1. Develop a simple java containerized application using Docker.

FROM openjdk:21

WORKDIR /app

COPY . /app

RUN javac Calci.java

CMD ["java","Calci"]

public class A

{

public static void main(String[] args)

{

System.out.println("hello");

}

}

---docker build -t img4 .

---docker run img4

---docker exec -it 23387488484848

---docker exec -it img4

2. Develop an interactive java containerized application (simple Calculator) using Docker.

import java.util.Scanner;

public class Calci{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.println("Simple Calculator");

System.out.print("Enter first number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter an operator (+, -, \*, /): ");

char operator = scanner.next().charAt(0);

System.out.print("Enter second number: ");

double num2 = scanner.nextDouble();

double result;

switch (operator) {

case '+':

result = num1 + num2;

System.out.println("Result: " + result);

break;

case '-':

result = num1 - num2;

System.out.println("Result: " + result);

break;

case '\*':

result = num1 \* num2;

System.out.println("Result: " + result);

break;

case '/':

if (num2 != 0) {

result = num1 / num2;

System.out.println("Result: " + result);

} else {

System.out.println("Error: Division by zero is not allowed.");

}

break;

default:

System.out.println("Error: Invalid operator.");

}

} catch (Exception e) {

System.out.println("Error: Invalid input. Please enter numbers and valid operators only.");

} finally {

scanner.close();

}

}

}

FROM openjdk:21

WORKDIR /app

COPY . /app

RUN javac Calci.java

CMD ["java","Calci"]

---docker build -t cal .

--docker run -it --name calciapp -d cal

---docker exec -t calciapp java Calci

3.Develop a simple webserver containerized application (Registration form) using Docker.

FROM nginx:latest

COPY index.html /usr/share/nginx/html/index.html

COPY styles.css /usr/share/nginx/html/styles.css

EXPOSE 8084

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registration Form</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="container">

<h1>Registration Form</h1>

<form id="registrationForm">

<div class="form-group">

<label for="firstName">First Name:</label>

<input type="text" id="firstName" name="firstName" required>

</div>

<div class="form-group">

<label for="lastName">Last Name:</label>

<input type="text" id="lastName" name="lastName" required>

</div>

<div class="form-group">

<label for="fatherName">Father's Name:</label>

<input type="text" id="fatherName" name="fatherName" required>

</div>

<div class="form-group">

<label for="motherName">Mother's Name:</label>

<input type="text" id="motherName" name="motherName" required>

</div>

<div class="form-group">

<label for="dob">Date of Birth:</label>

<input type="date" id="dob" name="dob" required>

</div>

<div class="form-group">

<label>Gender:</label>

<label><input type="radio" name="gender" value="Male" required> Male</label>

<label><input type="radio" name="gender" value="Female" required> Female</label>

<label><input type="radio" name="gender" value="Other" required> Other</label>

</div>

<div class="form-group">

<label for="address">Address:</label>

<textarea id="address" name="address" rows="4" required></textarea>

</div>

<div class="form-group">

<label for="qualification">Qualification:</label>

<input type="text" id="qualification" name="qualification" required>

</div>

<div class="form-group">

<label for="profilePhoto">Upload Profile Photo:</label>

<input type="file" id="profilePhoto" name="profilePhoto" accept="image/\*" required>

</div>

<div class="form-group">

<label for="signature">Upload Signature:</label>

<input type="file" id="signature" name="signature" accept="image/\*" required>

</div>

<div class="form-buttons">

<button type="submit">Submit</button>

<button type="reset">Reset</button>

</div>

</form>

</div>

<script src="script.js"></script>

</body>

</html>

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

}

.container {

max-width: 600px;

margin: 50px auto;

background: #fff;

padding: 20px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

border-radius: 8px;

}

h1 {

text-align: center;

color: #333;

}

.form-group {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

color: #555;

}

input[type="text"],

input[type="date"],

textarea,

input[type="file"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

border: 1px solid #ccc;

border-radius: 4px;

}

textarea {

resize: vertical;

}

input[type="radio"] {

margin-right: 5px;

}

.form-buttons {

display: flex;

justify-content: space-between;

}

button {

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

button[type="submit"] {

background-color: #28a745;

color: #fff;

}

button[type="reset"] {

background-color: #dc3545;

color: #fff;

}

document.getElementById("registrationForm").addEventListener("submit", function(event) {

event.preventDefault(); // Prevent form submission to server

const firstName = document.getElementById("firstName").value;

const lastName = document.getElementById("lastName").value;

alert(`Form submitted! Welcome, ${firstName} ${lastName}!`);

});

----docker build -t web .

-----docker run --name reg web

---docker run -d -p 8084:80 --name reg web

4.Write an installation steps for Integrate Kubernetes and Docker. (And some Kubernetes theory questions will be given).

5.Automate the process of running containerized web application using Kubernetes.

• docker build -t sanjaycheekati/kub:1.0 .

• docker push sanjaycheekati/kub:1.0

• kubectl run kub --image=sanjaycheekati/kub:1.0 --port=80

• kubectl get pods

• kubectl get svc

• kubectl port-forward service/kubsvc 8084:80

• Open browser and type localhost:8084

7.Install and Explore Selenium for Automated Testing.

Install NetBeans IDE 21 and JDK 17

Install Jar Files From The Selenium Downloads

Unzip Into Your Working Directory

Open NetBeans Go to the File>New Project

Give The File Name & Click On Finish.

Right Click on the Libraries And Click on Add JAR/Folder

Add all the Jar Files Of Selenium By Pressing CTRL+A

FirstSeleniumPro1.java

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.edge.EdgeDriver;

public class FirstSeleniumPro {

public static void main(String[] args) {

// Set the path to the ChromeDriver executable

// System.setProperty("webdriver.chrome.driver", "D:\\DEVOPS

LAB\\WebDrivers\\msgedgedriver.exe");

// Initialize ChromeDriver

WebDriver driver = new EdgeDriver();

try {

// Open a website

driver.get("https://www.google.com");

// Print the title of the page

System.out.println("Title: " + driver.getTitle());

// Wait for 60 seconds

Thread.sleep(60000);

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

} finally {

// Close the browser

driver.quit();

}

}

}

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.edge.EdgeDriver;

public class FirstSeleniumPro1 {

public static void main(String[] args) {

// Set the path to the ChromeDriver executable

//System.setProperty("webdriver.chrome.driver", "D:\\DEVOPS

LAB\\WebDrivers\\msgedgedriver.exe");

// Initialize ChromeDriver

WebDriver driver = new EdgeDriver();

try {

// Open a website

driver.get("D:\\DEVOPS LAB\\EventRegistrationForm.html");

// Print the title of the page

System.out.println("Title: " + driver.getTitle());

// Wait for 60 seconds

Thread.sleep(60000);

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

} finally {

// Close the browser

driver.quit();

}

}

}

8.Write a simple program in JavaScript and perform testing using Selenium.

JavaScriptCalculator.html

<!DOCTYPE html>

<html>

<head>

<title>Simple Calculator</title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.calculator {

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.calculator input[type="text"] {

width: 100%;

height: 50px;

font-size: 1.2em;

text-align: right;

margin-bottom: 10px;

border: none;

border-radius: 5px;

box-shadow: inset 0 0 5px rgba(0, 0, 0, 0.1);

padding-right: 10px;

}

.calculator button {

width: 22%;

height: 50px;

font-size: 1.2em;

margin: 2px;

border: none;

border-radius: 5px;

background-color: #f2f2f2;

cursor: pointer;

transition: background-color 0.2s;

}

.calculator button:hover {

background-color: #ddd;

}

.calculator button.operator {

background-color: #d4edda;

}

.calculator button.operator:hover {

background-color: #c3e6cb;

}

.calculator button.equals {

background-color: #76c7c0;

width: 48%;

color: #fff;

}

.calculator button.equals:hover {

background-color: #63b0b8;

}

.calculator button.zero {

width: 48%;

}

</style>

</head>

<body>

<div class="calculator">

<input type="text" id="display" readonly>

<br>

<button onclick="clearDisplay()">C</button>

<button onclick="deleteLast()">←</button>

<button class="operator" onclick="appendToDisplay('/')">/</button>

<button class="operator" onclick="appendToDisplay('\*')">\*</button>

<br>

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button class="operator" onclick="appendToDisplay('-')">-</button>

<br>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button class="operator" onclick="appendToDisplay('+')">+</button>

<br>

<button onclick="appendToDisplay('1')">1</button>

<button onclick="appendToDisplay('2')">2</button>

<button onclick="appendToDisplay('3')">3</button>

<button class="equals" onclick="calculateResult()">=</button>

<br>

<button class="zero" onclick="appendToDisplay('0')">0</button>

<button onclick="appendToDisplay('.')">.</button>

</div>

<script>

function clearDisplay() {

document.getElementById('display').value = '';

}

function deleteLast() {

var display = document.getElementById('display').value;

document.getElementById('display').value = display.substring(0, display.length - 1);

}

function appendToDisplay(value) {

document.getElementById('display').value += value;

}

function calculateResult() {

var display = document.getElementById('display').value;

try {

document.getElementById('display').value = eval(display);

} catch (e) {

document.getElementById('display').value = 'Error';

}

}

</script>

</body>

</html>

Java Code

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.edge.EdgeDriver;

public class FirstSeleniumPro {

public static void main(String[] args) {

// Set the path to the ChromeDriver executable

//System.setProperty("webdriver.chrome.driver", "D:\\DEVOPS

LAB\\WebDrivers\\msgedgedriver.exe");

// Initialize ChromeDriver

WebDriver driver = new EdgeDriver();

try {

// Open a website

driver.get("D:\\DEVOPS LAB\\JavaScriptCalculator.html");

// Print the title of the page

System.out.println("Title: " + driver.getTitle());

// Wait for 60 seconds

Thread.sleep(60000);

} catch (Exception e) {

System.out.println("Error: " + e.getMessage());

} finally {

// Close the browser

driver.quit();

}

}

}

9.

<!DOCTYPE html>

<html>

<head>

<title>Simple Calculator</title>

</head>

<body>

<div>

<input type="text" id="display" readonly><br>

<button onclick="clearDisplay()">C</button>

<button onclick="deleteLast()">←</button>

<button onclick="appendToDisplay('/')">/</button>

<button onclick="appendToDisplay('')"></button><br>

<button onclick="appendToDisplay('7')">7</button>

<button onclick="appendToDisplay('8')">8</button>

<button onclick="appendToDisplay('9')">9</button>

<button onclick="appendToDisplay('-')">-</button><br>

<button onclick="appendToDisplay('4')">4</button>

<button onclick="appendToDisplay('5')">5</button>

<button onclick="appendToDisplay('6')">6</button>

<button onclick="appendToDisplay('+')">+</button><br>

<button onclick="appendToDisplay('1')">1</button>

<button onclick="appendToDisplay('2')">2</button>

<button onclick="appendToDisplay('3')">3</button>

<button onclick="calculateResult()">=</button><br>

<button onclick="appendToDisplay('0')">0</button>

<button onclick="appendToDisplay('.')">.</button>

</div>

<script>

function clearDisplay() {

document.getElementById('display').value = '';

}

function deleteLast() {

let display = document.getElementById('display').value;

document.getElementById('display').value = display.slice(0, -1);

}

function appendToDisplay(value) {

document.getElementById('display').value += value;

}

function calculateResult() {

let display = document.getElementById('display').value;

try {

document.getElementById('display').value = eval(display);

} catch {

document.getElementById('display').value = 'Error';

}

}

</script>

</body>

</html>

from selenium import webdriver

import time

print("Test Execution Started")

options = webdriver.ChromeOptions()

driver = webdriver.Chrome(options=options)

try:

driver.maximize\_window()

driver.get("file:///C:/Users/abhir/index\_1.html")

time.sleep(2)

name\_field = driver.find\_element("name", "name")

name\_field.send\_keys("John Doe")

time.sleep(1)

age\_field = driver.find\_element("name", "age")

age\_field.send\_keys("25")

time.sleep(1)

submit\_button = driver.find\_element("id", "submit")

submit\_button.click()

time.sleep(2)

finally:

driver.quit()

print("Test Execution Completed Successfully!")

10.

import time

import logging

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from webdriver\_manager.microsoft import EdgeChromiumDriverManager

from selenium.webdriver.edge.service import Service

# Configure logging

logging.basicConfig(level=logging.INFO, format="%(asctime)s - %(levelname)s - %(message)s")

logger = logging.getLogger(\_name\_)

def setup\_webdriver():

"""Set up and initialize the Edge WebDriver."""

options = webdriver.EdgeOptions()

options.add\_argument('--ignore-ssl-errors=yes')

options.add\_argument('--ignore-certificate-errors')

options.add\_argument('--no-sandbox')

options.add\_argument('--disable-dev-shm-usage')

options.add\_argument('--start-maximized')

options.add\_argument('--disable-gpu')

# Uncomment below for headless mode

# options.add\_argument('--headless')

driver\_path = EdgeChromiumDriverManager().install()

service = Service(driver\_path) # Use Service for specifying the driver path

logger.info("Initializing WebDriver...")

return webdriver.Edge(service=service, options=options)

def fill\_registration\_form(driver):

"""Fill the registration form and submit."""

url = "file:///Users/indrakantirava/HTML/registration.html" # Use local file path instead of the server URL

logger.info(f"Navigating to the form at {url}...")

driver.get(url)

try:

# Wait for the form to load

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.ID, "firstName")))

logger.info("Form loaded successfully.")

# Fill out the form fields

logger.info("Filling out the form...")

driver.find\_element(By.ID, "firstName").send\_keys("Indra Kanti Rava")

time.sleep(2)

driver.find\_element(By.ID, "lastName").send\_keys("Vadde")

time.sleep(2)

driver.find\_element(By.ID, "email").send\_keys("vaddeindrakantirava@gmail.com")

time.sleep(2)

driver.find\_element(By.ID, "phone").send\_keys("8978296445")

time.sleep(2)

# Submit the form

logger.info("Submitting the form...")

WebDriverWait(driver, 10).until(EC.element\_to\_be\_clickable((By.XPATH, "//button[@type='submit']")))

driver.find\_element(By.XPATH, "//button[@type='submit']").click()

# Wait for and verify the success message

WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.ID, "successMessage")))

confirmation\_message = driver.find\_element(By.ID, "successMessage").text

assert "Thank you for registering!" in confirmation\_message, "Unexpected confirmation message!"

logger.info("Test Passed: Confirmation message verified.")

except Exception as e:

logger.error(f"Test Failed: {e}")

raise

finally:

time.sleep(2) # Optional pause before closing

def main():

"""Main function to execute the test."""

logger.info("Test Execution Started")

driver = None

try:

logger.info("Initializing WebDriver...")

driver = setup\_webdriver()

logger.info("WebDriver initialized successfully.")

# Execute the form fill and submission

fill\_registration\_form(driver)

except Exception as e:

logger.error(f"An error occurred during test execution: {e}")

finally:

if driver:

try:

logger.info("Closing the browser...")

driver.quit()

logger.info("Browser closed successfully.")

except Exception as e:

logger.error(f"Error while closing the browser: {e}")

logger.info("Test Execution Completed Successfully!")

if \_name\_ == "\_main\_":

main()

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registration Form</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

padding: 0;

background-color: #f9f9f9;

}

form {

max-width: 400px;

margin: auto;

background: #fff;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

label {

display: block;

margin-bottom: 8px;

font-weight: bold;

}

input[type="text"], input[type="email"], input[type="tel"] {

width: 100%;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

}

button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

button:hover {

background-color: #45a049;

}

#successMessage {

margin-top: 20px;

color: green;

font-weight: bold;

}

</style>

</head>

<body>

<h2 style="text-align: center;">Registration Form</h2>

<form id="registrationForm">

<label for="firstName">First Name</label>

<input type="text" id="firstName" name="firstName" required>

<label for="lastName">Last Name</label>

<input type="text" id="lastName" name="lastName" required>

<label for="email">Email</label>

<input type="email" id="email" name="email" required>

<label for="phone">Phone Number</label>

<input type="tel" id="phone" name="phone" required>

<button type="submit">Submit</button>

</form>

<!-- Placeholder for success message -->

<div id="successMessage" style="display: none;">

Thank you for registering!

</div>

<script>

// Simulating success message on form submission

document.getElementById('registrationForm').addEventListener('submit', function(event) {

event.preventDefault(); // Prevent actual form submission

document.getElementById('successMessage').style.display = 'block'; // Show success message

});

</script>

</body>

</html>