

Assignment-1

- Q. 1] Explore how a recent open-source project is being used in educational settings. Discuss its benefits, challenges and ~~also~~ potential for enhancing learning experiences.

Ans Project: Jupyter Notebook (Project Jupyter)

> Usage in Education:

- i) Widely used in universities and schools for teaching programming, data science and machine learning.
- ii) Enables interactive coding, visualization, and documentation in one environment.

> Benefits:

- i) Interactive learning (code + output + explanation together)
- ii) Supports multiple languages (Python, R, Julia)
- iii) Encourages collaborative learning through JupyterHub and cloud integration.
- iv) Free and open-source → accessible to students globally

> Challenges:

- i) Requires technical setup (servers, dependencies)
- ii) Performance issues with large datasets.
- iii) Security concerns when running untrusted code.

> Potential for Enhancement:

- i) Integration with AI ~~tutors~~ tutors for personalized feedback
- ii) Wider adoption in non-technical courses (mathematics, economics)
- iii) Cloud-based platforms to reduce infrastructure barriers.

- Q. 2] Your system crashes while testing an open-source software. Demonstrate how you would report this issue on GitHub. (Mention at least two things you must include in the report)

Ans Example GitHub Issue Report

Title: System crash on Ubuntu 22.04 when running RosBE v2.1.2 Installer.

Description: While testing RosBE (ReactOS Build Environment) version 2.1.2 on Ubuntu 22.04, the system crashes shortly after running the installer script. The crash requires a reboot.

Steps to reproduce:

i) Download RosBE v2.1.2 from the releases pages.

```
# wget https://github.com/reactos/RosBE/releases/download/2.1.2/RosBE-Unix-2.1.2.tar.gz
```

```
# tar -xvzf RosBE-Unix-2.1.2.tar.gz
```

```
# cd RosBE-Unix-2.1.2
```

ii) Run the installer script:

```
./RosBE-Builder.sh
```

iii) Accept the installation prompts.

iv) System freezes and requires reboot.

Expected Behaviour:

The build environment should install successfully without freezing the system.

Actual Behaviour:

The script begin dependency checks, but the entire system crashes during installation.

Environment:

- OS: Ubuntu 22.04 LTS
- Kernel: 5.15.0-88-generic
- RosBE version: 2.1.2
- Install method: Tarball (RosBE-Unix-2.1.2.tar.gz)
- Privileges: Tested as normal user (not root)

Logs & Screenshot:

Terminal output before crash:

User is not "root"

checking for the needed tools...

checking for as... OK

checking for bison... OK

checking for flex... OK

checking for gcc... OK

checking for g++... OK

checking for grep... OK

checking for makeinfo... OK

checking for python... OK

checking for GNU make... OK

checking for zlib... OK

(Screenshots are added below)



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2025-26

Sem: VII

Class/Branch: BEIT-A/B/C

Subject: ROSPL

```
nagios@worker01:~/RosBE/RosBE-Unix-2.1.2$ ./RosBE-Builder.sh
*****
* ReactOS Build Environment for Unix-based Operating Systems *
* Builder Tool for the Base package                          *
* by Colin Flinck <colin@reactos.org>                        *
*                                                            *
* Version 2.1.2                                              *
*****

This script compiles and installs a complete Build Environment for building
ReactOS.

User is not "root"
The default installation path for the Build Environment is "/usr/local/RosBE".
Therefore it is recommended to run this script as the "root" user. Otherwise you probably cannot create the necessary directories.
An alternative is to specify another installation directory at the installation process.
Do you really want to continue? (yes/no)
[no] yes

Checking for the needed tools...
Checking for as... OK
Checking for bison... OK
Checking for flex... OK
Checking for gcc... OK
Checking for g++... OK
Checking for grep... OK
Checking for makeInfo... OK
Checking for python... OK
Checking for GNU Make... OK
Checking for zlib... OK
```

```
Installation Directory
In which directory do you want to install it?
Enter the path to the directory here or simply press ENTER to install it into the default directory.
[/usr/local/RosBE] yes

The directory "yes" does not exist. The installation script will create it for you.

Ready to start
Ready to build and install the ReactOS Build Environment.
Press Return to continue or Ctrl+C to exit.
^C
nagios@worker01:~/RosBE/RosBE-Unix-2.1.2$ cd ~/reactos-0.4.15
nagios@worker01:~/reactos-0.4.15$ mkdir build && cd build
```

```
nagios@worker01:~/reactos-0.4.15/build$ cmake -G Ninja ..
CMake Error at CMakeLists.txt:2 (cmake_minimum_required):
  CMake 3.17.0 or higher is required. You are running version 3.10.2

-- Configuring incomplete, errors occurred!
```


Q.3] Consider the following scenarios and answers the questions accordingly;

a) A startup is developing a web application and wants to allow free usage but also prevent others from making it closed-source. Which open-source license should they choose and why?

b) Your team is releasing a machine learning library and wants maximum adoption, including ~~big~~ commercial companies. Between MIT, Apache and GPL which license would be most suitable?

c) A developer is contributing to an open-source project but also wants to use parts of it in proprietary software later. Which type of license would allow this flexibility?

d) A company wants to release a tool for community use, but they also want that any modification made by others must be shared back with the community. Which license enforces this condition?

Ans. a) ~~Free Usage + prevents closed-source~~ →

- License: GNU General Public License (GPL)
- Reason: Enforces "copyleft", meaning any modified versions must remain open-source.

b) ML library for maximum adoption (including commercial use) →

- License: Apache 2.0
- Reason: Permissive, allows commercial usage, includes patent protection, widely accepted in industry.

c) Developer wants to use in proprietary software later →

- License: MIT or Apache 2.0
- Reason: Both allow integration into proprietary projects without forcing source code disclosure.

d) Company wants modifications shared back →

- License: GPL or AGPL
- Reason: These require that derivative works must also be released as open-source.

Q.4) A government department is considering ReactOS for secure systems since it is open-source and not controlled by Microsoft. Analyze the risks and benefits of adopting React OS in such a mission-critical environment

Ans. > Benefits:

- i) Open-source: transparent code → potential to audit for security
- ii) Windows-compatible: can run legacy applications without Microsoft dependency
- iii) Cost-effective: no license fees.
- iv) Independence from proprietary vendors.

> Risks:

- i) Maturity: ReactOS is still in alpha stage; unstable for mission-critical tasks.
- ii) Security: Lacks the robust patching ecosystem of Linux/Windows.
- iii) Compatibility: Not all windows apps/drivers run reliably
- iv) Support: Limited community support compared to established OSes.

> Conclusion: While ReactOS offers independence and transparency, its immaturity and security concerns make it unsuitable for mission-critical government system at present.

Q.5) Demonstrate the procedure to contribute a bug fix or feature suggestion to the Cinnamon open-source OS. Outline the practical steps (from creating an issue on GitHub, providing system details, to submitting logs/screenshots) & explain how these steps help the developer community resolve the issue efficiently.

Ans Procedure:

1) Identify Issue / Feature →

Test Cinnamon on your system → note bug or improvement

2) Create GitHub issue: Go to Cinnamon's GitHub repo → Click "New Issue" → Provide title + clear description.

3) Include system details:

- OS version (Linux Mint 21.1, Ubuntu 22.04, etc.)
- Cinnamon version
- Hardware details (GPU, CPU, RAM)

4) Provide Logs & Screenshots:

- Attach system logs (~/.xsession-errors, journalctl).
- Include screenshots or screen recording showing the problem.

5) Suggest Fix / Feature Idea:

- Explain expected vs actual behavior.
- If you can code, fork the repo → create branch → fix issue → submit a pull request (PR).

6) Follow Up:

- Respond to maintainer queries.
- Test patches if requested.

This helps the developer community in following ways:

i) Structured issue reports = faster debugging.

ii) Logs/screenshots = better understanding.

iii) Pull requests = reduce developer workload.

iv) Collaboration improves quality and adoption of Cinnamon.