



Motorcycle Repair Shops

European Directory

Project Documentation

Generated: December 16, 2025

Table of Contents

1. Project Overview
2. Technology Stack
3. Development Timeline
4. Key Features
5. Architecture & Components
6. Data Management
7. Setup & Installation
8. Project Structure
9. Future Enhancements

1. Project Overview

The Motorcycle Repair Shops European Directory is a modern, completely FREE web application designed to help motorcycle enthusiasts and riders find reliable repair shops across 27 European countries. Built with cutting-edge web technologies, this application provides an intuitive interface for discovering, searching, and locating motorcycle service centers throughout Europe.

Key Highlights:

- ✓ Covers 27 European Union countries
- ✓ 3000+ motorcycle repair shops
- ✓ 100% FREE - No API costs, no billing required
- ✓ Interactive maps powered by OpenStreetMap
- ✓ Fully responsive design for all devices
- ✓ Real-time search and filtering capabilities
- ✓ Modern, gradient-based UI design

2. Technology Stack

Frontend Technologies:

Next.js 16: The latest version of React's premier framework, providing server-side rendering, static site generation, and optimized performance out of the box.

React 19: The newest version of React, offering improved performance, enhanced hooks, and better developer experience with concurrent features.

TypeScript: Provides type safety throughout the application, reducing bugs and improving code maintainability.

Tailwind CSS 4: Utility-first CSS framework enabling rapid UI development with consistent design patterns and responsive layouts.

Mapping Solution:

Leaflet + OpenStreetMap: A completely FREE mapping solution that requires no API keys, no billing setup, and has no usage limits. Leaflet is a lightweight JavaScript library for interactive maps, while OpenStreetMap provides community-driven, open-source map data.

Data Management:

CSV File Storage: Shop data is stored in CSV format for easy management and portability. The application includes a custom CSV parser that handles quoted fields and complex data structures.

Supabase (Optional): PostgreSQL-based backend service for advanced features and real-time capabilities. The free tier is sufficient for most use cases.

3. Development Timeline

The project was developed through several iterative phases, each adding significant functionality and improvements:

Phase	Development Focus	Key Achievements
Phase 1 Week 1	Initial Setup	<ul style="list-style-type: none">• Created Next.js project• Basic project structure• Initial commit
Phase 2 Week 2-3	Core Features	<ul style="list-style-type: none">• Country filtering system• Google Maps integration (later replaced)• Modern UI with gradients• Responsive design implementation
Phase 3 Week 4	Configuration & Docs	<ul style="list-style-type: none">• Comprehensive setup guides• API key configuration• Setup checker script• Quick start documentation
Phase 4 Week 5	Maps Migration	<ul style="list-style-type: none">• Replaced Google Maps with OpenStreetMap• Integrated Leaflet library• Removed API key requirements• Made app 100% FREE to use
Phase 5 Week 6-7	Data Enhancement	<ul style="list-style-type: none">• Added CSV file support• Built custom CSV parser• Migrated to CSV data storage• Data setup documentation
Phase 6 Recent	Optimization	<ul style="list-style-type: none">• Fixed shop list display issues• Optimized for 3000+ shops• Performance improvements• Bug fixes and refinements

4. Key Features

4.1 Interactive Maps

The application features a fully interactive map powered by Leaflet and OpenStreetMap. Users can:

- View all motorcycle shops as custom-styled markers on the map
- Zoom and pan to explore different regions
- Click on markers to see detailed shop information in popups
- Automatic map centering based on filtered results
- Smart bounds adjustment when filtering by country
- Custom motorcycle-themed markers with gradient styling

The map automatically adjusts its view based on the selected shops, ensuring users always see the most relevant information without manual navigation.

4.2 Country Filtering System

A sophisticated country filtering system allows users to narrow down their search:

- Desktop view displays an elegant grid of all 27 EU countries
- Mobile view uses a dropdown selector for space efficiency
- Each country displays its flag emoji and name
- Real-time shop count updates when countries are selected
- "All Countries" option to reset the filter
- Smooth animations and hover effects for better UX

The system covers: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, and Sweden.

4.3 Smart Search Functionality

The application includes a powerful search system that enables users to quickly find specific shops:

- Search by shop name, city, street address, or country code
- Real-time filtering as users type
- Case-insensitive search for better usability
- Search results counter showing matching shops
- Works in combination with country filtering

- Optimized for performance even with 3000+ shops

The search functionality uses React's useMemo hook to optimize performance, ensuring smooth filtering even with large datasets.

4.4 Dual View Modes

Users can switch between two distinct viewing modes:

Map View: Displays all shops on an interactive map with custom markers. Perfect for geographical exploration and finding shops in specific areas.

List View: Shows shops in a beautiful card-based grid layout. Each card includes:

- Shop name with gradient header
- Complete address information
- Contact details (phone, email, website)
- Country flag and name
- Direct link to Google Maps
- Hover effects and smooth transitions

The toggle between views is seamless, maintaining the current filter and search state.

4.5 Responsive Design

The application is fully responsive and optimized for all device sizes:

Desktop (1920px+): Full-featured interface with grid-based country selector, large map view, and 3-column shop cards.

Laptop (1024px - 1919px): Optimized layout with 2-column shop cards and adjusted spacing for comfortable viewing.

Tablet (768px - 1023px): Adaptive layout with 2-column cards, touch-optimized controls, and appropriate font sizes.

Mobile (320px - 767px): Single-column layout, dropdown country selector, stacked controls, and touch-friendly interface elements.

All interactions are optimized for touch and mouse input, ensuring a great experience regardless of device type.

5. Architecture & Components

The application follows a modern React architecture with clear separation of concerns:

5.1 MotorcycleShops Component

The main component that orchestrates the entire application:

- Manages application state (shops, filters, search, view mode)
- Fetches data from CSV file on component mount
- Implements useMemo for optimized filtering
- Renders header with statistics
- Coordinates child components
- Handles view mode switching

This component serves as the container for all other components and manages the data flow throughout the application.

5.2 CountrySelector Component

A smart component that provides country filtering functionality:

- Displays all 27 EU countries with flags
- Responsive grid layout (desktop) / dropdown (mobile)
- Highlights selected country
- Passes selection back to parent component
- Smooth animations and hover effects

The component automatically adapts its layout based on screen size, providing the best user experience for each device type.

5.3 ShopMap Component

An advanced mapping component built with React-Leaflet:

- Dynamically imported to prevent SSR issues
- Custom map markers with motorcycle icons
- Automatic bounds adjustment based on shops
- Interactive popups with shop information
- Links to external mapping services
- Optimized rendering for large datasets

The component includes a MapBoundsHandler sub-component that automatically adjusts the map view when the shop list changes, ensuring users always see relevant data.

5.4 CSV Parser Utility

A custom-built CSV parsing utility that handles data loading:

- Parses CSV files with proper quote handling
- Transforms flat CSV data into structured objects
- Maps country names to ISO codes
- Handles missing or malformed data gracefully
- Supports various CSV field formats

The parser is designed to handle the complexities of real-world CSV data, including quoted fields, commas within values, and varying data quality.

6. Data Management

The application uses a CSV-based data management system for simplicity and portability:

Data Source:

All motorcycle shop data is sourced from OpenStreetMap (OSM), the world's largest collaborative mapping project. The data includes shops tagged as:

- shop=motorcycle - Dedicated motorcycle shops
- craft=motorcycle - Motorcycle craft and repair services
- amenity=car_repair with motorcycle=yes - Multi-service repair shops

Data is fetched using the Overpass API, which allows querying OSM data with complex filters.

Data Structure:

Each shop record contains the following information:

- Unique ID
- Shop name
- Geographic coordinates (latitude, longitude)
- Complete address (street, house number, postal code, city)
- Country code (ISO 3166-1 alpha-2)
- Contact information (phone, email, website)
- Additional tags from OpenStreetMap

The data is stored in a CSV file located at `/public/data/eu_motorcycle_repairs.csv`, making it easy to update and maintain.

Data Updates:

The project includes a Python script (`scripts/fetch_osm_data.py`) that can fetch fresh data from OpenStreetMap. This allows the database to be updated with new shops and changes to existing entries. The script can be run with:

```
npm run fetch:data
```

This ensures the application always has access to the most current information.

7. Setup & Installation

Setting up the project is straightforward and requires minimal configuration:

Prerequisites:

- Node.js 18 or higher
- npm or yarn package manager
- Git for version control
- (Optional) Supabase account for advanced features

Important: No API keys are required for maps! The application uses OpenStreetMap, which is completely free and requires no registration.

Installation Steps:

```
1. Clone the repository: git clone <repository-url> cd motorcycle 2. Install dependencies: npm install 3. (Optional) Set up environment variables: cp .env.local.example .env.local # Edit .env.local with your Supabase credentials if needed 4. Start the development server: npm run dev 5. Open your browser to http://localhost:3000
```

Available Scripts:

- `npm run dev` - Start development server with hot reload
- `npm run build` - Create optimized production build
- `npm start` - Start production server
- `npm run fetch:data` - Fetch latest data from OpenStreetMap

8. Project Structure

```
motorcycle/ ■■■ src/ ■ ■■■ app/ # Next.js App Router ■ ■ ■■■ layout.tsx #
Root layout component ■ ■ ■■■ page.tsx # Home page ■ ■ ■■■ globals.css #
Global styles ■ ■ ■■■ favicon.ico # Site favicon ■ ■■■ components/ ■ ■
■■■ CountrySelector/ # Country filtering component ■ ■ ■■■ index.tsx ■ ■
■■■ MotorcycleShops/ # Main shops component ■ ■ ■■■ index.tsx ■ ■ ■■■
ShopMap/ # Interactive map component ■ ■ ■■■ index.tsx ■ ■■■ data/ ■ ■
■■■ countries.ts # EU countries configuration ■ ■■■ utils/ ■ ■■■
csvParser.ts # CSV parsing utility ■■■ public/ ■ ■■■ data/ ■ ■■■
eu_motorcycle_repairs.csv # Shop data (3000+ entries) ■■■ scripts/ ■ ■■■
fetch_osm_data.py # OSM data fetching script ■■■ supabase/ ■ ■■■
supabaseClient.js # Supabase configuration ■■■ package.json # Dependencies
and scripts ■■■ tsconfig.json # TypeScript configuration ■■■
tailwind.config.js # Tailwind CSS configuration ■■■ next.config.ts # Next.js
configuration ■■■ README.md # Project documentation
```

The project follows Next.js 13+ App Router conventions with a clear separation between presentation components, utilities, and data. This structure makes the codebase easy to navigate and maintain.

9. Future Enhancements

The project has a strong foundation and several potential enhancements could further improve its value:

Feature	Description	Benefit
User Reviews	Allow users to rate and review shops	Community-driven quality insights
Advanced Filters	Filter by services, certifications, ratings	More precise shop discovery
Route Planning	Integration with routing services	Help plan motorcycle trips
Favorites System	Save preferred shops to a personal list	Quick access to trusted shops
Mobile App	Native iOS/Android applications	Better mobile experience
Offline Support	Progressive Web App features	Access without internet connection
Multilingual	Support for multiple European languages	Wider accessibility
Business Portal	Allow shops to claim and update listings	More accurate, current information

Conclusion

The Motorcycle Repair Shops European Directory represents a modern approach to solving a real-world problem: helping motorcycle riders find reliable service centers across Europe. By leveraging cutting-edge web technologies and open-source mapping solutions, the project delivers a professional, feature-rich application that is completely free to use and deploy.

The development journey showcased in this document demonstrates the evolution from initial concept through multiple iterations, each adding value and improving the user experience. The decision to migrate from Google Maps to OpenStreetMap particularly highlights the project's commitment to accessibility and eliminating barriers to entry.

With 3000+ shops across 27 countries, responsive design, intelligent search and filtering, and an intuitive interface, this application provides real value to the European motorcycle community while serving as an excellent example of modern web development practices.



Project: Motorcycle Repair Shops - European Directory
Documentation Generated: December 16, 2025 at 11:00 AM
Technology: Next.js 16 • React 19 • TypeScript • Tailwind CSS 4
Repository: github.com/Svpriyaa2808/motorcycle