CrimeSpot Documentation

Executive Summary

CrimeSpot is a web app that aims to serve both regular users who is interested in crime data populated across the world, and law&crime department officers who which to see visualized crime data and perform CRUD operations on the data.

With CrimeSpot, user could search by year & crime type and then countries in the map will be colored in different degrees to indicate the severeness. This gives the user a prompt idea on the difference on crimes rates across countries.

Also, this app allows another type of user - admin, to add a pin point on the map with offender information. Admin user could search on the map, hit ‘enter’ to enter location information, then add offender name and description, by confirming it a marker with offender name would appear on the map, clicking this marker, a pop up window would show above the marker, in the pop up says the description of this offender.

The interaction with marker and map facilitate law enforcement people to visualize and keep track of offender data.

Project Background

The background of this project arises when international crime is getting more and more severe. Geological information could help with crime cases a lot.

The problem this app wish to address is to visualize offender information on the map, and reflect how severe a crime has been in each country. I chose this topic because the current available crime data management system in government and administrations are getting more outdated and in need of a more modernized system.

The objectives of this map would be to allow easy ways for users to add markers on the map for circumstances where, like allowing user to recognize how many similar type offender would show up in one area, where do offenders live and a short description of them.

Project Development

In the register page, ask user to input user name, user email, password, repeat password, get all these data and use passport to authenticate user, user information will be stored in mongodb. If user already have an account, user could also choose to jump to login from register page. During register, user can choose to register as admin or regular users. Once registered, user cannot change its user type.

Login is handled by passport to authenticate too. login requires user to enter username and password. In both register and login, I used bcrypt to ‘salt’ the password, and store the hashed password in mongodb.

After successful register or login, user would be directed to home page. In home page there is a map from mapbox api, next to it is a search bar that allows user to search by crime year and crime type, once user confirmed search, the map will be colored in different degrees to reflect in the specific year for the specific crime, what is the crime rate rate in that country. The coloring is realized from add layer in mapbox api, and the datasource for the layer comes from mongodb data. The map would also show some markers added by admin users to show offender information. By clicking on the marker, it will pop up offender name and description. The marker and popup are implemented by mapbox api.

On the navbar, user could jump from home page to admin page. This page is only allowed for admin users. I used a middleware to handle the “isloggedin” and “isadmin” identity check.

In the admin page, there is a map with no coloring layer, instead, it only shows markers that have been previously added by admins. next to the map are some input boxes that allow admin user to enter offender information, and add a marker with this information. On the map there is a search bar for searching for location, by search on this search bar and hit enter, the location input will automatically update its value with the result from the search bar. The search bar is implemented with the geocoding feature in mapbox.

The profile page constitute with a default icon image, user name and user email, by clicking editing profile, user could edit there name and email and update information.

Project Completeness

This project can be improved in many ways. What it has implemented so far demonstrates the potential in it. This app reflects crime rate data on a world map by the coloring degree. It can be improved in the way that when a user clicks the country, the map displays the crime rate number on the country, add a hover effect when a country is clicked, add color bars to indicate the range of crime rate based on the color degree. Moreover, since the data is stored in mongodb now, it's not dynamic data yet. I implemented in this simpler way for demonstration purpose. But with some pre-processing with the fetched data, the map can reflect the latest data. Another way to implement this would be keep data in the mongodb, but allow admin user to upload crime data. This is technically half-way implemented, because it's same logic for user to add marker user to mongodb. With some input area or with a file upload button it would facilitate the uploading data process for admin.

Project Reconsiderations

One assumption the app made is when user are updating profile, the input value is always not empty. Given more time I would implement a validateform function to check every form before submitting to avoid null value error.

I would also wish to have implemented this project in vue, so it’s more modern language. Somehow sometimes ejs does not load scripts file I included, so I have to put all script data in the ejs file. With vue it could be more modular, and independent.

I would also consider adding more routes. Like a route to user’s collections where it stores all the canvas pdf generated from the map centered in a place by user’s choice.