Haskell Earthquake Mapper

To compile from terminal:

• cd into HaskellEarthquakeMapper file

• run *stack build*

• install any dependencies you don’t have already

• run *stack exec HaskellEarthquakeMapper-exe*

Our program takes an earthquake API and lets you query it to produce a map in browser with points at each earthquake. If you hover over a pin in the map, you can see the magnitude, place and date of that earthquake. We used two months of data from the API since otherwise the dataset was too big and the API would return an error. This is because the creator of the API put a 20000 result limit on indivudal calls.

It works by first downloading the API then parsing the relevant information and putting this into two separate databases. The first database contains the year, month, day, place, magnitude, latitude, longitude, depth and URL for each earthquake. The URL takes you to a webpage with more information on the specific earthquake. The other database sorts the earthquakes by regions which we have predefined. “Database initialised!” will come up if the first database has been created successfully, and then “Regions database initialised!” indicates the regions database has also created. If either of these messages do not come up then there has been an error in the creation of the tables in the database.

The parser works by taking the JSON as a string and splitting it at appropriate points to retrieve relevant information. This information is found within an object called properties. All the rest of the information is ignored. These are then made into Haskell data types, taking into account that some of the properties are not there for all of our data by using maybe. The time is given as Epoch time in the JSON so that is transformed into UTC using the Data.UnixTime module the System.Time module.

The user is then asked if they would like to query by date, magnitude or region. Once the user has selected, they are then asked to narrow their search. If date was chosen, they will be asked for a day and then a month. The main database will then be queried for all earthquakes in the database after that date. If they picked magnitude, they will be asked to select a minimum magnitude based on the Richter scale, and the main database will be queried for all earthquakes with a magnitude greater than or equal to the chosen minimum. If region was picked, the user is asked to select one of our predefined regions. The regions database is then queried for all earthquakes within the selected region.

Using the Google Maps API, a pre made html page with inline Javascript exists in the root directory of the project. After querying the database for the users selected magnitude, region or time, a JSON file is created in the root of the project. This JSON data is created by taking a a list of lists containing the data returned from the database and selecting each element needed for the JSON file. This is then concatenated into a string and write to the JSON file. The JS in the html file reads from the JSON file and populates a Google Map. The user can hold his/her mouse over each pin to see the date, magnitude and exact location as well as zoom in to take a closer look at the location of the earthquake. A system command of “open index.hmtl” is sent to the users terminal on running the program. If the user has not set a default browser or is using a less common version of Linux this command will not open in the browser. Instead, the user will have to open the index.html manually to see the output. Finally, the program deletes the database to ensure that the program does not throw an error - because the database already exists - when re-running the program.