Gazi University

Faculty of Engineering
Department of Computer Engineering

CENG463

Geographic Information Systems

Dr. Tuba Çağlıkantar

Term Project

Alper Kürşat Ünver
151180066

Table of Contents

Introduction

GIS

“A geographic information system is a special case of information systems where the database consists of observations on spatially distributed features, activities or events, which are definable in space as points, lines, or areas. A geographic information system manipulates data about these points, lines, and areas to retrieve data for ad hoc queries and analyses” (Kenneth Dueker, Portland State University, 1979).

Purpose

Document

Purpose of the document is to give detailed information about GIS Term Project.

Project

The purpose of the project is exhibiting an earthquake dataset along with the heights(as much as accurate that Mapbox data allows us) of 3D buildings to create visual awareness about earthquake as a natural disaster that has very disastrous history in Turkey Republic which also happens very frequently.

Availability

Project files are available in https://github.com/alpeer/gis-project
Live production software is available on https://alpeer.github.io/gis-project/

Click to Launch Online Software

Technologies

This project is engineered with set of tools and libraries which allows us to exhibit visual and structural composition of map and the data programatically.

Visual Studio Code

(an Integrated Development Environment)

VSCode IDE which is fast, performant and plugin oriented has been chosen as IDE for fitting requirements.

Git

(a Version Control System)

VCS is a system, a file oriented database that holds changes between project files across different versions. Git VCS has been chosen for its compact structure.

GitHub

(a Git Hosting Platform)

GitHub has been chosen as Git hosting service to keep project.

JavaScript

(a Programming Language)

Flexible, portable, powerful and the most popular programming language; JavaScript has been used in both backend and frontend side of the project.

Node JS

(a V8 JavaScript Runtime)

Node JS version 16 (Long Term Support) used for static development environment that covers NPM, Webpack, Eslint, Node Sass

NPM

(a node.js package management software)

npm package manager used for organizing and installing JavaScript libraries as dependencies.

Static frontend development tools

Eslint

(a JavaScript linter)

Eslint has been used to enforce clean code style for enhancing readability and specific JS coding rules to avoid possible defects that can occur by using bad coding practices.

Sass

(a CSS transpiler)

NodeJS v16+ compatible version(v6) of node sass library has been used to write CSS files more readable, organised and easier.

Webpack

(a JavaScript Bundler)

Latest version(5.66) of Webpack bundler has been used to bundle and transpile ES6+ project files.

Frontend Runtime Libraries

React

(a JavaScript front end library)

Latest version (17.0.2) of React JS used for creating units of user interface other than map.
Also used for creating map control plugins -such as alternative two dimensional Compass- that are essential to Mapbox have been created with React.

Material UI

(a React UI components library)

Generic building blocks(components) of **Material UI for React** -such as *TextInput*, *TabView*…- has been used for creating project specific user interface components.

Map Libraries & Tools

Mapbox GL

(a WebGL based JavaScript front end map library)

Latest version of Mapbox GL map UI library used for creating and managing(flying to specific three dimensional viewport) three dimensional base map user interface.

Mapbox Studio

(an Online styling tool)

The map cartography has been created/curated with Mapbox Studio.
Built in plug-ins such as streets, 3D buildings, roads, areas, point of interests used.

Mapbox Rest API

(an Online Map Service)

Mapbox REST API allows fetching vector layers that has been created with Mapbox Studio.

Methods

Even this project has been developed as demo project of *the purpose* -that explained in Purpose section-, engineering principles below are followed to reduce barriers at time of decision for creating a commercial GIS product that can use this project as bootstrap.

React Functional Components

Functional approach has been used while creating React components(functional components) to enhance performance and reduce memory cost which is already drained by MapboxGL (which occurs naturally as the nature of drawing and managing complex layers are costly.)

SOLID Principles

Even the core approach is functional, SOLID principles still can and have been used while engineering the project.

Single Responsibility Principle:
Every component should have one single responsibility to avoid complexity

Open/Close Principle:
Components should be *open* to extensions *close* to changes.

Liskov's Substitution Principle:
"If it looks like a duck, quacks like a duck but it needs batteries, you probably made wrong abstraction" states the importance of proper abstraction while deriving functionality.

Interface Segregation:
Avoiding to enforce methods to use unnecessary methods by isolating behavioural interface.

Dependency Inversion:
Higher order component shouldn't depend on lower one.

Datasets

Earthquakes

Dataset (kandilli.csv) has been acquired manually from Bogazici Uni. Kandilli RETMC.
Detailed information: https://github.com/alpeer/gis-project/blob/master/datasets/earthquakes

Active Faults

Dataset (gem_active_faults.geojson) has been acquired from GEM Global Active Faults Database
Detailed information: https://github.com/alpeer/gis-project/blob/master/datasets/faults

Cartography

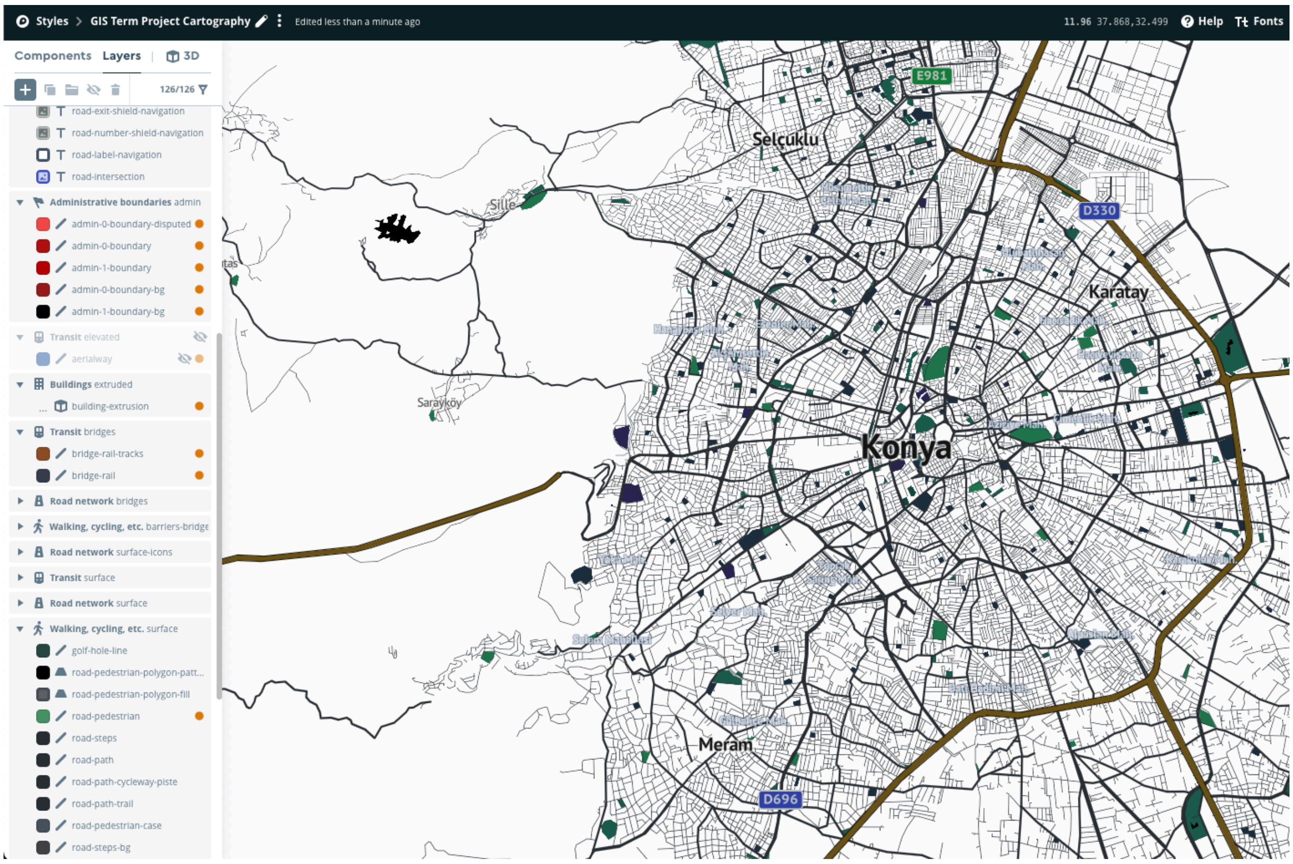
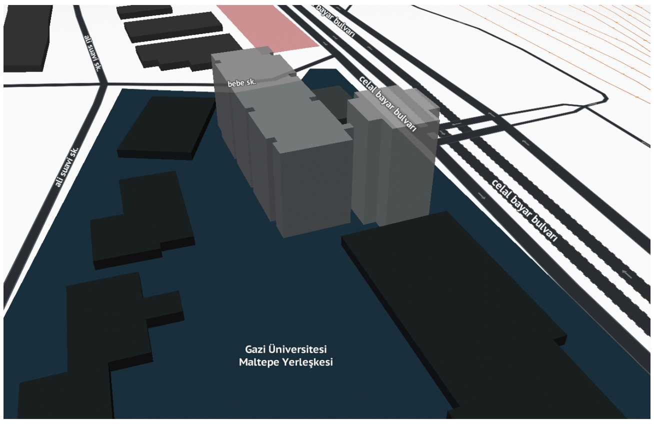


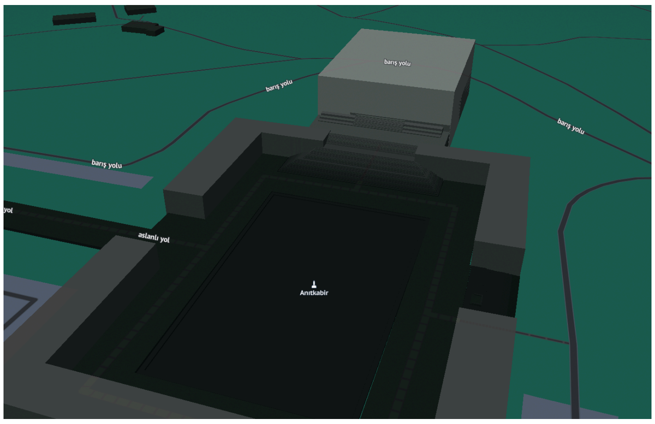
Figure 1. Creating Cartography via Mapbox Studio

Simple colors with high contrast selected while designing map to create minimalist style.

Features & Screenshots

3D Buildings

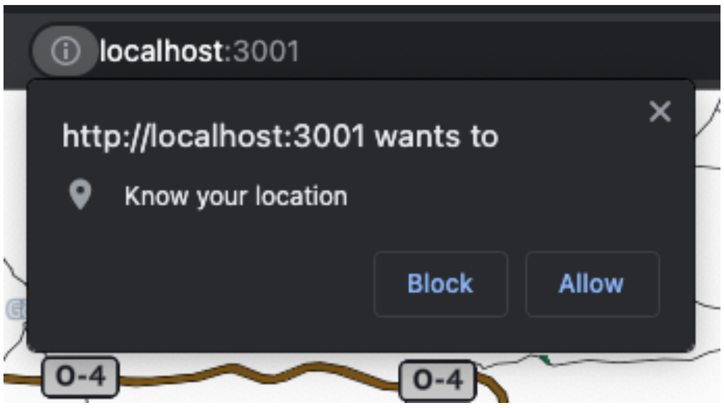




View 1. Gazi University, Maltepe Campus View 2. Ataturk's Mausoleum

User Location

User can let map to focus on it's coordinates with a single click.


localhost:3001

http://localhost:3001 wants to

Know your location

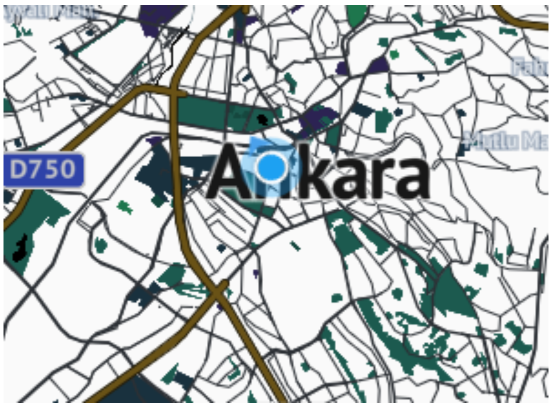


Figure 2. User Location Permission Prompt View 3. User Location

Compass

A Unique Compass has been designed *just for this project*.

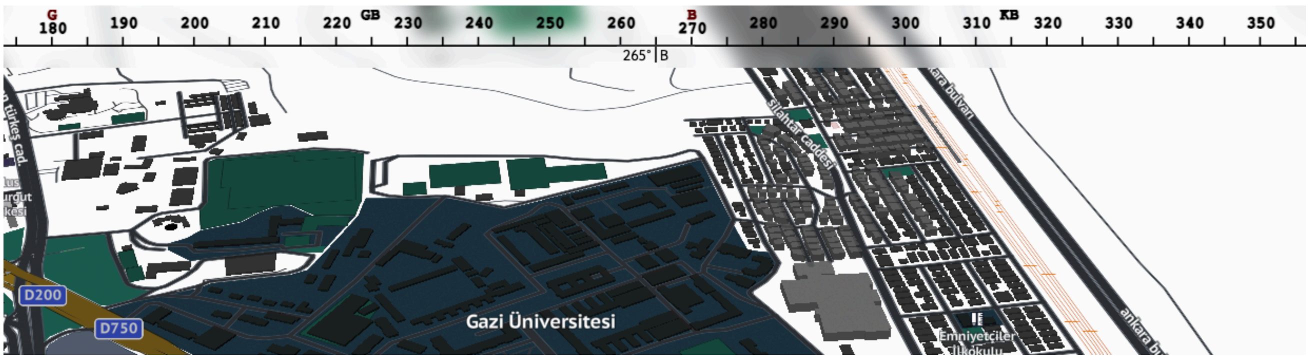
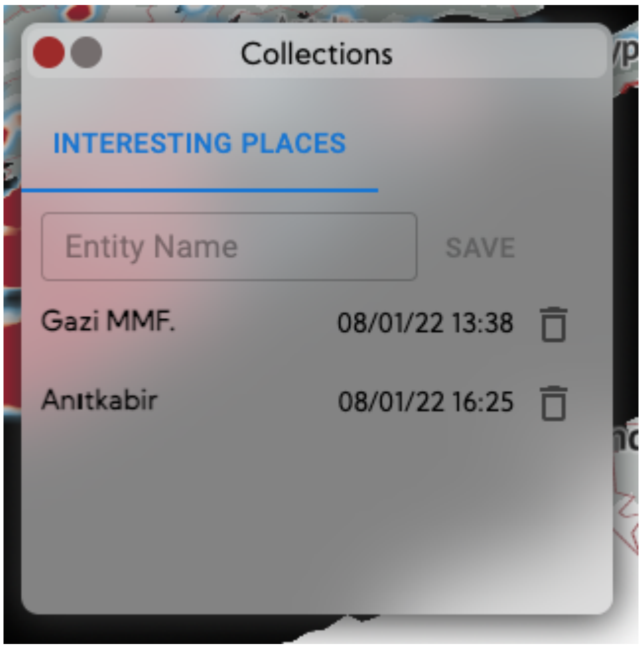


Figure 3. Unique Compass Design

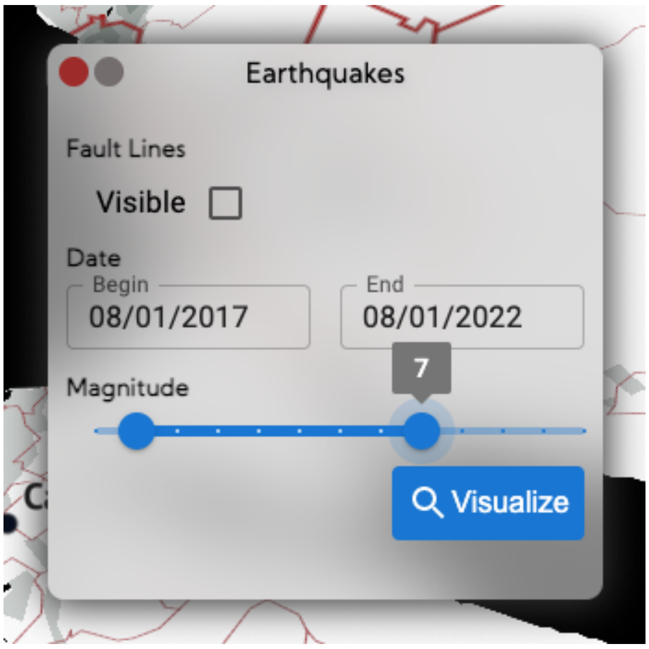
Windows

Draggable Windows allows user to interact map with modifying and storing data. Style and functionality designed in-house.



Gazi MMF. 08/01/22 13:38

Anıtkabir 08/01/22 16:25



Fault Lines

Visible ☐

Date

Begin 08/01/2017

End 08/01/2022

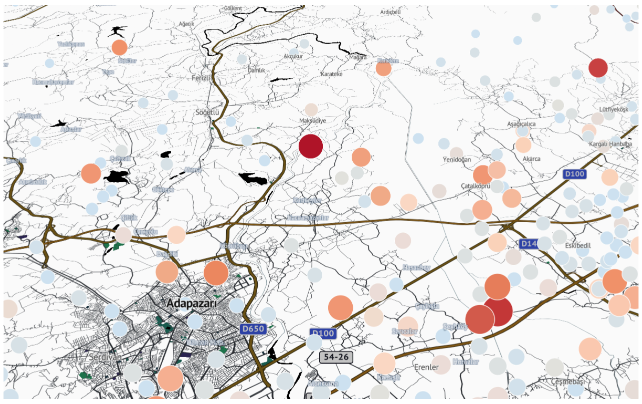
Magnitude 7

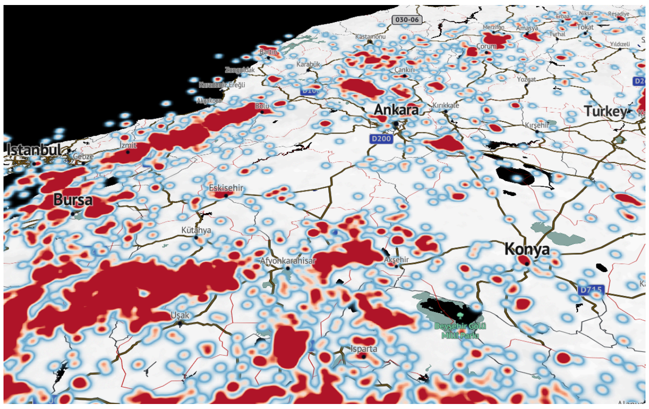
Figure 4: User can add views as favorites to collections

Figure 5. Earthquake Visualisation Settings Window

Earthquake Data Visualisation

User can see earthquake data as both heat-map (zoom<9) and circles (zoom>9).

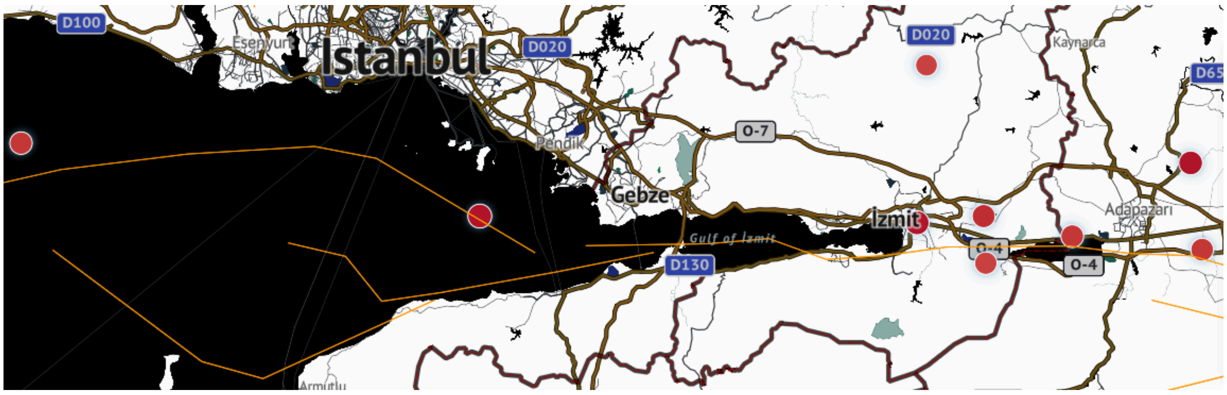




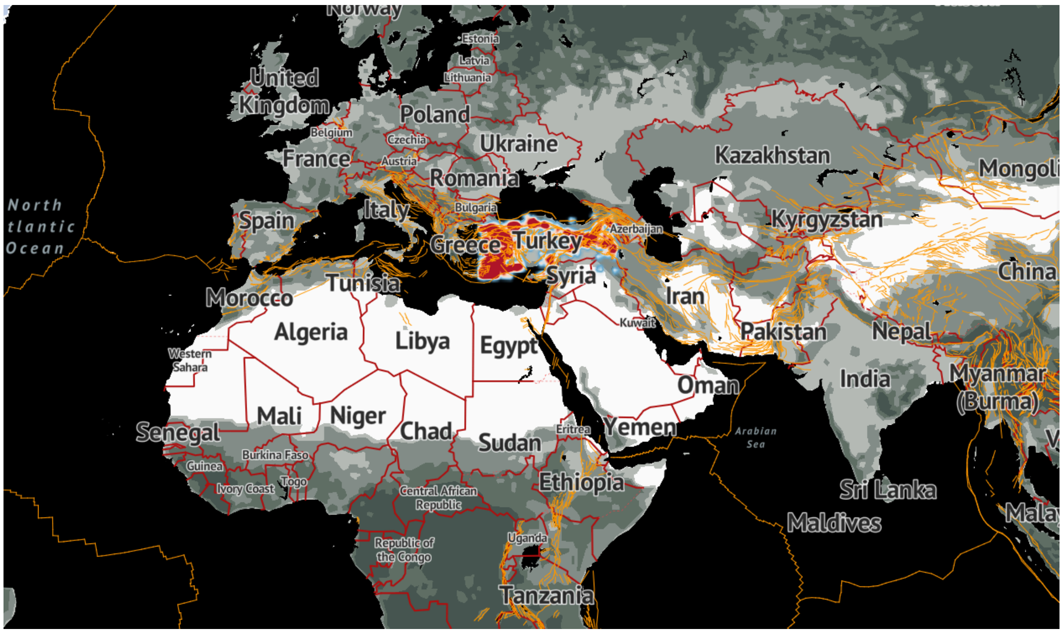
View 4. Earthquakes (Circles)

View 5. Earthquakes (Heatmap)

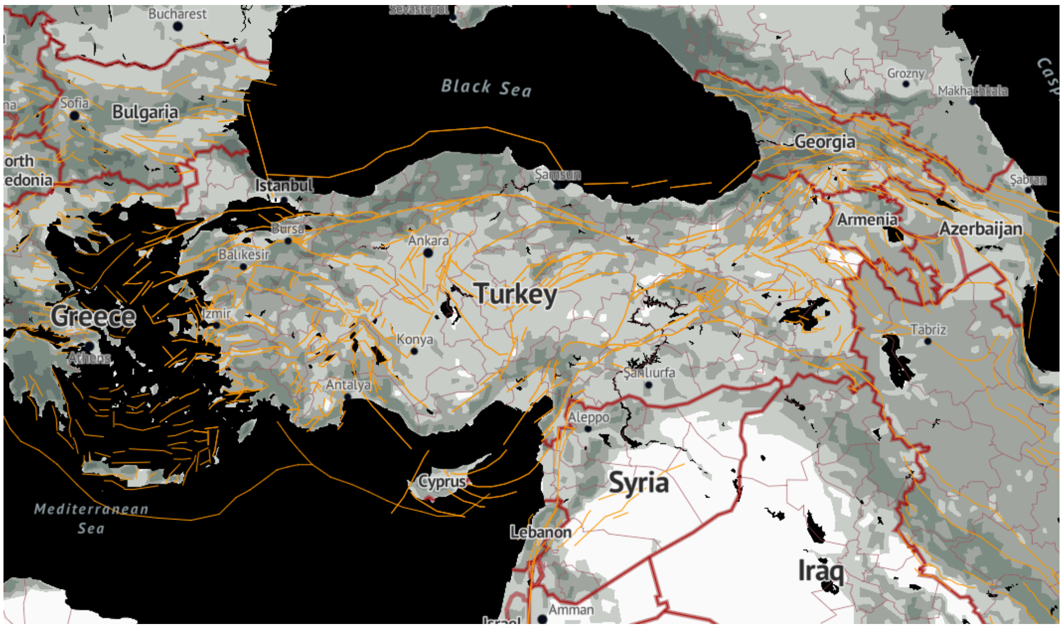
Active Faults Visualisation



View 6. Active Faults



View 7. Active Faults (Global View)



View 8. Active Faults (Republic of Turkey)

README file

As in project directory: https://github.com/alpeer/gis-project, readme file available for development purpose.

GIS Project

Installation

```
git clone https://github.com/alpeer/gis-project.git
cd gis-project
npm install
```

Commands

Start dev server

```
npm start
```

Production Build

```
npm run build
```

Analyze

Builds and generates source-map, launches analyzer

```
npm run analyze
```

Serve

```
npm run build
npm run serve
```