# Leaflet Web Map — Program Documentation

## 1. Overview

This application is a web-based interactive mapping tool built using the Leaflet JavaScript library. It integrates both Web Map Service (WMS) layers and vector-based GeoJSON datasets to display, analyze, and interact with geospatial data. The project focuses on user interaction, allowing users to explore geographic boundaries (ADM0–ADM2) of Latvia, add custom drawings, and import or export data in GeoJSON format.

## 2. Core Features

The application includes several main functional components:

• Base map switching between OpenStreetMap (OSM), an alternate OSM WMS background, and satellite imagery (Esri World Imagery).

• Overlay layers representing Latvia’s administrative boundaries (country, regional, and sub-regional).

• Dynamic feature popups and tooltips displaying attribute data such as name, area (km²), and coordinates.

• User-drawing tools powered by Leaflet.draw for creating points, polygons, circles, rectangles, and polylines.

• Import and export functionality for GeoJSON files to preserve user-drawn features.

• A responsive search system to locate geographic features by name across multiple layers.

• Layer opacity control for custom-drawn elements, enhancing visualization flexibility.

• Interactive legends and sidebars for better data organization and readability.

## 3. Technical Architecture

The system is entirely client-side, using HTML, CSS, and JavaScript. It loads the Leaflet library and its extensions (Leaflet.draw) directly from public CDNs. The code is modular, separating map initialization, data loading, drawing tools, search functionality, and user interface components. Key design elements include pane management for label overlays, event-driven interactivity, and on-demand fetching of GeoJSON boundary files.

Administrative boundaries (ADM0–ADM2) are loaded dynamically using the Fetch API. Each feature is styled based on deterministic color generation derived from feature names. The interface also provides live area calculation, centroid computation, and zoom/selection controls for each feature.

## 4. User Interaction Workflow

1. Launching the map loads a base layer centered on Valmiera, Latvia.

2. Users can toggle between base maps (OSM, OSM WMS, or Satellite) and display boundary overlays.

3. Clicking a region highlights it, brings it to front, and opens an attribute popup with calculated metrics.

4. Users can draw new features using the drawing toolbar, assign names/types, and save them via export.

5. Existing drawings can be imported, edited, or deleted interactively.

6. A search field allows finding features by name, automatically zooming and highlighting the result.

7. Legends and opacity sliders provide real-time control over visualization clarity.

## 5. Design and Usability Considerations

The program uses a minimalist, responsive interface for better usability. Compact floating controls, tooltips, and a clean color scheme ensure visual clarity. The system employs accessibility-friendly controls, including keyboard navigation for search results and descriptive tooltips for interactive layers.

## 6. Conclusion

This Leaflet-based mapping system demonstrates a flexible, interactive environment for spatial data exploration. It effectively combines open-source mapping services, vector data manipulation, and client-side processing into a cohesive application suitable for educational, municipal, or GIS demonstration purposes.