

3

Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Web Front-end (nâng cao)

Thực hành môn Lập trình ứng dụng Web

Tháng 4/2025

Lưu hành nội bộ

<Nghiêm cấm đăng tải trên internet dưới mọi hình thức>

A. TỔNG QUAN

1. Mục tiêu

- Có thể sử dụng HTML, CSS, JavaScript để lập trình giao diện cho websites.
- Làm quen với các cấu trúc cơ bản của JavaScript.
- Xây dựng website hoàn chỉnh có cấu trúc rõ ràng và tính thẩm mỹ cao.

2. Thời gian thực hành

- Thực hành tại lớp: 5 tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa 13 ngày.

3. Liên quan

- Sinh viên cần nắm các kiến thức nền tảng về HTML, CSS và JavaScript. Các kiến thức này đã được giới thiệu trong nội dung lý thuyết đã học do đó sẽ không được trình bày lại trong nội dung thực hành này.
- Tham khảo thêm tại website w3school.com để nắm vững kiến thức.

B. CHUẨN BỊ MÔI TRƯỜNG

- Cài đặt trình soạn thảo [Visual Studio Code](https://code.visualstudio.com/) hoặc các trình soạn thảo tương tự.
- Cài đặt trình duyệt web (Chrome, Firefox, Edge,...) để quan sát kết quả.

C. THỰC HÀNH

1. Giới thiệu JavaScript

JavaScript là gì?

JavaScript (JS) là một ngôn ngữ lập trình được sử dụng để tạo ra các trang web tương tác và động. JavaScript hỗ trợ phát triển cả máy khách (client-side) và máy chủ (server-side), đồng thời tích hợp liền mạch với HTML và CSS.

- HTML để xác định nội dung của trang web.
- CSS để chỉ định bố cục của trang web.
- JavaScript để lập trình hành vi của các trang web.

JavaScript dùng để làm gì?

Trước đây, các trang web có dạng tĩnh, tương tự như các trang trong một cuốn sách, chủ yếu hiển thị thông tin theo một bố cục cố định. JavaScript dần được biết đến như một công nghệ phía trình duyệt để làm cho các ứng dụng web linh hoạt hơn. Sử dụng JavaScript, các trình duyệt có thể phản hồi tương tác của người dùng và thay đổi bố cục của nội dung trên trang web. JavaScript giúp giảm tải cho server và tắc nghẽn mạng vì nó có thể chạy các phép toán logic và thực hiện nhiều công việc của server trên chính clients. Một số trường hợp ứng dụng phổ biến có thể kể tới là:

- Phát triển Web front-end
- Phát triển Web back-end
- Phát triển ứng dụng di động
- Phát triển ứng dụng web
- Lập trình game

JavaScript hoạt động như thế nào?

Cách thức hoạt động của JavaScript tại client-side:

1. Trình duyệt tải website khi bạn truy cập đến.
2. Trong khi tải, trình duyệt chuyển đổi trang và tất cả các phần tử của trang, như các buttons, labels và dropdown, thành một cấu trúc dữ liệu được gọi là Mô hình Đối tượng Tài liệu (DOM).
3. JavaScript engine của trình duyệt chuyển đổi mã JavaScript thành bytecode, đóng vai trò là trung gian giữa JavaScript syntax và máy.
4. Các sự kiện khác nhau, như nhấp vào nút, sẽ kích hoạt việc thực thi khối mã JavaScript được liên kết. Sau đó, engine sẽ diễn giải bytecode và thực hiện các thay đổi đối với DOM.
5. Trình duyệt hiển thị DOM mới.

Nhược điểm của JavaScript

- Vấn đề bảo mật: có thể bị hacker khai thác để thực thi mã độc ở trên máy tính của người dùng (cross-site scripting).
- Khó kiểm soát lỗi: Do JavaScript không khai báo kiểu dữ liệu cố định như Java hoặc C++.

Cách sử dụng JavaScript

Cách 1: Nhúng trực tiếp trong cặp thẻ `<script>` `</script>` của HTML

```
<!DOCTYPE html>
<html>
<body>
  <h2>I'm a Website</h2>
  <p id="name">Hello HTML!</p>
  <button onclick="sayHello()">Click Me!</button>
  <script>
    function sayHello() {
      document.getElementById('name').innerHTML = 'Hello JavaScript!';
    }
  </script>
</body>
</html>
```

Cách 2: Liên kết file `.js` bên ngoài

```
<body>
<script src="myScript.js"></script>
</body>
```

Lưu ý: Nên đặt code JavaScript ở phần dưới cùng của thẻ `<body>` để cải thiện tốc độ hiển thị, vì việc biên dịch JavaScript code sẽ làm chậm việc hiển thị nội dung của trang web.

2. Luyện tập với JavaScript

a) JavaScript Function

Mã bên trong hàm sẽ được thực thi khi "something" gọi tới hàm:

- Khi một sự kiện xảy ra (vd: khi người dùng nhấp vào nút)
- Khi nó được called từ mã JavaScript
- Tự động (self invoked)

```
<button onclick="handleClick()">Click Me</button>
<script>
  // 1. Khi có sự kiện xảy ra
  function handleClick() {
    console.log("Button clicked! Function invoked by event.");
  }

  // 2. Khi hàm được gọi từ mã JavaScript
  function invokedFromCode() {
    console.log("Function invoked from JavaScript code.");
  }
  invokedFromCode(); // Gọi trực tiếp từ mã JS

  // 3. Hàm tự động thực thi
  (function selfInvoked() {
    console.log("Function invoked automatically (self-invoked).");
  })();
</script>
```

So sánh:

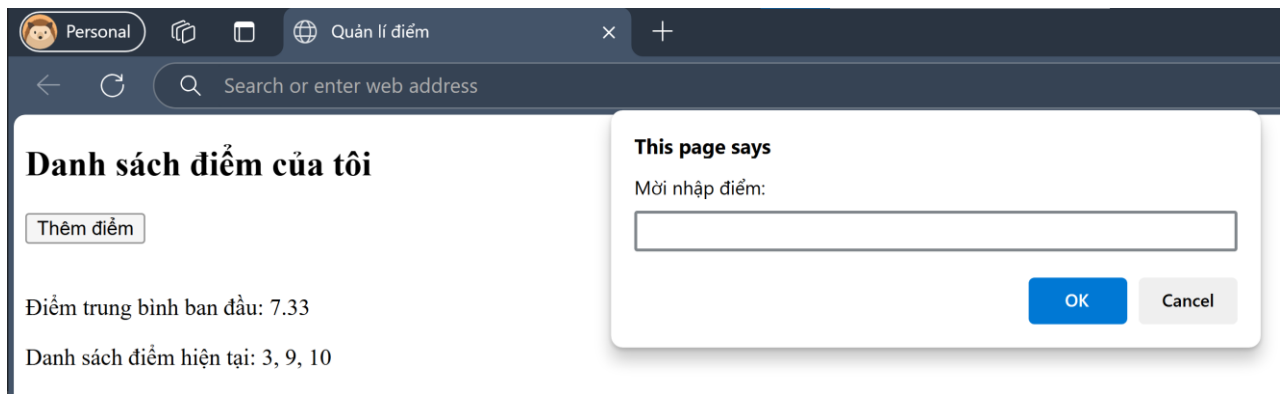
Đặc điểm	Loại 1	Loại 2	Loại 3
Cách gọi	Được kích hoạt bởi sự kiện (click, hover, input, v.v.)	Gọi bằng tên hàm	Thực thi ngay lập tức
Thời điểm thực thi	Khi sự kiện xảy ra	Khi mã chạy đến lệnh gọi hàm	Ngay khi trình duyệt tải mã
Tái sử dụng	Có thể gọi nhiều lần khi sự kiện xảy ra	Có thể gọi lại nhiều lần	Chỉ chạy một lần
Ứng dụng	Phản hồi thao tác của người dùng	Chủ động điều khiển việc thực thi	Muốn thực thi ngay lập tức

Task 1: Viết chương trình Quản lý điểm với các yêu cầu sau:

- Khởi tạo mảng (array) điểm ban đầu.
- Hàm 1 – Điểm trung bình ban đầu: Tính điểm trung bình từ danh sách trên và hiện thị ngay khi user tải trang.
- Hàm 2 – Sự kiện thêm điểm: Khi user click button thêm điểm, cho phép user nhập thêm điểm (có kiểm tra type Number, giá trị từ 0 đến 10), cảnh báo (alert) nếu người dùng nhập sai, nếu đúng thì thêm điểm mới vào mảng điểm.
- Hàm 3 – Danh sách điểm hiện tại: Hiển thị danh sách điểm (ban đầu và sau khi thêm điểm), các phần tử ngăn cách nhau bởi dấu phẩy + khoảng trắng [,].

Gợi ý: Tìm hiểu các hàm JavaScript reduce(), prompt(), push(), join()... JS array

Giao diện minh họa:



b) JavaScript Objects

JavaScript objects (đối tượng) cũng là biến nhưng có thể chứa nhiều giá trị. Objects là một tập hợp các thuộc tính (properties) và phương thức (methods). Mỗi thuộc tính có dạng `key: value`.

```
let student = {
  name: "Nam", // Thuộc tính (Property)
  age: 20, // Thuộc tính (Property)
  scores: [3, 9, 10], // Thuộc tính (Property - Array)
  greet: function() { // Phương thức (Method)
    return "Hello!";
  }
};
```

Để truy cập thuộc tính ta dùng 2 cách sau:

- Dùng dấu chấm `.`: `student.name`
- Dùng dấu ngoặc vuông `[]`: `student["age"]`

Để truy cập phương thức: `student.greet()`

Task 2: Viết chương trình Quản lý học sinh, tạo object student có các thuộc tính gồm: tên, tuổi và mảng điểm. Chương trình có các tính năng chính sau:

- Thêm điểm: Người dùng nhập điểm từ 0-10, có kiểm tra hợp lệ và đưa cảnh báo.
- Xóa điểm cuối: Nếu danh sách có điểm, xóa phần tử cuối. Nếu rỗng, báo lỗi.
- Xóa điểm đầu: Nếu danh sách có điểm, xóa phần tử đầu. Nếu rỗng, báo lỗi.
- Hiển thị điểm trung bình.
- Hiển thị điểm cao nhất/thấp nhất: Nếu không có phần tử nào, hiển thị “-”
- Hiển thị danh sách điểm: Nếu không có phần tử nào, hiển thị “Chưa có điểm”
- Nội dung cập nhật tự động: Khi điểm thay đổi, tất cả thông tin sẽ hiển thị lại ngay lập tức.

Gợi ý: Tìm hiểu các hàm `pop()`, `shift()`, `Math.max()`, `Math.min()`...

Giao diện minh họa:

Quản lý sinh viên

Họ tên: Nguyễn Thị Văn A

Tuổi: 21

Điểm trung bình: 7.33

Điểm cao nhất: 10

Điểm thấp nhất: 3

Danh sách điểm: 3, 9, 10

c) JavaScript object constructor

JavaScript object constructors là các hàm dùng để tạo nhiều đối tượng có cùng cấu trúc. Chúng hoạt động như một "khuôn mẫu" để khởi tạo đối tượng với các thuộc tính và phương thức.

```
<script>
function Person(first, last) {
    this.firstName = first;
    this.lastName = last;
    this.nationality = "Vietnamese";
    this.fullName = function() {
        return this.firstName + " " + this.lastName;
    };
}
</script>
```

Đây là một hàm tạo (constructor function) để tạo đối tượng `Person`.

- Hàm nhận 2 tham số (parameters) là `first` và `last`.
- `firstName` và `lastName` là thuộc tính của đối tượng, nhận giá trị từ `first` và `last` truyền vào.
- `nationality` là thuộc tính của đối tượng, có giá trị mặc định là "Vietnamese".
- `fullname()` là phương thức của đối tượng.

Task 3: Viết chương trình quản lý danh sách sinh viên, với các yêu cầu sau:

- 1. Tạo constructor** Student chung cho các đối tượng student (sinh viên) gồm thuộc tính:
 - `fullname` (Họ và tên đầy đủ, dạng chuỗi)
 - `scores` (mảng điểm)
 và các phương thức:
 - Tính điểm trung bình
 - Tìm điểm cao nhất
- 2. Button** Thêm sinh viên, khi nhấn vào, người dùng cần nhập các thông tin:
 - Họ tên đầy đủ
 - Mảng điểm (các điểm cách nhau bởi dấu phẩy và từ 1 đến 10)

3. Lưu danh sách sinh viên với **localStorage**, khi này dữ liệu sẽ được giữ dù tải lại trang. Đọc thêm tại [đây](#).

Gợi ý: localStorage chỉ hỗ trợ lưu trữ dữ liệu dưới dạng string tại bộ nhớ cục bộ của trình duyệt, khi lưu object có thể khiến phương thức của object gặp lỗi khi lấy lên sử dụng. Do đó chuyển dữ liệu thành dạng JSON, cú pháp `JSON.stringify()`, để lưu. Và chuyển lại thành object khi lấy ra `JSON.parse()`

Code tham khảo:

```
// Tạo constructor
function Student(fullName, scores) {
    this.fullName = formatName(fullName); // Gọi hàm formatName()
    this.scores = scores;
    Method A;
    Method B;
}

// Mảng toàn cục lưu danh sách sinh viên
let students = [];

// Lấy dữ liệu từ localStorage và chuyển đổi thành Student objects
function loadStudents() {
    // Nếu dữ liệu NULL gán mảng rỗng [] cho students
    students = JSON.parse(localStorage.getItem("savedStudents")) || [];
    // Duyệt mảng students, khôi phục dữ liệu thành đối tượng Student
    students = students.map(s => new Student(s.fullName, s.scores));
}

// Lưu danh sách vào localStorage dưới dạng JSON
function saveToLocalStorage() {
    localStorage.setItem("savedStudents", JSON.stringify(students));
}
```

4. Hàm hiển thị danh sách sinh viên:

- Họ tên (chuẩn hóa, viết hoa chữ cái đầu mỗi từ)
- Điểm trung bình
- Điểm cao nhất
- Button xóa sinh viên đó
- Ở mỗi sinh viên sẽ có link điều hướng tới trang Thông tin chi tiết của sinh viên đó

• Nguyễn Văn A - Điểm TB: 6.50 - Điểm cao nhất: 9	Xóa
• Bùi Thị B - Điểm TB: 6.50 - Điểm cao nhất: 9	Xóa
• Trần Kim C - Điểm TB: 4.60 - Điểm cao nhất: 6	Xóa

Gợi ý:

- Tìm hiểu **Template Literals** để hiển thị nội dung gồm chuỗi, biến, biểu thức,...
- Hàm `splice()` cho thêm/xóa phần tử trong mảng; save lại localStorage

Code tham khảo:

```
// Hiển thị sinh viên
function displayStudents() {
    let list = document.getElementById("ID");
    list.innerHTML = "";
    students.forEach((student, index) => {
```

```

let li = document.createElement("li");
// Sử dụng backtick (`) để tạo Template Literals,
// Sử dụng ${} để chèn biến/biểu thức
li.innerHTML = `
    // Điều hướng tới trang mới khi nhấn vào tên
    <strong onclick="viewInfo(${index})" style="cursor:pointer; color:blue;">
        ${student.fullName}
    </strong>
    - Điểm TB: ${Method A}
    - Điểm cao nhất: ${Method B}
    <button onclick="Function xóa">Xóa</button>
`;

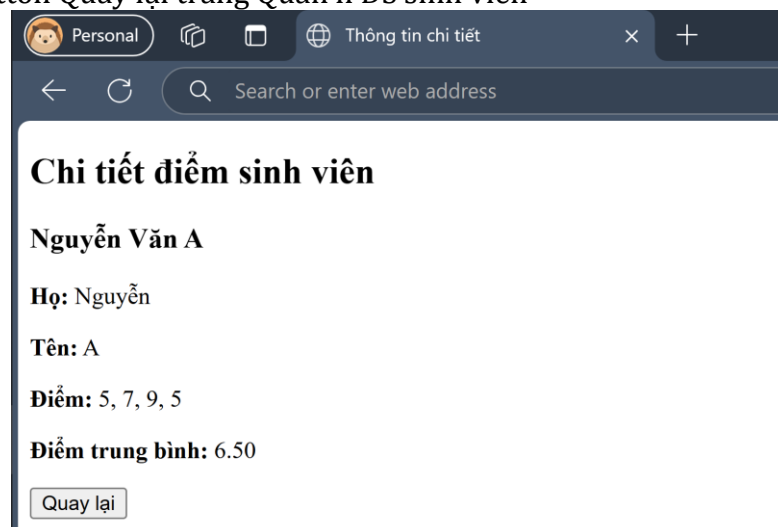
// Thêm phần tử li vừa tạo vào studentList và hiển thị
// thông tin sinh viên lên giao diện
list.appendChild(li);
}
);
}

// Điều hướng trang
function viewInfo(index) {
    // Lưu dữ liệu của SV có chỉ số index vào localStorage với key selectedStudent
    localStorage.setItem("selectedStudent", JSON.stringify(students[index]));
    // Chuyển hướng trình duyệt tới task3_info.html
    window.location.href = "task3_info.html";
}

```

5. Trang chuyển hướng Thông tin chi tiết cho mỗi sinh viên được thêm vào:

- Họ tên đầy đủ
- Họ (xử lý chuỗi)
- Tên (xử lý chuỗi)
- Mảng điểm
- Điểm trung bình
- Button Quay lại trang Quản lý DS sinh viên



6. Button sắp xếp theo điểm trung bình, Tăng và giảm dần trong cùng 1 button

7. Button xóa, xóa toàn bộ danh sách sinh viên, dữ liệu trong localStorage

8. Nội dung cập nhật tự động: Khi có thay đổi, tất cả thông tin sẽ được cập nhật và hiển thị lại ngay lập tức. Chương trình hoạt động logic, rõ ràng, có kiểm tra dữ liệu đầu vào.
Cộng điểm: Website có sử dụng css, bootstrap để tạo độ thẩm mỹ cao.

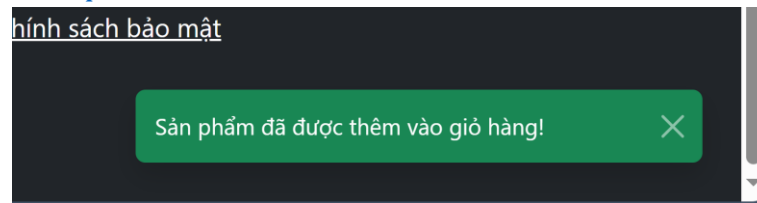
3. Bài tập về nhà

a) Xử lý sự kiện trong JavaScript - Bootstrap

Task 4: Cải tiến cho trang web Thương mại điện tử đã thực hiện tại task 3 – Lab 2. Sử dụng JavaScript và bootstrap để hoàn thành các yêu cầu tính năng sau:

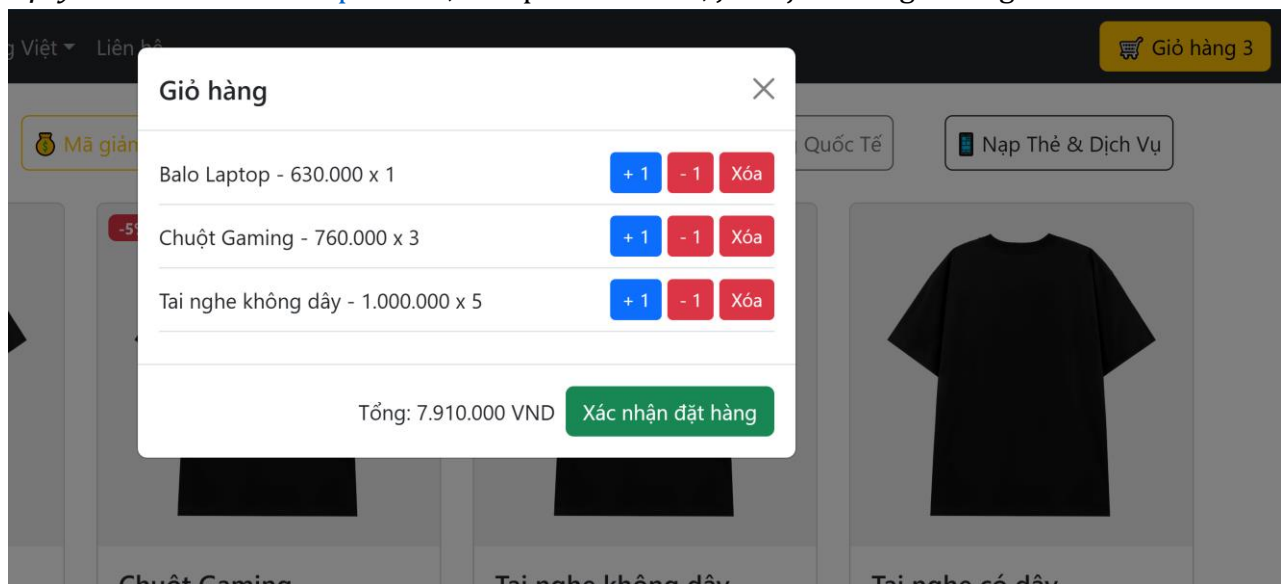
- Thông báo nổi (Toast) khi người dùng nhấn Mua ngay sản phẩm bất kì.

Gợi ý: Tìm hiểu [Bootstrap toast](#)



- Cập nhật số lượng mặt hàng trên giỏ hàng sau khi Mua ngay.
- Xuất hiện hộp thoại Thông tin giỏ hàng khi nhấn vào button giỏ hàng, có các thông tin:
 - Danh sách sản phẩm đã chọn, gồm tên, giá, số lượng.
 - Button tăng/giảm số lượng sản phẩm, xóa sản phẩm.
 - Hiện thị tổng giá trị các sản phẩm trong giỏ hàng.
 - Button xác nhận đặt hàng. Thông báo thành công khi giỏ hàng có sản phẩm và cảnh báo khi giỏ hàng rỗng.
 - Sử dụng Modal của bootstrap để thiết kế hộp thoại.

Gợi ý: Tìm hiểu [Bootstrap Modal](#), Template Literals, JS Object cho giỏ hàng.



b) Xử lý biểu mẫu và nâng cao trong JavaScript - Bootstrap

Task 5: Cải tiến cho trang web Bán vé sự kiện âm nhạc đã thực hiện tại task 4 – Lab 2. Sử dụng JavaScript và bootstrap để hoàn thành các yêu cầu tính năng sau:

1. Cải tiến Form đăng kí mua vé, gồm:

- Họ tên, số điện thoại, email, chọn loại vé, số lượng vé (đặt giới hạn)
- Dùng bootstrap validation để kiểm tra và hiện thị lỗi theo thời gian thực; báo lỗi nhập sai cho đến khi nhập đúng.
- Tính tổng tiền vé.
- Lưu thông tin vào localStorage sau khi submit form để giữ lại dữ liệu khi tải lại trang.
- Chặn gửi và cảnh báo khi nhập dữ liệu không hợp lệ trong form

Gợi ý: Tìm hiểu [Bootstrap Forms](#)

2. Thêm bộ đếm ngược thời gian sự kiện

- Hiện thị thời gian còn lại theo ngày, giờ, phút, giây
- Đồng hồ cập nhật mỗi giây

Gợi ý: Tìm hiểu [JavaScript Dates](#)

Về sự kiện

Concert 2025 là sự kiện âm nhạc hoành tráng với những tiết mục đẳng cấp.

Concert bắt đầu sau:

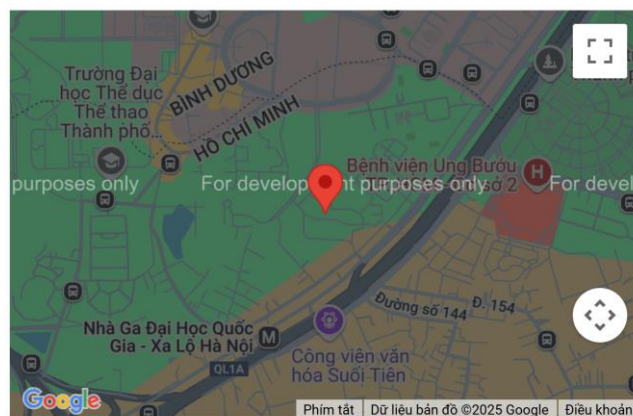
72 ngày 1 giờ 48 phút 56 giây

3. Tích hợp bản đồ Google Maps

- Hiện thị vị trí của nơi diễn ra sự kiện , ví dụ tại UIT
- Button Chỉ đường, dẫn link tới địa chỉ trên Google map

Gợi ý: Dùng [Maps JavaScript API](#)

Bản đồ



Task 6: Trong phần hướng dẫn của task 5, API key của Google Maps được hardcode trực tiếp vào mã nguồn, việc này tiềm ẩn những vấn đề về bảo mật. Vì bất kỳ ai truy cập trang web đều có thể xem mã nguồn của website và đánh cắp key, dẫn đến việc sử dụng trái phép API, làm tăng chi phí hoặc vượt hạn mức sử dụng. Hãy đề xuất tối thiểu 3 giải pháp để khắc phục vấn đề nêu trên, trình bày chi tiết cách thực hiện và giải thích. **Cộng điểm:** Triển khai thực tế giải pháp mà sinh viên đã đề xuất, có báo cáo bước thực hiện hoặc video minh chứng.



D. YÊU CẦU & ĐÁNH GIÁ

- Sinh viên tìm hiểu và thực hành theo hướng dẫn, thực hiện theo nhóm đã đăng ký.
- Nộp báo cáo kết quả gồm chi tiết những việc (Report) mà nhóm đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Báo cáo:
 - File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
 - Đặt tên theo định dạng: [Mã lớp]-LabX_NhomY.
 - Ví dụ: [NT208.P12.ANTT.1]-Lab1_Nhom1.
 - Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
 - Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT

Chúc các bạn hoàn thành tốt!