1.Git registry and registration form

Git push :
create a directory
place the files in the directory

push into git

```
git init
$git status
$git add .
$git status
$git commit -m “commit 1”
$git config --global user.name “github username”
$git config --global user.email “email id”
$git remote add origin “repo url”
$git push origin master
```

cloning and pull

```
$git clone “repo url”
$git pull “repo url”
$git config pull-rebase true //only if there is error
$git pull “repo url”
```

registration.html

```
<html>
<head>
<link href="register.css" rel="stylesheet" />
</head>
<body>
<h1> DevOps Lab</h1>
<h2> Student Registration Form</h1>
<form>
<table border="5" align="center" cellspacing="10" cellpadding="10" >
<tr>
<td>Name</td><td><input type="text"></td>
</tr>
<tr>
<td>Contact Number</td>
<td><input type="text"></td>
</tr>
<tr>
<td>Gender</td>
<td><input type="radio" name="g">Male
 <input type="radio" name="g">Female</td>
</tr>
<tr>
<td>Address</td>
<td><textarea rows="5" cols="15"></textarea></td>
</tr>
<tr>
<td>Hobbies</td>
<td><input type="checkbox">Singing
 <input type="checkbox">Travelling
 <input type="checkbox">Reading novels
</td>
</tr>
<tr>
<td>Skillset</td>
<td><input type="checkbox">C
```

```
 <input type="checkbox">Python
 <input type="checkbox">Java
</td>
</tr>
<tr>
<td>Highest Qualification</td>
<td><select>
 <option><--SELECT--></option>
 <option>Ph.D</option>
 <option>M.E/M.Tech</option>
 <option>B.E/B.Tech</option>
 <option>Diploma</option>
 <option>Inter</option>
 <option>SSC</option>
 </select>
 </td>
</tr>
<td>District</td>
<td><select>
 <option>--SELECT--></option>
 <option>Adilabad</option>
 <option>Zaheerabad</option>
 </select>
 </td>
</tr>
<tr>
<td><input type="submit" value="Register"></td> <td><input type="reset"
value="Clear"></td> </tr>
</table>
</body>
</html>
```

register.css

```
h1{
 color:green;
 text-align:center;
 }

h2{
 color:blue;
 text-align:center;
 }
p{
 color:red;
 }
table{
 background-color: cyan;
 }
td{
 color:red;
 font-size:24px;
 }
input
{
 color:blue;
 font-size:24px;
 text-align : center;
}
select

{
 color:blue;
 font-size:24px;
 text-align : center;
}
```

JENKINS QUESTION(2,3,4)

steps:
Create a file FileName.java , type the code and save file
Login into your GitHub Account and create a new repository
Upload the .java file into your GitHub repository and commit the changes.
Copy the URL of your Git Repository
In browser open https://localhost:8080 and login in using your account.
Create a new Jenkins job using the "Freestyle project" type.
o In the dash board, Click on New item to create job.
o Assign a meaningful name for the job and select free style project option and click on OK
button.
Provide a description about application and under Buildsteps option choose execute shell.
Type commands for execution of your java program.
o javac filename.java
o java filename
Cilck on Save button.
· Click on build now option available in dash board.
· To watch the output click on Console output

2. SumAvg.java

```
public class SumAvg {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 10; i++) {
            sum = sum + i;
        }
        double average = sum / 10.0;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);
    }
}
```

3.ArithmeticOperations.java

```
public class ArithmeticOperations {
public static void main(String[] args) {
int a = 12;
int b = 5;
System.out.println("a = " + a + ", b = " + b);
System.out.println("Addition: " + (a + b));
System.out.println("Subtraction: " + (a - b));
System.out.println("Multiplication: " + (a * b));
System.out.println("Division: " + (a / b));
System.out.println("Modulus: " + (a % b));
}
}
```

4.StudentDetails.java

```
public class StudentDetails {
    public static void main(String[] args) {
        String name = "John";
        int rollNo = 001;
        String department = "Computer Science";
        System.out.println("Student Details:");
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Department: " + department);
    }
}
```

DOCKER JAVA QUESTION(5,6,7)

in terminal :

```
$ sudo su
$ docker login -u [Dockerhubusername]
```

create 2 files

```
vi filename.java
vi Dockerfile
```

in vi filename write the code

for java Dockerfile

```
vi Dockerfile
FROM openjdk:11
WORKDIR /app
COPY ./filename.java .
RUN javac filename.java
CMD ["java", "filename"]
```

```
$ docker build -t htmlimage .
$ docker run -d -p 8082:80 htmlimage
$ docker tag imagename Dockerhubusername/htmlimage
$ docker push Dockerhubusername/htmlimage
```

5. SumAvg.java

```
public class SumAvg {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 1; i <= 10; i++) {
            sum = sum + i;
        }
        double average = sum / 10.0;
        System.out.println("Sum: " + sum);
        System.out.println("Average: " + average);
    }
}
```

6.ArithmeticOperations.java

```
public class ArithmeticOperations {
public static void main(String[] args) {
int a = 12;
int b = 5;
System.out.println("a = " + a + ", b = " + b);
System.out.println("Addition: " + (a + b));
System.out.println("Subtraction: " + (a - b));
System.out.println("Multiplication: " + (a * b));
System.out.println("Division: " + (a / b));
System.out.println("Modulus: " + (a % b));
}
}
```

7.StudentDetails.java

```
public class StudentDetails {
    public static void main(String[] args) {
        String name = "John";
        int rollNo = 001;
        String department = "Computer Science";
        System.out.println("Student Details:");
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Department: " + department);
    }
}
```

DOCKER PYTHON QUESTION(8,9,10)

in terminal :

```
$ sudo su
$ docker login -u [Dockerhubusername]
```

create 2 files

```
vi filename.py
vi Dockerfile
```

in vi filename write the code

for python Dockerfile

```
vi Dockerfile
FROM python: latest
WORKDIR /pythonapp
copy ./filename.py .
CMD ["python", "filename.py"]
```

```
$ docker build -t pythonimage .
$ docker run -it pythonimage
$ docker tag imagename Dockerhubusername/pythonimage
$ docker push Dockerhubusername/pythonimage
```

for pull Pull the code from Docker hub and execute

```
o $ docker pull sowjanyajindam/image
o $ docker run -it sowjanyajindam/image
```

8. SumAvg.py

```
total = 0
count = 10
for i in range(1, count + 1):
    total += i
average = total / count
print("Sum of first 10 numbers:", total)
print("Average of first 10 numbers:", average)
```

9.ArithmeticOperations.py

```
a = 20
b = 10
print("First Number:", a)
print("Second Number:", b)
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
print("Remainder:", a % b)
```

10.StudentDetails.py

```
name = "John Doe"
roll_no = 101
```

```
department = "Computer Science"
print("Student Details:")
print("Name:", name)
print("Roll No:", roll_no)
print("Department:", department)
```

DOCKER WEB QUESTION (11,12)

```
$ sudo su
$ mkdir HtmlDemo
o $ cd HtmlDemo
```

in vi filename write the code

```
o vi Dockerfile
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY ./index.html .
EXPOSE 80

 $ docker build –t htmlimage .
 $ docker run -it htmlimage

$ docker login -u [Dockerhubusername]
$ docker tag imagename Dockerhubusername/htmlimage
$ docker push Dockerhubusername/htmlimage
```

11.BG COLOR CHANGE

```
<html>
<head>
<script language="javascript">
function change(col)
{
switch(col)
{
case 'red':document.bgColor="red";
                   break;

 case 'green':document.bgColor="green";
                   break;
 case 'blue':document.bgColor="blue";
                   break;
  }
 }
 </script>
 </head>
 <body>
 <h1><input type="radio" name="c" onClick="change('red')"> RED</h1>
 <h1><input type="radio" name="c"  onClick="change('green')"> GREEN</h1>
 <h1><input type="radio" name="c" onClick="change('blue')"> BLUE<h1>
 </body>
 </html>
```

12.Login Form Validation

```
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
<script>
function validateLogin() {
var u=document.getElementById("username").value;
var p=document.getElementById("password").value;
if(u==="mvsr"&&p==="mvsr"){alert("Login successful!");return true;}
else{alert("Invalid credentials.");return false;}
}
</script>
</head>
<body>
<h2>Login Form</h2>
<form onsubmit="return validateLogin()">
<label>Username:</label><br>
<input type="text" id="username" required><br>
<label>Password:</label><br>
<input type="text" id="password" required><br><br>
<input type="submit" value="Login">
</form>
</body>
</html>
```

SELINIUM

13.Step 1: Create directory and add file.

```
o $ mkdir googleDemo
o $ cd googleDemo

o vi app.js
const {Builder, By, Key} = require("selenium-webdriver");
async function example(){
 let driver = await new Builder().forBrowser("chrome").build();  await
driver.get("https://www.google.com/");
 console.log("browser opened");
 await driver.quit();
}
example()
```

Step 2: Initialize the project and execute it

Execution Steps for Selenium:

```
· node -v
// check whether node is installed. If not, install using below commands.
// sudo apt-get update
//sudo apt install nodejs
· npm -v
// check whether npm is installed. If not, install using below commands.
//sudo apt install npm
· npm init // Initilaze the node package
· npm install selenium-webdriver // add selenium web driver as dependency
· npm init //check out for addition of selenium dependency
· node app.js //execute the program
```

14.create directory and add file.

```
o $ mkdir myloginDemo
o $ cd myloginDemo

o vi login.html
<html>
<head>
<title> Login Page</title>
<script language="javascript">
function validate()
{
var u=document.f1.u.value;
var p=document.f1.p.value;
if(u=="MVSREC" && p=="ITD")
{
window.open("loginsuccess.html");
}
else
{
window.open("loginfail.html");
}
}
</script>
</head>
<body>
<form name="f1">
<h1 align="center" style="color:blue">Login Page</h1>
<table align="center" bgcolor="pink">
<tr>
 <td>UserId</td>
 <td><input type="text" name="u" id="un"></td>
</tr>
<tr>
 <td>Password</td>
 <td><input type="password" name="p" id="pw"></td>
</tr>
<tr>
 <td><input type="button" value="Signin" id="s"
 onclick="validate()"></td>
 <td><input type="reset" value="Reset id="r"></td>
</tr>
</table>
</form>
</body>
</html>
```

o vi loginsucess.html

```
<html>
<head>
<title> Success </title>
</head>
<body>
<h1 align="center" style="color:red"> Login Succeess</h1> </body>
</html>
```

o vi loginfail.html

```
<html>
<head>
<title> Fail </title>
</head>
<body>
<h1 align="center" style="color:red"> Login Failed</h1> </body>
</html>
```

o vi mylogin.js

```
const { Builder, By, until } = require("selenium-webdriver"); const assert =
require("assert");
async function loginTest() {
 // launch the browser
let driver = await new Builder().forBrowser("chrome").build();  try {
await driver.get("file:///home/mvsr/myloginDemo/login.html"); await
driver.findElement(By.id("un")).sendKeys("MVSREC"); await
driver.findElement(By.id("pw")).sendKeys("ITD"); await
driver.findElement(By.id("s")).click();
const title = await driver.getTitle();
 assert.strictEqual(title,"Login Page");
 console.log("success");
 } finally {
 await driver.quit();
 }
}
loginTest();
```

Step 2: Initialize the project and execute it
Execution Steps for Selenium:

```
· node -v
// check whether node is installed. If not, install using below commands.
// sudo apt-get update
//sudo apt install nodejs
· npm -v
// check whether npm is installed. If not, install using below commands.
//sudo apt install npm
· npm init // Initilaze the node package
· npm install selenium-webdriver // add selenium web driver as dependency
· npm init //check out for addition of selenium dependency
· node app.js //execute the program
```

15. Create directory and add file.
o $ mkdir collegeloginDemo
o $ cd collegeloginDemo

o vi collegelogin.js

```
const { Builder, By, until } = require("selenium-webdriver");
const assert = require("assert");
async function loginTest() {
 // launch the browser
 let driver = await new Builder().forBrowser("chrome").build();  try {
 await driver.get("http://results.mvsrec.edu.in/SBLogin.aspx");  await
driver.findElement(By.id("txtUserName")).sendKeys("245121737129");  await
driver.findElement(By.id("txtPassword")).sendKeys("245121737129");  await
driver.findElement(By.id("btnSubmit")).click();
 const user = await driver.findElement(By.id("lblHTNo")).getText();
 assert.strictEqual(user, "245121737129");
 console.log("success");
```

```
 await driver.findElement(By.id("Stud_cpModules_imgbtnExams")).click();  await
driver.findElement(By.id("cpBody_lnkSem")).click();  const ur = await
driver.getCurrentUrl();
 assert.strictEqual(ur,
"http://results.mvsrec.edu.in/STUDENTLOGIN/Frm_SemwiseStudMarks.aspx");
 console.log("Display marks success");
 }
finally {
 await driver.quit();
 }
}
loginTest();
```

Step 2: Initialize the project and execute it
Execution Steps for Selenium:

```
· node -v
// check whether node is installed. If not, install using below commands.
// sudo apt-get update
//sudo apt install nodejs

· npm -v
// check whether npm is installed. If not, install using below commands.
//sudo apt install npm

· npm init // Initilaze the node package

· npm install selenium-webdriver // add selenium web driver as dependency
· npm init //check out for addition of selenium dependency
· node collegelogin.js //execute the program
```

For installation of higher version of node

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
source ~/.bashrc
nvm list-remote
nvm install v16.14.0
```