactions” tab on the sidebar.

**c)Installation on Linux**

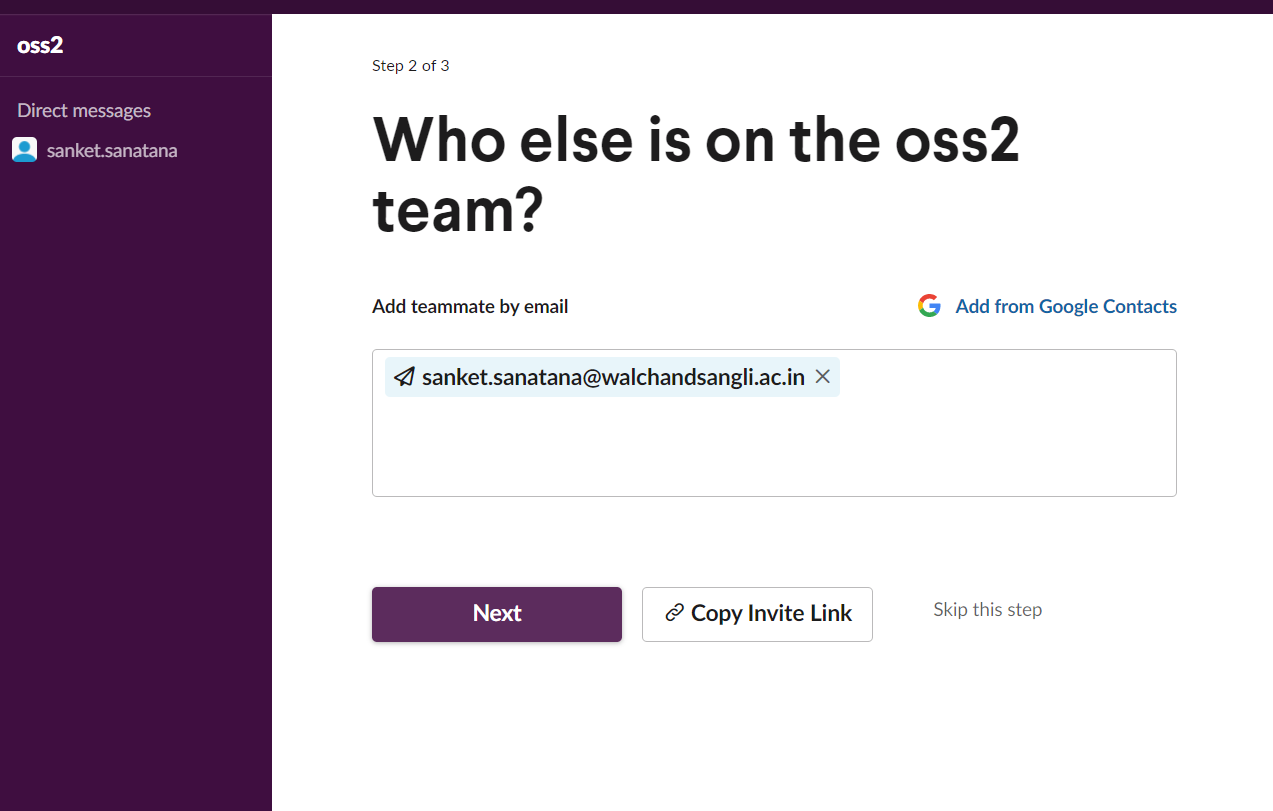
1. Visit slack.com/downloads.
2. Click on Download . ...
3. Locate the file in your Downloads folder (the downloaded file name will begin with slack-desktop).
4. Double-click the file to view Slack in the Ubuntu Software Centre.
5. Click Install.

**Snap Shots:**

**1)chatbox**



**2)Create Channel 3)Invitations for Teammates**



**Conclusion: Successfully installed and configured slack on windows and linux also created channel on slack.**

**Assignment no: 03**

**Assg. Name:** Bug tracking tool

**Tool Name:** Jira

**Purpose behind using tool:**

1. Jirahelps teams plan, assign, track, report, and manage work.
2. Jira brings teams together for everything from agile software development and customer support to start-ups and enterprises
3. Bug tracking.

**Various versions of Jira:**

1. Jira software 9.2
2. Jira software 9.1
3. Jira software 9

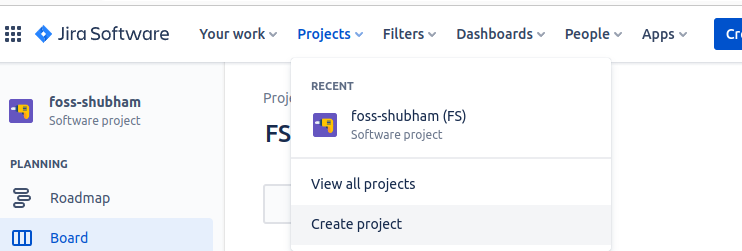
**Download and Installations:**

Used an instance of Jira provided by Atlassian.

**How to use Jira:**

We will be using Jira for creating bugs for a project and allocating to team members.

1. First if using instance on atlassian server.
2. Create your account
3. Configure account as per your usage and skills.
4. Create a project

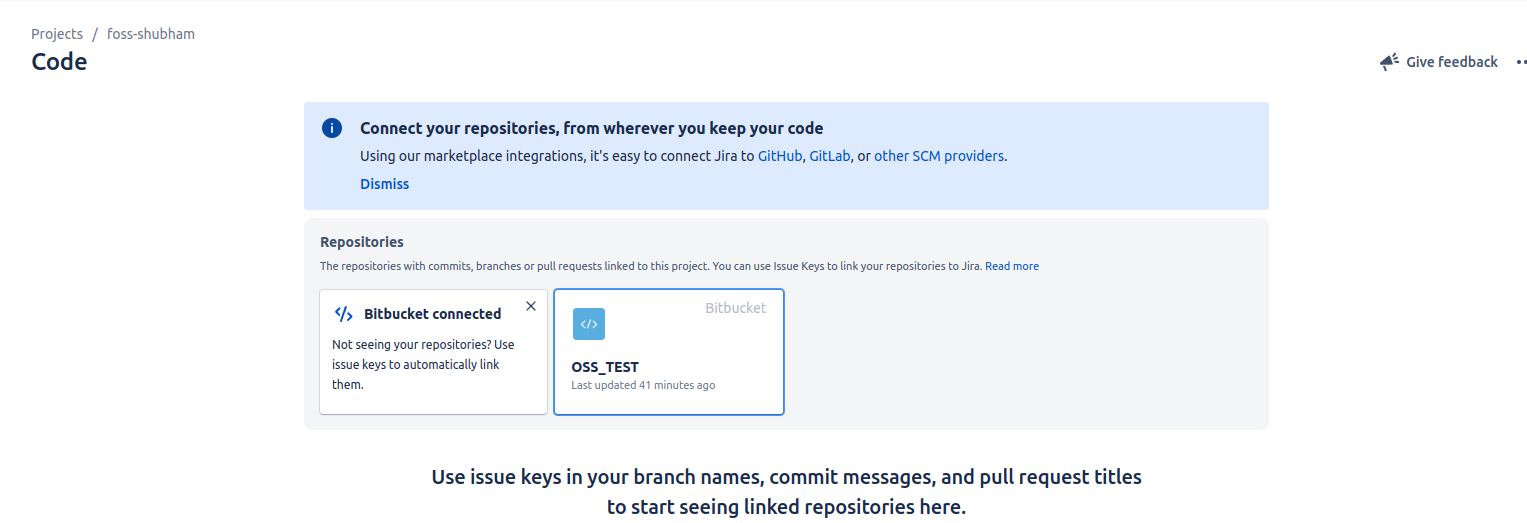


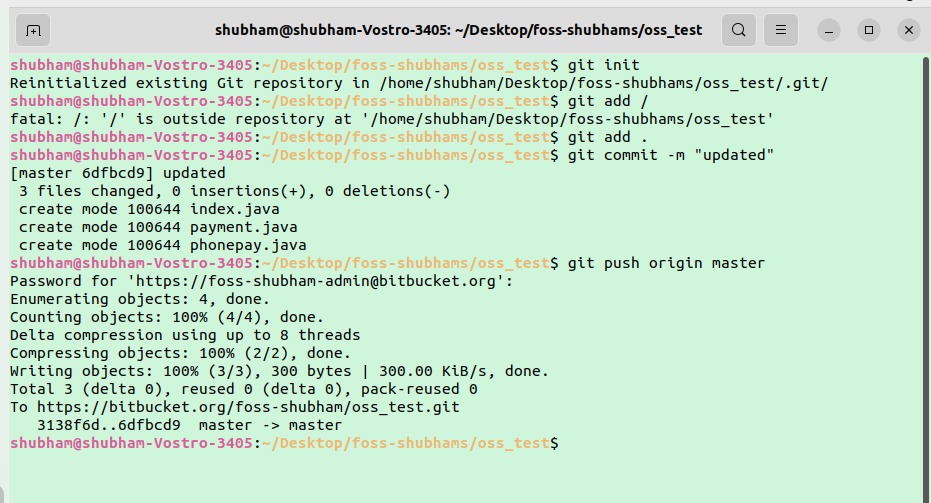


5. Give your project name, here it is foss-shubham

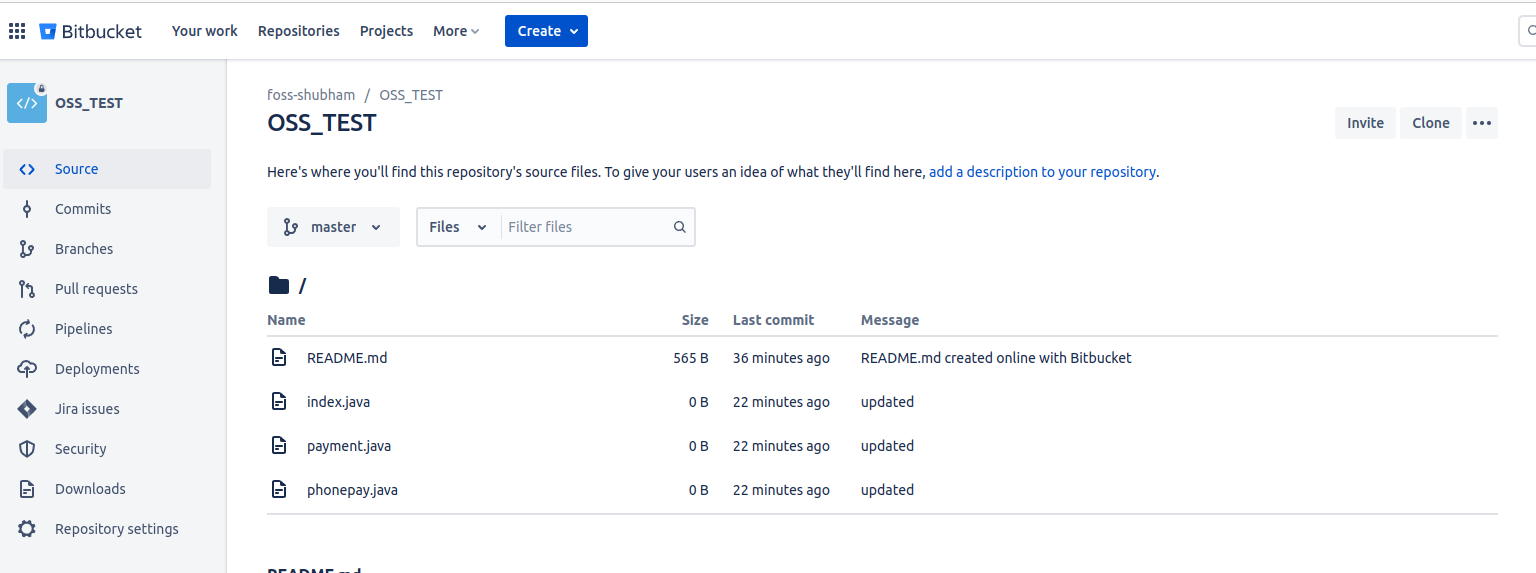
6. You can add your project code in bitbucket, to do so.

7. Go to code

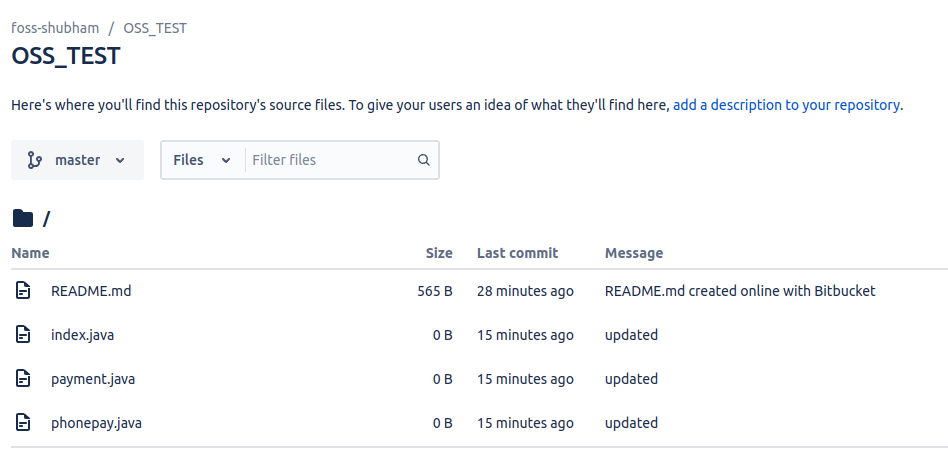


.

All files of code are now pushed to bitbucket as shown below.

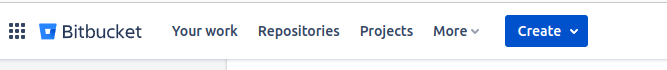


8. All code files are added to bitbucket. And this bitbucket account is integrated with jira account. So you can view all bitbucket repos in your jira account as shown below.



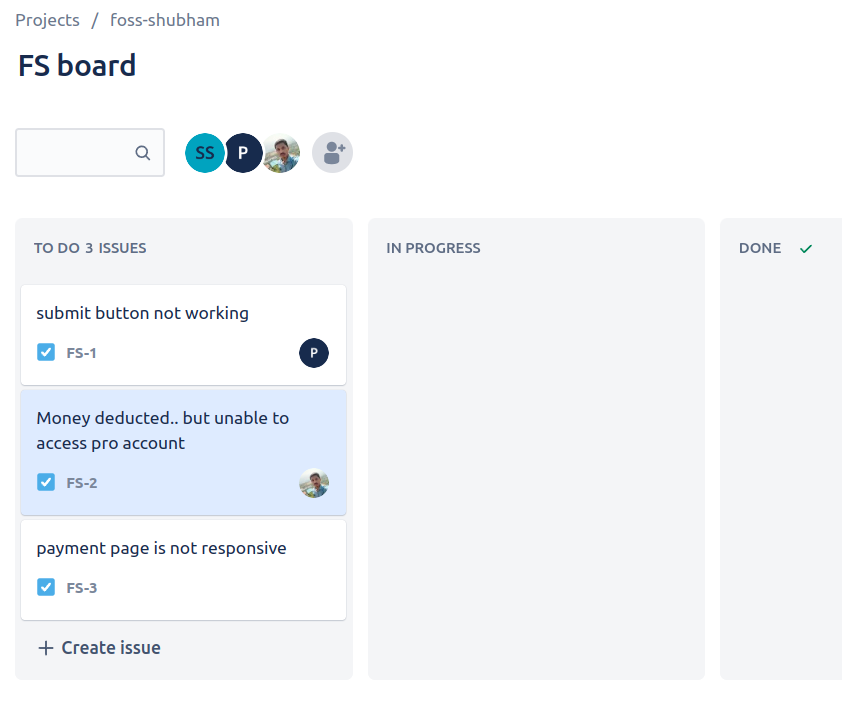
9. Now You can add bugs for your project.

10. For creating bugs click on create as shown below

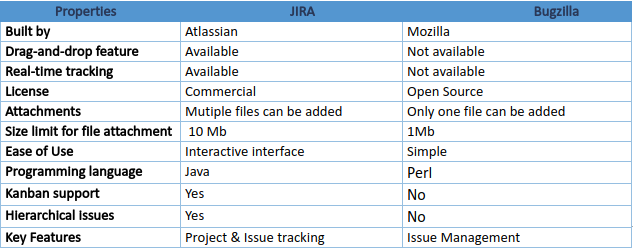


11. On clicking create you can create bug as shown below. I have created 3 bugs .

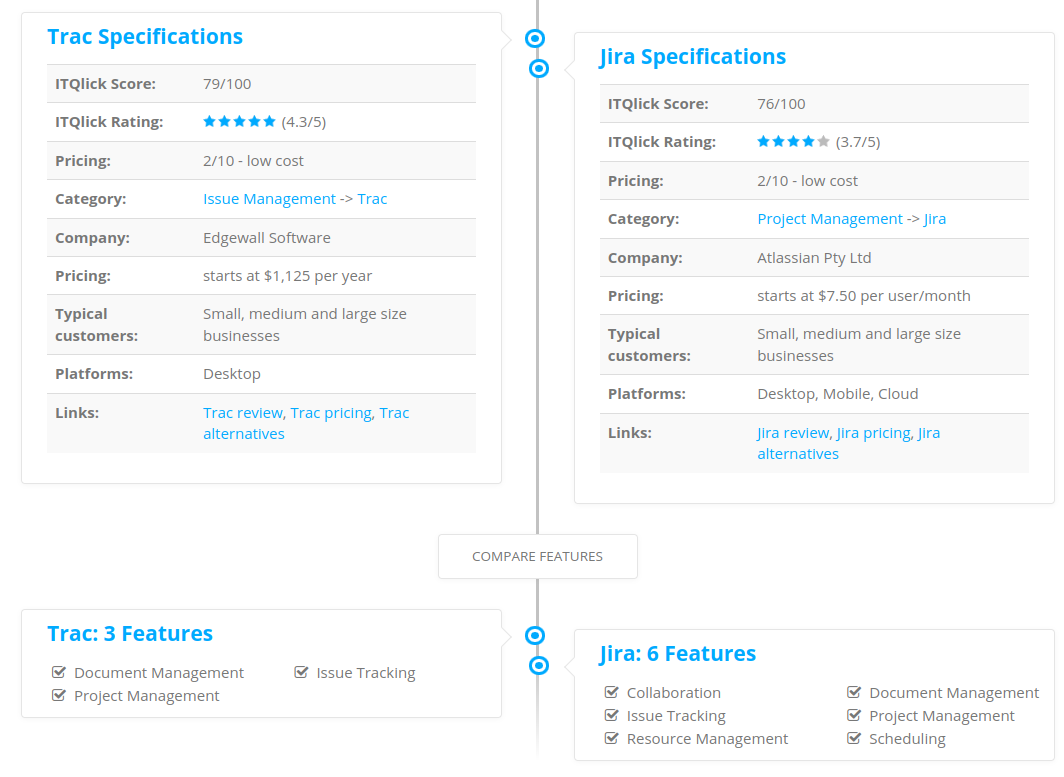
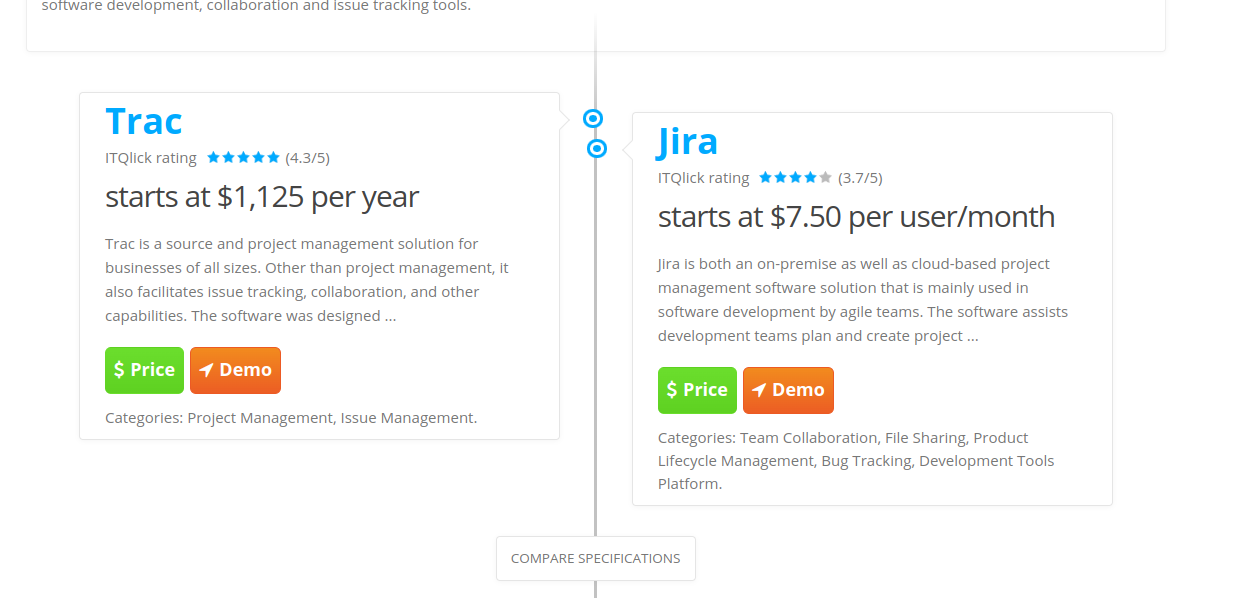
12.



**Comparision:**

****

TRAC VS JIRA

******

***Open Source Software Lab***

**Assignment 4**

**Topic : Version Control System**

**Objective : Use of version control system(git,github)**

**Theory:**

Git is a free and open source distributed version control system de- signed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like

Sub- version, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. Git is used by big faces like Google, Facebook, Microsoft, Twitter, Net- flix, Android, Gnome, QT, Rails, etc. Git is a member of Software Freedom Conservancy. It’s repository is hosted on GitHub.

**Features of GIT:**

* Git is free and Open Source.
* Git is based on branching and merging model, which makes it stand apart from every other

SCM.

* Git is small, fast and distributed.
* Git provides Data Assurance.
* Git has a staging area, where you can keep track of the development of your project.

# 1)Windows installation

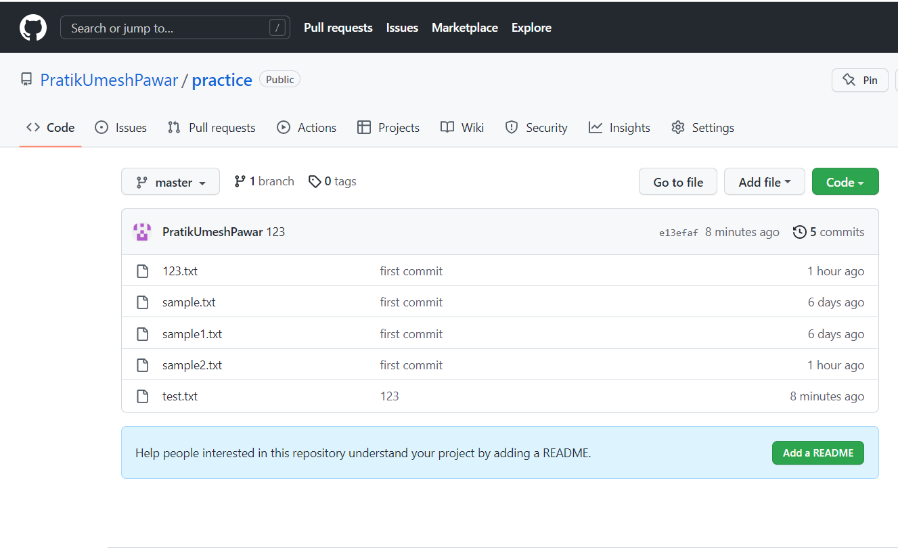
# A) installtion process

i)go to git official website and download for windows with 64 bit ii)use with gitbash instead of cmd

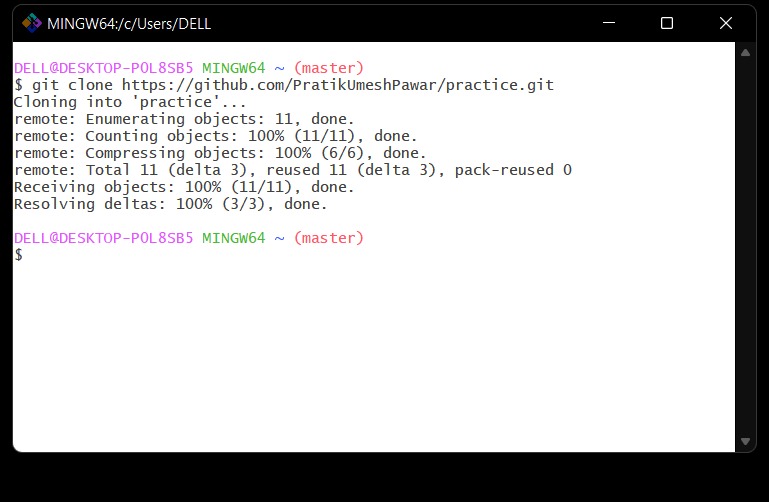
## B)making directory and adding txt and pdf file to local repository

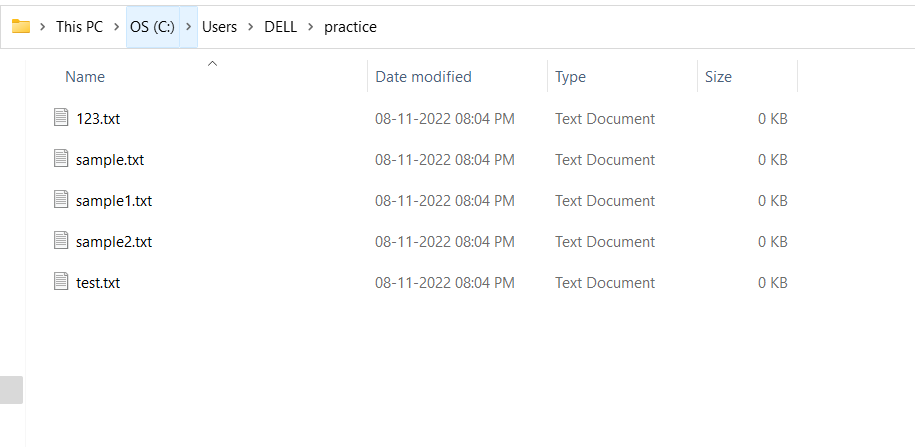


## C)creating new repository in github and linking with local repository and uploading document from local to github



## D)for cloning repository from remote to local



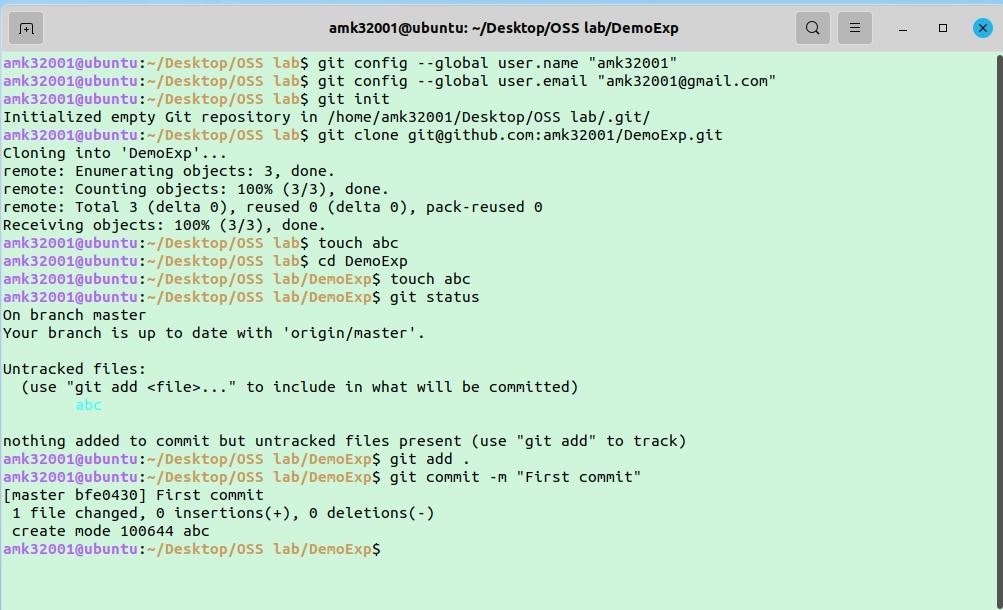


# 2)Linux Installation

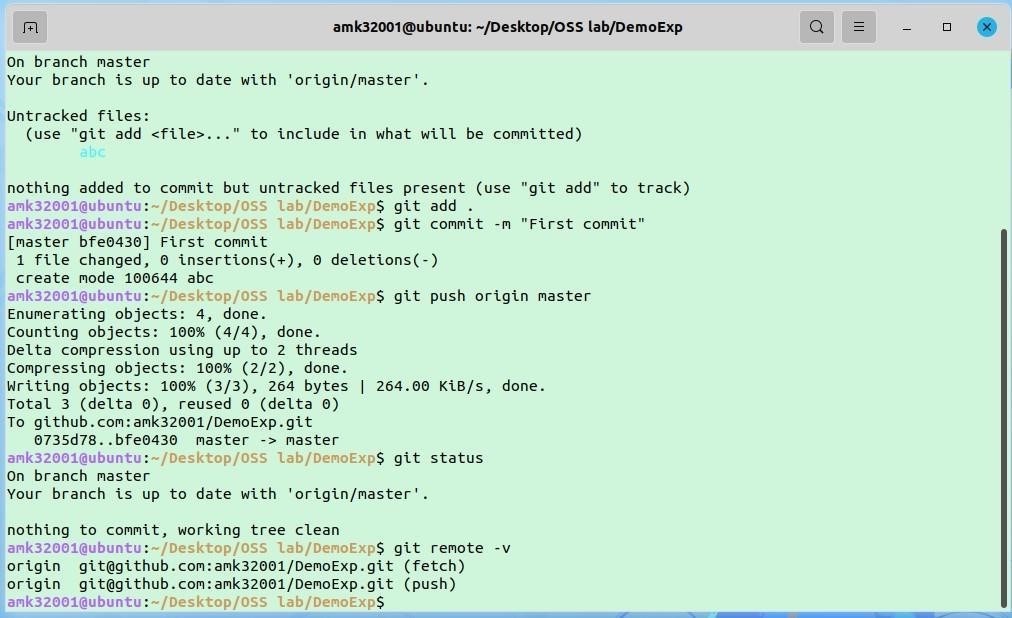
i) Install git on linux ubuntu by using terminal.

## → sudo apt-get install git

Or you can install it from the website directly. ii) Create one empty repository on github and make it public so that you can clone iteasily without any issue (In this case it is “DemoExp”). iii)Create one folder in the PC(“OSS lab”) and initialize and configure git in that location.



1. Clone the empty git repo and make some changes in it and then add and commit allchanges.
2. To make those changes visible on github, write a push command.



1. Then check the status using the “git status” command and also check if other commands are working fine or not.
2. After all commands get performed your repository will look like this.**Conclu**
3. **sion :**

**1.We learnt different VCS tools.**

**2.Compared Git with other VCS.**

**3.Overall use of GIT is topp-notch and productive.**

**Assignment No: 05**

**Installation and use of CMS software’s**

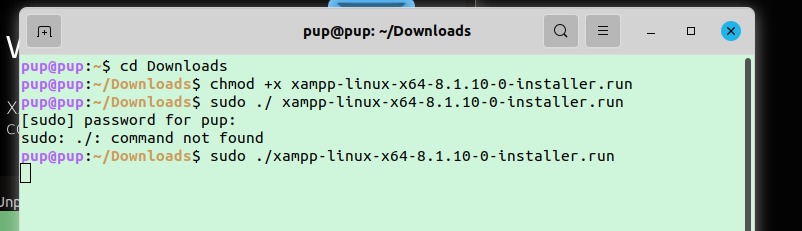
**Tools used: Xampp, Word-press**

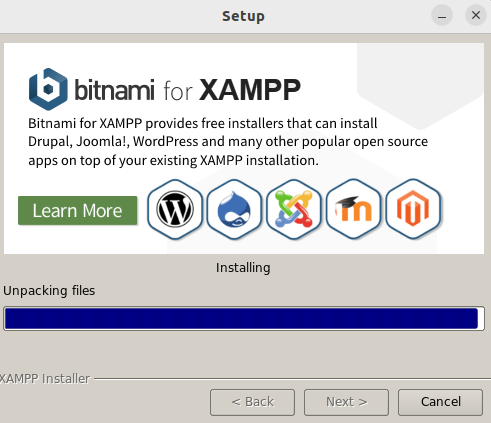
**PART A:**

Installation of Xampp: XAMPP is an abbreviation for Coss-platform, Apache, MySQL, PHP and Perl, and it allows you to build WordPress site offline, on a local web server on your computer.

1. Install Xampp from official website

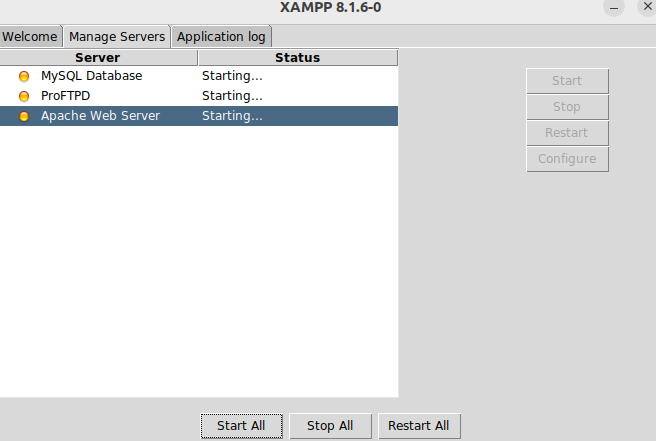
2. Change the permissions and run the file and install dialog box appears just click on next next.





3. It takes 10-15 mins for installing Xampp and then in manage servers start all servers.

4. Xampp will be installed at opt/lampp

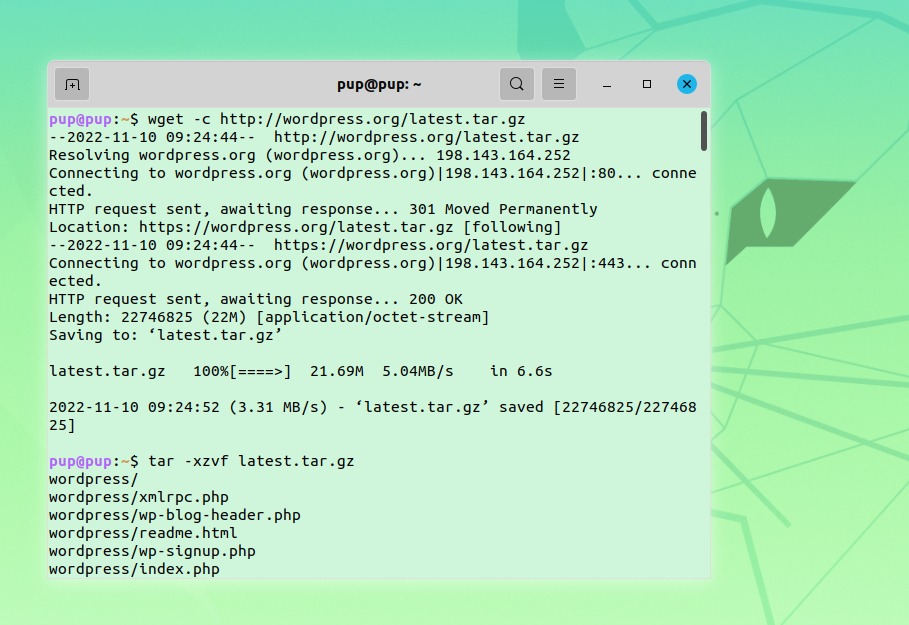


5. To check whether Apache server is installed properly click [http://localhost](http://localhost/) on your browser. If welcome page appears you installed properly.

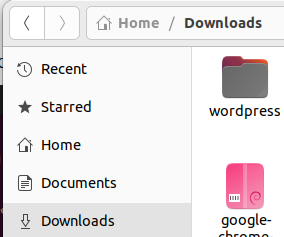


**PART B:**

1. Download Wordpress form official website and extract it

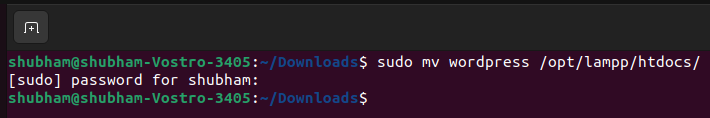


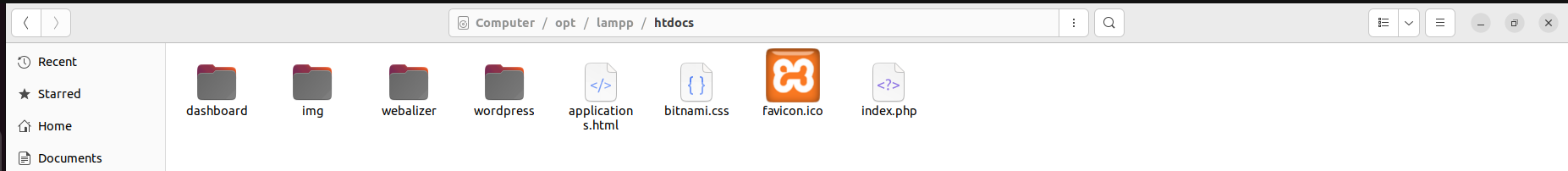
2. After extracting wordpress named folder must be there.



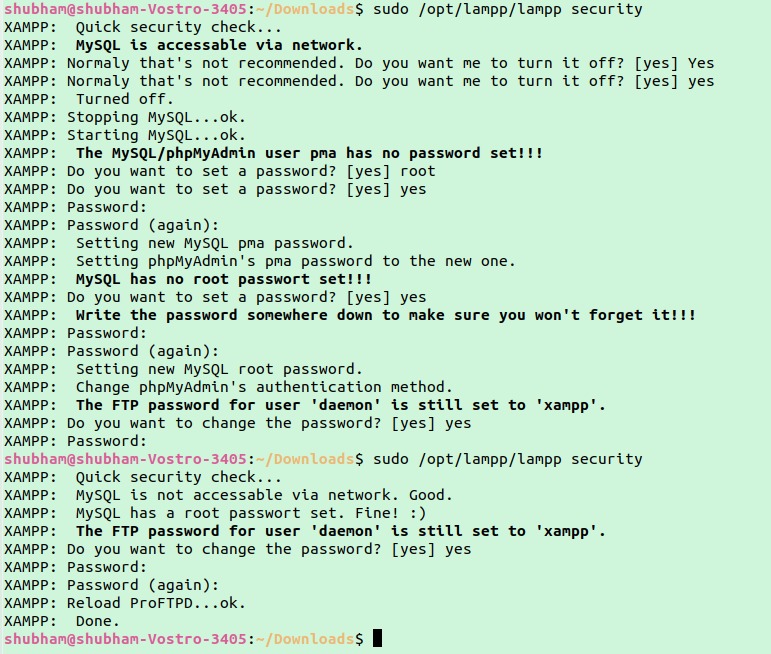
3. Open terminal from here and perform below command .

This command moves wordpresss folder to *opt*/lampp/htdocs folder

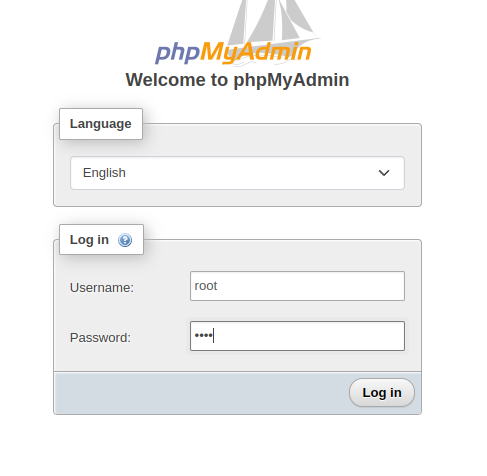




4. Execute the below commands

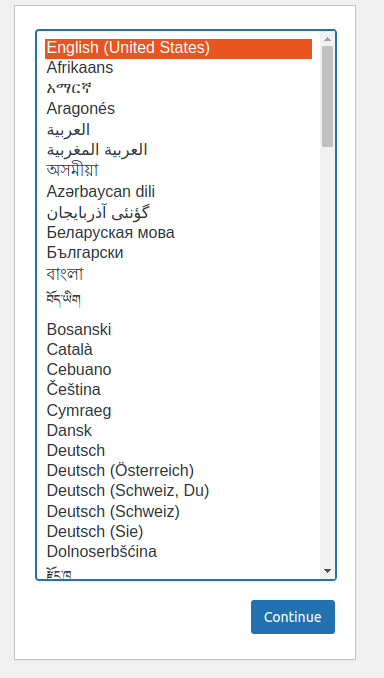


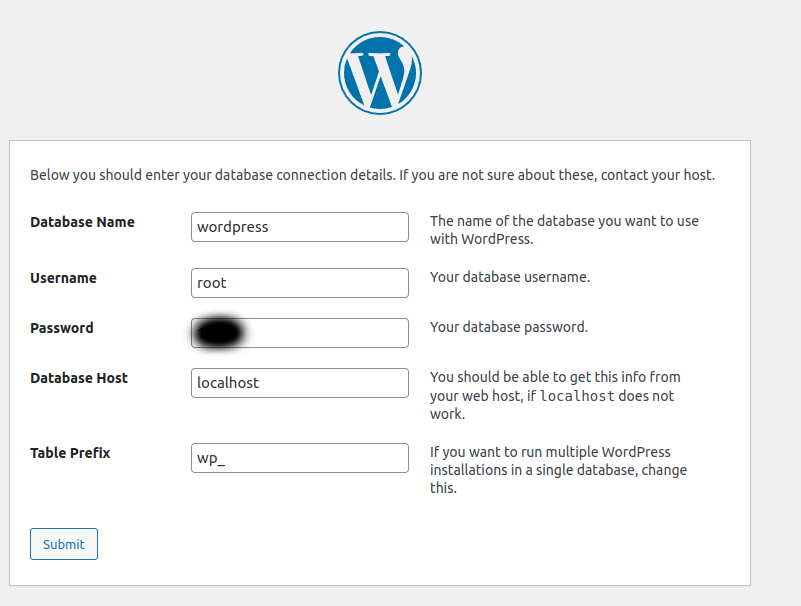
5.Go to <http://localhost/phpmyadmin> in your browser and create database.

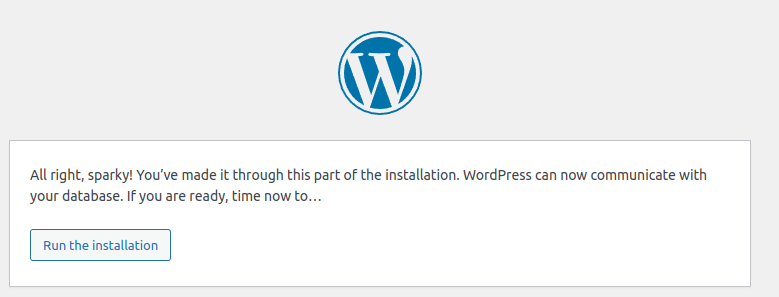




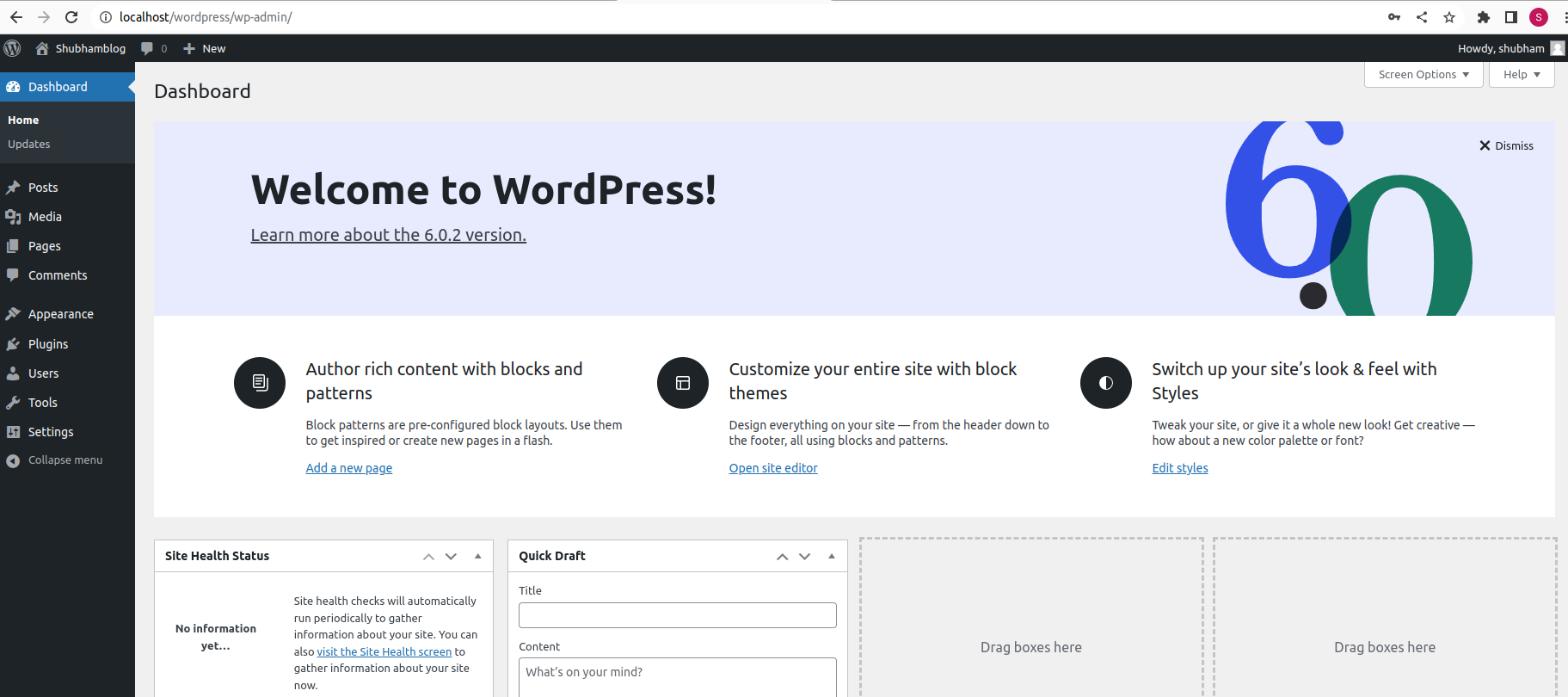
6. Last step go to <http://localhost/wordpress> and fill the fields as shown below:





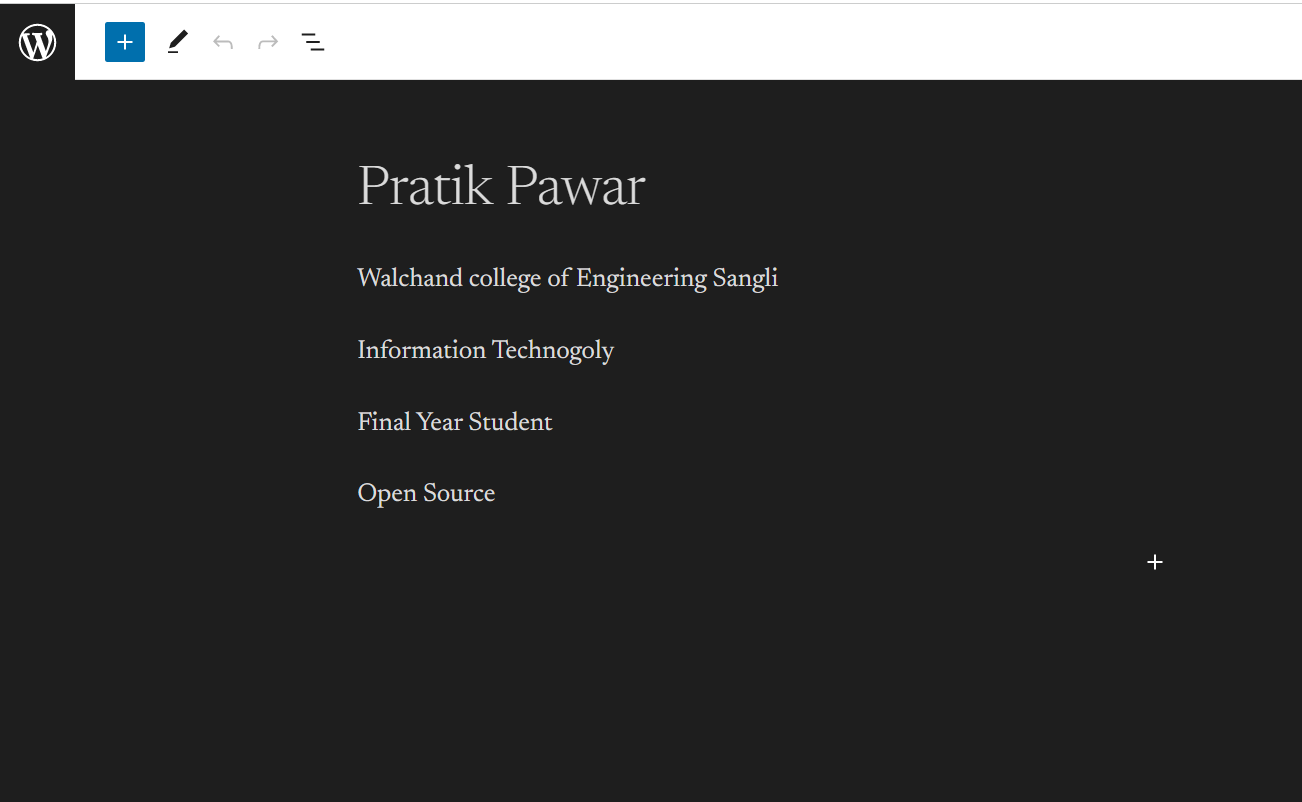


and submit this logins to wordpress and screen appears as shown below:



Now just create simple blog by going on pages options visible on left as shown above and host this to public/ private / protected as per your choice.

Simple demo



------------------------------------------------------------------------------------------

Blog published:)

**Assignment No: 6**

**Objective: To comprehend the open source software development**

**Distro: Ubuntu**

**Name of community:**

1. Ubuntu forums

2. Ubuntu Community Hub

3. Ask Ubuntu

4. Ubuntu LoCo

**Website:**

1. https://ubuntu.com/

**Mailing List:**

|  |  |
| --- | --- |
|  |  |
| Ubuntu user technical support: | <https://lists.ubuntu.com/mailman/listinfo/ubuntu-users> |
|  |  |
| Ubuntu news: | <https://lists.ubuntu.com/mailman/listinfo/ubuntu-news> |
|  |  |
| Ubuntu Security Announcement: | <https://lists.ubuntu.com/mailman/listinfo/ubuntu-security-announce> |
|  |  |
| Ubuntu Cloud Announcement: | <https://lists.ubuntu.com/mailman/listinfo/Ubuntu-cloud-announce> |
|  |  |
| Ubuntu Bugs: | <http://lists.ubuntu.com/mailman/listinfo/ubuntu-bugs> |
|  |  |

**OTHER:**

|  |  |
| --- | --- |
| WIKI**:** | <https://wiki.ubuntu.com/> |
|  |  |
| UBUNTU VERSION CONTROL: | <https://code.launchpad.net/ubuntu> |
|  |  |
| UBUNTU BUG TRACKING: | <https://bugs.launchpad.net/ubuntu> |
|  |  |
| UBUNTU LOCALIZATION: | <https://translations.launchpad.net/ubuntu> |
|  |  |
| UBUNTU DOCUMENTATION: | <https://help.ubuntu.com/community> |
|  |  |

**Assignment No: 8**

**8.1 Objective : To create package building process in Linux**

**8.2 Outcomes : (PO: b, I, j, k, l)**

8.3 Instructions for the Assignment: Students have to study RPM or DEB package building for their C, C++ or JAVA Codes.

They must build an rpm or debian package and install it through package manager such as YUM or APT-GET

8.4 Theory :

1. A Debian "package", or a Debian archive file, contains the executable files, libraries, and documentation associated with a particular suite of program or set of related programs. Normally, a Debian archive file has a filename that ends in .deb.

2. Debian control file: This file contains different fields which help to give the name of the package and so on.

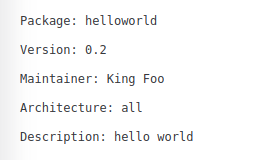
8.5 Steps :

1. Using terminal create a directory for eg sample

2. Inside this sample folder create another folder , i named

it DEBIAN

3. Inside this create control file and write the control code



Above are mandatory fields of control field .

4. In sample folder create usr folder

5. In usr foler create bin folder

6. In bin write a code of any programming language lets

say java

7. Save program compile it and run it check whether the

output is as expected.

8. Later come to home directory ie outside of sample and

write the command in terminal as

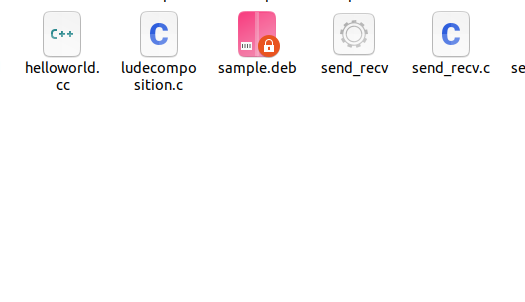
sudo apt-get install build-essential

NOTE: build-essential helps to create debian package.

And thats it debian package is create named helloworld as it is the name of package you have given in control file

Screenshots of above implementation





Conclusion :

1. We learned about Debian package creation

References :

1. <https://www.intracto.com/en-be/blog/creating-debianubuntu-deb-packages>

2. <https://www.youtube.com/watch?v=zyRWGzsFdq0>

3. <https://www.debian.org/doc/manuals/debian-faq/pkg-basics.en.html>

**Assignment 9**

# Install and demonstrate of various Serverbased services and theirUses.

# Objective : -

To Know server installation and configuration on linux platform

# Outcome : -

(PO: b, I, j, k, l)

# Instruction for the Assignment

* 1. **Theory (Functions of tool /How to Use /Drawbacks)**

1. **FTP :**

**FTP (file transfer protocol)** is an internet protocol that is used for transferring files between client and server over the internet or a computer network. It is similar to other internet protocols like SMTP which is used for emails and HTTP which is used for websites. **FTP server** enables the functionality of transferring files between server and client. A **client** connects to the **server** with credentials and depending upon the permissions it has, it can either read files or upload files to the server as well. In this article, we will see how to set up an FTP server, configure user permissions, configure a firewall and finally encrypt our FTP traffic with SSL.

### How it works :

FTP server facilitates the transfer of files between client and server. You can either upload a file to a server or download a file from the server. A client makes two types of connections with the server, one for giving commands and one for transferring data. The client issues the command to the FTP server on port 21, which is the command port for FTP. For transferring data, a ata port is used. There are two types of connection modes for transferring data:

* **Active mode:** In Active mode, the client opens a port and waits for the server to connect to it to transfer data. The server uses its port 20 to connect to the client for data transfer. Active mode is not set by default in most of the FTP clients because most firewalls block the connections which are initiated from outside, in this case, the connection initiated by our FTP server. To use this, you have to configure your firewall.
* **Passive mode:** In this, when a client requests a file from the server, the server opens a random port and tells the client to connect to that port. In this case, the connections are initiated by the client and this also solves the firewall issues. Most of the FTP clients use passive mode by default.

# Steps :

**Step 1: Install FTP server**

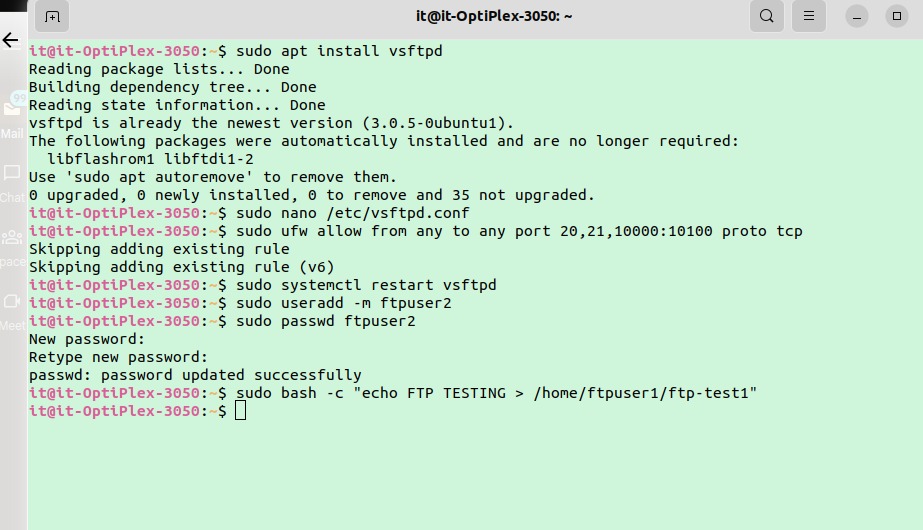
There are many FTP servers to choose from like ProFTPD, vsftpd, etc. We will be using vsftpd.

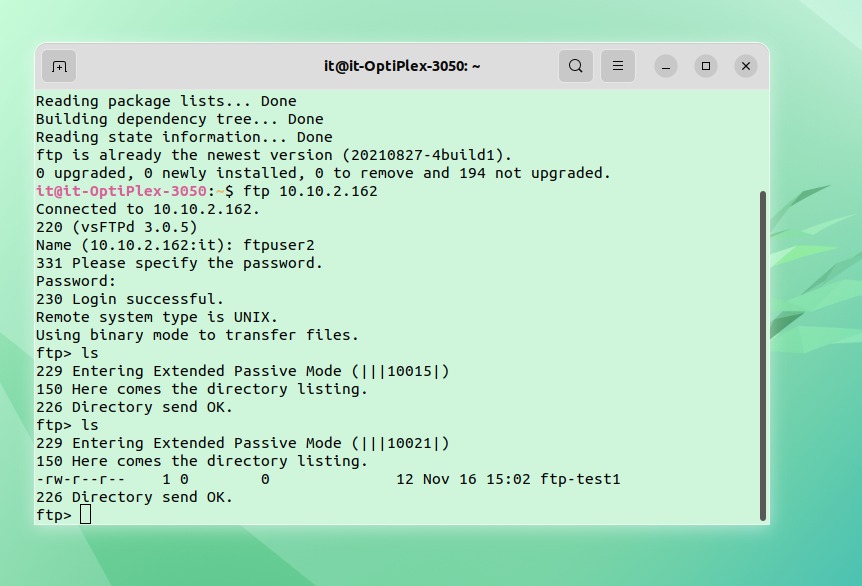
**Step 2:** Configure Firewall

**Step 3:** Configure Users

**Step 4:** Create the FTP folder and set permissions

**Step 5:** Configure and secure vsftpd **Step 6:** Securing vsftpd with SSL/TLS **Step 7:** Connecting to our FTP server



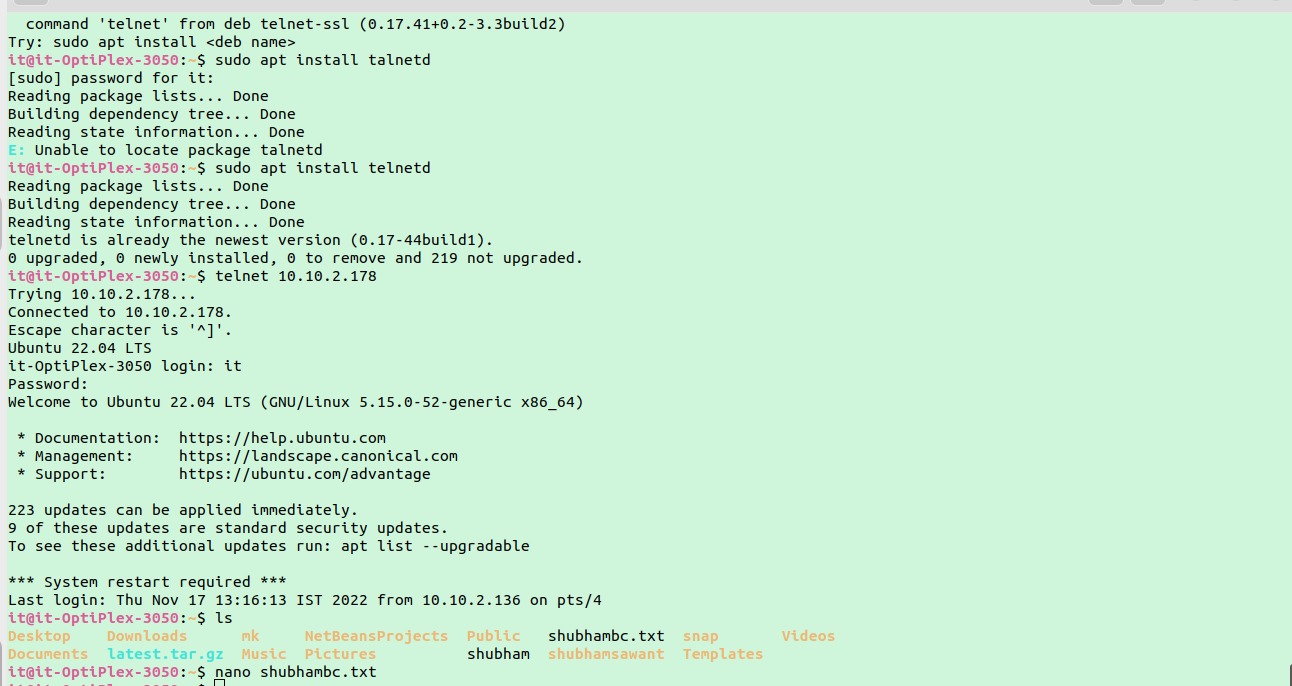


### TELNET :

**TELNET** stands for **TE**rmina**L NET**work. It is a type of protocol that enables one computer to connect to local computer. It is a used as a standard **TCP/IP protocol** for virtual terminal service which is given by.**ISO**. Computer which starts connection known as the **local computer**. Computer which is being connected to i.e. which accepts the connection known as **remote computer**. When the connection is established between local and remote computer. During telnet operation whatever that is being performed on the remote computer will be displayed by local computer. Telnet operates on client/server principle. Local computer uses telnet client program and the remote computers uses telnet server program.

**Step 1:** To install the telnet, execute the below command sudo apt install telnetd-y

**Step 2 :** verify the installation and whether the service is running or not, execute the

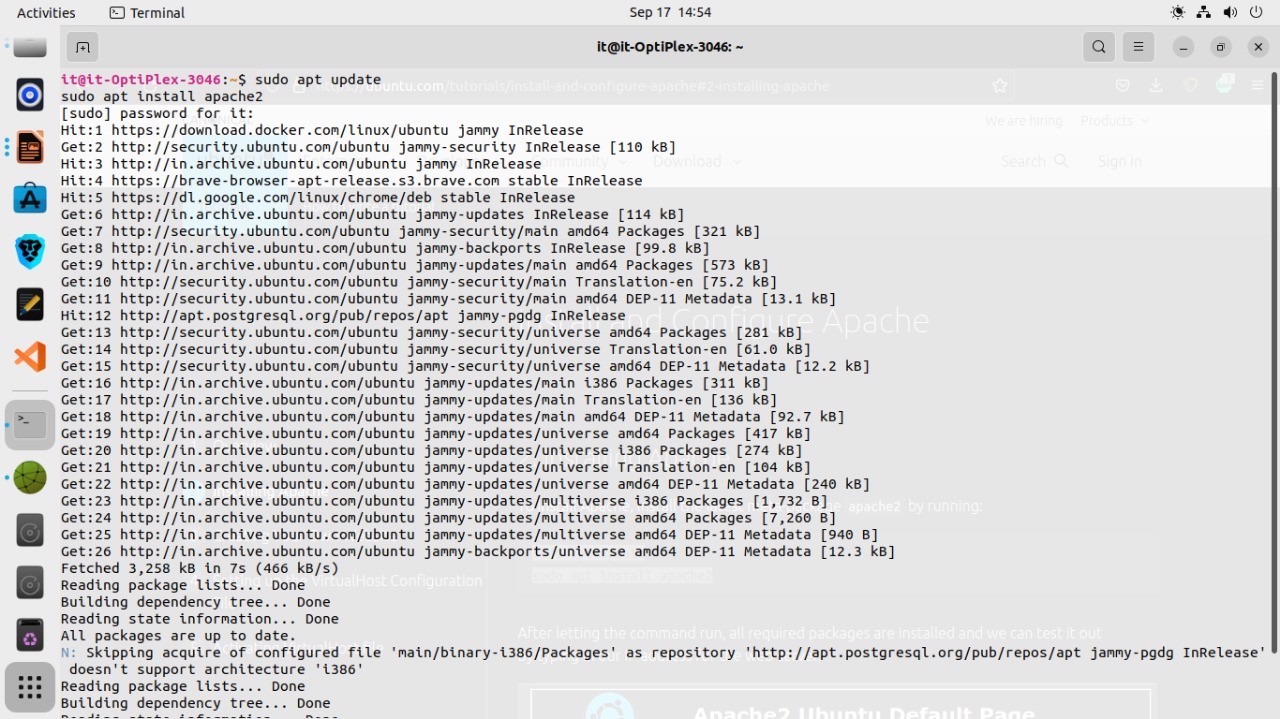


### HTTP :

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software.

**Step 1 :**Installing Apache

**Step 2 :** Adjusting the Firewall

**Step 3 :** Checking your Web Server

**Conclusion:**

**Successfully created Web server,Telnet Server,Ftp Server and use each .**

**Assignment No: 10**

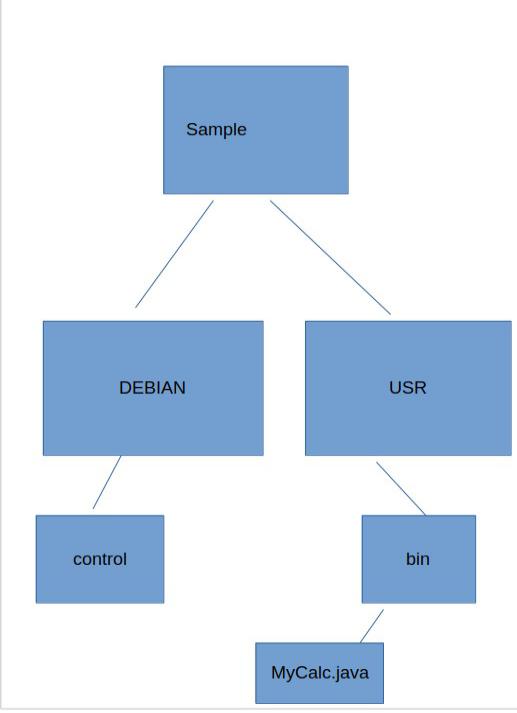
**Assignment Name: Development of new OSS or contribution to existing OSS (calculator )**

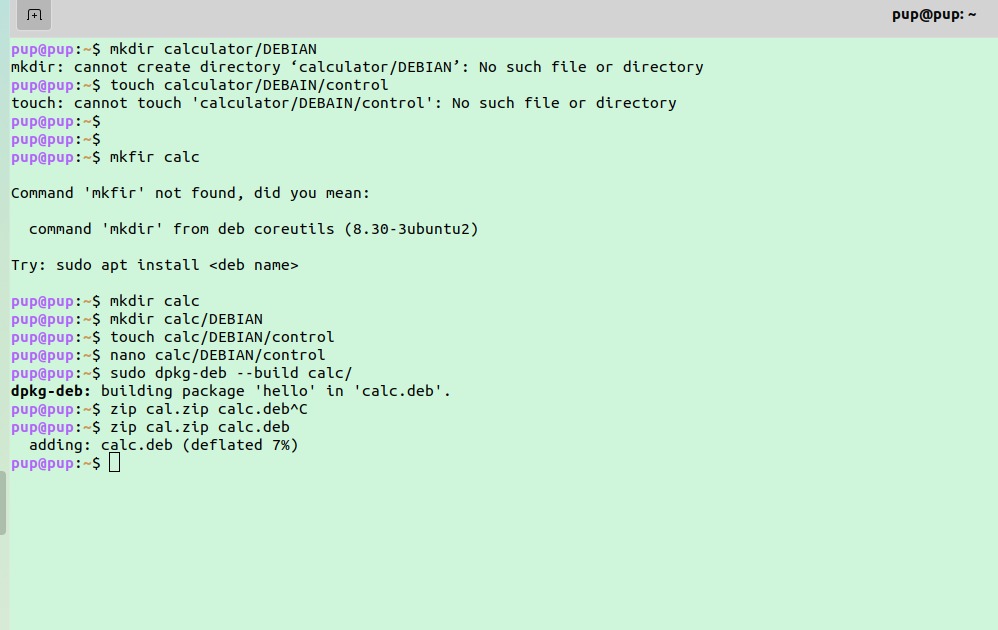
Language: Java

Git code: <https://github.com/PratikUmeshPawar/calculator.java.git>

Packaging building: Debian

Steps:

* We created folders in such hierarchy for creating debian packa



Control file

Package: Calc

Version: 0.2

Maintainer: Pratik

Architecture: all

Description: Calc

**Calc.java is uploaded on git. Link is provided above**

**References:**

**https://dev.to/rohitk570/creating-a-calculator-using-java-awt-16ll**

**Assignment No: 11**

**Docker container: An OSS virtualization command practice, use and understanding**

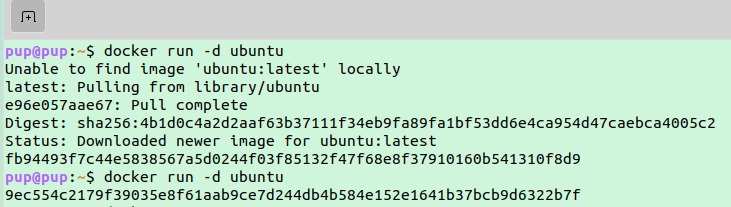
Tools used: Docker

Image used: Nginx

**A. SHOW LINUX DISTROS :**

1. Linux distros like ubuntu are available on Docker hub

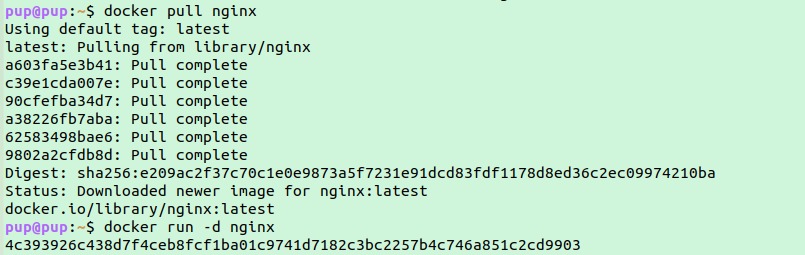
2. docker run -d ubuntu , creates a container



**B. Create image/Container of FOSS tool and upload on Docker Hub**.

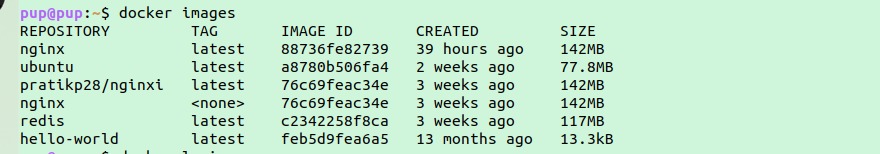
1. For this install docker engine on ubuntu.After installing

follow next step.

1. To pull image from docker hub use below.
2. To turn this image into container run the downloaded image

Now nginx container is running

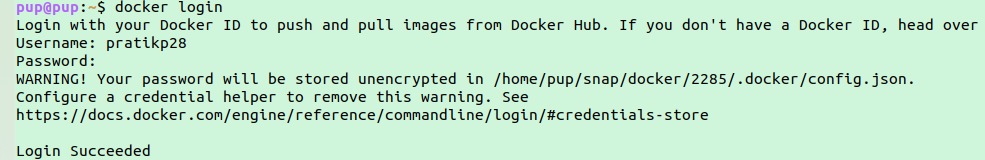
4. Now we will upload this image on docker hub. For this below commands need to executed



this show availables images in your local machine

1. Now signup at dockerhub if not yet created account

2. After creating account



3. Now we need to retag image that we are going to push

to dockerhub

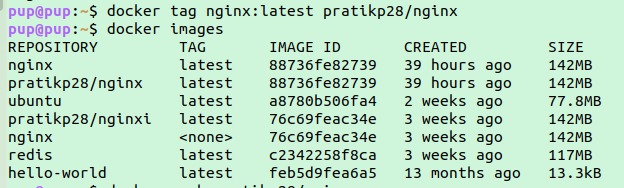
above command is

docker tag <image name>:<TAG > <dockerusername>/<image name>

Retagging simply means renaming your image

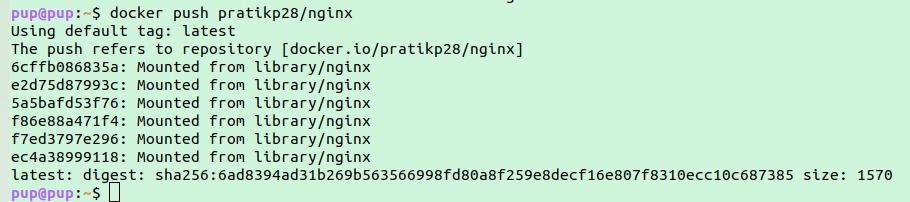
4. See all docker images and check whether image is

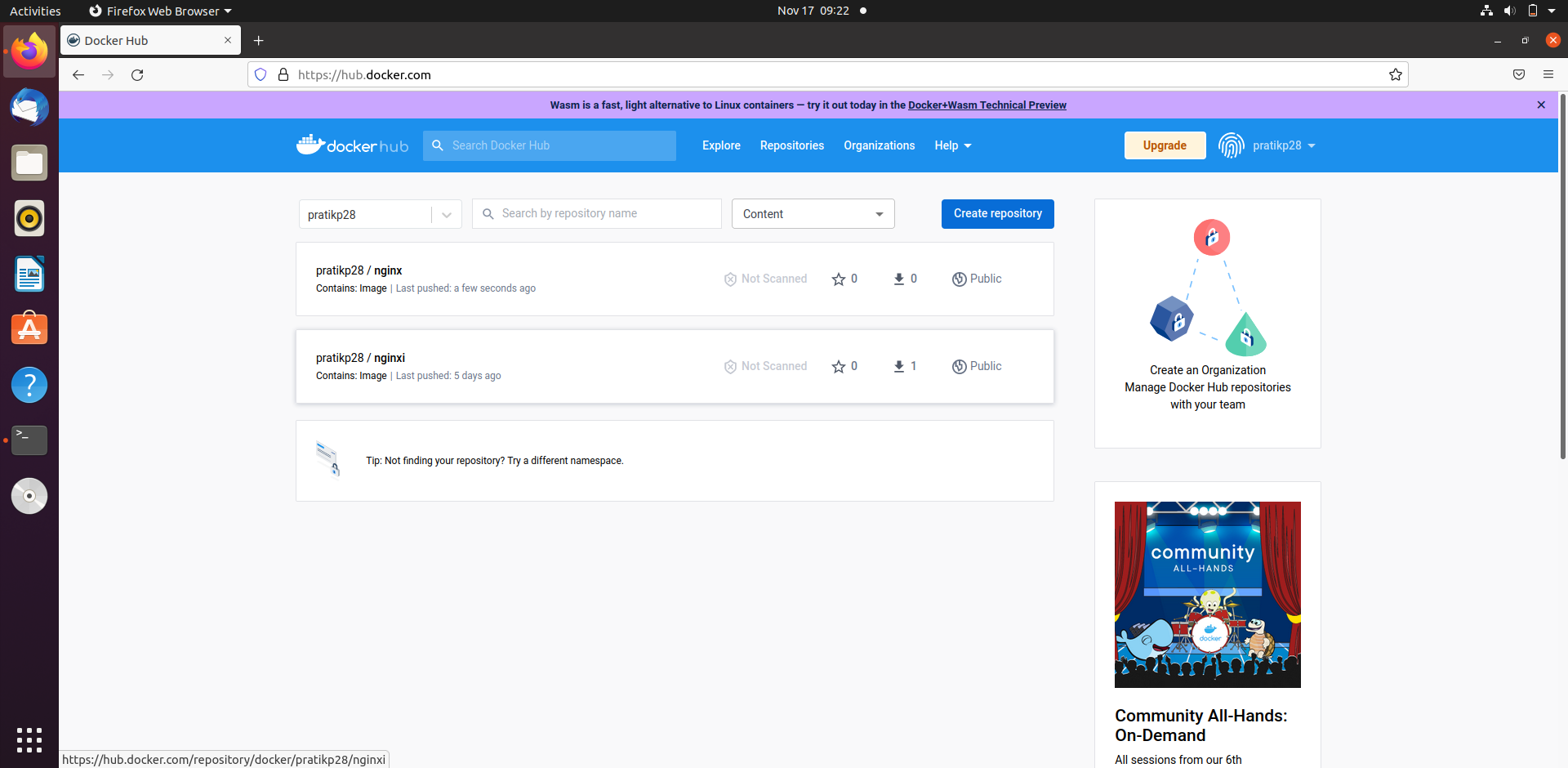
retagged or not.



Yes it is retagged

5. Push to dockerhub using below command





**C. IPC between two containers:**

1. Create one separate network named tulip-net



2. Create a container nginx of named tulipngnix in the network created above

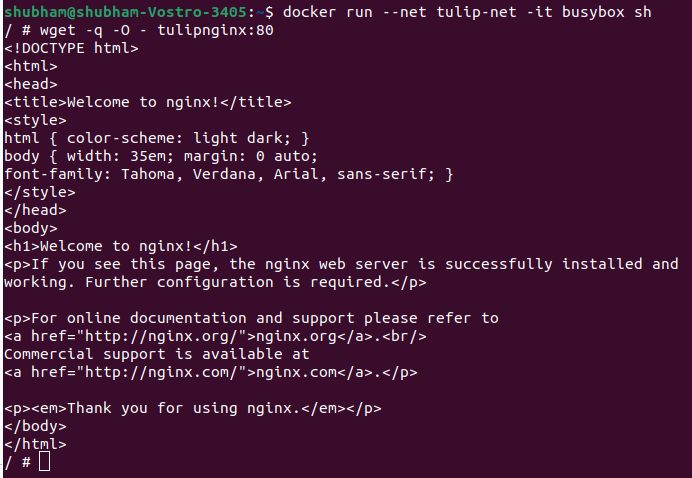


3. Go in the interactive mode of busybox container after creating it and it

should be also in the network created above and type the sh command as

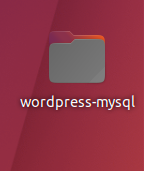
shown below

Then you will be accessed nginx container as shown below due to wget command used



**D. With the help of Docker-compose deploy the word-press and MySQL container and access the front end of word-press:**

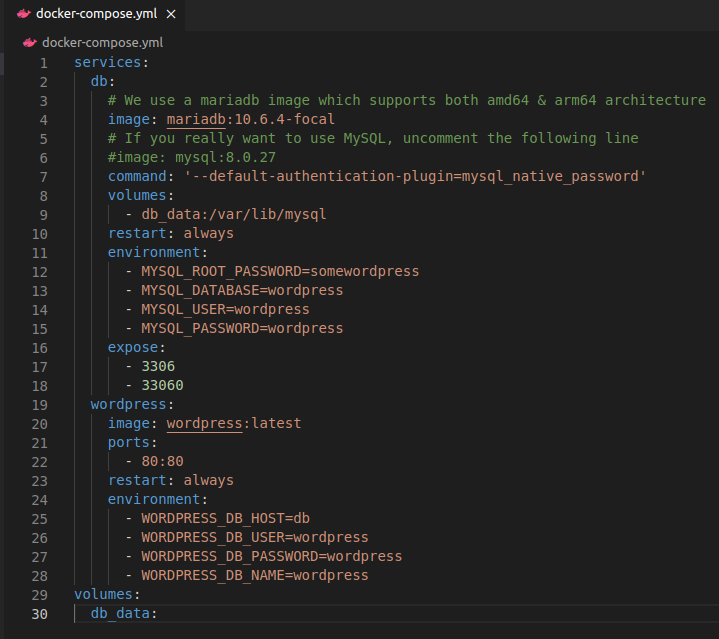
1. On the desktop create a folder.



2. Open VSC code with this wordpress-mysql folder

3. Create docker-wordpress.yaml file

4. Code it



this yaml or yml file pulls mariadb, mysql image from dockerhub. All the data of this databases is stored at *var*/lib/mysql folder and some security attributes are set.

Wordpress image is also pulled and environment variables are set.

To configure the container of this code. We need to build the project to do so, use below commands.

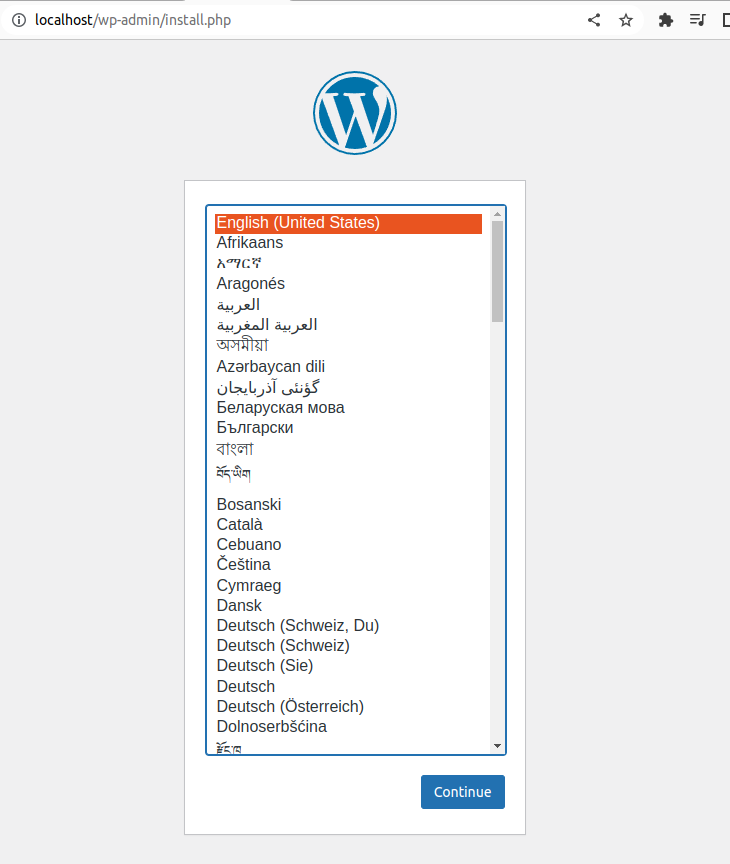


Container is now running..

We will access the running container from browser by typing

[http://localhost](http://localhost/)

You will see the front page of wordpress as follows:



By clicking on continue you can move ahead.

references

1. <https://www.geeksforgeeks.org/load-balancing-flask-application-using-nginx-and-docker/>

2. https://www.youtube.com/watch?v=wrMJoKpK2mk

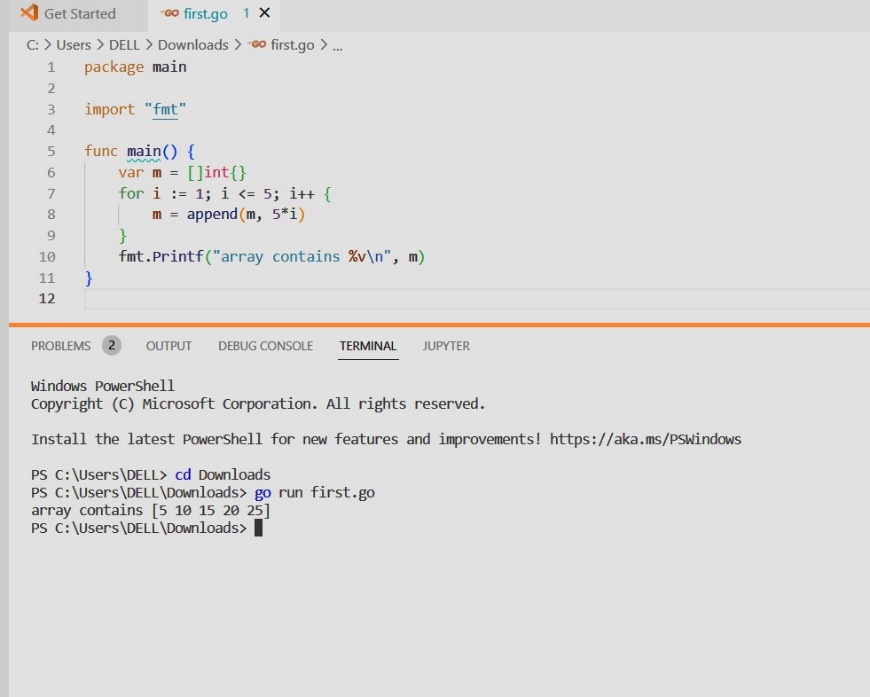
Assignment No: 14

Open source programming tutorial language GO

Name: Pratik U. Pawar

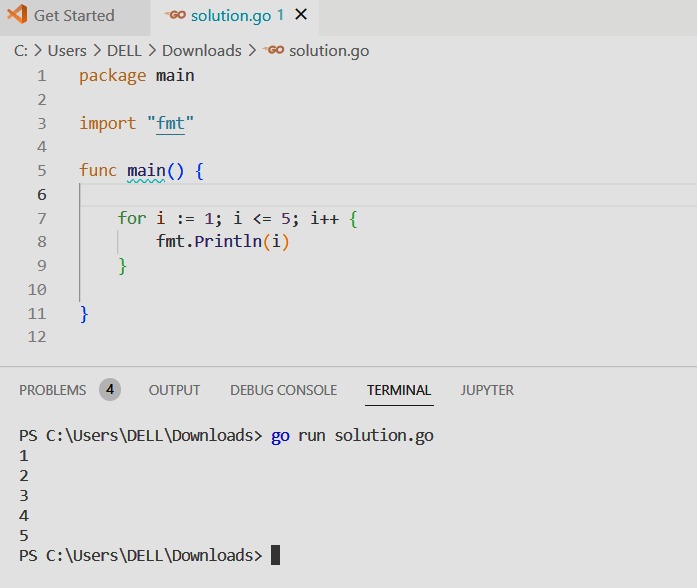
PRN : 2019bteit00006

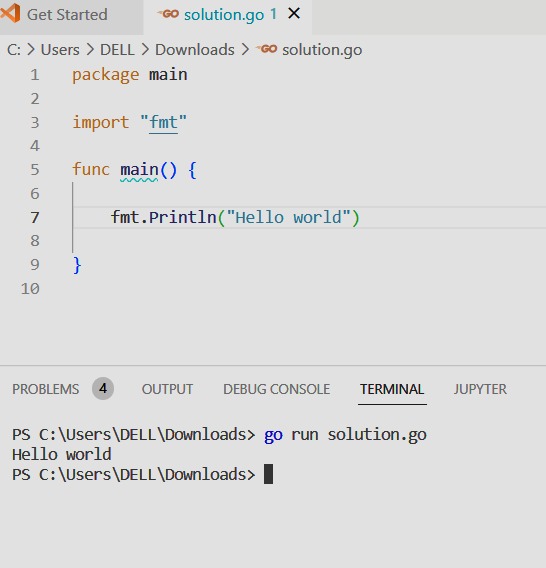
A. Three GO programs:

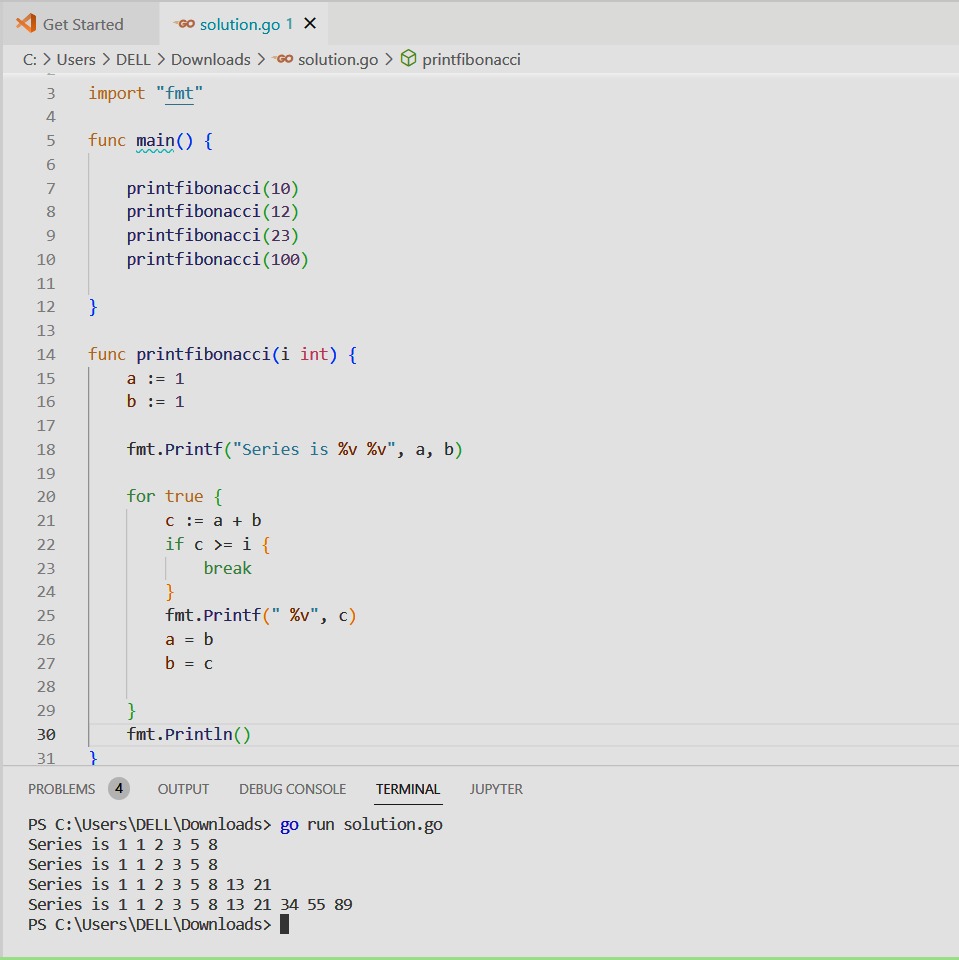
1.

2.

3.







4.

B. Compare Go language with C Functional and procedural based language

1Go is 1.procedural based language

2. It is neither functional nor object based language

3. Go doesnt support inheritance.

C. Features of Go language

1. Easy to learn

2. Open Source in nature

3. Language is fast

4. Powerful standard library

5. Static typing (errors are easily seen while typing)

6. Its binaries can be executed on any OS

D.Commands and compilers

1. To compile a file:

go run filename

2. Debuggers: VSC, Delve

E. Applications of Go language

1. It provides container services like Docker

2. Kubernetes

3. Big data

4. Machine learning

5. Network programming