# Personal Website - Interactive World Timeline Portfolio

## 1. Project Purpose

To create an immersive, interactive personal website that visualizes a 20-year history of travel. The site serves as a digital archive, combining a reverse-chronological narrative timeline with a real-time 3D geographical visualization. It is designed to allow users to explore travel history either narratively (by time) or geographically (by location), with a high-end, minimalist aesthetic.

## 2. Architecture & Views

### A. The Main Dashboard (Split-Screen)

This is the landing view and the core interface of the application.

- **Left Panel (The Timeline):** A scrollable vertical list of "Holiday Cards." Each card contains a title, date, brief description, and a single cover photo. Ordered from newest to oldest.
- **Right Panel (The Globe):** A 3D interactive Earth rendered in a minimalist/dark aesthetic. It features glowing markers for every visited location.
- **Mobile View:** The 3D globe sits in the background (reduced opacity) while the timeline scrolls over it to maximize screen real estate.

### B. The Detail Overlay (Holiday View)

A modal/overlay that appears on top of the dashboard when a specific holiday is selected.

- **Content:** Full holiday description and metadata.
- **Gallery:** A slideshow component displaying up to 50 high-resolution photos per trip.
- **Navigation:** Left/Right arrows and a thumbnail strip for browsing images.

## 3. User Flows & Interaction Logic

### Flow A: "Scrollytelling" (Narrative Mode)

- **Action:** The user scrolls down the vertical timeline.
- **System Response:** As a holiday card enters the viewport, the 3D globe automatically rotates and zooms to focus on the specific coordinates (Latitude/Longitude) of that trip.
- **Zoom Level:** The camera dynamically adjusts altitude based on location density (e.g., zooming in closer for "Lake District" vs. staying wider for a country-level trip).

### Flow B: Manual Exploration (Map Mode)

- **Action:** The user manually clicks and drags/spins the globe.

- **System Response:** The vertical timeline fades out (opacity: 0) to prevent distraction. The user can freely explore the globe.
- **Re-engagement:** Clicking a specific location marker re-centers the view. Clicking a "Back to Timeline" button or the active marker brings the timeline back into view, synced to that location.

### Flow C: Filtering

- **Action:** User applies a filter via the top navigation (e.g., "Father/Son", "Dorset") or by clicking a specific country on the globe.
- **System Response:**
    - **Globe:** Irrelevant location markers fade out/hide.
    - **Timeline:** The list updates to show *only* holiday cards matching the selected criteria.

## 4. Technology Stack & Use Cases

### Frontend Framework: Next.js (React)

- **Use Case:** Handles the core application structure, routing, and performance. It enables the "Overlay" architecture (intercepting routes) so the globe remains loaded in the background while viewing details.

### 3D Visualization: React-Globe.gl (Three.js)

- **Use Case:** Renders the interactive 3D Earth. Chosen specifically for its ability to handle precise coordinate mapping (Lat/Lng) for regional accuracy and its support for custom shaders (the "Minimalist" look).

### Content Management System (CMS): Sanity.io

- **Use Case:** Acts as the database for all holiday data.
    - **Free Tier:** Cost-effective for personal use.
    - **Geopoint Field:** Allows the user to drop a pin on a map to define coordinates for the 3D globe.
    - **Image CDN:** Automatically optimizes and resizes the upload of 50+ photos per holiday for performance.

### Animation Library: Framer Motion

- **Use Case:** Manages the complex state transitions, such as the timeline smooth-fading when the globe is dragged, and the modal expanding from the timeline card.

### State Management: Zustand

- **Use Case:** Connects the scrolling position of the Timeline to the rotation of the Globe, ensuring they stay perfectly in sync without performance lag.

## 5. Data Schema Specification (Sanity.io)

To support the functionality above, the CMS will require a single Document Type named `holiday` with the following field structure:

| Field Name | Type | Description | Usage in App |
|---|---|---|---|
| **Title** | string | The name of the holiday (e.g., "Skiing in Alps"). | Displayed on Timeline Card & Detail Modal header. |
| **Slug** | slug | Unique URL identifier (auto-generated from Title). | Used for routing (e.g., /holiday/skiing-alps). |
| **Date** | date | The start date of the trip. | Used to sort the timeline chronologically. |
| **Location** | geopoint | Latitude & Longitude coordinates. | **Crucial:** Used by the 3D Globe to position markers and calculate rotation. |
| **Main Image** | image | The single "Hero" image. | Displayed on the Timeline Card to reduce load time. |
| **Gallery** | array (of image) | Collection of up to 50 photos. | Loaded only when the specific Holiday Modal is opened. |

| | | | |
|---|---|---|---|
| **Tags** | array (of string) | Keywords (e.g., "Father/Son", "Dorset"). | Populates the top filter bar and enables filtering logic. |
| **Description** | text or block | Summary of the trip. | Displayed on the Timeline Card (truncated) and Modal (full). |