**Developer Guide: WOW · Wonders of the World**

**1. Overview**

**WOW (Wonders of the World) is a single‑page, interactive website that showcases the New Seven Wonders of the World. It combines rich historical narratives, immersive visuals, and several utility features (search, PDF download, rating, geolocation, booking simulation) into a polished, mobile‑responsive interface. This guide is intended for developers who wish to understand, maintain, or extend the codebase.**

**Purpose**

* **Educate users about each wonder with detailed historical facts and dates.**
* **Provide a visually engaging experience via full‑screen carousels and a dedicated gallery.**
* **Allow users to interact through feedback forms, location detection, and a booking demo.**
* **Demonstrate front‑end techniques: scroll‑snapping, dynamic modals, search filtering, and responsive design.**

**2. Technologies & Dependencies**

**The project uses a mix of standard web technologies and external libraries:**

| **Technology / Library** | **Purpose** |
| --- | --- |
| HTML5 | Structure and semantic markup |
| CSS3 | Custom styling, animations, responsive layout |
| JavaScript (ES5/jQuery) | DOM manipulation, event handling, AJAX (not used) |
| Bootstrap 5.3.2 | Responsive grid, navbar, modal, carousel, form controls |
| jQuery 3.7.1 | Simplified DOM traversal and event handling (loaded before Bootstrap’s JS) |
| jsPDF 2.5.1 | Client‑side PDF generation for downloading wonder histories |
| Google Fonts | Cinzel (serif) for titles, Inter (sans) for body text |
| Bootstrap Icons | Icon set (e.g., globe, stars, bus, etc.) |

All resources are loaded via CDN, so an internet connection is required for full functionality.

**3. File Structure**

The entire application is contained in a single index.html file. All styles, markup, and scripts are embedded. External assets (images) are referenced via relative paths (e.g., China wall/...). For deployment, ensure these image folders are present and correctly named.

**text**

**project-root/**

**├── index.html**

**├── China wall/**

**│ └── (image files)**

**├── Petra/**

**├── redeemer/**

**├── Machu/**

**├── Chicken/**

**├── Coloseum/**

**├── Taj Mehal/**

**└── (any other required image folders)**

**4. Core Structure & Layout**

**4.1 HTML Sections**

The body is divided into:

* Navbar – fixed top, with brand logo and navigation links.
* Advertise Card – bottom‑left click‑expandable element with travel offers.
* Booking Modal – form for demo bookings.
* Image Modal – full‑screen gallery image viewer.
* Search Box – appears only on Home and Gallery pages.
* Seven Page Sections (each with id="…-section"):
  + Home (home-section) – contains the full‑page scroll‑snapping wonder carousels.
  + Gallery (gallery-section) – grid of all wonder images.
  + Feedback (feedback-section) – form with star rating.
  + Contact (contact-section) – contact info and geolocation button.
  + About (about-section) – project description.
  + Sitemap (sitemap-section) – simple links.
  + Queries (queries-section) – placeholder query form.
* Detail Overlay – modal that displays full history and a carousel of images when a wonder is clicked.

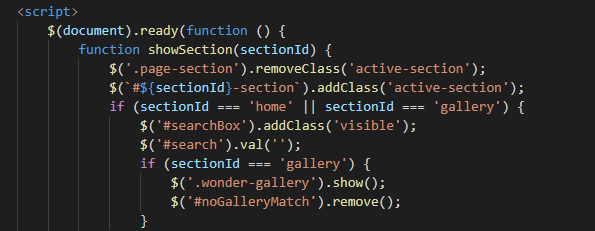
**4.2 CSS Organization**

Custom CSS follows a mobile‑first approach with:

* CSS variables for theming (--accent, --dark, etc.).
* Keyframe animations for fade‑in and slide‑in effects.
* Media queries for responsiveness at breakpoints (991.98px, 375px, 480px).
* Utility classes like .btn-gold for consistent styling.

**4.3 JavaScript / jQuery**

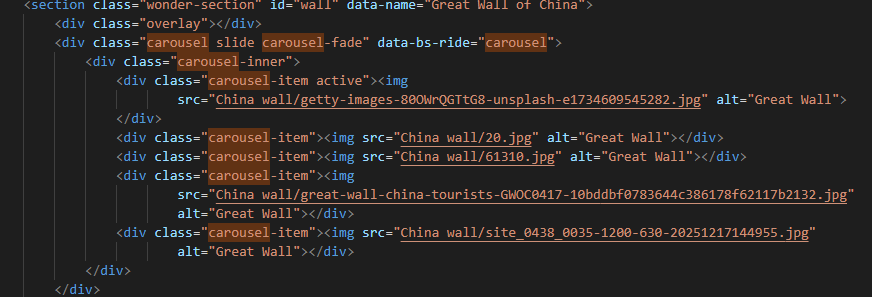
All scripts are placed inside a $(document).ready() block. Key functions:



* showSection(sectionId) – switches visible page sections and toggles search box.
* initHomeSidebar() – builds the right‑side navigation icons for each wonder.



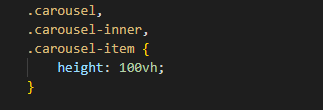
* Scroll‑spy logic for highlighting the active wonder icon.
* Event handlers for carousels, modals, search, rating, PDF download, etc.



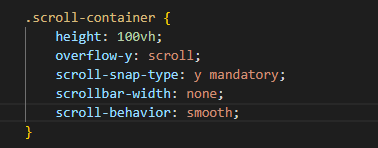
**5. Key Features Explained**

**5.1 Home Page – Scroll‑Snapping Wonder Sections**

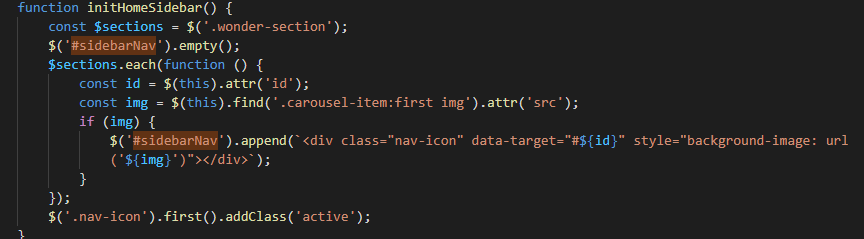
* Each wonder occupies a full viewport (height: 100vh) with a carousel of background images.



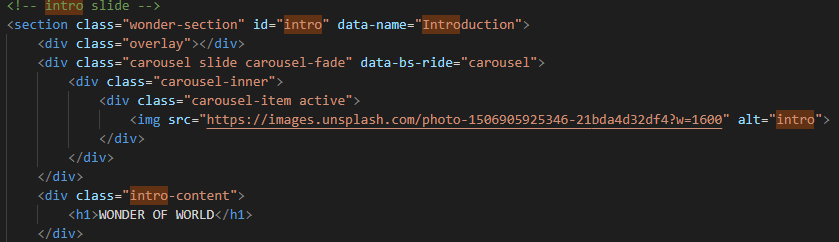
* Scroll‑snap (scroll-snap-type: y mandatory) ensures smooth section transitions.



* A right‑side navigation bar (#sidebarNav) contains circular thumbnails; clicking jumps to the corresponding wonder.



* The intro slide (first section) displays the site title with a glowing animation.



**5.2 Wonder Detail Overlay**

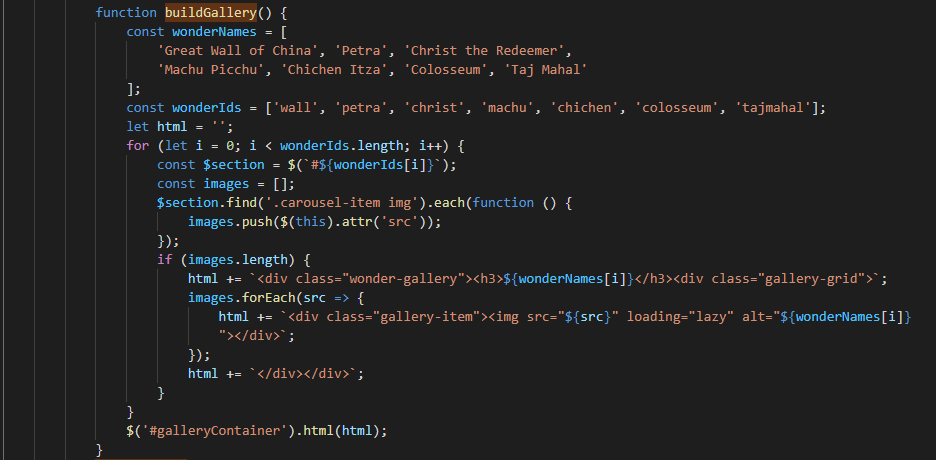
When a user clicks the content area of a wonder (excluding the PDF button), a full‑screen overlay appears:

* Left side: a carousel of all images from that wonder’s section.
* Right side: the wonder’s title and the complete historical text (stored in data-full attribute).
* Close by clicking the × button or outside the card.

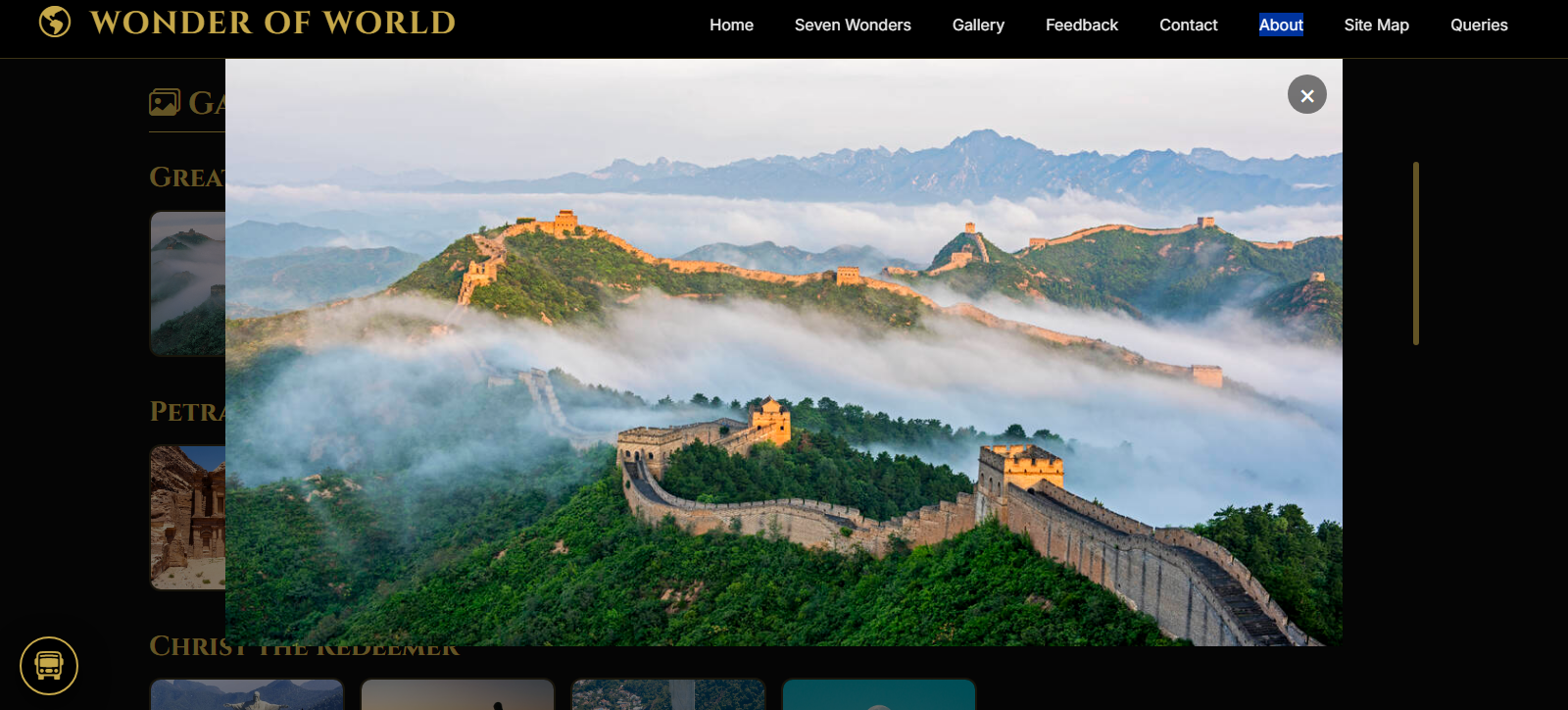


**5.3 Gallery Section**

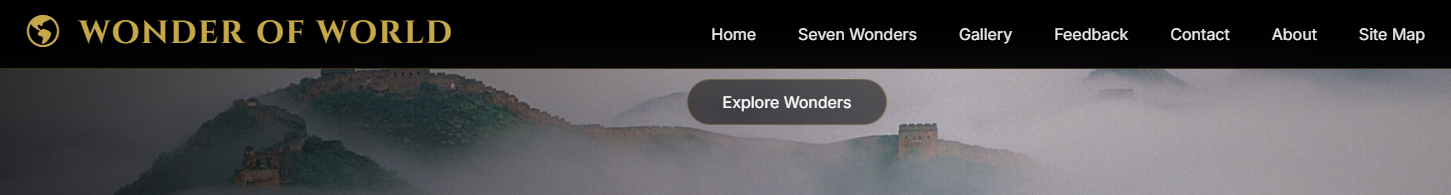
* Dynamically built from the image sources of each wonder section (function buildGallery()).



* Images are displayed in a responsive grid; clicking any image opens the Image Modal with a larger view.



* A search box filters galleries by wonder name (case‑insensitive). If no matches, a “no results” message appears.



**5.4 Feedback Form with Star Rating**

* Five clickable stars (Bootstrap Icons) that toggle from bi-star to bi-star-fill.





* On form submit, data is saved to localStorage as a JSON string (demo only).
* No server‑side processing is implemented.

**5.5 Geolocation (Contact Section)**

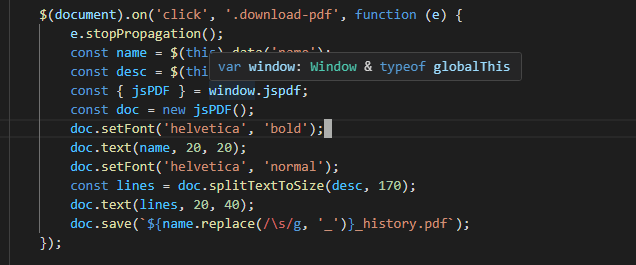
* Clicking “Find my location” triggers the browser’s geolocation API.
* Latitude and longitude are displayed below the button, along with accuracy.
* Graceful fallback for unsupported browsers or denied permission.



**5.6 PDF Download**

Each wonder’s content area includes a “Download PDF” button. When clicked:

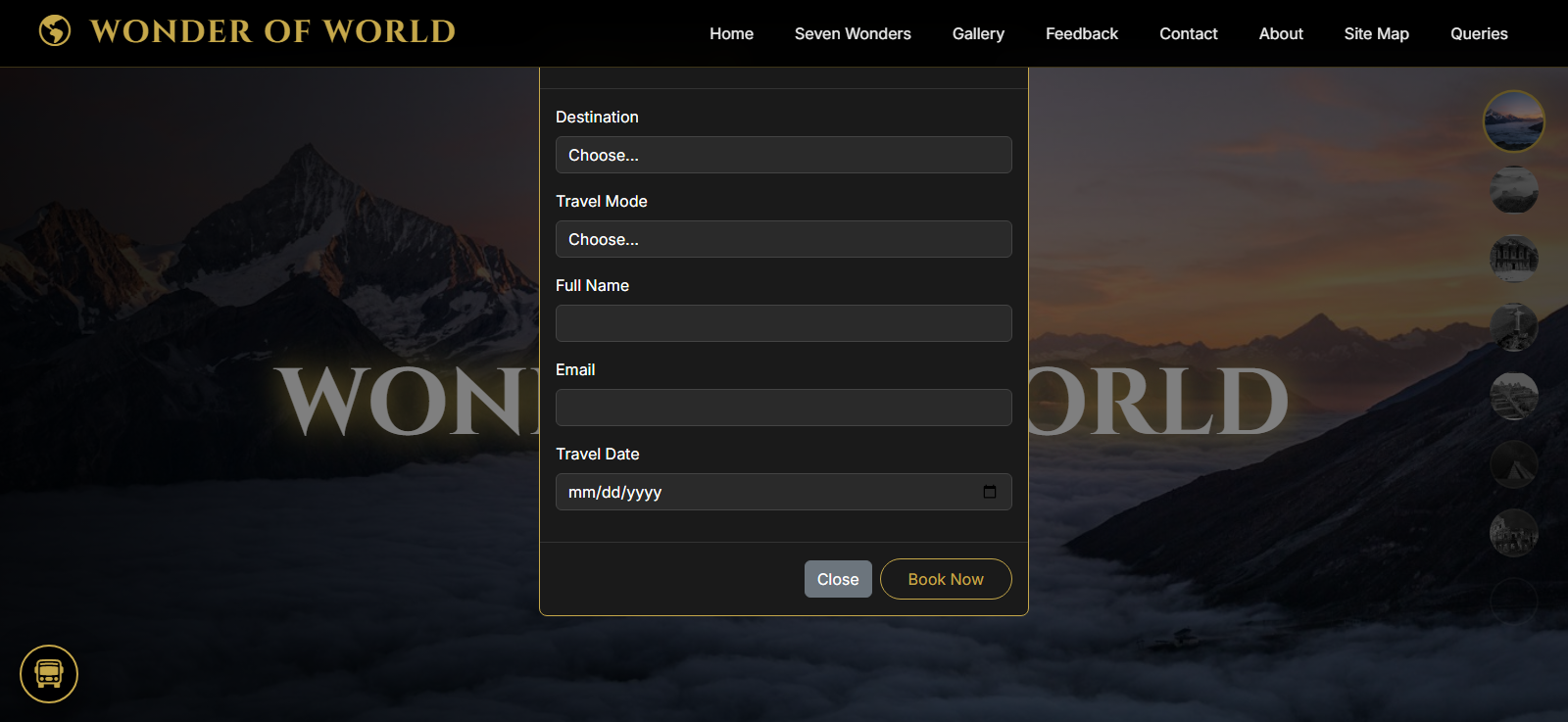
* It uses jsPDF to create a new document.



* The wonder’s name (as a title) and full history text (from data-desc) are added.
* The PDF is saved with a sanitised filename.

**5.7 Booking Modal**

* Triggered by clicking the “Bus to Petra” or “Flight to Machu Picchu” links inside the expandable ad card.



* A simple form with destination, travel mode, name, email, and date.
* On “Book Now”, basic validation is performed; if passed, a demo alert is shown and the form resets.

**5.8 Click‑Expandable Advert Card**

* The card in the bottom‑left starts as a small circle with a bus icon.



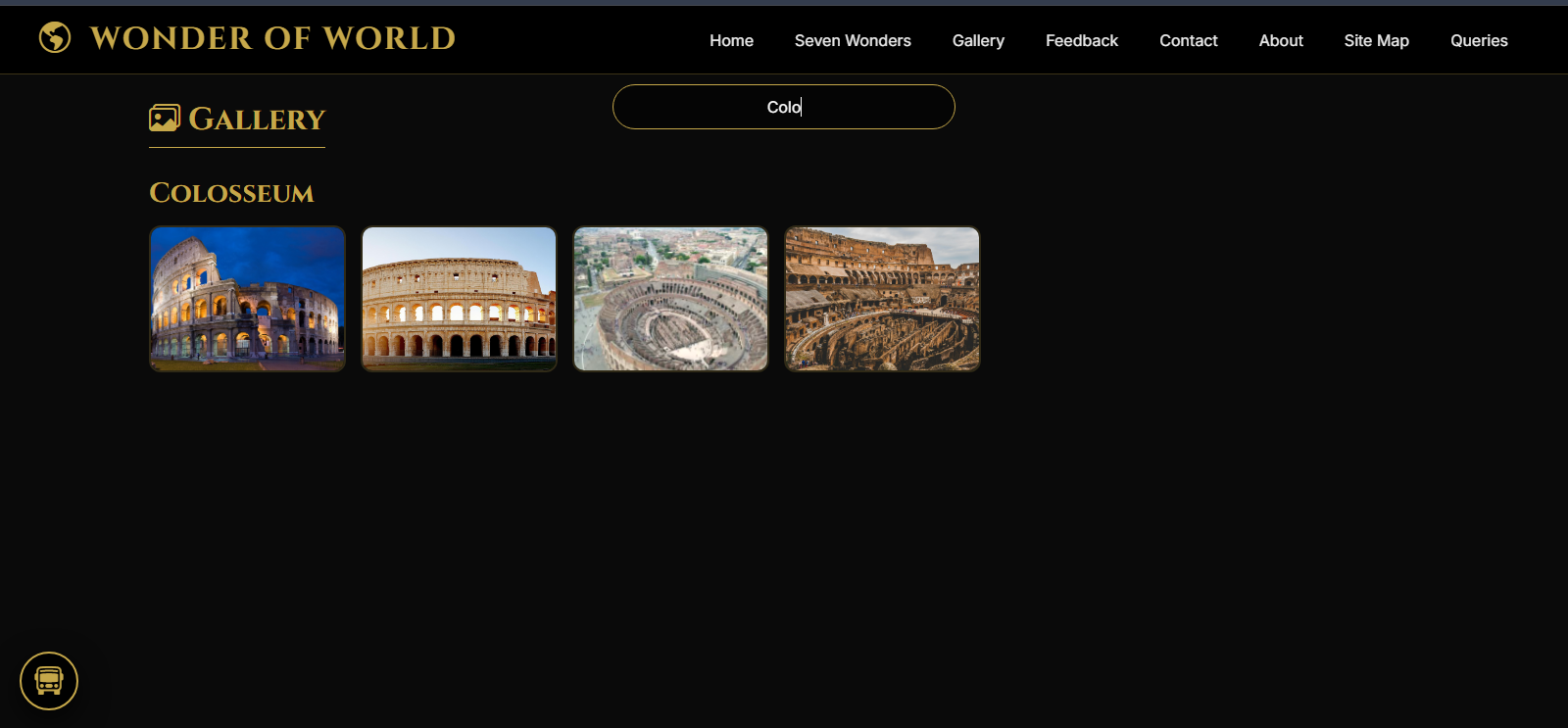
* Clicking it expands to reveal clickable links and a discount message.



* Clicking a link opens the booking modal (via Bootstrap’s modal).
* Clicking outside a link toggles the expanded state.

**5.9 Search Functionality**

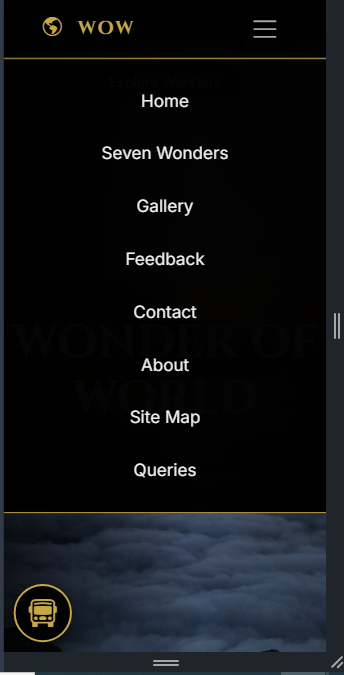
* The search box is visible only on Home and Gallery sections.
* On Home: searches wonder names and scrolls to the first matching section. If not found, the input briefly glows red.
* On Gallery: filters the gallery grid by wonder name; hides non‑matching groups.



**6. Responsive Design Highlights**

The site adapts to different screen sizes with careful media queries:

* Navbar collapses into a hamburger menu on mobile. When opened, the dropdown covers the full width below the navbar.



* Wonder navigation (side icons) moves to the bottom of the screen and becomes a horizontal row.
* Detail overlay switches to a single‑column layout (image above text) on narrow screens.
* Advertise card width reduces for very small devices (e.g., iPhone SE).
* Gallery grid columns adjust automatically (minmax(130px, 1fr) on mobile).
* Font sizes and spacing are reduced where appropriate.

All changes are implemented via CSS media queries; no separate mobile‑only JavaScript is used.

**7. Customization Guide**

**7.1 Changing Wonder Content**

Each wonder is defined inside a <section class="wonder-section"> with a unique id. To modify:



* Images: Add/remove <div class="carousel-item"> inside the carousel. Image paths are relative; ensure the files exist.



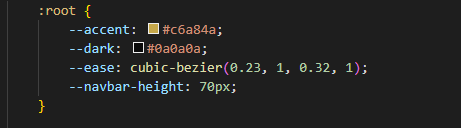
* Title: Edit the <h1 class="wonder-title">.
* Preview text: Change the text inside .wonder-history-preview.
* Full history: Update the data-full attribute on the .wonder-content div.
* PDF text: The data-desc attribute of the download button holds the text for the PDF. It can be identical to data-full or a shorter version.

**To add a new wonder:**

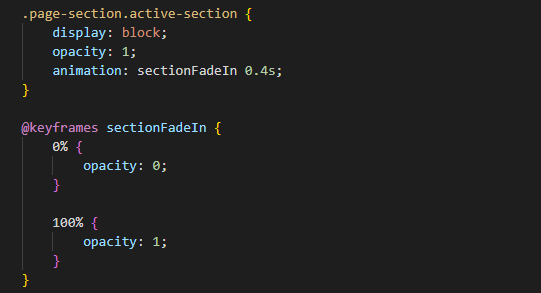
1. Copy an existing <section> block.
2. Change its id and data-name.
3. Update all image paths, title, preview, and data attributes.
4. Add the new wonder’s name to the wonderNames and wonderIds arrays inside buildGallery().
5. Ensure the corresponding image folder exists.

**7.2 Changing Styling**

* Theme colors: Modify the CSS variables at the top of the <style> block.
  + --accent: gold (#c6a84a) used for highlights.
  + --dark: background black.



* Fonts: Replace the Google Fonts link and update font-family in the CSS.
* Animations: Adjust keyframes (sectionFadeIn, slideFromRight, etc.) or durations.



**7.3 Updating Navigation Links**

The navbar links use data-section attributes to show sections. To add a new section:

1. Create a new <section> with id="new-section" and class page-section.
2. Add a link in the navbar with data-section="new" (note: the section id must be new-section).
3. The showSection() function will handle the rest.

**7.4 Customising the Advert Card**

* Text & links: Edit the .full-content div inside #advertiseCard.
* Collapsed icon: Change the <i> class (e.g., bi-bus-front to another Bootstrap icon).
* Expanded width: Adjust the .advertise-card.expanded width in CSS.

7.5 Enabling Real Booking / Backend Integration

Currently, the booking and feedback forms only simulate submission. To integrate with a backend:

* Replace the alert calls with AJAX requests (e.g., using $.post).
* Add proper form validation and CSRF tokens if needed.
* For PDF download, the jsPDF library can also be used to send the generated PDF to a server.

**7.6 Image Paths**

All images are referenced with relative paths like China wall/.... Ensure these folders exist and contain the images. For production, you may want to use absolute URLs or a CDN. Update the src attributes accordingly.

**8. Potential Improvements & Notes**

* Performance: The page loads many high‑resolution images. Consider lazy‑loading (already implemented via loading="lazy" on gallery images) and optimising image sizes.
* Accessibility: Add alt text to all images (currently missing in some carousel items). Ensure keyboard navigation works for modals and dropdowns.
* Browser Support: Modern browsers (Chrome, Firefox, Safari, Edge) are supported. Internet Explorer is not.
* Dependencies: All external libraries are loaded via CDN. If offline use is required, download and serve local copies.
* Code Organisation: The JavaScript is monolithic; for larger projects, consider splitting into modules. However, for a single‑page demo, it remains manageable.
* Placeholder Content: Some sections (Queries, Sitemap) contain minimal content. Expand as needed.

**9. Testing & Deployment**

**9.1 Local Testing**

* Clone or download the index.html and image folders.
* Open index.html in a modern web browser.
* All features (except geolocation, which requires HTTPS in some browsers) should work.

**9.2 Deployment**

* Upload the index.html file and all image folders to your web server.
* Ensure file permissions allow reading of images.
* If using a subdirectory, update relative paths if necessary.
* For HTTPS sites, geolocation will work without issues.

**9.3 Common Issues**

* Images not loading: Check file paths and case sensitivity.
* PDF download fails: Ensure jsPDF script loaded correctly (check console for errors).
* Bootstrap carousel not working: jQuery must be loaded before Bootstrap’s JavaScript.
* Mobile menu not closing after click: The code adds a click handler to .navbar-nav .nav-link to collapse the menu; ensure no other scripts interfere.

**10. Conclusion**

This WOW project is a self‑contained, feature‑rich demonstration of modern front‑end techniques. It can serve as a template for educational or promotional websites about landmarks, or as a learning resource for developers exploring interactive web design. The code is intentionally kept simple and well‑commented to facilitate understanding and modification.