

BÀI KIỂM TRA GIỮA KỲ

HỌC PHẦN: Lập trình WEB

I. THÔNG TIN SINH VIÊN

Nhóm: INT1434.N2

STT	MÃ SỐ	HỌ ĐỆM	TÊN	LỚP
76	B24DTCN442	Phạm Hương	Quỳnh	D24TXCN08-B

II. NỘI DUNG BÀI LÀM

- ĐỀ KIỂM TRA SỐ 2

Câu 1.

1. Khái niệm về HTML, CSS và JavaScript

- HTML (HyperText Markup Language): là ngôn ngữ đánh dấu dùng để tạo cấu trúc và nội dung của trang web. Nó xác định các phần như tiêu đề, đoạn văn, hình ảnh, liên kết... HTML chỉ mô tả “cái gì” xuất hiện trên trang, không xử lý logic hay tương tác.
- CSS (Cascading Style Sheets): là ngôn ngữ dùng để định dạng và trình bày nội dung HTML. Nó điều khiển màu sắc, font chữ, bố cục, khoảng cách, hiệu ứng... giúp trang web trở nên đẹp và dễ sử dụng.
- JavaScript: là ngôn ngữ lập trình dùng để tạo tính năng tương tác và động trên trang web. JavaScript có thể xử lý sự kiện, thay đổi nội dung, kiểm tra dữ liệu, gọi API... giúp trang web “sống” và phản ứng với người dùng.

2. Phân biệt vai trò của các ngôn ngữ lập trình web phía Client: HTML, CSS và JavaScript.

Ngôn ngữ	Vai trò chính	Mô tả chi tiết
----------	---------------	----------------

HTML	Cấu trúc	Xác định nội dung và cấu trúc của trang web: tiêu đề, đoạn văn, hình ảnh, liên kết... Nó là “xương sống” của trang web
CSS	Trình bày (giao diện)	Định dạng giao diện: màu sắc, font chữ, khoảng cách, bố cục, responsive. Giúp trang web đẹp và dễ dùng
JavaScript	Tương tác & logic	Tạo tính năng động, xử lý sự kiện, thay đổi nội dung, validate form, gọi API. Giúp trang web “sống” và phản ứng với người dùng

3. Trình bày cách ba ngôn ngữ này tương tác với nhau để tạo nên một trang web hoàn chỉnh.

1. **HTML** tạo **cấu trúc** của trang: xác định các phần tử như tiêu đề, đoạn văn, hình ảnh, nút bấm...
 2. **CSS** định dạng các phần tử HTML: màu sắc, font chữ, khoảng cách, bố cục, responsive... làm trang web “đẹp” và dễ nhìn.
 3. **JavaScript** thêm **tương tác**: xử lý sự kiện người dùng (nhấn nút, nhập dữ liệu), thay đổi nội dung động, validate form, tạo hiệu ứng...
- Lấy ví dụ minh họa về một trang web đơn giản và giải thích vai trò của mỗi ngôn ngữ trên trang đó.

❖ Ví dụ minh họa: Trang web “Chào mừng người dùng”

- index.html

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
```

```

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Form inputmode</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>Hiển thị chính xác Bàn phím trên Mobile với </h1>
<code>inputmode</code>
<h2>Note: Kiểm tra chúng trên di động để thấy sự khác
nhau!!</h2>

<form id="demoForm">
    <div class="form-group">
        <label for="name">Name:</label>
        <input id="name" inputmode="text"
placeholder="Nhập tên của bạn">

        <label for="email">Email:</label>
        <input id="email" inputmode="email" type="email"
placeholder="Nhập email">

        <label for="tel">Telephone:</label>
        <input id="tel" inputmode="tel" placeholder="Nhập
số điện thoại">

        <label for="url">Url:</label>
        <input id="url" inputmode="url" type="url"
placeholder="Nhập địa chỉ website">

        <button type="submit">Gửi</button>
    </div>
</form>

<div id="message"></div>

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

```

- styles.css

```

body {
    background-color: lightgoldenrodyellow;
    display: flex;

```

```
    flex-direction: column;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    margin: 0;
    font-family: "Berkshire Swash", cursive;
}

h1 {
    font-size: 2.5rem;
    color: salmon;
    text-align: center;
    margin-bottom: 10px;
}

h2 {
    font-size: 1.4rem;
    color: #333;
    text-align: center;
    margin-bottom: 20px;
}

code {
    background: salmon;
    color: #fff;
    border-radius: 5px;
    padding: 3px 6px;
    font-size: 1rem;
}

form {
    background-color: #fff;
    padding: 25px 35px;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    width: 270px;
}

form div {
    display: flex;
    flex-direction: column;
    gap: 15px;
}

label {
    font-size: 1rem;
```

```
    color: #333;
}

input {
    padding: 8px 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 1rem;
}

input:focus {
    border-color: salmon;
    outline: none;
}

button {
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    background-color: salmon;
    color: white;
    cursor: pointer;
    font-size: 1rem;
    margin-top: 10px;
}

button:hover {
    background-color: darkred;
}

#message {
    margin-top: 20px;
    font-size: 1.2rem;
    color: green;
}
```

- script.js

```
const form = document.getElementById('demoForm')
const message = document.getElementById('message')

form.addEventListener('submit', function(e) {
    e.preventDefault()
    const name = document.getElementById('name').value.trim()
    const email = document.getElementById('email').value.trim()
```

```

const tel = document.getElementById('tel').value.trim()
const url = document.getElementById('url').value.trim()

if(!name || !email || !tel || !url) {
    message.style.color = 'red'
    message.textContent = 'Vui lòng điền đầy đủ thông tin!'
    return
}

message.style.color = 'green'
message.textContent = `Cảm ơn ${name}! Thông tin của bạn đã
được ghi nhận.`

form.reset()
})

```

- Kết quả (khi bấm “Gửi” nhưng input rỗng):

Hiện thị chính xác Bàn phím trên Mobile với

inputmode

Note: Kiểm tra chúng trên di động để thấy sự khác nhau!!

Name:

Email:

Telephone:

Url:

Vui lòng điền đầy đủ thông tin!

- Vai trò của 3 ngôn ngữ html, css, js trên trang đó.

Thành phần	Vai trò	Ví dụ trên trang
HTML	Cấu trúc nội dung, xác định các phần tử tương tác	<ul style="list-style-type: none"> - <code><h1></code>, <code><h2></code> → tiêu đề hướng dẫn - <code><form></code> + <code><label></code> + <code><input></code> + <code><button></code> → form nhập liệu - <code><div id="message"></code> → hiển thị thông báo
CSS	Trang trí và bố cục, định dạng các phần tử, tạo trải nghiệm trực quan	<ul style="list-style-type: none"> - body căn giữa form; - form background, padding, border-radius; - input:focus đổi màu viền; - button: hover hiệu ứng hover
JavaScript	Tương tác động, xử lý dữ liệu người dùng, phản hồi ngay trên trang	<ul style="list-style-type: none"> - Bắt sự kiện submit form; - Kiểm tra input rỗng; - Hiển thị thông báo lỗi hoặc thành công; - Reset form

Câu 2.

1. Khái niệm và vai trò của XML trong lập trình web

Khái niệm	Vai trò
- XML (eXtensible Markup Language) là ngôn ngữ đánh dấu mở rộng, dùng để mô tả và lưu trữ dữ liệu theo cấu trúc cây, có thể đọc được bởi con người và máy tính	<ul style="list-style-type: none"> - Lưu trữ dữ liệu: lưu trữ thông tin có cấu trúc như danh sách sản phẩm, dữ liệu người dùng - Trao đổi dữ liệu: giữa các hệ thống khác nhau (Web service, API) - Tách dữ liệu và hiển thị: dữ liệu được lưu riêng, HTML/CSS/JS chỉ dùng để hiển thị

2. So sánh XML và HTML

Tiêu chí	HTML	XML
Mục đích	Hiển thị nội dung trên web	Lưu trữ và trao đổi dữ liệu
Cú pháp	Có tập hợp thẻ cố định (<h1>, <p>, <a>, ...)	Thẻ do người dùng tự định nghĩa (<product>, <name>, <price>)
Tính linh hoạt	Thấp, chỉ dùng cho trình duyệt	Cao, có thể mô tả bất kỳ dữ liệu nào
Kiểm tra hợp lệ	Không bắt buộc XML schema	Có thể sử dụng DTD hoặc XSD để kiểm tra tính hợp lệ

3. Khi nào nên dùng XML thay vì HTML là:

- Khi cần trao đổi dữ liệu giữa các hệ thống (Web service, API, mobile ↔ server)
- Khi dữ liệu có cấu trúc phức tạp (danh sách, cây phân cấp, nhiều thuộc tính)
- Khi muốn tách dữ liệu và hiển thị (HTML/JS/CSS chỉ hiển thị, XML lưu trữ dữ liệu)
- Khi cần tính linh hoạt cao, có thể mở rộng thẻ tùy theo nhu cầu ứng dụng

Câu 3.

- Giải thích và phân tích:

1. File login.html:

```
<!DOCTYPE html>
<!-- Khai báo tài liệu HTML5, giúp trình duyệt nhận diện định dạng HTML chuẩn. -->
<html>
<!-- Thẻ bắt đầu tài liệu HTML, chứa toàn bộ nội dung trang web. -->
<head>
  <!-- Thẻ chứa các siêu dữ liệu và tài nguyên liên quan như tiêu đề, CSS, v.v. -->
  <meta charset="UTF-8">
  <!-- Thiết lập mã hóa ký tự UTF-8 để hỗ trợ hiển thị tiếng Việt. -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Thiết lập viewport để trang web responsive, điều chỉnh chiều rộng theo thiết bị và tỷ lệ zoom ban đầu là 1. -->
```



```

-->
    <link rel="stylesheet" href="style.css">
    <!-- Liên kết đến file CSS bên ngoài (style.css) để định
kiểu giao diện form. -->
</head>
<!-- Kết thúc thẻ head, bắt đầu phần nội dung hiển thị. -->
<body>
<!-- Thẻ chứa nội dung chính của trang web, hiển thị trên
trình duyệt. -->
<div>
    <!-- Thẻ div nhóm nội dung form đăng nhập, có thể áp dụng
kiểu CSS. -->
    <h2>ĐĂNG NHẬP</h2>
    <!-- Thẻ h2 hiển thị tiêu đề cấp 2 "ĐĂNG NHẬP" cho form.
-->
    <form action="welcome.jsp" method="post">
        <!-- Thẻ form gửi dữ liệu đến file "welcome.jsp" bằng
phương thức POST, dữ liệu không hiển thị trên URL. -->
        <label for="username">Tên đăng nhập :</label>
        <!-- Thẻ label liên kết với trường nhập tên đăng nhập
qua thuộc tính "for", hiển thị văn bản "Tên đăng nhập :". -->
        <input type="text" id="username" name="username"
required>
        <!-- Thẻ input tạo trường nhập văn bản với
id="username" để liên kết với label, name="username" để gửi
dữ liệu, thuộc tính "required" bắt buộc nhập trước khi gửi
form. -->
        <label for="password">Mật khẩu :</label>
        <!-- Thẻ label liên kết với trường nhập mật khẩu qua
thuộc tính "for", hiển thị văn bản "Mật khẩu :". -->
        <input type="password" id="password" name="password"
required>
        <!-- Thẻ input tạo trường nhập mật khẩu, hiển thị ký
tự dạng dấu *, với id="password" để liên kết với label,
name="password" để gửi dữ liệu, thuộc tính "required" bắt
buộc nhập. -->
        <input type="submit" value="Đăng nhập">
        <!-- Thẻ input tạo nút gửi form, với type="submit" để
kích hoạt gửi dữ liệu, value="Đăng nhập" hiển thị nhãn nút.
-->
    </form>
    <!-- Kết thúc thẻ form. -->
</div>
<!-- Kết thúc thẻ div chứa form đăng nhập. -->
</body>
<!-- Kết thúc thẻ body, kết thúc nội dung chính của trang.
-->
</html>
<!-- Kết thúc tài liệu HTML. -->

```

2. File welcome.jsp:

```
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!-- Chỉ thị JSP xác định nội dung trả về là HTML với mã hóa
UTF-8 để hỗ trợ tiếng Việt và ký tự đặc biệt. -->
<!DOCTYPE html>
<!-- Khai báo tài liệu HTML5, giúp trình duyệt nhận diện định
dạng HTML chuẩn. -->
<html>
<!-- Thẻ bắt đầu tài liệu HTML, chứa toàn bộ nội dung trang
web. -->
<head>
<!-- Thẻ chứa các siêu dữ liệu và tài nguyên liên quan như
tiêu đề, CSS, v.v. -->
    <meta charset="UTF-8">
    <!-- Thiết lập mã hóa ký tự UTF-8 để trình duyệt hiển thị
đúng ký tự tiếng Việt. -->
    <title>Chào mừng</title>
    <!-- Đặt tiêu đề trang là "Chào mừng", hiển thị trên tab
trình duyệt. -->
    <link rel="stylesheet" href="style.css">
    <!-- Liên kết đến file CSS bên ngoài (style.css) để định
kiểu giao diện trang. -->
</head>
<!-- Kết thúc thẻ head, bắt đầu phần nội dung hiển thị. -->
<body>
<!-- Thẻ chứa nội dung chính của trang web, hiển thị trên
trình duyệt. -->
    <div class="welcome-container">
    <!-- Thẻ div với class "welcome-container" để áp dụng kiểu
CSS, nhóm nội dung chào mừng. -->
        <h1>Chào mừng, <%= request.getParameter("username")
%>!</h1>
        <!-- Thẻ h1 hiển thị tiêu đề cấp 1. JSP expression
<%= request.getParameter("username") %> lấy giá trị tham số
"username" từ request (dữ liệu từ form) và hiển thị trong
tiêu đề. -->
    </div>
    <!-- Kết thúc thẻ div chứa nội dung chào mừng. -->
</body>
<!-- Kết thúc thẻ body, kết thúc nội dung chính của trang.
-->
</html>
<!-- Kết thúc tài liệu HTML. -->
```

- Luồng xử lý khi người dùng nhập tên và nhấn nút "Đăng nhập" là:

-
1. Hiển thị form đăng nhập:
 - Trình duyệt tải file HTML, hiển thị form với hai trường nhập: "Tên đăng nhập" và "Mật khẩu", cùng nút "Đăng nhập". Giao diện được định kiểu bởi style.css.
 2. Người dùng nhập dữ liệu và gửi:
 - Người dùng nhập giá trị vào trường username (ví dụ: "Quynh") và password (ví dụ: "123456").
 - Thuộc tính required đảm bảo cả hai trường phải được điền trước khi gửi.
 - Khi nhấn nút "Đăng nhập", trình duyệt gửi yêu cầu HTTP POST đến welcome.jsp với dữ liệu: username=Quynh và password=123456.
 3. Xử lý tại server (welcome.jsp):-
 - Server (ví dụ: Apache Tomcat) nhận yêu cầu POST và chạy welcome.jsp.
 - Chỉ thị `<%@ page contentType="text/html" pageEncoding="UTF-8"%>` báo cho server trả về nội dung HTML với mã hóa UTF-8.
 - Dòng `<%= request.getParameter("username") %>` lấy giá trị username (ví dụ: "Quynh") từ request.
 - ➔ Server tạo HTML động với nội dung: `<h1>Chào mừng, Quynh!</h1>`.
 4. Hiển thị trang chào mừng:
 - Server gửi HTML được tạo về trình duyệt.
 - Trình duyệt hiển thị trang với tiêu đề "Chào mừng, Quynh!" được định kiểu bởi style.css.
 - Mật khẩu được gửi nhưng không được sử dụng trong welcome.jsp.
 5. Trường hợp lỗi:
 - Nếu người dùng không nhập username, thuộc tính required ngăn gửi form.
 - Nếu có lỗi trong request (ví dụ: không gửi username), `request.getParameter("username")` trả về null, dẫn đến hiển thị "Chào mừng, null!".
 - Giao diện khi chạy chương trình hiển thị:

ĐĂNG NHẬP

Tên đăng nhập :

Quynh

Mật khẩu :

Đăng nhập

Chào mừng, Quynh!

=====END=====