WK4: SIMPLE SOURCE CODE

Part 5

Anila Naz

University of Phoenix

BSA-425: BSIT Capstone

Professor: Dr. Reid

20$^{th}$. July 2024

1. **Login and View Account Balance**:

A simple example of a web application consisting of HTML JavaScript, and CSS code is presented in this document. This example demonstrates a basic web application where users can log in and view their account balance.

- **HTML** (index.html)

```
<!DOCTYPE html>
<html lang=”en’>
<head>
    <meta charset=”UTF8”>
    <meta name=”viewport” content=”width=device-width, initial – scale=1.0”>
    <title>Internet Bank Login</title>
    <link rel=”stylesheet” href=”styles.css”>
</head>
<body>
    <h1>Internet Bank</h1>
    <div id="loginForm">
        <h2>Login</h2>
            <form onsubmit="return login();">
                <label for="username">Username:</label>
                <input type="text" id="username" name="username"
                required><br><br> <label for="password">Password:</label>
                <input type="password" id="password" name="password"
                required><br><br>
                <button type="submit">Login</button>
        </form>
    </div>
    <div id="accountInfo" style="display:none;">
        <h2>Account Information</h2>
        <p id="accountBalance"></p>
        <button onclick="logout();">Logout</button>
    </div>
    <script src="scripts.js"></script>
</body>
</html>
```

- **JavaScript** (scripts.js)

```
function login() {
    var username = document.getElementById('username').value;
    var password = document.getElementById('password').value;

    // Simulating a login process
    if (username === "user" && password === "password") {
        document.getElementById('loginForm').style.display = 'none';
        document.getElementById('accountInfo').style.display = 'block';
        document.getElementById('accountBalance').innerText = 'Account Balance: $1,000';
    } else {
        alert('Invalid credentials');
    }

    return false;
}

function logout() {
    document.getElementById('loginForm').style.display = 'block';
    document.getElementById('accountInfo').style.display = 'none';
    document.getElementById('username').value = ";
    document.getElementById('password').value = ";
}
```

- **CSS (styles.css)**:

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
    padding: 50px;
}

h1 {
    color: #333;
}
```

```
form {
    display: inline-block;
    text-align: left;
    background: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

input {
    margin-bottom: 10px;
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    padding: 10px 20px;
    background: #28a745;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background: #218838;
}
```

**Summary**

- **index.html**: Provides the basic structure of the web page with a login form.
- **scripts.js**: Handles the login functionality and simulates displaying account balance after a successful login.
- **styles.css**: Adds some basic styling to the HTML elements.

2. **Financial Transaction**:

The following code is handling of a financial transaction in a web application. The code includes HTML form for entering transaction details, JavaScript for client-side validation and interaction and Java for server-side processing.

- **HTML (transaction.html)**:

This file provides a form for users to enter transaction details of type of transaction, such as deposit or withdrawal, amount, and account ID.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Transaction</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Make a Transaction</h1>
    <div id="transactionForm">
        <form onsubmit="return submitTransaction();">
            <label for="accountID">Account ID:</label>
            <input type="text" id="accountID" name="accountID" required><br><br>
            <label for="transactionType">Transaction Type:</label>
            <select id="transactionType" name="transactionType" required>
                <option value="deposit">Deposit</option>
                <option value="withdrawal">Withdrawal</option>
            </select><br><br>
            <label for="amount">Amount:</label>
            <input type="number" id="amount" name="amount" step="0.01" required><br><br>
            <button type="submit">Submit</button>
        </form>
    </div>
    <div id="transactionResult" style="display:none;">
        <h2>Transaction Result</h2>
        <p id="resultMessage"></p>
    <button onclick="newTransaction();">New Transaction</button>
  </div>
  <script src="transaction.js"></script>
</body>
</html>
```

- **JavaScript (transaction.js)**:

This script handles the form submission, performs client-side validation and interacts with the server.

```
function submitTransaction() {
  var accountID = document.getElementById('accountID').value;
  var transactionType = document.getElementById('transactionType').value;
  var amount = document.getElementById('amount').value;

  // Basic validation
  if (accountID === "" || amount <= 0) {
    alert("Please enter valid transaction details.");
    return false;
  }

  var xhr = new XMLHttpRequest();
  xhr.open("POST", "TransactionServlet", true);
  xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
```

```
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            document.getElementById('transactionForm').style.display = 'none';
            document.getElementById('transactionResult').style.display = 'block';
            document.getElementById('resultMessage').innerText = xhr.responseText;
        }
    };

    xhr.send("accountID=" + accountID + "&transactionType=" + transactionType + "&amount="
+ amount);
    return false; // Prevent default form submission
}

function newTransaction() {
    document.getElementById('transactionForm').style.display = 'block';
    document.getElementById('transactionResult').style.display = 'none';
    document.getElementById('accountID').value = '';
    document.getElementById('amount').value = '';
}
```

- **Java (TransactionServlet.java)**:

This Java servlet handles the transaction request, processes the transaction, and returns the results.

```
import java.io.IOException;
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("/TransactionServlet")
public class TransactionServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        String accountID = request.getParameter("accountID");
        String transactionType = request.getParameter("transactionType");
        double amount = Double.parseDouble(request.getParameter("amount"));

        // Simulate transaction processing
        String resultMessage;
        if (transactionType.equals("deposit")) {
            resultMessage = "Deposit of $" + amount + " to account " + accountID + " was
successful.";
        } else if (transactionType.equals("withdrawal")) {
            resultMessage = "Withdrawal of $" + amount + " from account " + accountID + " was successful.";
        } else {
            resultMessage = "Invalid transaction type.";
        }

        response.setContentType("text/plain");
        response.getWriter().write(resultMessage);
    }
}
```

**Summary**

- **transaction.html**: Contains the form for entering transaction details.
- **transaction.js**: Manages the form submission and communication with the server.
- **TransactionServlet.java**: Handles server-side processing of the transaction and returns the result.

# REFRENCES:

Chellman, J. (2023, April 3). *JavaScript for Web Designers*. LinkedIn. https://www.linkedin.com/learning/javascript-for-web-designers-3/what-is-javascript?leis=LTI13&u=74653506

Chellman, J. (2023, May 10). *Learning JavaScript Language*. LinkedIn. https://www.linkedin.com/learning/html-essential-training-4/links?leis=LTI13&u=74653506

Envato Tuts+. (n.d.). *HTML & CSS for Beginners*. https://youtu.be/iG2jotQo9NI?si=HTuamvLSRFHQgg78

Peterson, C. (2017, August 17). *HTML & CSS: Creating Forms*. LinkedIn. https://www.linkedin.com/learning/html-css-creating-forms/welcome?leis=LTI13&u=74653506

Simmons, J. (2020, February 19). *HTML & CSS Essential Training*. LinkedIn. https://www.linkedin.com/learning/html-essential-training-4/links?leis=LTI13&u=74653506

W3 Schools. (n.d.). *CSS Tutorial*. https://www.w3schools.com/css/default.asp

W3 Schools. (n.d.). *HTML Tutorial*. Tutorial https://www.w3schools.com/html/default.asp

W3 Schools. (n.d.). *JavaScript Tutorial*. Tutorial https://www.w3schools.com/js/default.asp