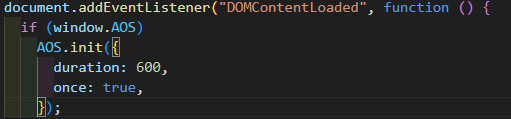
**JAVASCRIPT**

**File main.js**

**\* Khởi tạo thư viện AOS (tạo chuyển động biến mất và xuất hiện của các phần tử khi cuộn trang)**

***document.addEventListener("DOMContentLoaded", function () { ... });***

* Đây là **sự kiện DOMContentLoaded**.
* Nó đảm bảo **mã bên trong chỉ chạy khi toàn bộ nội dung HTML của trang đã được tải xong**, **trước khi** hình ảnh, video hoặc CSS tải xong.
* Mục đích: tránh lỗi “element chưa tồn tại” khi script chạy quá sớm.

***if (window.AOS)***

* window là đối tượng toàn cục trong trình duyệt (chứa tất cả biến và thư viện được nạp vào trang).
* AOS là **thư viện hiệu ứng chuyển động khi cuộn trang** (Animate On Scroll).
* Dòng này kiểm tra xem thư viện AOS có được tải chưa (tức là có tồn tại trong window không).

Nếu chưa import thư viện AOS (<script src="...aos.js">), thì phần còn lại sẽ **không chạy** để tránh lỗi.

***AOS.init({ ... })***

* Đây là lệnh **khởi tạo thư viện AOS**.
* Hàm init() cho phép cấu hình hiệu ứng cho các phần tử HTML có thuộc tính data-aos="...".

***duration: 600***

* Thiết lập **thời gian kéo dài của hiệu ứng** là 600 mili-giây (0.6 giây).
* Ví dụ: nếu một phần tử có hiệu ứng “fade-up”, nó sẽ mất 0.6 giây để hoàn tất chuyển động khi xuất hiện trên màn hình.

***once: true***

* Nghĩa là **hiệu ứng chỉ xảy ra một lần duy nhất** khi phần tử đó xuất hiện lần đầu trong vùng nhìn thấy của người dùng.
* Nếu once: false, thì mỗi lần người dùng cuộn xuống rồi cuộn lên, hiệu ứng sẽ **lặp lại**.

A screen shot of a computer code

AI-generated content may be incorrect.**\* Làm nút chuyển đổi Dark mode/Light mode**

***const btn = document.getElementById("theme-toggle");***

* Dòng này **tìm phần tử HTML có id="theme-toggle"**, ví dụ
* Kết quả được lưu vào biến btn.

***if (btn) {…}***

* Đây là **cách viết ngắn gọn** để kiểm tra xem btn có tồn tại không.
* Nghĩa là: Nếu btn tồn tại (khác null), thì thực hiện phần sau (addEventListener(...)).

***btn.addEventListener("click", function () { ... });***

* Gắn **sự kiện click** cho nút btn.
* Mỗi khi người dùng **bấm nút**, hàm bên trong sẽ được chạy.

***document.body.classList.toggle("dark");***

* ***document.body*** là thẻ <body> trong HTML.
* ***.classList.toggle("dark")*** sẽ:
  + Nếu thẻ <body> **chưa có class dark**, thì **thêm vào**.
  + Nếu **đã có rồi**, thì **xóa đi**.

***btn.textContent = document.body.classList.contains("dark") ? "Light" : "Dark";***

* Dòng này thay đổi **chữ hiển thị trên nút** sau mỗi lần bấm.
* ***document.body.classList.contains("dark")*** kiểm tra xem body có class "dark" hay không.
* Toán tử ***? :*** là **toán tử 3 ngôi (ternary operator)**:
  + Nếu **có class "dark"**, thì nút hiển thị chữ "Light".
  + Nếu **không có**, thì hiển thị "Dark".

**A screen shot of a computer code

AI-generated content may be incorrect.\* Hàm xử lí form “Đăng kí nhận tư vấn miễn phí” ở cuối trang index.html**

***const sub = document.getElementById("subscribe-form");***

* Lấy phần tử HTML có id là "subscribe-form".
* Gán phần tử này vào biến ***sub*** để tiện thao tác sau này.

***if (sub) { ... }***

* Kiểm tra xem biến sub có tồn tại hay không.
* Nếu có (tức là trang có chứa form này), thì mới chạy phần mã bên trong.
* Mục đích: **tránh lỗi** khi đoạn script được tải ở trang không có form đăng ký.

***sub.addEventListener("submit", function (e) { ... });***

* Gắn **sự kiện “submit”** cho form sub.
* Nghĩa là: mỗi khi người dùng **bấm nút gửi form (Submit)**,  
  → hàm bên trong sẽ được thực thi.

***e.preventDefault();***

* Ngăn hành vi mặc định của form — là **gửi dữ liệu lên server** và **tải lại trang**.
* Nhờ đó, form vẫn giữ nguyên trên trang và bạn có thể xử lý kết quả bằng JavaScript.

***alert("Cảm ơn bạn đã đăng ký!");***

* Hiển thị thông báo cảm ơn sau khi người dùng nhấn gửi.
* Mục đích: xác nhận thao tác đăng ký đã được ghi nhận (về mặt giao diện).

***sub.reset();***

* Đặt lại tất cả các ô nhập trong form về **trạng thái trống ban đầu**.
* Giúp form sẵn sàng cho lần đăng ký mới.

A computer screen with text on it

AI-generated content may be incorrect.**\* Hàm xử lí form “Liên hệ” ở cuối trang contact.html**

***const contact = document.getElementById("contact-form");***

* Lấy phần tử HTML có id là "contact-form".
* Gán phần tử này vào biến contact để dễ thao tác.

***if (contact) { ... }***

* Kiểm tra xem biến contact có tồn tại không.
* Nếu form thật sự có mặt trên trang, đoạn mã bên trong mới được thực thi.
* Mục đích: **tránh lỗi** khi script chạy ở trang không chứa form liên hệ.

***contact.addEventListener("submit", function (e) { ... });***

* Gắn sự kiện “submit” cho form contact.
* Nghĩa là: khi người dùng bấm nút Gửi (Submit) thì hàm bên trong sẽ được chạy.

***e.preventDefault();***

* Ngăn hành vi mặc định của form là **gửi dữ liệu lên server** và **tải lại trang**.
* Nhờ vậy, trang vẫn giữ nguyên và bạn có thể xử lý thao tác bằng JavaScript mà không cần tải lại.

***alert("Đã gửi liên hệ. Chúng tôi sẽ liên hệ lại sớm!");***

* Hiển thị thông báo cảm ơn sau khi người dùng bấm gửi.
* Câu thông báo giúp xác nhận rằng yêu cầu liên hệ của người dùng đã được ghi nhận.

***contact.reset();***

* Xóa toàn bộ dữ liệu người dùng vừa nhập trong form.
* Giúp form trở lại **trạng thái trống ban đầu**, sẵn sàng cho lần nhập mới.

A screen shot of a computer code

AI-generated content may be incorrect.**\* Giúp tất cả các liên kết nội trang hoạt động mượt mà hơn:**

***document.querySelectorAll('a[href^="#"]')***

* document.querySelectorAll(...) sẽ chọn **tất cả các thẻ <a>** (liên kết) trong trang mà:
  + Thuộc tính href **bắt đầu bằng dấu “#”** (^="#" nghĩa là “bắt đầu bằng”).

***.forEach((a) => { ... });***

* Duyệt qua **từng phần tử <a>** tìm được ở trên.
* Với mỗi liên kết, biến a đại diện cho phần tử đó.
* Bên trong, ta sẽ gắn sự kiện click cho từng thẻ <a>.

***a.addEventListener("click", function (e) { ... });***

* Gắn sự kiện **click** cho từng liên kết nội trang.
* Khi người dùng bấm vào một thẻ <a>, hàm bên trong sẽ được chạy.

***e.preventDefault();***

* Ngăn hành vi mặc định của liên kết, vốn là **nhảy ngay lập tức** đến phần có id tương ứng.
* Thay vì nhảy đột ngột, ta sẽ điều khiển việc **cuộn mượt (smooth scroll)** bằng JavaScript.

***document.querySelector(a.getAttribute("href")).scrollIntoView({ behavior: "smooth" });***

* ***a.getAttribute("href")***: lấy giá trị của thuộc tính href trong thẻ <a>.
* ***document.querySelector(...)***: tìm phần tử có id tương ứng trong trang
* ***.scrollIntoView({ behavior: "smooth" })***: cuộn trang **một cách mượt mà** đến phần tử đó.

**\* (Tạo bằng JavaScript) một nút liên kết “Đặt tour ngay”.**

***A screen shot of a computer

AI-generated content may be incorrect.const cta = document.createElement("a");***

* Tạo một **phần tử HTML mới** loại <a> (thẻ liên kết).
* document.createElement("a"): sinh ra một thẻ <a> chưa được gắn vào trang.
* Gán phần tử này vào biến cta để có thể chỉnh sửa và thêm thuộc tính cho nó.

***cta.href = "tour-detail.html";***

* Gán đường dẫn cho thẻ <a>.
* Khi người dùng bấm vào, trình duyệt sẽ mở trang **tour-detail.html**.
* Đây có thể là **trang chi tiết của tour du lịch**.

***cta.id = "floating-cta";***

* Đặt thuộc tính id cho thẻ <a> là "floating-cta".

***cta.textContent = "Đặt tour ngay";***

* Gán **nội dung văn bản hiển thị** bên trong thẻ <a>.
* Kết quả: người dùng sẽ thấy dòng chữ **“Đặt tour ngay”** trên nút.

***document.body.appendChild(cta);***

* Thêm phần tử <a> mới vừa tạo vào **cuối phần <body>** của trang.
* Sau lệnh này, nút “Đặt tour ngay” chính thức **xuất hiện trên giao diện**.

A computer screen with text on it

AI-generated content may be incorrect.**\* Hiệu ứng load trang**

***window.addEventListener("load", () => { ... });***

* Gắn **sự kiện “load”** cho đối tượng window.
* Sự kiện này sẽ kích hoạt **khi toàn bộ trang web (bao gồm hình ảnh, CSS, script...) đã tải xong hoàn toàn**.
* Khi đó, hàm bên trong sẽ được thực thi.

***const pre = document.getElementById("preloader");***

* Tìm và lấy phần tử HTML có id="preloader".
* Đây thường là **màn hình chờ (loading screen)** xuất hiện trong lúc trang đang tải.
* Biến pre sẽ lưu phần tử này để thao tác tiếp theo.

***if (pre) { ... }***

* Kiểm tra xem phần tử preloader có tồn tại không.
* Nếu có, mới chạy các lệnh bên trong để tránh lỗi khi trang không có preloader.

***setTimeout(() => {***

***pre.classList.add("fade-out");***

***}, 2200);***

* setTimeout() dùng để **thực hiện một hành động sau một khoảng thời gian nhất định** (tính bằng mili-giây).
* Sau **2.200 mili-giây** đoạn code trong hàm sẽ được chạy.
* pre.classList.add("fade-out"): thêm class "fade-out" vào phần tử preloader.
* Mục đích: kích hoạt **hiệu ứng mờ dần** (fade-out) do CSS quy định, giúp màn hình chờ biến mất dần thay vì biến mất đột ngột.

***setTimeout(() => {***

***pre.remove();***

***}, 3000);***

* Sau **3.000 mili-giây (≈ 3 giây)**, phần tử preloader sẽ được **xóa hoàn toàn khỏi trang** (.remove()).
* Thời điểm này được chọn để đảm bảo **hiệu ứng mờ dần đã chạy xong** trước khi xóa phần tử.

A screen shot of a computer

AI-generated content may be incorrect.**\* Tạo hiệu ứng cuộn mượt xuống phần tiếp theo của trang khi người dùng bấm vào nút mũi tên xuống ở trang index.html**

***document.querySelector(".scroll-down")***

* Tìm **phần tử đầu tiên** trong trang có **class là scroll-down**.

***?.addEventListener("click", () => { ... });***

* Dấu ?. gọi là **optional chaining** trong JavaScript.
* Nó có nghĩa là:
  + **Chỉ thêm sự kiện “click” nếu phần tử .scroll-down tồn tại**.
  + Nếu không có phần tử đó, chương trình **sẽ không bị lỗi** (bỏ qua đoạn này).
* addEventListener("click", ...): gắn sự kiện **khi người dùng nhấp chuột** vào nút.

***window.scrollTo({ top: window.innerHeight, behavior: "smooth" });***

* Đây là hàm sẽ chạy khi người dùng bấm vào nút .scroll-down.
* window.scrollTo(...): cuộn cửa sổ trình duyệt đến một vị trí nhất định.
* Các tùy chọn:
  + top: window.innerHeight → cuộn xuống **bằng đúng chiều cao của màn hình hiện tại**,  
    tức là cuộn **vừa đủ sang phần tiếp theo** bên dưới.
  + behavior: "smooth" → tạo **hiệu ứng cuộn mượt** (trơn tru, không giật).

**File tour-detail (JavaScript)**

**\*Tạo đối tượng dữ liệu**

**A computer screen shot of a program code

AI-generated content may be incorrect.**

***const dsChuyenDi = { ... }***

* Đây là một **đối tượng (object)** tổng hợp chứa **nhiều chuyến đi (tour)**.
* Tên dsChuyenDi viết tắt của **“Danh sách chuyến đi”**.

→ Mỗi tour được định danh bằng một **mã riêng** (ví dụ "TL-01").

***"TL-01": { ... }***

* "TL-01" là **khóa (key)** trong đối tượng.
* Giá trị là **một object chứa toàn bộ thông tin chi tiết** của tour đó.

**Các thuộc tính bên trong "TL-01"**

| **Thuộc tính** | **Ý nghĩa** | **Ví dụ** |
| --- | --- | --- |
| ma | Mã định danh tour | "TL-01" |
| ten | Tên đầy đủ của tour | "Đà Lạt 3N2Đ - Thành phố ngàn hoa" |
| giaNguoiLonText | Giá hiển thị dạng chữ (có đơn vị “đ”) | "3.890.000đ" |
| giaNguoiLonNum | Giá dạng số để dễ tính toán | 3890000 |
| phanTramGiamTreEm | % giảm giá cho trẻ em | 40 (tức giảm 40%) |
| gioiHanTuoi | Tuổi tối đa được tính là trẻ em | 10 |
| duration | Thời lượng chuyến đi | "3 ngày 2 đêm" |
| phuongTien | Phương tiện di chuyển | "Xe giường nằm" |
| anh | Đường dẫn ảnh minh họa tour | "img/dalat.jpg" |
| diemDen | Các điểm du lịch nổi bật | "Nhà thờ Con Gà, Quảng trường Lâm Viên, Vườn hoa" |
| amThuc | Ẩm thực đặc trưng trong chuyến đi | "Lẩu gà lá é, bánh căn" |
| doiTuong | Đối tượng phù hợp | "Gia đình, cặp đôi" |
| timeSuggest | Thời gian lý tưởng để đi | "Quanh năm" |
| khuyenMai | Thông tin khuyến mãi | "Giảm 5% khi đặt trước 15 ngày" |

**Thuộc tính lichTrinh**

lichTrinh: [

{ ngay: 1, noidung: ["Khởi hành...", "Tham quan...", "Dạo chợ đêm..."] },

{ ngay: 2, noidung: ["Vườn hoa...", "Đồi chè...", "Buổi tối tự do..."] },

{ ngay: 3, noidung: ["Check-out...", "Khởi hành về..."] }

]

* lichTrinh là **một mảng (array)** chứa **nhiều ngày trong chuyến đi**.
* Mỗi phần tử là **một object** gồm:
  + ngay: số thứ tự của ngày (1, 2, 3, …)
  + noidung: danh sách các hoạt động trong ngày (dạng mảng chuỗi).

**\*Lấy id từ URL**

***window.location.search***

* Thuộc tính này trả về **chuỗi chứa phần “query string”** của URL, tức là **phần bắt đầu từ dấu ?**.

***new URLSearchParams(window.location.search)***

* Đây là **đối tượng tiện ích có sẵn trong JavaScript** giúp bạn dễ dàng **xử lý và truy vấn tham số trên URL**.

***.get('id')***

* Phương thức .get() lấy ra **giá trị của một tham số cụ thể** trong query string.

***let maChuyenDi = ...***

* Lưu giá trị vào biến maChuyenDi.

A black background with white text

AI-generated content may be incorrect.**\*Lấy các phần tử (element) trong trang HTML**

***const mainEl = document.getElementById('mainContent');***

* Lấy phần tử có id="mainContent" trong HTML.

***const giaBenPhaiEl = document.getElementById('giaBenPhai');***

* Lấy phần tử hiển thị giá tour ở bên phải màn hình

***const maChuongTrinhEl = document.getElementById('maChuongTrinh');***

* Lấy phần tử có id="maChuongTrinh"

**\*Hiển thị các thẻ trong danh sách chuyến đi**

A screen shot of a computer program

AI-generated content may be incorrect.

***document.title = "Danh sách chuyến đi | HappyTrip";***

* Cập nhật **tiêu đề tab trình duyệt** thành "Danh sách chuyến đi | HappyTrip".

***mainEl.innerHTML = `***

***<h3>Danh sách chuyến đi</h3>***

***<div class="row g-3" id="danhSach">***

***</div>`;***

* Ghi nội dung mới vào phần tử mainEl (ví dụ <div id="mainContent">):
* Một tiêu đề: **“Danh sách chuyến đi”**
* Một khối rỗng <div id="danhSach"></div> — nơi các tour sẽ được thêm vào sau.

***const el = document.getElementById("danhSach");***

* Lấy lại phần tử <div id="danhSach"> vừa mới tạo ở trên để **chuẩn bị thêm các tour vào trong đó.**

***for (const k in dsChuyenDi) {***

***const cd = dsChuyenDi[k];***

* Duyệt qua tất cả các **thuộc tính (tour)** trong đối tượng dsChuyenDi.

***const col = document.createElement("div");***

***col.className = "col-12 col-md-6 mb-3";***

* Tạo một <div> mới, đặt class Bootstrap để chia cột:
* col-12: chiếm toàn bộ chiều rộng trên màn nhỏ
* col-md-6: chia 2 cột trên màn trung bình trở lên
* mb-3: khoảng cách dưới (margin-bottom: 3)

***col.innerHTML = `***

***<div class="card h-100">***

***...***

***</div>`;***

* Gắn nội dung HTML cho mỗi thẻ **tour card**, bao gồm:
* Ảnh tour
* Tên tour
* Điểm đến
* Giá
* Hai nút:
  + **Chi tiết**: mở trang có ?id=${k} để xem chi tiết tour
  + **Đặt**: gọi hàm moDatNhanh('${k}') để đặt tour ngay

***el.appendChild(col);***

* Thêm card vừa tạo vào trong danh sách #danhSach.

**\*Hiển thị thông tin chi tiết cho từng chuyến đi**

**A computer screen shot of text

AI-generated content may be incorrect.**

***function hienChiTietChuyenDi(cd) { ...}***

* function → khai báo một hàm mới.
* hienChiTietChuyenDi → tên hàm, có nghĩa là “hiển thị chi tiết chuyến đi”.
* (cd) → tham số đầu vào (viết tắt của **chuyến đi**), là một **đối tượng chứa thông tin tour**

***document.title = cd.ten + " | HappyTrip";***

* Thay đổi **tiêu đề của trang web (tab trên trình duyệt)**.
* document.title là thuộc tính để **đọc hoặc gán tiêu đề trang**.
* cd.ten là tên chuyến đi (ví dụ: "Đà Lạt 3N2Đ - Thành phố ngàn hoa").
* " | HappyTrip" là phần cố định để thêm thương hiệu.

***document.getElementById("phanTramGiam").innerText = cd.phanTramGiamTreEm;***

* Gán giá trị phần trăm giảm giá cho **phần tử HTML có id="phanTramGiam"**.
* document.getElementById("phanTramGiam") → tìm thẻ HTML, ví dụ:
* <span id="phanTramGiam"></span>
* .innerText = cd.phanTramGiamTreEm; → chèn số % giảm giá (ví dụ 40).

***document.getElementById("gioiHanTuoi").innerText = cd.gioiHanTuoi;***

* Chèn **giới hạn tuổi** của tour vào phần tử HTML có id="gioiHanTuoi".

***const mainEl = document.getElementById('mainContent');***

* mainEl là **phần tử HTML** nơi toàn bộ nội dung chi tiết tour sẽ được hiển thị. Khi gán innerHTML = ..., bạn đang “vẽ” giao diện chi tiết chuyến đi vào chỗ đó.

**Cấu trúc nội dung được gán vào innerHTML**

* Toàn bộ phần giữa dấu backtick (``) là **một template HTML**.
* Trong đó, có các **biến chèn nội dung động** bằng cú pháp ${cd.xxx}.

**giaBenPhaiEl.innerText = cd.giaNguoiLonText;** //

* Hiển thị giá tiền ở bên phải

***maChuongTrinhEl.innerText = "Mã chương trình: " + cd.ma;***

* Hiển thị mã chương trình ở bên phải

**A screen shot of a computer screen

AI-generated content may be incorrect.\*Lấy các giá trị nhập vào từ form**

***const formDat = document.getElementById("formDat");***

* Lấy phần tử form có id="formDat" trong HTML (chính là biểu mẫu đặt tour trong modal “Đặt tour”).

***const tenChuyenDiModal = document.getElementById("tenChuyenDiModal");***

* Lấy ô input hiển thị tên chuyến đi trong modal đặt tour.

***const giaNguoiLonModal = document.getElementById("giaNguoiLonModal");***

* Tham chiếu đến phần tử hiển thị giá cho người lớn trong modal.

***const giaTreEmModal = document.getElementById("giaTreEmModal");***

* Phần tử hiển thị giá vé cho trẻ em, thường được tính dựa trên phần trăm giảm giá

***const tongTienModal = document.getElementById("tongTienModal");***

* Tham chiếu đến ô hiển thị tổng tiền phải thanh toán

***const soNguoiLonEl = document.getElementById("soNguoiLon");***

* Input nhập số lượng người lớn (trong form đặt tour).

***const soTreEmEl = document.getElementById("soTreEm");***

* Input nhập số lượng trẻ em đi cùng.

***const ngayKhoiHanhEl = document.getElementById("ngayKhoiHanh");***

* Input chọn ngày khởi hành (type="date").

***const hoTenEl = getElementById("hoTen");***

* Ô họ và tên khách hàng.

***const soDienThoaiEl = document.getElementById("soDienThoai");***

* Ô nhập số điện thoại liên hệ.

**A computer code on a black background

AI-generated content may be incorrect.\***

***function moDatNhanh(ma) {***

Khai báo hàm moDatNhanh với tham số ma.

***maChuyenDi = ma;***

Gán giá trị ma (mã tour được chọn) cho biến toàn cục maChuyenDi.

***history.replaceState(null, "", "?id=" + ma);***

* Cập nhật URL trên thanh địa chỉ của trình duyệt thành dạng: ***?id=TL-01*** mà không tải lại trang.
* Dòng này dùng History API (replaceState) để thay đổi URL tạm thời.
* Mục đích: nếu người dùng muốn reload trang, trình duyệt vẫn biết đang mở tour nào (vì id có trong URL).

***hienChiTietChuyenDi(dsChuyenDi[ma]);***

* Gọi hàm hienChiTietChuyenDi() để hiển thị thông tin chi tiết của tour được chọn.
* dsChuyenDi[ma] truy xuất đúng đối tượng dữ liệu tour tương ứng trong danh sách dsChuyenDi.

***const modal = new bootstrap.Modal(document.getElementById("modalDat"));***

* Tạo một đối tượng modal Bootstrap từ phần tử có id="modalDat".
* Đây chính là hộp thoại đặt tour (popup) nằm trong HTML.

***dienThongTinModal(dsChuyenDi[ma]);***

* Gọi hàm dienThongTinModal() để điền thông tin tour được chọn vào các ô trong modal:
* modal.show();
* Hiển thị modal đặt tour ra màn hình (mở popup).

A screen shot of a computer code

AI-generated content may be incorrect.**\*Xác định và trả về đối tượng chuyến đi hiện đang được hiển thị**

**if (maChuyenDi && dsChuyenDi[maChuyenDi])**

**return dsChuyenDi[maChuyenDi];**

* Kiểm tra nếu biến maChuyenDi đang có giá trị và tồn tại trong danh sách dsChuyenDi để trả về đối tượng tour tương ứng.

***const ten = document.querySelector("#mainContent h2")?.innerText;***

* Nếu maChuyenDi không có (hoặc không hợp lệ), hàm sẽ tìm tiêu đề (<h2>) trong phần nội dung chính của trang (#mainContent).
* Dấu ?. (optional chaining) giúp tránh lỗi nếu không tìm thấy phần tử.

***for (const k in dsChuyenDi)***

***if (dsChuyenDi[k].ten === ten)***

***return dsChuyenDi[k];***

* Duyệt toàn bộ danh sách dsChuyenDi.
* Nếu tìm thấy tour có tên trùng với tiêu đề đang hiển thị, trả về tour đó.

***return Object.values(dsChuyenDi)[0];***

* Nếu vẫn không tìm thấy tour phù hợp, trả về tour đầu tiên trong danh sách làm mặc định.
* Đảm bảo hàm luôn trả về một tour hợp lệ, tránh lỗi “undefined”.

**A computer screen shot of text

AI-generated content may be incorrect.\*Điền một số thông tin tour được chọn vào form đặt tour**

***tenChuyenDiModal.value = cd.ten;***

* Gán tên chuyến đi (cd.ten) vào ô input hiển thị tên tour trong modal (<input id="tenChuyenDiModal">).
* Ô này có thuộc tính readonly, chỉ để người dùng xem, không chỉnh sửa.

***giaNguoiLonModal.innerText = formatVND(cd.giaNguoiLonNum);***

* Hiển thị giá vé người lớn trong modal.
* Gọi hàm formatVND() để định dạng số tiền theo kiểu tiền Việt (ví dụ: 3.890.000 đ).

***const giaTre = Math.round(***

***cd.giaNguoiLonNum \* (1 - cd.phanTramGiamTreEm / 100)***

***);***

* Tính giá vé trẻ em dựa trên phần trăm giảm giá (phanTramGiamTreEm).

***giaTreEmModal.innerText = formatVND(giaTre);***

* Hiển thị giá vé trẻ em đã tính được lên modal.
* Cũng định dạng tiền giống giá người lớn.

***document.getElementById("gioiHanTuoi").innerText = cd.gioiHanTuoi;***

* Cập nhật giới hạn tuổi của trẻ em hiển thị trong modal.
* Giúp người dùng biết độ tuổi nào được tính là “trẻ em”.

***soNguoiLonEl.value = 2;***

***soTreEmEl.value = 0;***

***ngayKhoiHanhEl.value = "";***

* Đặt giá trị mặc định cho các ô nhập trong form:
* soNguoiLonEl: mặc định 2 người lớn.
* soTreEmEl: mặc định 0 trẻ em.

***capNhatTong();***

* Gọi hàm capNhatTong() để tính và hiển thị tổng tiền ban đầu trong modal dựa trên số người mặc định.
* Đảm bảo khi modal vừa mở, người dùng thấy ngay tổng chi phí tạm tính.

A computer screen shot of code

AI-generated content may be incorrect.**\*Tính và hiển thị tổng số tiền phải trả**

***const cd = getChuyenDiHienTai();***

* Gọi hàm getChuyenDiHienTai() để lấy thông tin tour hiện tại (đang được đặt).
* Biến cd chứa toàn bộ dữ liệu của chuyến đi như giá người lớn, giảm giá trẻ em, v.v.

***const adults = Math.max(***

***0,***

***Math.min(10, Number(soNguoiLonEl.value || 0))***

***);***

* Lấy số người lớn từ ô nhập (soNguoiLonEl.value).
* Dùng Number() để chuyển từ chuỗi sang số.
* **“|| 0”** đảm bảo nếu ô trống thì tính là 0.
* Math.min(10, …) và Math.max(0, …) giúp giới hạn số người lớn trong khoảng 0–10 (tránh nhập sai hoặc vượt giới hạn).

***const children = Math.max(***

***0,***

***Math.min(10, Number(soTreEmEl.value || 0))***

***);***

* Làm tương tự cho số lượng trẻ em:

***const childPrice = Math.round(***

***cd.giaNguoiLonNum \* (1 - cd.phanTramGiamTreEm / 100)***

***);***

* Tính giá vé cho trẻ em, dựa theo phần trăm giảm giá (phanTramGiamTreEm).
* Làm tròn số bằng Math.round() để tránh số lẻ.

***const total = adults \* cd.giaNguoiLonNum + children \* childPrice;***

* Tính tổng tiền: tổng = (số người lớn × giá người lớn) + (số trẻ em × giá trẻ em)

***tongTienModal.innerText = formatVND(total);***

* Hiển thị tổng tiền tính được lên phần tử tongTienModal.
* Dùng hàm formatVND() để định dạng tiền tệ Việt Nam (ví dụ: 5.580.000 đ).

**\*Điền sẵn thông tin của tour hiện tại vào form đặt tour**

**A screen shot of a computer code

AI-generated content may be incorrect.**

***document***

***.getElementById("moDat")***

***.addEventListener("click", () =>***

***dienThongTinModal(getChuyenDiHienTai())***

***);***

* Lấy phần tử HTML có id="moDat" (thường là nút “Đặt tour” nằm ở trang chi tiết chuyến đi).
* Thêm trình lắng nghe sự kiện “click” (nghe khi người dùng nhấn nút).
* Khi click:
* Gọi getChuyenDiHienTai() để xác định tour đang hiển thị.
* Gọi dienThongTinModal() để điền thông tin tour (tên, giá, giới hạn tuổi, số lượng mặc định, tổng tiền ban đầu) vào modal.

***document***

***.getElementById("nutDat")***

***.addEventListener("click", () =>***

***dienThongTinModal(getChuyenDiHienTai())***

***);***

* Tương tự dòng trên, nhưng dành cho nút khác có id="nutDat" (thường là nút “Đặt ngay” ở vị trí khác, ví dụ dưới phần mô tả hoặc banner).
* Khi người dùng nhấn nút này, modal cũng sẽ được điền sẵn thông tin tour hiện tại.

**\*Cập nhật tổng tiền tự động khi thay đổi số lượng người lớn/trẻ em, và đảm bảo số điện thoại nhập vào hợp lệ**

**A screen shot of a computer program

AI-generated content may be incorrect.**

***soNguoiLonEl.addEventListener("input", capNhatTong);***

* Gắn sự kiện lắng nghe “input” cho ô nhập số người lớn (soNguoiLonEl).
* Mỗi khi người dùng thay đổi giá trị (tăng, giảm hoặc nhập số mới), hàm capNhatTong() sẽ được gọi.
* Kết quả: tổng tiền hiển thị trong modal được cập nhật ngay lập tức theo số người lớn mới.

***soTreEmEl.addEventListener("input", capNhatTong);***

* Tương tự như trên, nhưng áp dụng cho ô nhập số lượng trẻ em (soTreEmEl).
* Khi thay đổi số trẻ em, hệ thống cũng tự động tính lại tổng tiền.

***soDienThoaiEl.addEventListener("input", function () {***

***this.value = this.value.replace(/[^0-9]/g, "").slice(0, 10);***

***});***

* Lắng nghe sự kiện “input” trên ô nhập số điện thoại (soDienThoaiEl).
* Mỗi khi người dùng nhập dữ liệu:
* ***.replace(/[^0-9]/g, "")*** :loại bỏ mọi ký tự không phải là số (0–9).
* ***.slice(0, 10)***: giới hạn chỉ cho phép tối đa 10 chữ số.
* ***this.value = ...*** cập nhật lại giá trị sạch sẽ vào ô nhập.
* Kết quả: ô “Số điện thoại” chỉ nhận đúng 10 số hợp lệ, tránh nhập chữ hoặc ký hiệu.

**\*Gửi form đặt tour**

A black screen with a black background

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

***formDat.addEventListener("submit", function (e) {***

***e.preventDefault();***

* Lắng nghe sự kiện submit của form đặt tour (formDat).

***e.preventDefault()*** ngăn trình duyệt tải lại trang (hành vi mặc định khi gửi form) cho phép ta xử lý logic đặt tour bằng JavaScript thay vì gửi dữ liệu thật lên server.

***const cd = getChuyenDiHienTai();***

* Lấy thông tin tour hiện tại bằng hàm getChuyenDiHienTai().
* Biến cd chứa toàn bộ dữ liệu tour đang được đặt (mã, tên, giá...).

***const booking = {***

***ma: cd.ma,***

***ten: cd.ten,***

***ngay: ngayKhoiHanhEl.value,***

***nguoiLon: Number(soNguoiLonEl.value),***

***treEm: Number(soTreEmEl.value),***

***hoTen: hoTenEl.value.trim(),***

***soDienThoai: soDienThoaiEl.value.trim(),***

***tong:***

***Number(soNguoiLonEl.value) \* cd.giaNguoiLonNum +***

***Number(soTreEmEl.value) \****

***Math.round(cd.giaNguoiLonNum \* (1 - cd.phanTramGiamTreEm / 100)),***

***};***

* Tạo một đối tượng booking lưu thông tin đơn đặt tour gồm:
* ma, ten: mã & tên tour.
* ngay: ngày khởi hành người dùng chọn.
* nguoiLon, treEm: số lượng người lớn & trẻ em.
* hoTen, soDienThoai: thông tin liên hệ khách hàng.
* tong: tổng tiền (tính từ số lượng và giá vé).
* trim() loại bỏ khoảng trắng thừa ở đầu/cuối khi nhập tên hoặc số điện thoại.

***if (!booking.ngay || !booking.hoTen || !booking.soDienThoai) {***

***alert("Vui lòng nhập đầy đủ thông tin.");***

***return;***

***}***

* Kiểm tra dữ liệu hợp lệ: nếu người dùng chưa nhập ngày, họ tên hoặc số điện thoại → báo lỗi.
* Dùng alert hiển thị thông báo yêu cầu nhập đầy đủ.
* return để dừng hàm, không tiếp tục xử lý.

***const bs = bootstrap.Modal.getInstance(***

***document.getElementById("modalDat")***

***);***

***bs.hide();***

* Lấy thể hiện modal Bootstrap đang mở (modalDat).
* Gọi .hide() để đóng modal sau khi đặt tour thành công.

***alert(***

***"Đặt tour thành công! Chúng tôi sẽ liên hệ lại sớm.\n\nChi tiết:\n" +***

***booking.ma +***

***" • " +***

***booking.ngay +***

***" • " +***

***booking.nguoiLon +***

***" người lớn, " +***

***booking.treEm +***

***" trẻ em\nTổng: " +***

***formatVND(booking.tong)***

***);***

* Hiển thị thông báo thành công cho người dùng bằng alert().

***console.log("ĐẶT TOUR", booking);***

* Ghi dữ liệu đơn đặt tour ra console (trình duyệt) để theo dõi, phục vụ cho việc kiểm tra hoặc lưu trữ sau này.

***formDat.reset();***

***capNhatTong();***

* Reset form về trạng thái trống (xoá tất cả dữ liệu nhập trước đó).
* Gọi lại capNhatTong() để đặt lại tổng tiền về 0 hoặc giá trị mặc định.