**Exp.No. 1 Create and access git hub repositories with various commands and web hosting applications**

**Date:**

**Aim:**

To develop and access git repository is to familiarize participants with essential Git concepts and commands, enabling them to effectively use Git for version control and collaboration

**Requirements:**

* Github
* Java

**Installation:**

* Github – login
* Git bash – commandline
* Java jdk – 17
* Eclipse IDE – 4.3.2

# Key Concepts:

* **Version Control:** GitHub uses Git, a distributed version control system, to track changes to source code over time.
* **Repositories**: A repository (or repo) is a collection of files, folders, and the entire history of a project. Repositories on GitHub serve as the central place where code and project-related assets are stored.
* **Remote Repository:** A remote repository is a copy of the Git repository stored on a server, enabling collaboration among multiple developers.
* **Forks and Clones:** Developers can create copies (forks) of repositories to work on their own versions of a project. Cloning a repository allows developers to create a local copy of the project on their machine.
* **Commit:** A commit is a snapshot of the changes made to the files in the repository at a specific point in time.
* **Branch:** A branch is a separate line of development within a repository. It allows developers to work on new features or bug fixes without affecting the main codebase
* **Merge:** Merging is the process of combining changes from one branch into another.
* **Pull Request:** In Git hosting platforms like GitHub, a pull request is a feature that allows developers to propose changes from one branch to another
* **GitHub Pages:** This feature allows users to publish web content directly from a GitHub repository, making it easy to create websites (web hosting) and documentation for projects.

1. Procedure to access files with git and github commands

Preparing Git Bash command line:

* Git status
* Cd “D://CSE/Lab”
* Mkdir CSEA
* Cd CSEA
* nano f1.c
* nano f2.c

Step 1: Setting Up Git Repository

* git init
* ls -l
* ls -la
* git status

Step 2: Creating and Committing Changes

* git add f1.c
* git add .
* git status
* git commit -m “First Version of file”
* ls -la

Step 3: Exploring History

* git status
* git diff
* git log

Go to github.com – sign in with login user & password

* create a repository – (GitAccess)
* new
* public
* create repository
* copy https of the repository
* Go back to git bash and paste it with a command

Step 4: Connect with remote repository and push

* git remote add origin “https://github.com/user/user\_repo.git”
* git push -u origin master
* git clone “https://github.com/user/user\_repo.git”
* git branch: Lists, creates, or deletes branches within the repository.
* git checkout: Switches between branches, commits, or tags. It's used to navigate through the repository's history.
* git merge: Combines changes from different branches, integrating them into the current branch.
* git pull: Fetches changes from a remote repository and merges them into the current branch.

Step 5: Branching and merging

* git branch feature (or)
* git checkout -b feature
* git checkout master
* git merge feature

Step 6: Cloning a Repository

* git clone “https://github.com/user/user\_repo.git”

Check it with local repository

* cd user\_repo

1. Procedure to access web hosting via github monitoring version control system

Step 1: Create a html file in notepad and store it in any disk as .html extension

Step 2: Program

Fn.html

<html>

<head>

<title>Welcome page for web hosting</title>

</head>

<body>

<h1> Welcome to Alliance University</h1>

<h2 style="color:green;font-style:italic;">

Welcome to CSE Department</h2>

<h5> Welcome you all for the International Conference </h5>

<pre>

Sample Program for web hosting

Developed by

CSE Team

</pre>

</body>

</html>

**Inferences:**

Understanding github and its commands by performing transition for storage and access from online environment, and integrating data from multiple resources.

**Conclusion:**

Thus the way of accessing files from remote repository to local repository and its vice-versa was done successfully and web hosting applications done successfully with all the key commands of git distributed systems

**Expt:2 Program to execute in Java Application using github**

**Date:**

**Aim: To implement java program and servlet application using eclipse IDE via github monitoring applications.**

**Requirements:**

* Github
* Java EE

**Installation:**

* Github – login
* Git bash – commandline
* Java jdk – 17
* Eclipse IDE – 4.3.2

1. Procedure to access Java files with git and github in eclipse IDE

**Program:**

import java.util.Scanner;

public class FirstJavaProj

{ Scanner s;

int regno;

String name, address;

void input()

{ s=new Scanner(System.*in*);

System.*out*.println("Enter Regno of the student:");

regno=s.nextInt();

System.*out*.println("Enter the Name of the student:");

name=s.next();

System.*out*.println("Enter the Address of the student:");

address=s.next();

}

void compute()

{ System.*out*.println("Reg No is:"+regno);

System.*out*.println("Name is:"+name);

System.*out*.println("Address is:"+address);

}

public static void main(String[] args)

{

FirstJavaProj fp=new FirstJavaProj();

fp.input();

fp.compute();

}

**}**

1. Procedure to access Servlet files with git and github in eclipse IDE

Program for servlet:

**Fn.html**

**<!DOCTYPE html>**

**<html><head>**

**<meta charset=*"UTF-8"*>**

**<title>Insert title here</title>**

**</head><body>**

**<form name=*f1* method=*get* action=*"ServProj"*>**

**Enter Username:<input type=*text* name=*un*><br>**

**Enter Password:<input type=*text* name=*pwd*><br>**

**<input type=*submit* value=*Enter*>**

**</form></body></html>**

**ServProj**

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/ServProj")

**public** **class** ServProj **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

response.setContentType("text/html");

PrintWriter out=response.getWriter();

String user=request.getParameter("un");

String pwd=request.getParameter("pwd");

out.println("<html><body><h1>Welcome to Alliance university</h1>");

out.println("<h1>Welcome to Login page program</h1>");

out.println("<h4> user is:"+ user);

out.println("<br>Password is:"+ pwd+"</h4>");

}

}

**Conclusion:**

**Thus the program to edit and access java and Servlet in eclipse IDE and github was executed successfully**