OPEN SOURCE TECHNOLOGIES (01CE0618) Lab Manual

Name: GAE ABLE JEAN
Enrolment No: 92301703215
Class: EC5
Batch: **A**

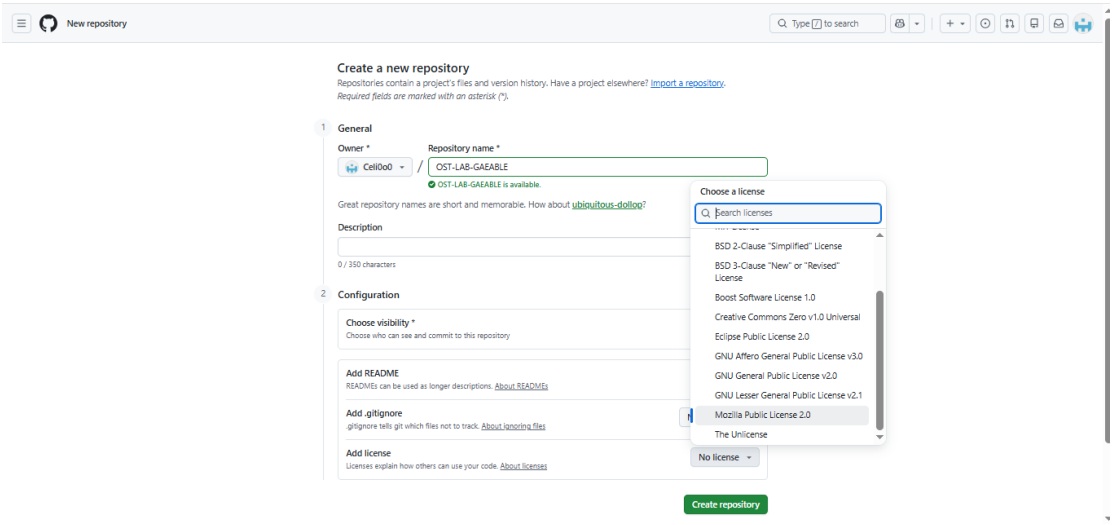
INDEX

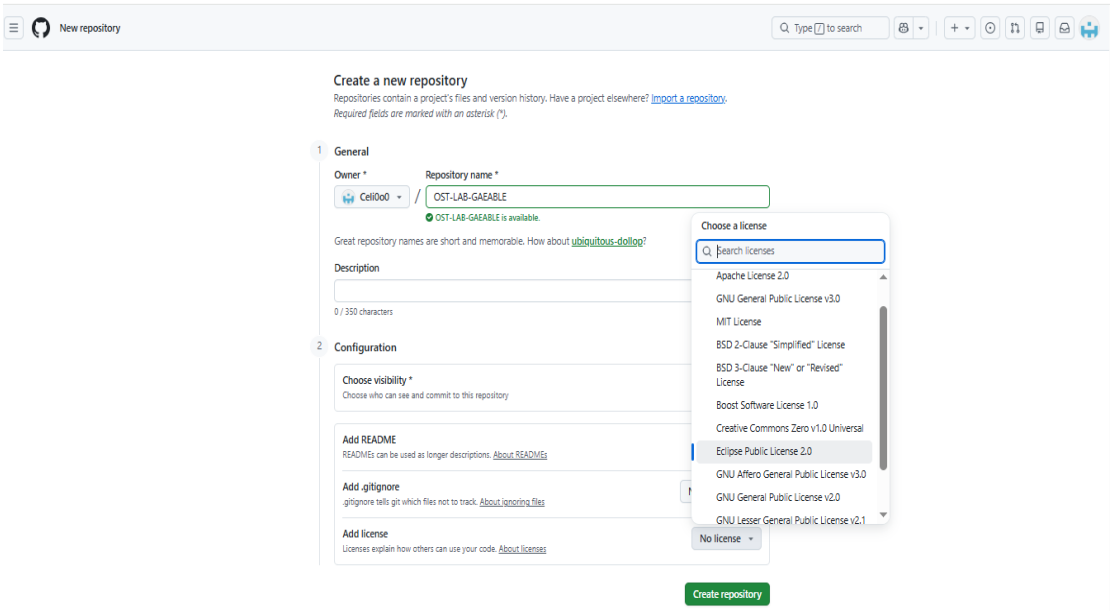
Lab	Program	Date	Marks	Signature
1.	Explore GitHub/GitLab for open-source projects with different licenses			
2.	Setup Git and explore commands related to Version Control System (VCS)			
3.	Create a GitHub/GitLab repository and upload sample code			
4.	Use npm / yarn / pip to install and manage packages			
5.	Deploy a simple application using Nginx / Apache			
6.	Setup Nginx to handle proxy requests and load balancing			
7.	Develop a Laravel / Django based web application			
8.	Use pytest to test a Python application			
9.	Use Selenium to create browser-based tests			
10.	Use Postman to test a sample API			
11.	Use OWASP ZAP to check security integrity			
12.	Modify any open-source desktop application			
	Contribute to any web-based open-source project			
	Use source code of an open-source web application and deploy it locally			

Experiment 1

AIM: Explore GitHub/GitLab for open-source projects with different licenses

List of GitHub Licenses





1. Licenses and Short Description Table

License	Brief Description
Apache License 2.0	Permissive license allowing use, modification, and distribution (including commercial). Requires preservation of notices and provides explicit patent protection.
MIT License	Very permissive and simple license. Allows almost unrestricted use, modification, and distribution with only attribution required.
GNU General Public License v3.0 (GPL v3)	Strong copyleft license. Any modified or derived work must be released under the same license. Includes patent protection.
GNU General Public License v2.0 (GPL v2)	Older strong copyleft license. Derived works must also be open source under GPL v2. No explicit patent protection.
GNU Lesser General Public License v2.1 (LGPL v2.1)	Weak copyleft license mainly for libraries. Allows linking with proprietary software; modifications to the library must remain open source.
GNU Affero General Public License v3.0 (AGPL v3)	Strong copyleft license for network/server software. Requires sharing source code even when used over a network (e.g., web apps).
BSD 2-Clause "Simplified" License	Permissive license with minimal restrictions. Allows reuse and redistribution with attribution.
BSD 3-Clause "New" or "Revised" License	Similar to BSD 2-Clause but prevents using the author's name for promotion without permission.
Boost Software License 1.0	Very permissive license, mainly for libraries. Allows commercial and proprietary use with minimal conditions.
Creative Commons Zero v1.0 Universal (CC0)	Places the work in the public domain. No restrictions or attribution required.
Eclipse Public License 2.0 (EPL 2.0)	Weak copyleft license. Modifications to EPL-covered files must be shared; can be combined with proprietary code.
Mozilla Public License 2.0 (MPL 2.0)	File-level copyleft license. Modified files must be open source, but the entire project does not need to be.
The Unlicense	Public-domain–like license. Allows anyone to do anything with the code without restrictions.
No License	Code is fully copyrighted by default. Others cannot legally use, modify, or distribute it.

2. Licenses Comparison Table

License	Type	Commercial Use	Modification Allowed	Redistribution	Must Disclose Source?	Attribution Required	Patent Protection	Best Use Case
MIT	Permissive	Yes	Yes	Yes	No	Yes	No	College projects, simple OSS
Apache 2.0	Permissive	Yes	Yes	Yes	No	Yes	Yes	Enterprise & commercial apps
BSD 2-Clause	Permissive	Yes	Yes	Yes	No	Yes	No	Academic & research projects
BSD 3-Clause	Permissive	Yes	Yes	Yes	No	Yes	No	Open projects with author protection
Boost 1.0	Permissive	Yes	Yes	Yes	No	Yes	No	Libraries & frameworks
GPL v2	Copyleft (Strong)	Yes	Yes	Yes	Yes	Yes	No	Traditional open-source software
GPL v3	Copyleft (Strong)	Yes	Yes	Yes	Yes	Yes	Yes	Freedom-focused OSS
LGPL v2.1	Copyleft (Weak)	Yes	Yes	Yes	Partial	Yes	No	Open-source libraries
AGPL v3	Copyleft (Very Strong)	Yes	Yes	Yes	Yes (Network use)	Yes	Yes	Web & SaaS apps
MPL 2.0	Copyleft (File-level)	Yes	Yes	Yes	File-level only	Yes	No	Mixed open/closed projects
EPL 2.0	Copyleft (Weak)	Yes	Yes	Yes	Partial	Yes	No	Enterprise & Java projects
CC0	Public Domain	Yes	Yes	Yes	No	No	No	Data, research, no ownership
Unlicense	Public Domain	Yes	Yes	Yes	No	No	No	Personal or free-use projects
No License	Copyrighted	No	No	No	No	No	No	Private code

3. List of GitHub Alternatives

Platform	Best For	Highlights
GitLab	DevOps & CI/CD	Full DevOps lifecycle, self-hosted or cloud, strong CI/CD integration
Bitbucket	Team collaboration	Integrates with Jira, free for small teams, supports Git & Mercurial
SourceForge	Open-source projects	Longstanding platform, project visibility, community support
AWS CodeCommit	Cloud-native teams	Fully managed by AWS, integrates with other AWS services
Gitea	Lightweight self-hosting	Open-source, simple setup, low resource usage
Gogs	Minimalist hosting	Extremely lightweight, easy to deploy on personal servers
Gitbucket	Java-based hosting	GitHub-like interface, plugins available
TaraVault	Free private repos	Unlimited repositories, free trial, secure hosting
Launchpad	Ubuntu ecosystem	Focused on open-source collaboration, bug tracking
Azure DevOps Repos	Enterprise workflows	Deep integration with Microsoft ecosystem, CI/CD pipelines

4. GitHub vs GitLab Table

Feature	GitHub	GitLab
Ownership	Owned by Microsoft	Independent company (GitLab Inc.)
Primary Focus	Code hosting & collaboration	Complete DevOps lifecycle (code + CI/CD + deployment)
CI/CD Integration	Requires external tools (e.g., GitHub Actions, Jenkins)	Built-in, fully integrated CI/CD system
Open Source	Proprietary platform	Open-source Community Edition available
Project Management	Basic issue tracking, Kanban boards	Advanced project management tools (issues, epics, milestones)
Popularity	Larger community, widely used for open-source projects	Smaller but growing community, strong in enterprise DevOps
Pricing	Free tier + paid plans	Free tier + paid plans, more features in free tier
Security & Compliance	Strong security features, integrates with enterprise tools	Advanced compliance, vulnerability management, DevSecOps focus
Integration Ecosystem	Extensive marketplace with third-party apps	Fewer integrations, but strong built-in features
Best For	Open-source collaboration, community-driven projects	Enterprises needing end-to-end DevOps platform

5. Open-Source vs Proprietary vs Freeware

Category	Definition	Key Features	Examples
Open-Source	Software whose source code is publicly available for anyone to view, modify, and distribute	- Free to use and modify - Community-driven development - Transparency and flexibility - Often licensed under GPL, MIT, Apache	Linux, Firefox, Apache, Android
Proprietary	Software owned by a company/individual, with restricted access to source code	- Source code hidden - Licensed with usage restrictions - Updates and support controlled by vendor - Often requires paid license	Microsoft Windows, Adobe Photoshop, macOS
Freeware	Software distributed free of charge, but source code is not available	- Free to use (no cost) - Source code closed - May have limited features compared to paid versions - Often used for marketing or user base expansion	Skype, WinRAR (trial), Google Chrome