# Presentation

Hi I’m Mark Packham. The site is a geographic educational app intended to provide users with information on countries. It uses a Vue.js front end and for the backend the NoSQL database, MongoDB with routing carried out by the Node.js web application framework, Express. All the code is stored in Github along with setup documentation. The educational site takes in data from 4 apis, 3 from external sites and one created by ourselves that our backend framework handles. Cameron created a seed file we can run whenever we need to repopulate our database after testing to make sure our app is retrieving the correct information. The external apis are the country one we used during labs, REST Countries. One that displays maps, the Open Street Map and one that provides our information for the country quiz that Neil and Nathan worked on, the Open Trivia Database.

We’re using fairly commonly used packages on our backend, nodemon to save us from constantly having to restart Express when dealing with updates, body-parser to handle the JSON, obviously a MongoDB driver to interact with our database and cors so we can pass information between our locally hosted Vue and Express apps. The REST request testing app, Insomia proved helpful for experimenting with our Express based CRUD requests before we got round to building the necessary web forms in Vue.

[Load the front page]

The front end loads with a lot of gimmicky CSS animations mainly to make it more memorable to an audience, a bit like a cheesy advert. A simple color scheme of green and blue to represent land and sea along with the use of standard web safe fonts. A fair amount of html symbols are to give the users eyes a break from the dry serious levels of data.

The bulk of components are held in the homepage App.vue, [App homepage] thankfully we’re not dealing with vast amounts of data and Vue has the ability to minify its CSS and JavaScript in a script intended for production environments to improve performance. CSS is kept in the in CSS specific files and imported into components rather than using inline styling. This is to stop components from getting too bloated and hard to debug containing both CSS and JavaScript and Html.

Countries are divided into specific regions with sub regions and users can click on a country in a sub region to receive more information about it. [Demo France]

There’s also a modal made purely with CSS to provide the user with even greater amounts of information about a country. [Modal Example - New Zealand]

The countries’ latitude and longitude data from the api is used in order to populate the url of an open source map site that giving us the map location. We have to use string interpolation and Vue’s v-bind: on the src file to achieve this. And place the map inside an iFrame.

[Map String Interpolation]

The site is capable of displaying random facts about countries by clicking a button [Country Facts - Get]. If you’re not fond of that specific fact you can delete it or create your own in a text field. So these are typical Get, Post and Delete Requests. There’s also a nuclear option where you can erase all data in the database which shan’t be used for the purposes of this demo. [Destroy All] whilst it doesn’t serve much practical purpose I was curious to see how to wipe out a collection devoid of just dropping the entire database and using MongoDB’s .remove() with an empty object proved to be the answer.

[Neil & Nathan to go in depth over the quiz and Cameron talks about working with maps & generating the data - introduce selves with full names like Melinda said]