# SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

Dhavalagiri, Dharwad-580002, Karnataka State, India.

**Email: cse.sdmcet@gmail.com**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING


# A Report on

# Selenium- A software Testing Tool

**COURSE CODE: xxxxx COURSE TITLE:**
**SEMESTER: DIVISION:**
**COURSE TEACHER: YYYYYYY**


**[ Academic Year- 2023-24]**

**Date of Submission: dd-mm-yyyy**

Submitted
By


| **Team Members:** | **USN:** |
|---|---|
| Kavana B B | 2SD22CS042 |
| Keerthi Naik | 2SD22CS043 |
| K Bhanushankar | 2SD22CS044 |
| Krishna | 2SD22CS045 |
| Kushaal | 2SD22CS046 |

# Table of Contents

## About GitHub software testing tool.

### What is GitHub?

GitHub is a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time. As Carpenter notes, because it allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.

### How can GitHub help my team and me?

GitHub allows multiple developers to work on a single project at the same time, reduces the risk of duplicative or conflicting work, and can help decrease production time. With GitHub, developers can build code, track changes, and innovate solutions to problems that might arise during the site development process simultaneously. Non-developers can also use it to create, edit, and update website content, which Carpenter demonstrates in her tutorial.

## Features of GitHub:

There are some common terms teams will need to understand when using GitHub.

They are:

- **Git** — a tool that allows developers and others to use version control
- **GitHub** — one of many web interfaces for using Git
- **Organization (org)** — a grouping mechanism allowing teams to collaborate across many projects at once
- **Repository (repo)** — a folder in which all files and their version histories are stored
- **Branch** — a version of the repo that allows work without affecting other branches. Repos may have many branches for different possible changes being tested or considered, along with a default branch that serves as the source of truth.
- **Fork** — a new repository that inherits from a parent “upstream” repo. It is used to suggest changes to an “upstream” public repo by someone who doesn’t have access to edit in the repo’s home org.
- **Commit Changes** — a saved record of a change made to a file within the repo.
- **Pull Request (PR)** — a request for changes made to a branch to be pulled into another branch. Allows multiple users to see, discuss and review work being suggested.
- **Merge** — after a pull request is approved, the commit will be pulled in (or merged) from one branch to another and then, deployed on the live site
- **Issues** — allow users to report issues or bugs and track progress of assigning the fix for the issues.
- **Projects** — allows you to use GitHub for project management and tracking a set of issues, either for a specific repo or an entire org
- **Wiki** — a section of a repo made for hosting documentation. Documentation may be in the repo’s README files instead.

# Installation procedure:

**Step 1:Downloading GitHub:**

- Visit the download page for GitHub Desktop.
- Click Download for Windows.
- In your computer's Downloads folder, double-click the GitHub Desktop setup file.
- GitHub Desktop will launch after installation is complete.
- For more information, see "Installing GitHub Desktop."

**Step 2:Creating an account**

- If you do not already have an account on GitHub.com, create one now. For more information, see "Creating an account on GitHub."
- Authenticating to GitHub
- To connect to GitHub Desktop with GitHub, you'll need to authenticate your account.
- For more information, see "Authenticating to GitHub in GitHub Desktop."
- After authenticating your account, you are ready to manage and contribute to projects with GitHub Desktop.

**Step 3: Configuring Git**

- You must have Git installed before using GitHub Desktop.
- After you have Git installed, you'll need to configure Git for GitHub Desktop.

**Step 4: Customizing GitHub Desktop**

- You can adjust defaults and settings to tailor GitHub Desktop to your needs.
- Choosing a default text editor

  You can open a text editor from GitHub Desktop to manipulate files and repositories. GitHub Desktop supports a variety of text editors and integrated development environments (IDEs) for Windows and macOS. You can choose a default editor in the GitHub Desktop settings.
- Choosing a theme
- Press alt+up to activate

# Illustration of features and environment:

**Team Creation:**

- The upper-right corner of GitHub.com, select your profile photo, then click  Your organizations.
- Click the name of your organization.
- Under your organization name, click  Teams.
- A tab, labeled with the people icon and "Teams," is outlined in dark orange.
- At the top of the page, click New team.
- Under "Create new team", type the name for your new team.
- Optionally, in the "Description" field, type a description of the team.
- Optionally, if you're creating a child team, under "Parent team", select the Select a parent team
- dropdown menu and click a parent team.
- Under "Team visibility", select a visibility for the team.
- Under "Team notifications", select Enabled or Disabled.
- This setting does not affect notifications for reviews requested from the team.
- Click Create team.
- Optionally, give the team access to organization repositories.
- Press alt+up to activate

# References.

<< Contents to be written here in IEE format>>