

Project Overview

Build a modern, interactive web mapping application using React with TypeScript that integrates Maptiler SDK for base maps, implements drawing functionality, and provides sophisticated layer management capabilities.

Core Technical Requirements

1. Base Map Implementation

- Integrate Maptiler SDK to display interactive base maps
- Ensure proper TypeScript typing for map instances and events
- Implement basic map controls (zoom, pan, rotation)

2. Drawing Operations

- Implement polygon drawing functionality on the map
- Support for creating, editing, and deleting geometric shapes
- Real-time visual feedback during drawing operations

3. Layer Management System

- Create a layer management interface for users to add/remove layers
- Each layer must contain its own set of drawn features (polygons, shapes)
- Implement layer-specific isolation - operations affect only the active layer

4. Layer Visibility Controls

- Provide toggle buttons/switches for each layer to show/hide features
- Visual indicators for layer visibility status
- Maintain layer state when toggling visibility

5. State Management

- Implement Zustand for global state management
- Store layer data, active layer state, and drawing mode status
- Ensure type safety with proper TypeScript interfaces

Design & UX Requirements

- Create a sleek, modern interface focused on layer functionality
- Intuitive layer management panel with clear visual hierarchy
- Responsive design that works across different screen sizes
- Professional color scheme and typography
- Smooth transitions and interactive feedback

Optional: Hosting

- Deploy the application to a hosting platform of choice (Vercel, Netlify, AWS, etc.)
- Configure production environment variables for Maptiler API keys
- Ensure proper build optimization and performance
- Set up custom domain if available (optional)
- Implement proper CORS configuration if needed

Deliverables

1. Fully functional React TypeScript application
2. Source code with proper documentation
3. Live hosted demo (if hosting option is chosen)
4. Brief documentation on setup and usage

Technical Stack

- React 18+ with TypeScript
- Maptiler SDK
- Zustand for state management
- Modern CSS/styling solution

Success Criteria

- Smooth drawing experience on the map
- Reliable layer isolation and management
- Clean, professional user interface
- Type-safe implementation throughout
- Hosted and accessible application (if hosting implemented)

Notes

- Hosting is optional but recommended for demonstration purposes
- Focus on core functionality first, then proceed with deployment
- Ensure sensitive keys are properly handled in environment variables
- Pushing it to GitHub is Mandatory.