Lab 2: Building First Servlet & Handling Form Requests

Lab Overview

In this lab, you will learn how to build and deploy your first servlet and handle basic HTTP form requests. You will develop a simple servlet-based web application that accepts form data from users, processes the request on the server side, and displays a response. This lab will introduce the core concepts of servlets and HTTP request handling in a Java EE web environment.

You are required to complete this lab individually. The final submission will involve storing your code in a GitHub repository and submitting the repository link for evaluation.

Learning Objectives

By the end of this lab, students should be able to:

- Understand the structure of a Java servlet.
- Build and deploy a basic servlet using Apache Tomcat.
- Handle HTTP GET and POST requests in servlets.
- Process form data and simulate basic business logic transactions.
- Use GitHub for version control and project submission.

Lab Tasks

1. **Setup and Environment Configuration**
    - Set up Apache Tomcat 10 as your servlet container.
    - Configure your development environment (Eclipse) to deploy Java servlets.
2. **Business Case: Socket Machine Web Application**
    - **Scenario**: You are tasked with creating a simple web application for a fictional company, **Socket Machines Ltd.**, which sells industrial sockets. The application should allow users to submit a form to request a quote for specific types of sockets.
    - **Form Fields**: The form should include fields for the socket type (e.g., Type A, Type B, Type C), quantity, and customer contact information (name and email).

3. **Creating Your First Servlet**

    - Write a servlet class that extends HttpServlet.
    - Implement the doGet and doPost methods. In the doPost method, handle the form submission, validate the input (e.g., ensure quantity is numeric), and generate a response with the quote details.

4. **Form Handling**

    - Create an HTML form that allows users to input their socket type, quantity, and contact information. The form should submit data using the POST method.
    - In the servlet, retrieve the form data using request.getParameter().
    - Implement business logic to calculate a price quote based on the socket type and quantity and generate a response to display the quote to the user.

5. **Servlet Deployment**

    - Package your servlet into a WAR file and deploy it on Apache Tomcat.
    - Test the application by submitting data through the HTML form and verify the servlet’s response.

6. **Version Control with GitHub**

    - Initialize a Git repository for your project.
    - Commit your code frequently, ensuring each commit has a meaningful message.
    - Push your final code to a GitHub repository.

Submission Guidelines

- **Deadline**: The GitHub repository link must be submitted by **September 12, 2024 11:59pm**

- **Submission Platform**: Submit the GitHub repository link via the course management system.

- **Version Control**: Ensure your final code is properly committed to GitHub, and the repository is public or accessible to the instructor.

- **Screenshots**: Create screenshots, illustrating your Web Application and store then in the same GitHub Repository with codes

Evaluation Criteria

Your lab will be evaluated based on the following criteria:

Criteria	**Points**
Servlet functionality	30
Correct handling of HTTP requests	20
Business logic implementation (quote calculation)	20
Code organization and structure	10
Proper deployment and testing	10
Correct use of GitHub for version control	10

Additional Notes

- **Academic Integrity**: This is an individual assignment. Sharing code or collaborating with others on this lab is strictly prohibited. Any violations will be dealt with according to the institution's academic integrity policy.
- **Instructor Availability**: The instructor will be available during office hours and by appointment. Please feel free to reach out for any clarification or help.
- **GitHub Requirement**: You must use GitHub for storing your project and submitting the repository link. Ensure your repository is accessible for instructor. A GitHub Repository MUST NOT be modified after lab submission. If a repository is modified after submission of the lab – it will not be graded.