Building a Solar Potential Mapper: From Scratch to Deployment

Here's a comprehensive step-by-step guide to build a solar potential mapper web application from scratch to deployment:

Phase 1: Planning and Setup

Step 1: Define Requirements

1. Outline core features (roof analysis, solar potential calculation, energy savings estimates)
2. Define target users (homeowners, solar installers, urban planners)
3. Create wireframes/mockups of the user interface
4. Determine what data sources you'll need (elevation data, weather data, satellite imagery)

Step 2: Set Up Development Environment

1. Install code editor (VS Code, WebStorm, etc.)
2. Install Git for version control
3. Create a GitHub/GitLab repository
4. Set up a local development server

Step 3: Select Technology Stack

1. Choose frontend framework (React recommended for interactive maps)
2. Select backend framework (Node.js with Express or Python with Django/Flask)
3. Choose database (PostgreSQL with PostGIS extension for spatial data)
4. Select mapping library (Mapbox GL JS or Leaflet)

Phase 2: Backend Development

Step 4: Create Database Schema

1. Design tables for user data, saved projects, and calculation results
2. Set up spatial data tables for map features
3. Implement PostgreSQL with PostGIS extension
4. Create database migration scripts

Step 5: Develop Core API Endpoints

1. Build user authentication/authorization system
2. Create endpoints for address geocoding
3. Develop API for retrieving building footprints
4. Build endpoints for solar calculations

Step 6: Implement Solar Calculation Engine

1. Create algorithms for solar radiation estimation
2. Implement roof orientation and slope detection
3. Develop shading analysis functionality
4. Build energy production estimation algorithm
5. Create endpoints to access these calculations

Step 7: Data Integration

1. Set up connection to elevation data sources (USGS, national services)
2. Integrate weather data API (NREL, Solcast, or similar)
3. Implement satellite imagery retrieval
4. Set up data preprocessing pipeline

Phase 3: Frontend Development

Step 8: Set Up Frontend Project

1. Initialize project with Create React App or Vite
2. Set up routing (React Router)
3. Configure state management (Redux/Context API)
4. Implement basic styling framework (Tailwind/Material UI)

Step 9: Build Core UI Components

1. Create header, footer, and navigation components
2. Implement responsive layout
3. Build form components for address input
4. Design results display components

Step 10: Develop Interactive Map

1. Implement map using Mapbox GL JS or Leaflet
2. Add base layers (satellite, street maps)
3. Create building footprint overlay
4. Implement drawing tools for manual roof outlining

5. Build the 3D visualization of buildings (if applicable)

Step 11: Create Solar Analysis UI

1. Develop interface for displaying solar potential
2. Create visualizations for sun path and shading
3. Build financial calculation components (ROI, savings)
4. Implement system size recommendation UI

Phase 4: Integration and Testing

Step 12: Connect Frontend to Backend

1. Implement API service layer
2. Set up authentication flow
3. Create data fetching logic
4. Implement error handling

Step 13: Testing

1. Write unit tests for critical functions
2. Perform integration testing of API endpoints
3. Conduct browser compatibility testing
4. Test on different devices for responsiveness
5. Run performance testing, especially for map rendering

Step 14: Optimization

1. Implement caching for map tiles and calculation results
2. Optimize database queries
3. Minify and bundle frontend assets
4. Implement lazy loading for components
5. Add progress indicators for long-running calculations

Phase 5: Deployment

Step 15: Prepare for Deployment

1. Set up environment variables
2. Configure production database
3. Set up logging and monitoring
4. Create deployment scripts

Step 16: Backend Deployment

1. Provision server (AWS EC2, Google Cloud, etc.)
2. Set up Docker containers for services
3. Configure Nginx as reverse proxy
4. Set up SSL certificates for HTTPS

Step 17: Frontend Deployment

1. Build production version of frontend
2. Deploy to cloud storage (AWS S3, Vercel, Netlify)
3. Configure CDN for static assets
4. Set up custom domain

Step 18: Database Deployment

1. Set up production database instance
2. Configure backups and replication
3. Implement database migration strategy
4. Set up database monitoring

Step 19: Launch and Monitor

1. Perform final end-to-end testing
2. Launch the application
3. Set up application monitoring (New Relic, Datadog)
4. Configure error tracking (Sentry)

Phase 6: Post-Launch

Step 20: Ongoing Maintenance

1. Monitor application performance
2. Fix bugs as they arise
3. Collect user feedback
4. Plan for future feature enhancements
5. Regularly update dependencies and security patches

This guide provides a comprehensive roadmap for building your solar potential mapping application. Each step can be further broken down into smaller tasks as you progress through development.