

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN
THÔNG**



**BÁO CÁO THỰC TẬP CƠ SỞ
TUẦN 3**

**ĐỀ TÀI
XÂY DỰNG WEBSITE BÃI ĐỖ XE THÔNG
MINH**

Giảng viên hướng dẫn :	Kim Ngọc Bách
Họ và tên :	Trần Xuân Kiên
Mã sinh viên :	B22DCVT269
Lớp :	E22CQCN02-B

I. Những công nghệ dùng trong hệ thống.

1. HTML

a. Định nghĩa.

- HTML (HyperText Markup Language) là ngôn ngữ đánh dấu dùng để tạo trang web trên World Wide Web. Nó kết hợp với CSS để tạo giao diện và JavaScript để xử lý hành vi trang web.
- Đặc điểm chính của HTML:
 - Cấu trúc trang web: HTML mô tả bố cục và nội dung của trang.
 - Phần tử HTML: Gồm các thẻ như `<p>` (đoạn văn), `` (hình ảnh), `<a>` (liên kết), `<input />` (biểu mẫu)...
 - Trình duyệt web: Nhận tài liệu HTML từ máy chủ và hiển thị thành trang web đa phương tiện.
 - Tương tác với CSS & JavaScript:
 - CSS giúp định dạng giao diện trang web.
 - JavaScript điều khiển hành vi và nội dung động.
- W3C (World Wide Web Consortium) khuyến khích sử dụng CSS thay vì HTML để thiết kế giao diện từ năm 1997.

b. Lịch sử phát triển của HTML.

- 1989: Tim Berners-Lee, một nhà khoa học tại CERN, đề xuất ý tưởng về World Wide Web (WWW).
- 1991: Ông tạo ra phiên bản đầu tiên của HTML (HTML 1.0), chỉ gồm các thẻ cơ bản như `<p>`, `<a>`, `<h1>`...
- 1995: HTML 2.0 ra đời, bổ sung biểu mẫu (`<form>`) và bảng (`<table>`).
- 1997: HTML 3.2 được W3C phát hành, hỗ trợ JavaScript và CSS.
- 1999: HTML 4.01 trở thành tiêu chuẩn, tách biệt nội dung (HTML) và định dạng (CSS).
- 2004: Nhóm WHATWG phát triển HTML5 để cải thiện khả năng đa phương tiện.
- 2014: HTML5 chính thức ra mắt, hỗ trợ video (`<video>`), audio (`<audio>`) và ứng dụng web hiện đại.

c. Cấu trúc trang HTML.

```
<!DOCTYPE html>
<html lang="vi">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Trang Web Mẫu</title>
  <link rel="stylesheet" href="/style.css"> <!-- Liên kết file CSS
-->
</head>
<body>

  <header>
    <h1>Chào mừng đến với trang web của tôi</h1>
    <nav>
      <ul>
        <li><a href="#">Trang chủ</a></li>
        <li><a href="#">Giới thiệu</a></li>
        <li><a href="#">Dịch vụ</a></li>
        <li><a href="#">Liên hệ</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <h2>Giới thiệu</h2>
      <p>Đây là một trang web HTML cơ bản với cấu trúc đầy
đủ.</p>
    </section>

    <section>
      <h2>Dịch vụ</h2>
      <ul>
        <li>Dịch vụ 1</li>
        <li>Dịch vụ 2</li>
        <li>Dịch vụ 3</li>
      </ul>
    </section>
  </main>

  <footer>
```

```
<p>&copy; 2025 Trang Web Mẫu. Mọi quyền được bảo lưu.</p>
</footer>

<script src="/script.js"></script> <!-- Liên kết file JavaScript
-->
</body>
</html>
```

- Giải thích cấu trúc:
 - **<!DOCTYPE html>**: Khai báo phiên bản HTML5.
 - **<head>**: Chứa thông tin meta, tiêu đề và liên kết CSS.
 - **<body>**: Phần nội dung hiển thị trên trang web.
 - **<header>**: Thanh tiêu đề, bao gồm tên trang và menu điều hướng.
 - **<main>**: Nội dung chính của trang web, chia thành nhiều **<section>**.
 - **<footer>**: Chân trang, chứa thông tin bản quyền.
 - CSS (**styles.css**) để tạo kiểu, JavaScript (**script.js**) để thêm tương tác.
- Khi đó ta sẽ có trang HTML (chưa có CSS và JavaScript) như này:

Chào mừng đến với trang web của tôi

- [Trang chủ](#)
- [Giới thiệu](#)
- [Dịch vụ](#)
- [Liên hệ](#)

Giới thiệu

Đây là một trang web HTML cơ bản với cấu trúc đầy đủ.

Dịch vụ

- Dịch vụ 1
- Dịch vụ 2
- Dịch vụ 3

© 2025 Trang Web Mẫu. Mọi quyền được bảo lưu.

2. CSS

a. Định nghĩa

- CSS (Cascading Style Sheets) là ngôn ngữ tạo kiểu dùng để định dạng giao diện và bố cục của trang web. Nó giúp tách biệt nội dung (HTML) và thiết kế, làm cho trang web đẹp mắt và nhất quán.

- Chức năng chính của CSS:
 - Thiết kế giao diện: Thay đổi màu sắc, font chữ, kích thước, khoảng cách, viền...
 - Bố cục trang web: Xác định vị trí, căn chỉnh các phần tử trên trang.
 - Hiệu ứng động: Tạo hiệu ứng hover, animation, transition...
 - Tương thích đa thiết bị: Dùng Media Queries để hiển thị tốt trên PC, tablet, mobile.
- Tác dụng của CSS:
 - Hạn chế tối thiểu việc làm rối mã HTML của trang Web bằng các thẻ quy định kiểu dáng (chữ đậm, chữ in nghiêng, chữ có gạch chân, chữ màu), khiến mã nguồn của trang Web được gọn gàng hơn, tách nội dung của trang Web và định dạng hiển thị, dễ dàng cho việc cập nhật nội dung.
 - Tạo ra các kiểu dáng có thể áp dụng cho nhiều trang Web, giúp tránh phải lặp lại việc định dạng cho các trang Web giống nhau.

b. Lịch sử phát triển của CSS

- 1994: Håkon Wium Lie đề xuất ý tưởng về CSS (Cascading Style Sheets) để tách biệt nội dung (HTML) và thiết kế trang web.
- 1996: CSS1 ra mắt, cung cấp các thuộc tính cơ bản như màu sắc, font chữ, căn lề.
- 1998: CSS2 bổ sung nhiều tính năng như định dạng bảng (display), vị trí (position), và hỗ trợ media types.
- 2001 - 2011: CSS2.1 ra đời để sửa lỗi và cải tiến từ CSS2.
- 2011 - nay: CSS3 phát triển với các tính năng hiện đại như animation, gradient, flexbox, grid, giúp thiết kế web linh hoạt hơn.
- Hiện tại, CSS tiếp tục được cập nhật với các công nghệ như CSS Variables, Subgrid, Container Queries, giúp tối ưu trải nghiệm người dùng trên nhiều thiết bị.

c. Cấu trúc CSS

```
/* Reset CSS */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

/* Body */
body {
    background-color: #f4f4f4;
    color: #333;
    line-height: 1.6;
    padding: 20px;
}

/* Header */
header {
    background-color: #007bff;
    color: white;
    padding: 20px;
    text-align: center;
}

header h1 {
    margin-bottom: 10px;
}

/* Navigation */
nav ul {
    list-style: none;
    padding: 0;
}

nav ul li {
    display: inline;
    margin: 0 15px;
}

nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}
```

```

}

nav ul li a:hover {
    text-decoration: underline;
}

/* Main content */
main {
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    background: white;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
}

/* Sections */
section {
    margin-bottom: 20px;
}

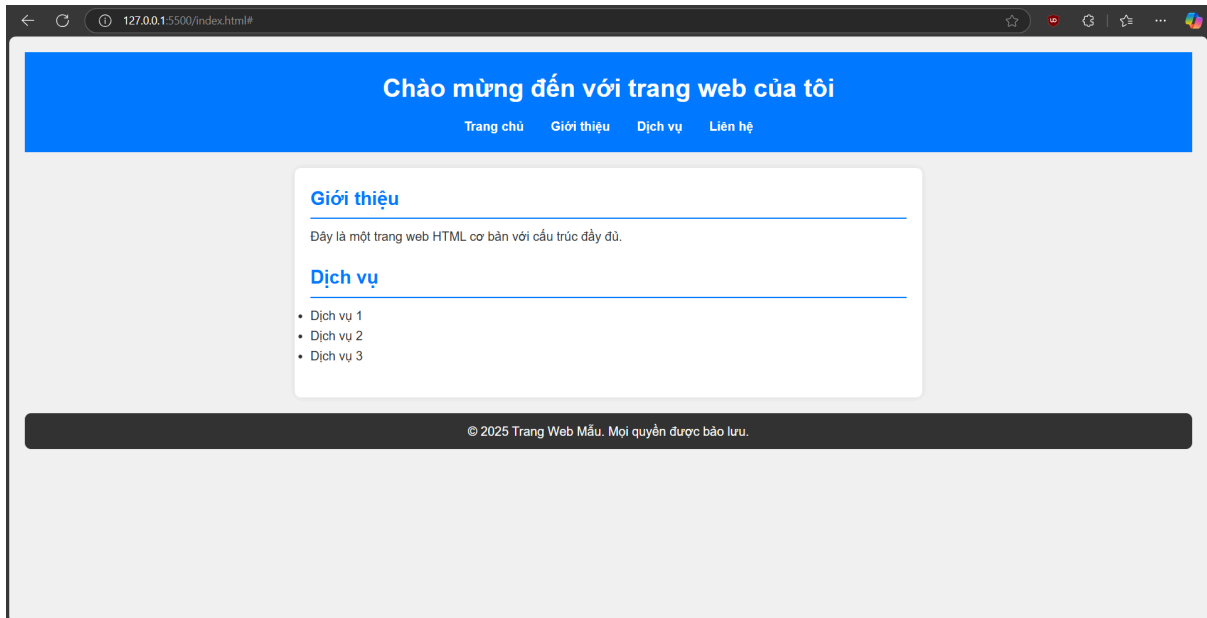
h2 {
    color: #007bff;
    border-bottom: 2px solid #007bff;
    padding-bottom: 5px;
    margin-bottom: 10px;
}

/* Footer */
footer {
    text-align: center;
    background: #333;
    color: white;
    padding: 10px;
    margin-top: 20px;
    border-radius: 8px;
}

```

- Giải thích cấu trúc:
 - Reset CSS để xóa margin, padding mặc định.
 - Header có nền xanh, chữ trắng, căn giữa.

- Navigation hiển thị menu ngang, chữ đậm, hiệu ứng hover.
- Main có khung bo tròn, đổ bóng, nền trắng.
- Footer có nền tối, chữ trắng, bo tròn.
- Khi đó trang HTML (sau khi thêm CSS) sẽ trông như này:



3. JavaScript

a. Định nghĩa

- JavaScript (JS) là một ngôn ngữ lập trình dùng để tạo nội dung động cho trang web. Nó giúp trang web tương tác, linh hoạt và hấp dẫn hơn bằng cách xử lý sự kiện, thay đổi nội dung, và thao tác với DOM.
- Chức năng chính của JavaScript:
 - Tương tác với người dùng: Xử lý sự kiện (click, nhập dữ liệu, cuộn trang...).
 - Thay đổi nội dung động: Cập nhật văn bản, hình ảnh mà không cần tải lại trang.
 - Xử lý dữ liệu: Tính toán, kiểm tra form, gửi & nhận dữ liệu từ server (AJAX, Fetch API).
 - Hoạt động trên mọi trình duyệt: Chạy trực tiếp trên trình duyệt mà không cần cài đặt thêm.

b. Lịch sử phát triển của JavaScript

- 1995: Brendan Eich (làm việc tại Netscape) phát triển JavaScript chỉ trong 10 ngày với tên ban đầu là Mocha, sau đó đổi thành LiveScript, rồi JavaScript để tận dụng độ phổ biến của Java.
- 1996: Microsoft phát triển một phiên bản tương tự tên JScript, dẫn đến sự thiếu nhất quán giữa các trình duyệt.
- 1997: ECMAScript (ES) ra đời, được chuẩn hóa bởi ECMA International để đảm bảo JavaScript hoạt động đồng nhất.
- 2009: ES5 được phát hành, cải tiến cú pháp và hỗ trợ JSON.
- 2015: ES6 (ECMAScript 2015) mang đến những thay đổi lớn như let, const, arrow functions, class và Promise.
- 2016 - nay: ECMAScript tiếp tục cập nhật hàng năm với nhiều tính năng như async/await, optional chaining, modules, giúp JavaScript trở thành một ngôn ngữ mạnh mẽ để phát triển web, server (Node.js) và ứng dụng di động.

c. Cấu trúc JavaScript

```
// Hiển thị thông báo khi trang được tải
window.onload = function() {
    alert("Chào mừng bạn đến với trang web của tôi!");
};

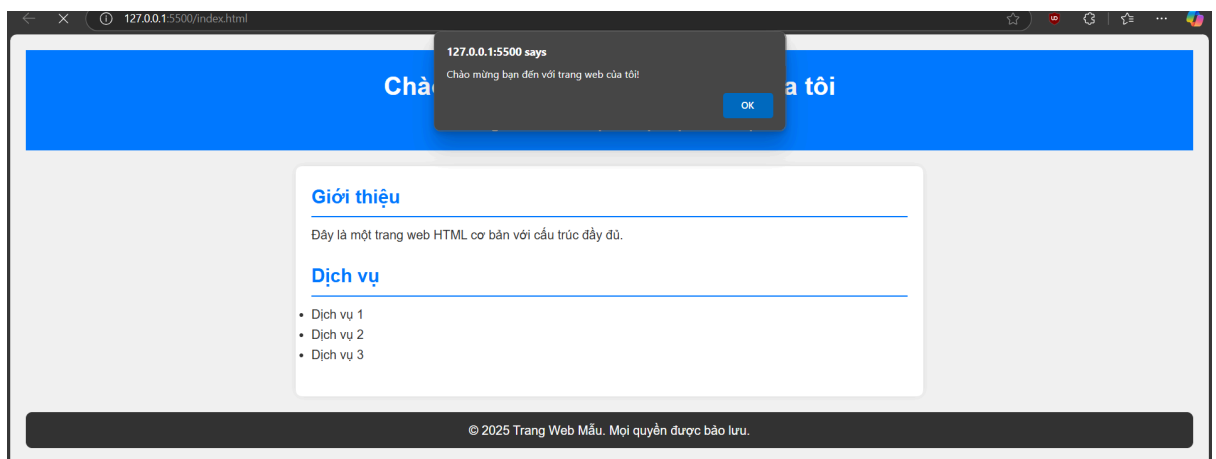
// Thêm hiệu ứng đổi màu header khi cuộn trang
window.addEventListener("scroll", function() {
    let header = document.querySelector("header");
    if (window.scrollY > 50) {
        header.style.backgroundColor = "#0056b3";
    } else {
        header.style.backgroundColor = "#007bff";
    }
});

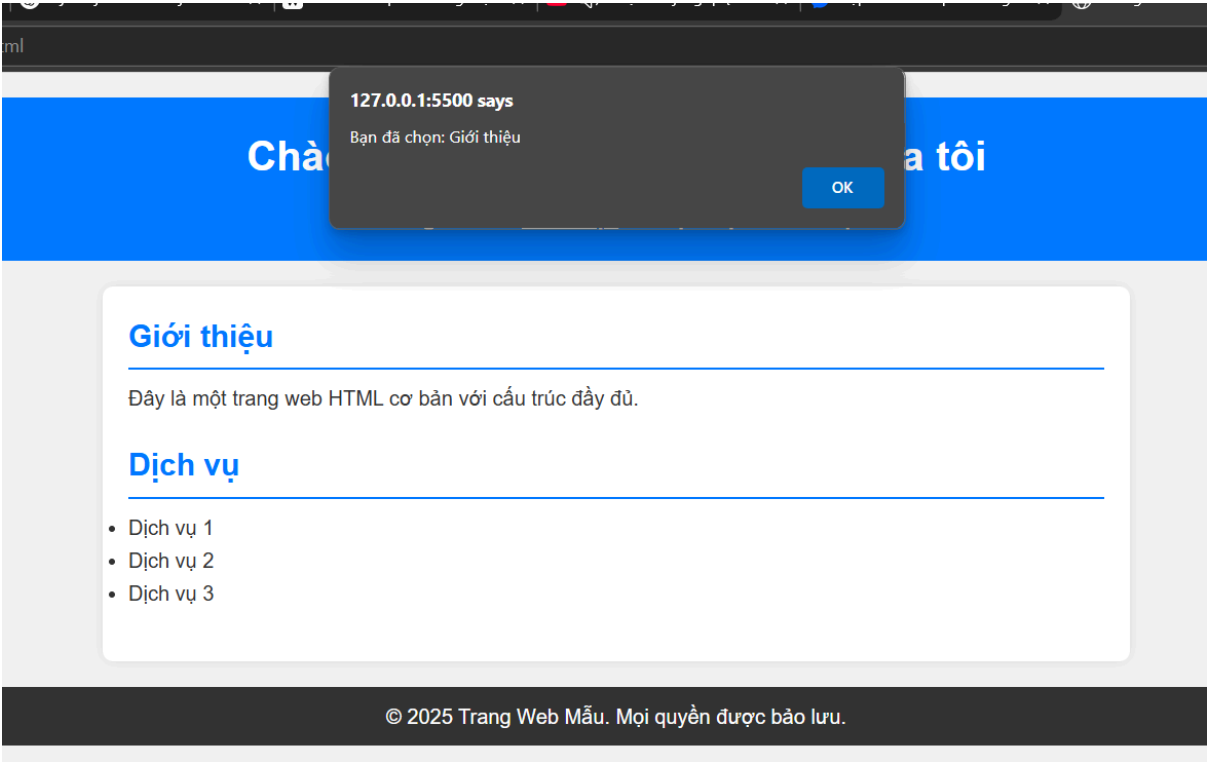
// Thêm sự kiện click cho các mục menu
document.querySelectorAll("nav ul li a").forEach(item => {
    item.addEventListener("click", function(event) {
        event.preventDefault();
        alert(`Bạn đã chọn: ${this.textContent}`);
    });
});
```

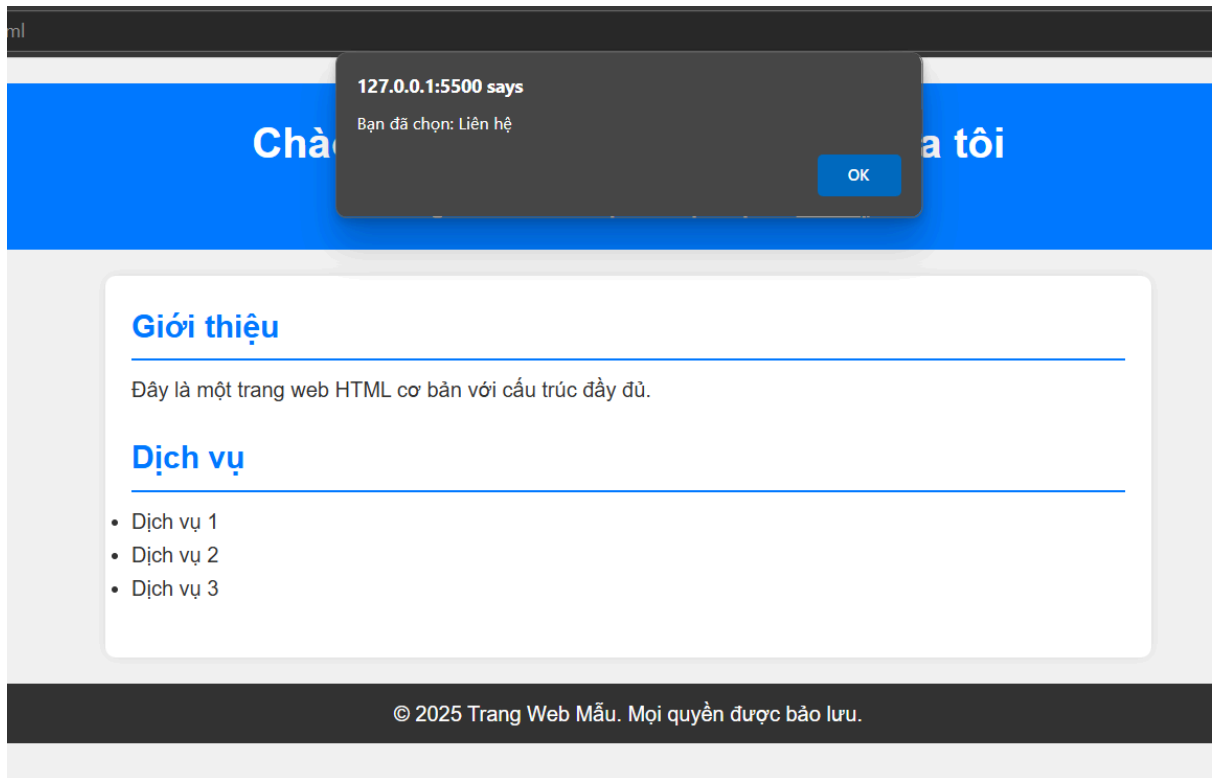
```
});

// Tạo hiệu ứng đổi màu cho footer khi di chuột vào
let footer = document.querySelector("footer");
footer.addEventListener("mouseenter", function() {
    footer.style.backgroundColor = "#444";
});
footer.addEventListener("mouseleave", function() {
    footer.style.backgroundColor = "#333";
});
```

- Giải thích:
 - Hiện thị thông báo chào mừng khi trang web được tải.
 - Đổi màu header khi cuộn trang để tạo hiệu ứng trực quan.
 - Bắt sự kiện click vào menu, hiển thị tên menu đã chọn.
 - Thay đổi màu nền footer khi di chuột vào, tạo hiệu ứng sinh động.
- Trang HTML sau khi thêm CSS và JavaScript:







4. Node.js

a. Node.js là gì?

- Node.js là một môi trường chạy JavaScript trên máy chủ (server-side), giúp xây dựng ứng dụng web mạnh mẽ và hiệu suất cao. Nó sử dụng V8 Engine (của Chrome) để thực thi JavaScript bên ngoài trình duyệt.
 - Non-blocking, asynchronous: Xử lý nhiều yêu cầu cùng lúc mà không bị chặn.
 - Single-threaded: Sử dụng mô hình event-driven, thích hợp cho ứng dụng thời gian thực.
 - Hỗ trợ nhiều module: Có npm (Node Package Manager) với hàng ngàn thư viện hữu ích.

b. Express.js là gì?

- Express.js là một framework web cho Node.js, giúp xây dựng API và ứng dụng web dễ dàng hơn.
- Đơn giản và nhanh chóng: Hỗ trợ định tuyến (routing), middleware.

- Xây dựng API RESTful dễ dàng.
- Tương thích với nhiều CSDL như MySQL, MongoDB.

c. Cấu trúc ứng dụng Express.js

- Ví dụ tạo server với Express.js:

```
const express = require("express");
const app = express();

// Định nghĩa một route cơ bản
app.get("/", (req, res) => {
  res.send("Chào mừng đến với Express.js!");
});

// Khởi động server
app.listen(3000, () => {
  console.log("Server đang chạy tại http://localhost:3000");
});
```

→ Node.js + Express.js là một lựa chọn mạnh mẽ để xây dựng hệ thống backend cho các ứng dụng web và API.

5. MySQL

a. MySQL là gì?

- MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, sử dụng ngôn ngữ SQL (Structured Query Language) để thao tác dữ liệu. Nó phổ biến trong các hệ thống web, ứng dụng doanh nghiệp và lưu trữ dữ liệu.
 - Nhanh, mạnh mẽ, dễ sử dụng
 - Hỗ trợ giao dịch (ACID), khóa (LOCK), truy vấn tối ưu
 - Tích hợp với nhiều ngôn ngữ lập trình như PHP, Node.js, Python...

b. Cấu trúc dữ liệu trong MySQL

- Database: Tập hợp các bảng.
- Table: Lưu trữ dữ liệu theo hàng và cột.
- Row (Bản ghi - Record): Một dòng dữ liệu trong bảng.

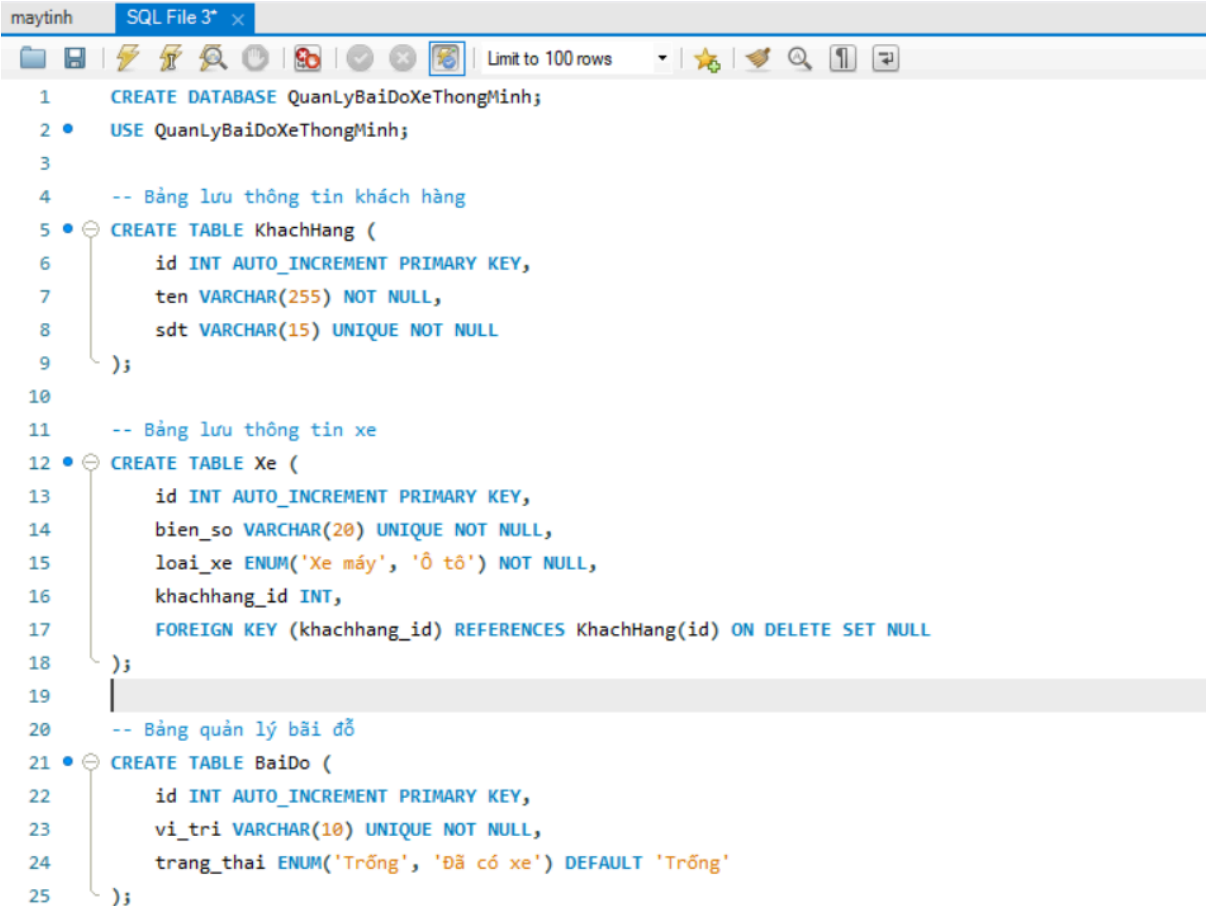
- Column (Trường - Field): Đại diện cho một loại dữ liệu cụ thể.

c. Các kiểu dữ liệu trong MySQL

- Số: INT, BIGINT, FLOAT, DECIMAL
- Chuỗi: VARCHAR, TEXT, CHAR
- Ngày/Giờ: DATE, DATETIME, TIMESTAMP
- Logic: BOOLEAN (TRUE/FALSE)

d. Các câu lệnh SQL cơ bản trong MySQL

- Tạo database & bảng:



```

1  CREATE DATABASE QuanLyBaiDoXeThongMinh;
2  • USE QuanLyBaiDoXeThongMinh;
3
4  -- Bảng lưu thông tin khách hàng
5  • CREATE TABLE KhachHang (
6      id INT AUTO_INCREMENT PRIMARY KEY,
7      ten VARCHAR(255) NOT NULL,
8      sdt VARCHAR(15) UNIQUE NOT NULL
9  );
10
11  -- Bảng lưu thông tin xe
12  • CREATE TABLE Xe (
13      id INT AUTO_INCREMENT PRIMARY KEY,
14      bien_so VARCHAR(20) UNIQUE NOT NULL,
15      loai_xe ENUM('Xe máy', 'Ô tô') NOT NULL,
16      khachhang_id INT,
17      FOREIGN KEY (khachhang_id) REFERENCES KhachHang(id) ON DELETE SET NULL
18  );
19
20  -- Bảng quản lý bãi đỗ
21  • CREATE TABLE BaiDo (
22      id INT AUTO_INCREMENT PRIMARY KEY,
23      vi_tri VARCHAR(10) UNIQUE NOT NULL,
24      trang_thai ENUM('Trống', 'Đã có xe') DEFAULT 'Trống'
25  );
  
```

```

-- Bảng lưu thông tin về xe
CREATE TABLE VeXe (
    id INT AUTO_INCREMENT PRIMARY KEY,
    xe_id INT,
    bai_do_id INT,
    thoi_gian_vao DATETIME DEFAULT CURRENT_TIMESTAMP,
    thoi_gian_ra DATETIME NULL,
    trang_thai ENUM('Đang gửi', 'Đã lấy xe') DEFAULT 'Đang gửi',
    FOREIGN KEY (xe_id) REFERENCES Xe(id) ON DELETE CASCADE,
    FOREIGN KEY (bai_do_id) REFERENCES BaiDo(id) ON DELETE CASCADE
);

```

- Thêm dữ liệu:

```

9 -- Thêm khách hàng
0 • INSERT INTO KháchHang (ten, sdt) VALUES
1 ('Nguyễn Văn A', '0909123456'),
2 ('Trần Thị B', '0912456789');
3
4 -- Thêm xe của khách hàng
5 • INSERT INTO Xe (bien_so, loai_xe, khachhang_id) VALUES
6 ('51A-12345', 'Ô tô', 1),
7 ('59X1-67890', 'Xe máy', 2);
8
9 -- Thêm vị trí bãi đỗ
0 • INSERT INTO BaiDo (vi_tri, trang_thai) VALUES
1 ('A1', 'Trống'),
2 ('A2', 'Trống'),
3 ('B1', 'Trống');
4
5 -- Thêm vé xe khi khách hàng vào bãi đỗ
6 • INSERT INTO VeXe (xe_id, bai_do_id, thoi_gian_vao) VALUES
7 (1, 1, NOW()),
8 (2, 2, NOW());

```

- Lấy dữ liệu:

```

) -- Lấy danh sách xe đang gửi trong bãi
• SELECT VeXe.id AS ma_ve, Xe.bien_so, Xe.loai_xe, BaiDo.vi_tri, VeXe.thoi_gian_vao
FROM VeXe
JOIN Xe ON VeXe.xe_id = Xe.id
JOIN BaiDo ON VeXe.bai_do_id = BaiDo.id
WHERE VeXe.trang_thai = 'Đang gửi';

```

- Cập nhật dữ liệu:

```

-- Cập nhật trạng thái khi xe rời bãi
UPDATE VeXe
SET thoi_gian_ra = NOW(), trang_thai = 'Đã lấy xe'
WHERE id = 1;

```

- Xóa dữ liệu:

```

-- Xóa thông tin vé xe khi hết hạn lưu trữ
DELETE FROM VeXe WHERE thoi_gian_ra < NOW() - INTERVAL 30 DAY;

```

- Liên kết bảng (JOIN):

```

-- Lấy danh sách khách hàng cùng thông tin xe của họ
SELECT KháchHang.ten, KháchHang.sdt, Xe.bien_so, Xe.loai_xe
FROM KháchHang
JOIN Xe ON KháchHang.id = Xe.khachhang_id;

```

- Tạo khóa chính, khóa ngoại:


```

0 ● ○ CREATE TABLE NhanVien (
1     id INT AUTO_INCREMENT PRIMARY KEY,
2     ten VARCHAR(255) NOT NULL,
3     ca_truc ENUM('Ca sáng', 'Ca chiều', 'Ca đêm') NOT NULL
4 );
5
6 ● ○ CREATE TABLE QuanLyBaiDo (
7     id INT AUTO_INCREMENT PRIMARY KEY,
8     nhanvien_id INT,
9     ve_id INT,
0     FOREIGN KEY (nhanvien_id) REFERENCES NhanVien(id),
1     FOREIGN KEY (ve_id) REFERENCES VeXe(id)
2 );

```

e. Quản lý người dùng và quyền (Phân quyền trong hệ thống quản lý bãi đỗ xe)

- Tạo tài khoản cho nhân viên quản lý:

```

▶ CREATE USER 'admin_baido'@'localhost' IDENTIFIED BY 'matkhau123';

```

- Cấp quyền cho nhân viên quản lý bãi đỗ:

```

GRANT SELECT, INSERT, UPDATE ON QuanLyBaiDoXeThongMinh.* TO 'admin_baido'@'localhost';
FLUSH PRIVILEGES;

```

- Cấp quyền cho nhân viên bảo vệ (chỉ xem danh sách vé xe):

```
GRANT SELECT ON VeXe TO 'baove'@'localhost';  
FLUSH PRIVILEGES;
```

f. Ưu điểm của MySQL trong hệ thống quản lý bãi đỗ xe thông minh

- Quản lý dữ liệu hiệu quả, tối ưu truy vấn khi có nhiều xe vào/ra cùng lúc.
- Bảo mật dữ liệu, phân quyền cho nhân viên quản lý và bảo vệ.
- Tích hợp tốt với Node.js/Express.js, giúp xây dựng hệ thống nhanh chóng.
- Hỗ trợ truy vấn phức tạp, tối ưu bãi đỗ bằng cách tìm chỗ trống tự động.