

travel-planner.js — JavaScript for Travel Planner Website

```
/* travel-planner.js
   JavaScript for a simple Travel Planner Website
   Features:
   - Search form handling (destination, dates, travelers)
   - Mock "search results" rendering as cards
   - Add to itinerary and store in localStorage
   - Simple itinerary sidebar with remove and clear
   - Basic client-side filtering and sorting
   - Map placeholder integration hook (for future map provider)
   Save as travel-planner.js and include in your HTML.
*/

// --- Utilities
const el = (selector, parent = document) => parent.querySelector(selector);
const elAll = (selector, parent = document) => Array.from(parent.querySelectorAll(selector));
const formatDate = d => new Date(d).toLocaleDateString();

// --- Mock Data (replace with real API calls)
const MOCK_DESTINATIONS = [
  { id: 'paris', title: 'Paris, France', description: 'City of light. Museums, cafés, and the Eiffel Tower.', days: 4, price: 520 },
  { id: 'rome', title: 'Rome, Italy', description: 'Ancient ruins and legendary cuisine.', days: 3, price: 430 },
  { id: 'bali', title: 'Bali, Indonesia', description: 'Tropical beaches, temples, and rice terraces.', days: 6, price: 640 },
  { id: 'tokyo', title: 'Tokyo, Japan', description: 'Neon streets, world-class food, and culture.', days: 5, price: 790 },
  { id: 'nyc', title: 'New York City, USA', description: 'The city that never sleeps – theater, dining, museums.', days: 4, price: 610 }
];

// --- State
const state = {
  results: [],
  itinerary: JSON.parse(localStorage.getItem('tp_itinerary')) || '[]'
};

// --- DOM refs (expect these IDs/classes in HTML)
const refs = {
  searchForm: el('#searchForm'),
  resultsContainer: el('#results'),
  itineraryContainer: el('#itineraryList'),
  clearBtn: el('#clearItinerary'),
  sortSelect: el('#sortBy'),
  mapPlaceholder: el('#map')
};

// --- Initialize app
function init() {
  bindEvents();
  renderItinerary();
  renderResults(MOCK_DESTINATIONS); // show some suggestions by default
  console.log('Travel Planner initialized');
}

// --- Event bindings
function bindEvents() {
  if (refs.searchForm) {
    refs.searchForm.addEventListener('submit', onSearch);
  }
  if (refs.clearBtn) {
    refs.clearBtn.addEventListener('click', clearItinerary);
  }
  if (refs.sortSelect) {
    refs.sortSelect.addEventListener('change', () => renderResults(state.results));
  }
  // delegate add-to-itinerary clicks
  if (refs.resultsContainer) {
    refs.resultsContainer.addEventListener('click', (e) => {
      const btn = e.target.closest('[data-action]');
      if (!btn) return;
      const action = btn.dataset.action;
      const id = btn.dataset.id;
      if (action === 'add') addToItinerary(id);
    });
  }
  // delegate remove in itinerary
  if (refs.itineraryContainer) {
    refs.itineraryContainer.addEventListener('click', (e) => {
      const btn = e.target.closest('[data-action]');
    });
  }
}
```