# Route Planner Web App - Project Requirements Document

## Project Overview

### Purpose

Create a live web application for interactive route planning with elevation profiling that runs entirely in the browser without requiring a backend database.

### Target Users

* Outdoor enthusiasts
* Cyclists and hikers
* Route planners
* Anyone needing elevation-aware route planning

## Functional Requirements

### 1. Interactive Map Interface

#### 1.1 Map Display

* Map Provider: OpenStreetMap tiles via open source services
* Initial View: Centered on user's location (or default location if geolocation unavailable)
* Zoom Controls: Standard zoom in/out functionality
* Pan Support: Click and drag map navigation

#### 1.2 Pin Placement System

* Pin Creation: Single click on map to place pins
* First Pin: Automatically designated as starting point (different visual style)
* Pin Sequence: Subsequent pins create waypoints in chronological order
* Pin Management:
  + Right-click to remove pins
  + Drag pins to reposition
  + Maximum 20 pins to prevent performance issues

#### 1.3 Route Optimization

* Routing Engine: Use OpenRouteService API or similar open source routing
* Route Type: Road network only (no off-road routing)
* Route Display: Clear line visualization between pins
* Real-time Updates: Route recalculates automatically when pins are moved/added/removed

### 2. Elevation Profile Display

#### 2.1 Profile Visualization

* Position: Below the map interface
* Style: Similar to Komoot's elevation profile
* Data Source: Elevation data from OpenStreetMap or open elevation APIs
* X-axis: Distance along route
* Y-axis: Elevation in meters

#### 2.2 Gradient Visualization

* Color Coding:
  + Normal sections: Blue/green color
  + Steep sections (>5° gradient): Red/orange color
* Gradient Calculation: Calculate slope between elevation points
* Smooth Transitions: Gradual color transitions between different gradient zones

#### 2.3 Interactive Features

* Hover Effects: Show exact elevation and distance on hover
* Map Synchronization: Clicking on elevation profile highlights corresponding map location

### 3. Technical Requirements

#### 3.1 Architecture

* Type: Single Page Application (SPA)
* Framework: Vanilla JavaScript, React, or Vue.js
* No Backend: All processing client-side
* No Database: No persistent data storage

#### 3.2 APIs and Services

* Map Tiles: OpenStreetMap
* Routing: OpenRouteService API or GraphHopper
* Elevation Data: Open-Elevation API or similar
* Rate Limiting: Implement request throttling to respect API limits

#### 3.3 Performance Requirements

* Load Time: Initial page load < 3 seconds
* Route Calculation: < 2 seconds for routes up to 100km
* Elevation Profile: Generate within 1 second of route calculation
* Memory Usage: Efficient cleanup of unused route data

## User Interface Requirements

### 4.1 Layout

* Top Section: Interactive map (70% of viewport height)
* Bottom Section: Elevation profile (30% of viewport height)
* Responsive: Adapt to different screen sizes
* Mobile Friendly: Touch-optimized for tablets/phones

### 4.2 Controls

* Clear Route Button: Remove all pins and reset
* Download Route: Export GPX file (optional)
* Route Statistics: Display total distance, elevation gain/loss, estimated time

### 4.3 Visual Design

* Color Scheme: Modern, high-contrast colors
* Pin Styling:
  + Start pin: Green with number "1"
  + Waypoint pins: Blue with sequential numbers
  + Selected pin: Highlighted border
* Route Line: Bold, easily visible color (e.g., blue or purple)

## Technical Constraints

### 5.1 Browser Support

* Modern Browsers: Chrome 80+, Firefox 75+, Safari 13+, Edge 80+
* JavaScript: ES6+ features acceptable
* Mobile Browsers: iOS Safari, Chrome Mobile

### 5.2 API Dependencies

* Primary Routing: OpenRouteService (free tier: 2000 requests/day)
* Fallback Routing: GraphHopper or MapBox (if needed)
* Elevation: Open-Elevation API (unlimited, open source)
* Error Handling: Graceful degradation if APIs unavailable

### 5.3 Data Storage

* Session Only: Route data stored in browser memory only
* No Persistence: Data lost on page refresh (by design)
* Local Storage: Only for user preferences (map center, zoom level)

## Success Criteria

### 6.1 Functionality

* [ ] User can place pins on map with single click
* [ ] Route automatically calculates between pins using road network
* [ ] Elevation profile displays below map
* [ ] Gradient sections >5° show in different color
* [ ] Application runs smoothly without backend dependencies

### 6.2 Performance

* [ ] Route calculation completes within 2 seconds
* [ ] Elevation profile renders within 1 second
* [ ] No memory leaks during extended use
* [ ] Responsive on mobile devices

### 6.3 User Experience

* [ ] Intuitive pin placement and management
* [ ] Clear visual feedback for all interactions
* [ ] Smooth transitions and animations
* [ ] Error messages when APIs fail

## Optional Enhancements (Future Iterations)

* Route Profiles: Walking, cycling, driving options
* Weather Integration: Current weather along route
* Photo Integration: Add photos to pins
* Route Sharing: Share routes via URL
* Offline Mode: Cache map tiles for offline use

## Development Notes

### Technology Stack Recommendations

* Frontend: React with Leaflet.js for mapping
* Styling: Tailwind CSS or styled-components
* Charts: Chart.js or D3.js for elevation profile
* Build Tool: Vite or Create React App

### API Keys Required

* OpenRouteService API key (free registration)
* Consider backup services for reliability

### Testing Requirements

* Test with various route lengths (1km to 100km+)
* Test gradient calculation accuracy
* Test API error handling
* Cross-browser compatibility testing