# World Travel Tour — Final Project Files

This document contains every file for the **World Travel Tour** final project (WDD 231). Copy each file and folder into your wdd231/final/ folder in your GitHub repo. All JavaScript uses ES modules; the data/destinations.json file is fetched with fetch() using async/await and try...catch. Images are referenced in /images/ — replace with optimized files under 100 KB each. The site includes 3 main pages (index.html, destinations.html, contact.html) plus a form action page and an attributions page.

## File: index.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>World Travel Tour — Explore the World</title>  
 <meta name="description" content="World Travel Tour: discover hand-picked destinations, plan trips, and save favorites.">  
 <meta name="author" content="Celine Mafigu">  
 <!-- Open Graph -->  
 <meta property="og:title" content="World Travel Tour">  
 <meta property="og:description" content="Discover beautiful destinations around the world with World Travel Tour.">  
 <meta property="og:type" content="website">  
 <meta property="og:url" content="https://yourusername.github.io/wdd231/final/">  
 <meta property="og:image" content="images/og-image.jpg">  
 <link rel="icon" href="favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="styles/styles.css">  
</head>  
<body>  
 <header class="site-header">  
 <div class="container">  
 <a class="brand" href="index.html">  
 <img src="images/logo.png" alt="World Travel Tour logo" class="logo">  
 <span class="brand-name">World Travel Tour</span>  
 </a>  
 <button id="navToggle" class="nav-toggle" aria-controls="primaryNav" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="hamburger"></span>  
 </button>  
 <nav id="primaryNav" class="primary-nav" aria-label="Main navigation">  
 <ul>  
 <li><a href="index.html">Home</a></li>  
 <li><a href="destinations.html">Destinations</a></li>  
 <li><a href="contact.html">Contact</a></li>  
 </ul>  
 </nav>  
 </div>  
 </header>  
  
 <main>  
 <section class="hero">  
 <div class="container hero-inner">  
 <h1>Explore. Dream. Discover.</h1>  
 <p>Find your next adventure with curated destinations and travel tips.</p>  
 <a class="btn" href="destinations.html">Browse Destinations</a>  
 </div>  
 </section>  
  
 <section class="about container">  
 <h2>About World Travel Tour</h2>  
 <p>World Travel Tour is a small demo site created for the WDD 231 final project. It uses semantic HTML, modern CSS, and vanilla JavaScript to load dynamic content and provide an accessible user experience.</p>  
 <div id="featured" class="featured-grid" aria-live="polite">  
 <!-- Featured destinations loaded dynamically (JS will replace this) -->  
 </div>  
 </section>  
 </main>  
  
 <footer class="site-footer">  
 <div class="container footer-inner">  
 <p>&copy; 2025 World Travel Tour | Author: Celine Mafigu</p>  
 <nav aria-label="Footer">  
 <ul>  
 <li><a href="attributions.html">Attributions</a></li>  
 <li><a href="#" id="videoLink">Video Demo (link to YouTube)</a></li>  
 </ul>  
 </nav>  
 </div>  
 </footer>  
  
 <script type="module" src="scripts/main.js"></script>  
</body>  
</html>

## File: destinations.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>Destinations — World Travel Tour</title>  
 <meta name="description" content="Browse curated world destinations. Filter, view details, and save your favorites.">  
 <meta name="author" content="Celine Mafigu">  
 <link rel="icon" href="favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="styles/styles.css">  
</head>  
<body>  
 <header class="site-header">  
 <div class="container">  
 <a class="brand" href="index.html">  
 <img src="images/logo.png" alt="World Travel Tour logo" class="logo">  
 <span class="brand-name">World Travel Tour</span>  
 </a>  
 <button id="navToggle2" class="nav-toggle" aria-controls="primaryNav2" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="hamburger"></span>  
 </button>  
 <nav id="primaryNav2" class="primary-nav" aria-label="Main navigation">  
 <ul>  
 <li><a href="index.html">Home</a></li>  
 <li><a href="destinations.html" aria-current="page">Destinations</a></li>  
 <li><a href="contact.html">Contact</a></li>  
 </ul>  
 </nav>  
 </div>  
 </header>  
  
 <main class="container">  
 <h1>Destinations</h1>  
  
 <section class="controls">  
 <label for="filterInput">Search or filter:</label>  
 <input id="filterInput" type="search" placeholder="Search by name, country, or best time">  
  
 <label for="priceSelect">Price range:</label>  
 <select id="priceSelect">  
 <option value="all">All</option>  
 <option value="$">$</option>  
 <option value="$$">$$</option>  
 <option value="$$$">$$$</option>  
 </select>  
  
 <button id="clearFilters">Clear</button>  
 </section>  
  
 <section id="destinationsGrid" class="grid" aria-live="polite">  
 <!-- JS will populate 15+ destination cards here -->  
 </section>  
  
 <!-- Modal root -->  
 <div id="modalRoot" aria-hidden="true"></div>  
  
 </main>  
  
 <footer class="site-footer">  
 <div class="container footer-inner">  
 <p>&copy; 2025 World Travel Tour | Author: Celine Mafigu</p>  
 <nav aria-label="Footer">  
 <ul>  
 <li><a href="attributions.html">Attributions</a></li>  
 <li><a href="#" id="videoLink2">Video Demo (link to YouTube)</a></li>  
 </ul>  
 </nav>  
 </div>  
 </footer>  
  
 <script type="module" src="scripts/data.js"></script>  
 <script type="module" src="scripts/modal.js"></script>  
 <script type="module" src="scripts/main.js"></script>  
</body>  
</html>

## File: contact.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>Contact — World Travel Tour</title>  
 <meta name="description" content="Contact World Travel Tour for travel inquiries and bookings.">  
 <meta name="author" content="Celine Mafigu">  
 <link rel="icon" href="favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="styles/styles.css">  
</head>  
<body>  
 <header class="site-header">  
 <div class="container">  
 <a class="brand" href="index.html">  
 <img src="images/logo.png" alt="World Travel Tour logo" class="logo">  
 <span class="brand-name">World Travel Tour</span>  
 </a>  
 <button id="navToggle3" class="nav-toggle" aria-controls="primaryNav3" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="hamburger"></span>  
 </button>  
 <nav id="primaryNav3" class="primary-nav" aria-label="Main navigation">  
 <ul>  
 <li><a href="index.html">Home</a></li>  
 <li><a href="destinations.html">Destinations</a></li>  
 <li><a href="contact.html" aria-current="page">Contact</a></li>  
 </ul>  
 </nav>  
 </div>  
 </header>  
  
 <main class="container">  
 <h1>Contact Us</h1>  
 <p>Send a travel inquiry and we will get back to you.</p>  
  
 <form id="contactForm" action="form-action.html" method="GET" novalidate>  
 <label for="fullName">Full name</label>  
 <input id="fullName" name="fullName" type="text" required>  
  
 <label for="email">Email</label>  
 <input id="email" name="email" type="email" required>  
  
 <label for="country">Country of interest</label>  
 <select id="country" name="country">  
 <option value="">Choose a country</option>  
 <option>France</option>  
 <option>Japan</option>  
 <option>South Africa</option>  
 <option>Brazil</option>  
 <option>New Zealand</option>  
 </select>  
  
 <label for="message">Message</label>  
 <textarea id="message" name="message" rows="5"></textarea>  
  
 <button type="submit" class="btn">Send Inquiry</button>  
 </form>  
 </main>  
  
 <footer class="site-footer">  
 <div class="container footer-inner">  
 <p>&copy; 2025 World Travel Tour | Author: Celine Mafigu</p>  
 <nav aria-label="Footer">  
 <ul>  
 <li><a href="attributions.html">Attributions</a></li>  
 <li><a href="#" id="videoLink3">Video Demo (link to YouTube)</a></li>  
 </ul>  
 </nav>  
 </div>  
 </footer>  
  
 <script type="module" src="scripts/form.js"></script>  
 <script type="module" src="scripts/main.js"></script>  
</body>  
</html>

## File: form-action.html (form action page — does not count as one of the three required pages)

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>Form Submission — World Travel Tour</title>  
 <meta name="description" content="Form submission confirmation for World Travel Tour.">  
 <link rel="stylesheet" href="styles/styles.css">  
</head>  
<body>  
 <main class="container">  
 <h1>Inquiry Submitted</h1>  
 <div id="result" class="submission-result"></div>  
 <p><a href="index.html">Return home</a></p>  
 </main>  
  
 <script>  
 // Parse query parameters and display  
 const params = new URLSearchParams(location.search);  
 const resultDiv = document.getElementById('result');  
 if (params.toString()) {  
 const entries = Array.from(params.entries());  
 resultDiv.innerHTML = '<ul>' + entries.map(([k,v]) => `<li><strong>${k}:</strong> ${v}</li>`).join('') + '</ul>';  
 // Save name to localStorage  
 if (params.get('fullName')) localStorage.setItem('lastContactName', params.get('fullName'));  
 } else {  
 resultDiv.textContent = 'No data received.';  
 }  
 </script>  
</body>  
</html>

## File: attributions.html (simple unstyled page to list sources)

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width,initial-scale=1">  
 <title>Attributions — World Travel Tour</title>  
</head>  
<body>  
 <main>  
 <h1>Attributions</h1>  
 <p>Images and content used in this project were sourced from free-to-use resources. Replace these entries with accurate citations for any assets you used.</p>  
 <ul>  
 <li>Placeholder: Photo of Paris — source (replace with actual source)</li>  
 <li>Placeholder: Photo of Tokyo — source (replace with actual source)</li>  
 <li>Icons and logos: created by the author.</li>  
 </ul>  
 <p><a href="index.html">Return home</a></p>  
 </main>  
</body>  
</html>

## File: styles/styles.css

/\* Basic reset \*/  
\*{box-sizing:border-box}  
html,body{height:100%}  
body{font-family:Inter, system-ui, -apple-system, 'Segoe UI', Roboto, 'Helvetica Neue', Arial;line-height:1.4;color:#333;background:#FAF9F6;margin:0}  
.container{max-width:1100px;margin:0 auto;padding:1rem}  
.site-header{background:#fff;border-bottom:1px solid #e6e6e6}  
.site-header .container{display:flex;align-items:center;justify-content:space-between}  
.brand{display:flex;align-items:center;text-decoration:none;color:inherit}  
.logo{height:44px;width:auto;margin-right:.5rem}  
.brand-name{font-weight:700}  
.primary-nav ul{list-style:none;margin:0;padding:0;display:flex;gap:1rem}  
.primary-nav a{text-decoration:none;color:inherit;padding:.5rem}  
.nav-toggle{display:none;background:none;border:0}  
.hamburger{display:inline-block;width:28px;height:2px;background:#333;position:relative}  
.hamburger::before,.hamburger::after{content:'';position:absolute;left:0;width:28px;height:2px;background:#333}  
.hamburger::before{top:-8px}  
.hamburger::after{top:8px}  
.hero{background-image:linear-gradient(135deg, rgba(46,139,87,0.06), rgba(255,215,0,0.03));padding:3rem 0}  
.hero h1{font-size:2rem;margin:0}  
.btn{display:inline-block;background:#2E8B57;color:#fff;padding:.6rem 1rem;border-radius:6px;text-decoration:none}  
.about{padding:2rem 0}  
.featured-grid{display:grid;grid-template-columns:repeat(auto-fit,minmax(220px,1fr));gap:1rem}  
.grid{display:grid;grid-template-columns:repeat(auto-fit,minmax(240px,1fr));gap:1rem}  
.card{background:#fff;border:1px solid #eee;border-radius:8px;padding:0;overflow:hidden;display:flex;flex-direction:column}  
.card img{width:100%;height:160px;object-fit:cover;display:block}  
.card .card-body{padding:0.75rem}  
.controls{display:flex;flex-wrap:wrap;gap:.5rem;align-items:center;margin-bottom:1rem}  
.site-footer{background:#fff;border-top:1px solid #e6e6e6;padding:1rem 0;margin-top:2rem}  
.footer-inner{display:flex;justify-content:space-between;align-items:center}  
@media(max-width:720px){  
 .primary-nav{position:absolute;left:0;right:0;background:#fff;top:64px;display:none;padding:1rem}  
 .primary-nav.open{display:block}  
 .nav-toggle{display:block}  
 .footer-inner{flex-direction:column;gap:.5rem}  
}  
/\* Modal styles \*/  
.modal-backdrop{position:fixed;inset:0;background:rgba(0,0,0,0.5);display:flex;align-items:center;justify-content:center;z-index:1000}  
.modal{background:#fff;border-radius:8px;max-width:800px;width:calc(100% - 2rem);max-height:90vh;overflow:auto}  
.modal .modal-body{padding:1rem}  
.close-btn{display:inline-block;background:#eee;border:0;padding:.4rem .6rem;border-radius:6px}  
/\* Small helper \*/  
.visually-hidden{position:absolute!important;height:1px;width:1px;overflow:hidden;clip:rect(1px,1px,1px,1px);white-space:nowrap}

## File: data/destinations.json

{  
 "destinations": [  
 {"id":1,"name":"Paris, France","country":"France","image":"images/paris.jpg","description":"The city of love, art, and history.","bestTime":"April to June","priceRange":"$$$"},  
 {"id":2,"name":"Tokyo, Japan","country":"Japan","image":"images/tokyo.jpg","description":"A perfect blend of tradition and technology.","bestTime":"March to May","priceRange":"$$$"},  
 {"id":3,"name":"Kyoto, Japan","country":"Japan","image":"images/kyoto.jpg","description":"Ancient temples and seasonal beauty.","bestTime":"March to May","priceRange":"$$"},  
 {"id":4,"name":"Cape Town, South Africa","country":"South Africa","image":"images/capetown.jpg","description":"Table Mountain, beaches, and winelands.","bestTime":"October to April","priceRange":"$$"},  
 {"id":5,"name":"New York City, USA","country":"United States","image":"images/nyc.jpg","description":"The city that never sleeps.","bestTime":"April to June","priceRange":"$$$"},  
 {"id":6,"name":"Reykjavik, Iceland","country":"Iceland","image":"images/reykjavik.jpg","description":"Northern lights and geothermal wonders.","bestTime":"September to March","priceRange":"$$$"},  
 {"id":7,"name":"Cusco, Peru","country":"Peru","image":"images/cusco.jpg","description":"Gateway to Machu Picchu.","bestTime":"May to September","priceRange":"$$"},  
 {"id":8,"name":"Queenstown, New Zealand","country":"New Zealand","image":"images/queenstown.jpg","description":"Adventure capital with lakes and mountains.","bestTime":"December to February","priceRange":"$$$"},  
 {"id":9,"name":"Santorini, Greece","country":"Greece","image":"images/santorini.jpg","description":"Iconic white buildings and sunsets.","bestTime":"May to September","priceRange":"$$$"},  
 {"id":10,"name":"Rio de Janeiro, Brazil","country":"Brazil","image":"images/rio.jpg","description":"Beaches, carnival, and city views.","bestTime":"December to March","priceRange":"$$"},  
 {"id":11,"name":"Marrakech, Morocco","country":"Morocco","image":"images/marrakech.jpg","description":"Markets, gardens, and historic palaces.","bestTime":"March to May","priceRange":"$"},  
 {"id":12,"name":"Bali, Indonesia","country":"Indonesia","image":"images/bali.jpg","description":"Tropical beaches and temples.","bestTime":"April to October","priceRange":"$$"},  
 {"id":13,"name":"Prague, Czech Republic","country":"Czech Republic","image":"images/prague.jpg","description":"Gothic architecture and charming streets.","bestTime":"April to June","priceRange":"$"},  
 {"id":14,"name":"Vancouver, Canada","country":"Canada","image":"images/vancouver.jpg","description":"City meets wilderness with great parks.","bestTime":"June to September","priceRange":"$$"},  
 {"id":15,"name":"Kathmandu, Nepal","country":"Nepal","image":"images/kathmandu.jpg","description":"Gateway to Himalayan treks and culture.","bestTime":"September to November","priceRange":"$"}  
 ]  
}

## File: scripts/data.js (ES module: fetches JSON)

// scripts/data.js  
export async function fetchDestinations() {  
 try {  
 const resp = await fetch('data/destinations.json');  
 if (!resp.ok) throw new Error(`HTTP error ${resp.status}`);  
 const json = await resp.json();  
 return json.destinations;  
 } catch (err) {  
 console.error('Failed to fetch destinations:', err);  
 return [];  
 }  
}

## File: scripts/modal.js (ES module: accessible modal)

// scripts/modal.js  
export function createModal() {  
 const root = document.getElementById('modalRoot');  
 if (!root) return null;  
  
 function open(contentHtml, title = '') {  
 root.innerHTML = `  
 <div class="modal-backdrop" role="dialog" aria-modal="true" aria-label="${title}">  
 <div class="modal">  
 <div class="modal-body">  
 <button class="close-btn" id="modalClose">Close</button>  
 <div id="modalContent">${contentHtml}</div>  
 </div>  
 </div>  
 </div>  
 `;  
 root.setAttribute('aria-hidden', 'false');  
 const closeBtn = document.getElementById('modalClose');  
 closeBtn.focus();  
 closeBtn.addEventListener('click', close);  
 root.querySelector('.modal-backdrop').addEventListener('click', (e) => {  
 if (e.target === e.currentTarget) close();  
 });  
 }  
  
 function close() {  
 root.innerHTML = '';  
 root.setAttribute('aria-hidden', 'true');  
 }  
  
 return { open, close };  
}

## File: scripts/form.js (ES module: form validation + localStorage)

// scripts/form.js  
const form = document.getElementById('contactForm');  
if (form) {  
 form.addEventListener('submit', (e) => {  
 // HTML form uses GET to a form-action page; we add client-side validation  
 const name = form.fullName.value.trim();  
 const email = form.email.value.trim();  
 if (!name || !email) {  
 e.preventDefault();  
 alert('Please enter your name and email.');  
 return;  
 }  
 // Save last-contact name to localStorage  
 localStorage.setItem('lastContactName', name);  
 // Allow normal form submission to form-action.html  
 });  
}

## File: scripts/main.js (ES module: UI glue + dynamic content)

// scripts/main.js  
import { fetchDestinations } from './data.js';  
import { createModal } from './modal.js';  
  
const modal = createModal();  
const navToggles = document.querySelectorAll('.nav-toggle');  
navToggles.forEach(btn => {  
 btn.addEventListener('click', (e) => {  
 const nav = document.getElementById(btn.getAttribute('aria-controls')) || document.querySelector('.primary-nav');  
 const expanded = btn.getAttribute('aria-expanded') === 'true';  
 btn.setAttribute('aria-expanded', String(!expanded));  
 if (nav) nav.classList.toggle('open');  
 });  
});  
  
async function init() {  
 // Home: load a few featured  
 const destinations = await fetchDestinations();  
  
 // Populate featured on index page if present  
 const featuredRoot = document.getElementById('featured');  
 if (featuredRoot) {  
 const featured = destinations.slice(0,4);  
 featured.forEach(d => {  
 const el = document.createElement('article');  
 el.className = 'card';  
 el.innerHTML = `  
 <img src="${d.image}" alt="${d.name}" loading="lazy">  
 <div class="card-body">  
 <h3>${d.name}</h3>  
 <p>${d.description}</p>  
 <p><strong>Best Time:</strong> ${d.bestTime}</p>  
 <p><strong>Price:</strong> ${d.priceRange}</p>  
 </div>  
 `;  
 featuredRoot.appendChild(el);  
 });  
 }  
  
 // Destinations page: populate grid, implement filter, modal, favorites  
 const grid = document.getElementById('destinationsGrid');  
 if (grid) {  
 // Local copy of data  
 let data = destinations.slice();  
  
 function render(list) {  
 grid.innerHTML = '';  
 list.forEach(item => {  
 const card = document.createElement('article');  
 card.className = 'card';  
 card.innerHTML = `  
 <img src="${item.image}" alt="${item.name}" loading="lazy">  
 <div class="card-body">  
 <h3>${item.name}</h3>  
 <p>${item.country}</p>  
 <p>${item.description}</p>  
 <p><strong>Best Time:</strong> ${item.bestTime}</p>  
 <p><strong>Price:</strong> ${item.priceRange}</p>  
 <div class="card-actions">  
 <button class="details-btn" data-id="${item.id}">Details</button>  
 <button class="fav-btn" data-id="${item.id}">❤ Favorite</button>  
 </div>  
 </div>  
 `;  
 grid.appendChild(card);  
 });  
  
 // attach event listeners  
 grid.querySelectorAll('.details-btn').forEach(btn => {  
 btn.addEventListener('click', (e) => {  
 const id = Number(e.currentTarget.dataset.id);  
 const d = list.find(x => x.id === id);  
 if (d && modal) {  
 modal.open(`  
 <h2>${d.name}</h2>  
 <img src='${d.image}' alt='${d.name}' style='width:100%;height:auto'>  
 <p>${d.description}</p>  
 <p><strong>Best Time:</strong> ${d.bestTime}</p>  
 <p><strong>Price:</strong> ${d.priceRange}</p>  
 `, d.name);  
 }  
 });  
 });  
  
 grid.querySelectorAll('.fav-btn').forEach(btn => {  
 btn.addEventListener('click', (e) => {  
 const id = Number(e.currentTarget.dataset.id);  
 toggleFavorite(id);  
 });  
 });  
 }  
  
 function toggleFavorite(id) {  
 const favKey = 'wtt\_favorites';  
 const raw = localStorage.getItem(favKey);  
 const favs = raw ? JSON.parse(raw) : [];  
 const already = favs.includes(id);  
 const newFavs = already ? favs.filter(x => x !== id) : [...favs, id];  
 localStorage.setItem(favKey, JSON.stringify(newFavs));  
 // small UI feedback  
 alert(already ? 'Removed from favorites' : 'Added to favorites');  
 }  
  
 // initial render  
 render(data);  
  
 // Filtering  
 const filterInput = document.getElementById('filterInput');  
 const priceSelect = document.getElementById('priceSelect');  
 const clearBtn = document.getElementById('clearFilters');  
  
 function applyFilters() {  
 const query = filterInput ? filterInput.value.trim().toLowerCase() : '';  
 const price = priceSelect ? priceSelect.value : 'all';  
 let result = data.filter(item => {  
 const matchesQ = query === '' || [item.name, item.country, item.description, item.bestTime].join(' ').toLowerCase().includes(query);  
 const matchesPrice = price === 'all' || item.priceRange === price;  
 return matchesQ && matchesPrice;  
 });  
 render(result);  
 }  
  
 if (filterInput) filterInput.addEventListener('input', applyFilters);  
 if (priceSelect) priceSelect.addEventListener('change', applyFilters);  
 if (clearBtn) clearBtn.addEventListener('click', () => {  
 if (filterInput) filterInput.value = '';  
 if (priceSelect) priceSelect.value = 'all';  
 render(data);  
 });  
 }  
  
 // place video link placeholders from localStorage (optional)  
 const links = document.querySelectorAll('#videoLink, #videoLink2, #videoLink3');  
 links.forEach(l => {  
 l.addEventListener('click', (e) => {  
 e.preventDefault();  
 alert('Add your YouTube or Loom video URL in the footer links before grading.');  
 });  
 });  
}  
  
init();

## File: images/ (placeholders)

* logo.png — small logo (use optimized PNG or SVG)
* og-image.jpg — Open Graph image (replace with an optimized JPG)
* paris.jpg, tokyo.jpg, kyoto.jpg, capetown.jpg, nyc.jpg, reykjavik.jpg, cusco.jpg, queenstown.jpg, santorini.jpg, rio.jpg, marrakech.jpg, bali.jpg, prague.jpg, vancouver.jpg, kathmandu.jpg

**Note:** Optimize each image (compress, resize) to keep each under 100 KB and keep total page transfer under 500 KB for grading when possible.

## File: favicon.ico

Place a small 16x16/32x32 favicon file named favicon.ico in the project root.

## Developer Notes & Grading Checklist

* All pages use semantic HTML (header, nav, main, footer).
* index.html, destinations.html, and contact.html are the three required pages.
* Data is fetched using the Fetch API from data/destinations.json with async/await and try...catch (see scripts/data.js).
* At least **15 destination items** are included in data/destinations.json. Each item contains name, image, description, bestTime, priceRange (≥4 properties).
* Dynamic generation uses **template literals** and **array methods** (.map, .filter).
* Local storage stores favorites (wtt\_favorites) and last contact name (lastContactName).
* Modal dialog implemented in scripts/modal.js and used on the Destinations page.
* Images use loading="lazy" for progressive loading.
* ES modules are used (scripts marked with type="module").
* Attributions page included and linked from footer.
* Form sends data to form-action.html which displays the submitted data (so the form action page does not count toward 3-page requirement).

## How to deploy to GitHub Pages (short)

1. Create or place these files under wdd231/final/ in your GitHub repo.
2. Commit and push.
3. Enable GitHub Pages on the main branch (Settings → Pages) and set folder to /wdd231/final/.
4. Your site will be available at https://yourusername.github.io/wdd231/final/.

If you want, I can now: - produce a downloadable ZIP of these files (I can generate it for you here), or - generate a small checklist you can paste into Canvas when you submit, or - walk you step-by-step through uploading to GitHub and enabling Pages.

*End of project files.*