1)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;
}
```

GIT COMMANDS

```
1.git init
2.git status
3.git add .
4. git commit -m “ project”
5.git remote add origin “repository link”
$ git config --global user.name “username”
$ git config --global user.email “user@mail.com”
6.git push origin master
```

2. #sum and average java

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

3. # arithmetic operations

```
Vi ArithmeticOperations.java
public class ArithmeticOperations {
    public static void main(String[] args) {
        int a = 20;
        int b = 10;

        System.out.println("Addition = " + (a + b));
        System.out.println("Subtraction = " + (a - b));
        System.out.println("Multiplication = " + (a * b));
        System.out.println("Division = " + (a / b));
    }
}
```

4. Student details

```
public class StudentDetails {
    public static void main(String[] args) {
        String name = "Name";
        String rollNo = "24512274800";
        String department = "Computer Science";

        System.out.println("Student Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Department: " + department);
    }
}
```

5. Docker sum and average

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

Dockerfile

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

Docker commands

```
1.$ sudo su
2.$ docker build –t javaimage .
o $ docker run -it javaimage
3.$ docker login -u [Dockerhubusername]
4.docker tag javaimage Dockerhubusername/javaimage
5.$ docker push Dockerhubusername/javaimage
```

6.arthimetic operations docker

```
public class ArithmeticOperations {
    public static void main(String[] args) {
        int a = 15;
        int b = 5;

        System.out.println("Addition = " + (a + b));
        System.out.println("Subtraction = " + (a - b));
        System.out.println("Multiplication = " + (a * b));
        System.out.println("Division = " + (a / b));
    }
}
```

Dockerfile

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

Docker commands

```
1.$ sudo su
2.$ docker build –t javaimage .
o $ docker run -it javaimage
3.$ docker login -u [Dockerhubusername]
4.docker tag javaimage Dockerhubusername/javaimage
5.$ docker push Dockerhubusername/javaimage
```

7.docker student details

```
public class StudentDetails {
    public static void main(String[] args) {
        String name = "Name";
        String rollNo = "24512274800";
        String department = "Computer Science";

        System.out.println("Student Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Department: " + department);
    }
}
```

Docker file

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

Docker commands

```
1.$ sudo su
2.$ docker build –t javaimage .
o $ docker run -it javaimage
3.$ docker login -u [Dockerhubusername]
4.docker tag javaimage Dockerhubusername/javaimage
5.$ docker push Dockerhubusername/javaimage
```

8. Sum and average docker

sum_avg.py

```
total = 0
for i in range(1, 11):
    total += i
```

```
average = total / 10

print("Sum =", total)
print("Average =", average)
```

Docker file

```
FROM python:3.10
WORKDIR /app
COPY sum_average.py .
CMD ["python", "sum_average.py"]
```

Docker commands

```
1.$ sudo su
2.$ docker build –t pythonimage .
o $ docker run -it pythonimage
3.$ docker login -u [Dockerhubusername]
4.docker tag pythonimage Dockerhubusername/pythonimage
5.$ docker push Dockerhubusername/pythonimage
```

9.arthimetic docker python

arithmetic.py

```
a = 12
b = 4

print("Addition =", a + b)
print("Subtraction =", a - b)
print("Multiplication =", a * b)
print("Division =", a / b)
```

Docker file

```
FROM python:3.10
WORKDIR /app
COPY arithmetic_operations.py .
CMD ["python", "arithmetic_operations.py"]
```

Docker commands

```
1.$ sudo su
2.$ docker build –t pythonimage .
o $ docker run -it pythonimage
3.$ docker login -u [Dockerhubusername]
4.docker tag pythonimage Dockerhubusername/pythonimage
5.$ docker push Dockerhubusername/pythonimage
```

10.student details docker py

```
name = "name"
roll_no = "21CS123"
department = "Computer Science"

print("Student Name:", name)
print("Roll No:", roll_no)
print("Department:", department)
```

Docker file

```
FROM python:3.10
```

```
WORKDIR /app
COPY student_details.py .
CMD ["python", "student_details.py"]
```

Docker commands

```
1.$ sudo su
2.$ docker build –t pythonimage .
o $ docker run -it pythonimage
3.$ docker login -u [Dockerhubusername]
4.docker tag pythonimage Dockerhubusername/pythonimage
5.$ docker push Dockerhubusername/pythonimage
```

11.webpage docker

bgcolordemo1.html

```
<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>
```

Dockerfile

```
FROM nginx:latest
WORKDIR /usr/share/nginx/html
COPY ./bgcolordemo1.html .
EXPOSE 80
```

Docker commands

```
1.$ sudo su
2.$ docker build –t htmlimage .
```

```
o $ docker run -d -p 8080:80 htmlimage
3.$ docker login -u [Dockerhubusername]
4.docker tag htmlimage Dockerhubusername/htmlimage
5.$ docker push Dockerhubusername/htmlimage
```

12. Login validation docker

```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
  <script>
    function validateForm() {
      var username = document.forms["loginForm"]["username"].value;
      var password = document.forms["loginForm"]["password"].value;
      if (username == "" || password == "") {
        alert("Username and Password must be filled out");
        return false;
      } else {
        alert("Login successful");
        return true;
      }
    }
  </script>
</head>
<body>
  <h2>Login Form</h2>
  <form name="loginForm" onsubmit="return validateForm()">
    Username: <input type="text" name="username"><br><br>
    Password: <input type="password" name="password"><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```

Docker File

```
FROM nginx:alpine
RUN rm -rf /usr/share/nginx/html/*
COPY login.html /usr/share/nginx/html/index.html
EXPOSE 80
```

Docker commands

```
1.$ sudo su
2.$ docker build –t htmlimage .
o $ docker run -d -p 8080:80 htmlimage
3.$ docker login -u [Dockerhubusername]
4.docker tag htmlimage Dockerhubusername/htmlimage
5.$ docker push Dockerhubusername/htmlimage
```

13.SIMPLE PROGRAM IN JS google

```
$ 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()
```

COMMANDS

```
npm init
npm install selenium-webdriver
npm init
node app.js
```

14.login form testing selenium

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();
```

COMMANDS

```
npm init
npm install selenium-webdriver
npm init
node mylogin.js
```

15.results.mvsr using selenium

```
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();
```

COMMANDS

```
npm init
npm install selenium-webdriver
npm init
node collegelogin.js
```