Travel Planner — Project Blueprint

A polished, multipage, responsive travel itinerary planner showcasing JS/HTML/CSS with dynamic DOM elements, LocalStorage persistence, event-driven interactions, and bonus API + animations. Built to impress on LinkedIn and align with course requirements.

---

1) Elevator Pitch

A sleek travel micro-portal where users pick a destination, generate a day-by-day plan with photos and weather, tweak activities, and save multiple trips locally. No account needed — everything persists in the browser.

Why it impresses: - Visual: destination imagery, animated cards, smooth page transitions. - Interactive: dynamic itinerary builder, drag-and-drop reordering (stretch goal), filters. - Integrated: 2–3 third-party APIs.

---

2) Sitemap & Pages

- **/index.html (Home)** — Hero, value props, “Plan a Trip” CTA, highlights of saved trips.
- **/plan.html (Plan Trip)** — Interactive form (destination, dates, preferences), live validation, results grid of recommended activities, “Build Itinerary” wizard.
- **/explore.html (Explore)** — Search destinations; dynamic cards with image + quick facts; filter/sort.
- **/saved.html (Saved Trips)** — Saved itineraries listed as cards; view, duplicate, delete.
- **/settings.html (Settings)** — Units (°C/°F), theme, animation preferences; all persisted.
- **/about.html (About/Contact)** — Contact form with validation + success state.

  SPA-ish feel with client-side routing is optional, but project will be truly multipage.

---

3) Core Features (by requirement)

Responsive Design: - Mobile-first; grid on desktop, stacked on mobile. CSS Grid for page sections; Flexbox for card interiors.

Interactive Form: - Plan form: destination autocomplete, dates, traveler count, interests (culture, food, outdoors, nightlife, kid-friendly). - Contact form on About page.

DOM-Connected Elements (≥2): 1) Dynamic **Itinerary Day** sections created from form input. 2) **Activity cards** injected/removed and re-ordered; details shown via modal. 3) Optional **Map module** (Leaflet) that highlights points of interest.

Event Handlers: - Submit/validate forms, filter chips toggles, like/save buttons, card expand/collapse, modal open/close, theme switch, scroll-to-top, pagination/infinite-scroll on Explore.

LocalStorage: - Persist trips, user settings (theme, units, animation toggles), and last search.

Bonus (APIs + Animations): - **APIs:** - Photos: Unsplash (or Pexels) for destination images. - Weather: Open-Meteo (no key) or WeatherAPI for forecast by dates. - Places/Geo: GeoDB Cities (autocomplete) or Teleport/Wikivoyage summaries. - **Animations:** - Page transitions (opacity/slide via CSS or WAAPI). - Card hover micro-interactions (transform, shadow, clip-path). - Scroll-triggered reveals with IntersectionObserver. - Modal open/close spring (WAAPI). Optional parallax hero.

---

4) Data Model & Storage

LocalStorage Keys:

```
trips           -> Array<Trip>
settings        -> { theme: 'light'|'dark', units: 'C'|'F', reduceMotion:
boolean }
lastSearch      -> { destination: string, dates: {start, end}, interests:
string[] }
```

Trip:

```
Trip = {
  id: string,                 // uuid
  name: string,               // "Rome Weekend"
  destination: {
    city: string,
    country: string,
    lat: number,
    lon: number,
    imageUrl?: string
  },
  dates: { start: string, end: string },
  travelers: number,
  interests: string[],        // ["food","history"]
  days: Day[]                 // generated itinerary
}

Day = {
  date: string,
  activities: Activity[]
}

Activity = {
  id: string,
```

```
  title: string,
  category: 'food'|'culture'|'outdoors'|'nightlife'|'kids'|'misc',
  time?: string,
  location?: string,
  coordinates?: [number, number],
  note?: string,
  imageUrl?: string,
  externalUrl?: string
}
```

---

5) UI Components (JS + DOM)

- **Navbar** (active link highlight, sticky).
- **Hero** (parallax/clip mask; CTA buttons).
- **Search/Plan Form** (autocomplete, date pickers, chips, validation messages).
- **Card** (destination/activity/trip variants with badges + “save” toggle).
- **Itinerary Day** (accordion + add/remove activity buttons).
- **Modal** (activity details, add note, set time).
- **Toast** (Saved/Deleted/Restored confirmations).
- **Map** (Leaflet map with markers; optional).
- **Settings Toggles** (theme, units, reduce motion).

---

6) UX Flows

1) **Create Trip** → user fills form → suggested activities appear → user selects → itinerary preview → save → success toast → visible under Saved. 2) **Edit Trip** → open trip → add/remove/reorder activities → auto-save. 3) **Explore** → search city → infinite scroll image cards → open details → add to wishlist or start plan.

---

7) Validation & Error Handling

- Required fields: destination, start/end dates.
- Logic: end ≥ start; max length for trip name; travelers ≥ 1.
- API states: loading skeletons, graceful fallbacks when API rate-limits or offline.

---

8) Visual Design System

- **Layout:** CSS Grid for page templates; Flex for card internals.
- **Spacing:** 8px scale; generous card padding; responsive gutters.
- **Type:** 2-font pairing; clamp() for fluid headings.
- **Color:** Light/dark themes; accessible contrast.

- **Components:** Rounded-xl cards; soft shadows; subtle borders.
- **Iconography:** SVG sprite or icon font.

---

9) Accessibility (a11y)

- Semantic HTML (nav, main, section, article, button, form, label).
- Keyboard navigable modals/cards; focus trap; skip-to-content.
- ARIA for dynamic regions; reduced-motion preference honored.
- Color contrast ≥ WCAG AA; visible focus outlines.

---

10) Performance

- Lazy-load images; responsive srcset.
- Preload critical fonts; limit blocking scripts.
- Cache API responses in memory; debounce search.

---

11) Testing Checklist

- Form validation (happy/sad paths).
- LocalStorage (save/update/delete; clear data → empty state).
- Responsiveness (320px → 1440px+).
- Keyboard navigation + screen reader labels.
- API fallback (offline/429/timeout).

---

12) Roadmap & Milestones

M1 — Skeleton & Styles (1–2 days) - File structure; base HTML for all pages; shared navbar/footer. - Color tokens, type, grid/flex utilities; hero section.

M2 — Forms & LocalStorage (1–2 days) - Plan form + validation; save/read settings; basic trip model.

M3 — Dynamic DOM & Itinerary (2–3 days) - Generate day sections; add/remove activities; modals; toasts.

M4 — Explore Search + Cards (1–2 days) - Destination search, results grid, filters; save to wishlist.

M5 — APIs + Enhancements (2–3 days) - Hook Unsplash/photos, Open-Meteo weather by trip dates, GeoDB autocomplete. - Skeleton loaders; IntersectionObserver reveals; optional Leaflet map.

M6 — Polish & A11y (1 day) - Reduced motion, focus states, error states, 404 page.

M7 — Deploy & LinkedIn Pack (0.5 day) - Build/optimize; deploy to GitHub Pages/Netlify; screenshots & GIF.

---

13) Tech Stack & Libraries

- **Vanilla JS** (modules + ES6), **HTML5**, **CSS3** (Grid/Flex + custom properties).
- **Optional:** Leaflet (map), a lightweight date picker, and a tiny UUID helper.
- **No heavy frameworks** to keep focus on core skills.

---

14) LinkedIn Showcase Plan

- 10–15s screen capture of key flows (create trip, edit, weather/imagery).
- Before/after stills: mobile vs desktop.
- Short write-up: problem, approach, tech, what you learned, next steps.

---

15) Stretch Goals

- Drag-and-drop to reorder activities.
- Export itinerary to PDF/ICS.
- Offline-first via Service Worker.
- Basic unit tests with Vitest.

---

16) Next Step (Implementation Plan)

1) Create repo + base file structure. 2) Build **index.html** hero + shared navbar/footer. 3) Scaffold **plan.html** form (HTML only), then wire up JS validation. 4) Add LocalStorage helpers and trip model factory. 5) Implement dynamic itinerary DOM creation with sample data (no API yet). 6) Integrate photo + weather APIs; add loaders and error states. 7) Finalize Saved Trips management; Settings persistence; polish and deploy.