



General Assembly - Software Engineering Immersive

GreenWorld - A MERN stack app

Overview

This was the third project in the General Assembly Software Engineering Immersive course. We built a MERN stack app - which means we used Mongo, Express, React and Node to put together a fully functioning application that we deployed via MongoDB and Heroku. Four of us worked on the app for seven days, and we then presented the completed app to the rest of the class.

Together we developed the full backend of the website and the full frontend, and so it was an immersive experience that allowed us to practice a wide variety the coding skills that we've developed over the past few months.

You can see the files for the completed app on GitHub pages [here](#).

Brief

We had to:

- Work in a team of four, using Git to code collaboratively
- Build a full-stack application by making our own backend and our own frontend
- Use an Express API to serve our data from a Mongo database
- Consume our API with a separate frontend built with React
- Create a complete product, which meant multiple relationships and CRUD functionality
- Implement thoughtful user stories/wireframes, identifying the features that were core to MVP
- Ensure we developed a visually impressive design
- Deploy our app online

Technologies Used

- HTML
- CSS
- JavaScript (ES6)
- React
- APIs
- Insomnia
- Express.js
- Webpack
- Node.js
- Axios
- Bulma
- Mapbox
- MongoDB
- Heroku
- Git and GitHub
- VS Code

Approach

The brief for this project was quite open and so we thought carefully about how we could make something that was both useful and relevant. We zeroed in on an idea to create a website that would help the user to locate environmentally friendly products near to their home.

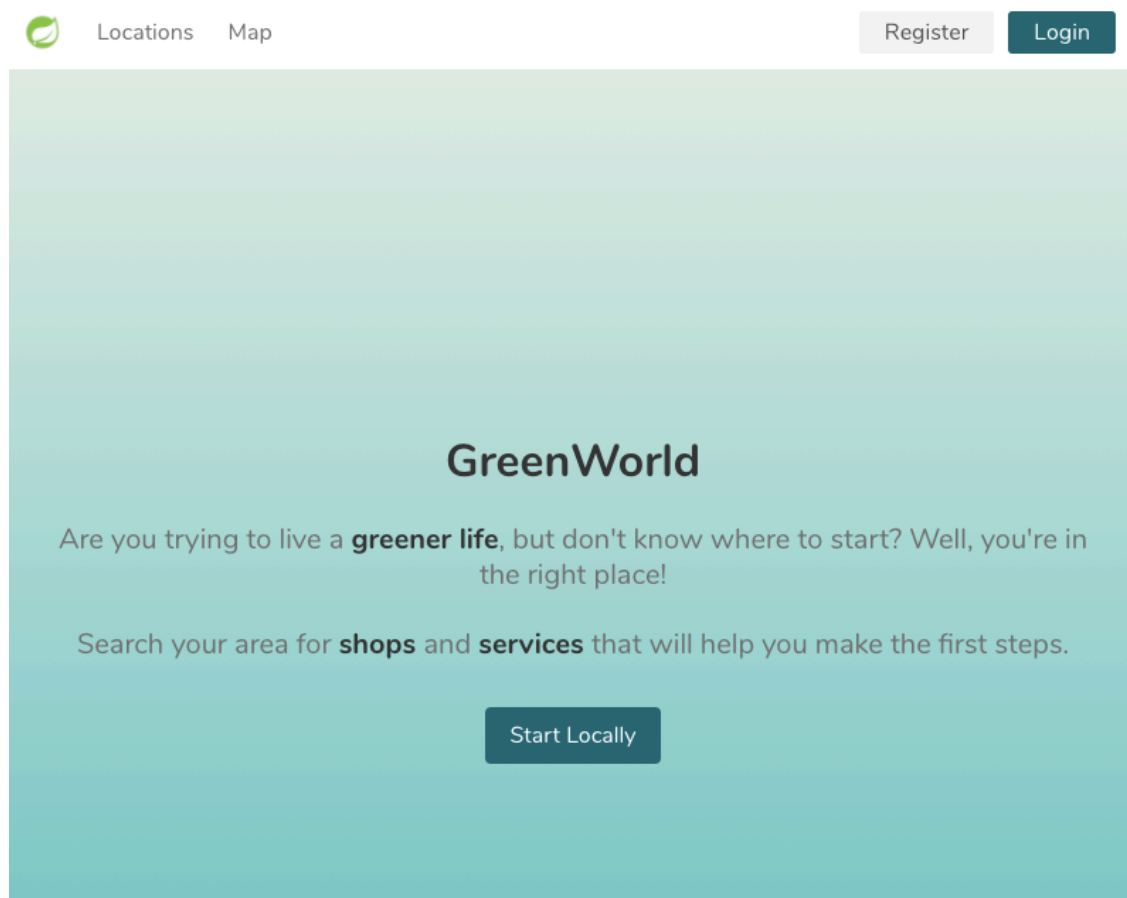
We laid out the user experience and then considered the code that we would need. The next step was to produce seed data that included locations that were near to where team members lived. This was used to test the code in the early stages, before we introduced the API. We then wrote the code for the backend and tested our seed API in Insomnia, before looking for a greater depth of data to introduce.

I discovered that the website Yelp produces a powerful API with thousands of businesses on it. I tested it in Insomnia and found it had exactly what we needed, so I pitched it to the rest of the team for incorporation into our project. We tested it further in Insomnia and then commenced the building of the frontend.


We wanted to create a complete user experience, and so the app includes functionality such as signing up, creating a user profile and logging in. It also includes the ability to post, edit and delete comments, and users can add new locations as well. They can also rate the venues that they have visited by giving them from 1 to 5 stars.

Screenshots

This is the Home page, where the 'Start Locally' button takes the user to the Map page, which launches with their own coordinates. Note also the 'Register' and the 'Login' buttons on the navigation bar:



This is the Locations page, which contains a list of venues near to the user:

 [Locations](#) [Map](#) [Register](#) [Login](#)


[Sustainable Groceries](#) [Farmers Market](#) [Farm Shop](#) [Restaurant](#) [Circular Economy](#) [Cycling](#) [Upcycling/Repair](#)

Dash Vegan

Sustainable Groceries

City: Nottingham

About: We are a vegan food delivery service providing great quality and wholesome vegan, gluten free food and plastic free options in Nottingham and the surrounding area through VDelivery and throughout the UK on this site.




Shop Zero

Sustainable Groceries

City: Nottingham

About: We are here to help you shop more consciously and reduce your environmental



And this is the Login page:

 [Locations](#) [Map](#) [Register](#) [Login](#)

Email

e.g. example@example.com

Password

Please enter your password

[Log In](#)

Lessons Learned

A lot of lessons were learnt. The need to start the project by carefully considering the user experience, as well as all the coding that would entail, was a key lesson. Whiteboarding the overall plan for the website was an important part of that and something we did well, but perhaps could have done in greater detail.

Dividing the tasks up amongst the team and setting clear goals for the completion of code blocks was something we could have spent more time on, particularly in the earlier days of the project. However, we stayed on track on and met our targets as the project progressed. Spending time sharing our completed code through Git was crucial to the success of this and taking time to do it paid off.

We hadn't used APIs with authentication keys prior to this project, so the detail of making that work gave us useful knowledge for the future. I worked on this part myself, and this involved understanding how to manipulate the API data so that it served our needs (mapping, etc.) and that built on the strong level of knowledge developed during project 2. The Yelp API was particularly advanced but it was exciting to see the ease in which I could make it work as the project progressed.

In addition, we included a second API that turned postcodes into coordinates (longitude and latitude), which enhances the user experience because if they add a new location they don't need to also look for the coordinates - this is done automatically as the location renders on the map.

Writing the code for adding comments and ratings was a fiddly exercise that involved close attention to detail and a keen eye for spotting minor errors in the coding.

Most of the styling was done through Bulma, and that was also a learning experience. Manipulating the features of the website so that they displayed just as we wanted them to wasn't easy. For example, I was working on making the search bar line up correctly with the 'hot filter' buttons on the Locations page and that took some careful thought. But once you've worked with Bulma on this sort of project the next time it will be much easier because I have this experience.

I worked on the Locations page and ended up with a complex code block for the Bulma styling:

```
<div className="notifications is-primary">
  {filterLocations().map((location, index) => {
    return <div key={index} >
      <Link to={`/locations/${location._id}`}>
        <div className="tile is-parent px-0">
          <div className="tile is-child box">
            <div className="columns">
              <div className="column">
                <p className="title is-4">{location.name}</p>
                <div className="tags">
                  {location.category.map((category, index) => {
                    return <div className="tag is-warning"
                      key={index}>
                      {category}
                    </div>
                  })}
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    )}
  </div>
```

Another part of the code that was difficult to get right was the way the map page interacted with the user's location. Making the map open at the location the user is based at required careful reading of the Map GL documentation, and the same was true of the editing the map location based on the API that we put in to convert a postcode to a set of coordinates:

```
// Updating position of map based on browser location
function useLocation() {
  const newViewport = {
    ...viewport,
    latitude: latitude,
    longitude: longitude,
    zoom: 12
  }
  setViewport(newViewport)
}

// Updating position of map based on postcode search
function handleSubmit() {
  event.preventDefault()
  axios
    .get(`https://api.mapbox.com/geocoding/v5/mapbox.places/${searchText}.json?access_token=${process.env.MapBoxKey}`)
    .then(resp => {
      const postCodeViewport = {
        ...viewport,
        longitude: resp.data.features[0].center[0],
        latitude: resp.data.features[0].center[1]
      }
      setViewport(postCodeViewport)
    })
}
```

One of the main successes of this project came from the deployment. Prior to this project we hadn't deployed a project online and so working with Heroku and MongoDB was an extremely useful experience. Making it work in just a couple of hours was a great success, and brought a strong sense of satisfaction.

Bugs

The site functions incredibly well, but the lack of data outside of London means the site is not very useful for people based elsewhere in the country. The Yelp API only covers London. We hardcoded the API string to search the whole of the UK, but when we were testing it in Insomnia we didn't realise that it was only bringing back details of businesses in the capital. We considered many other APIs, but for example Google's data has to be paid for and so that wasn't possible.

Potential Future Features

GreenWorld already has a lot of functionality. With more data available it will be a genuinely useable website. Further functionality could include adding a more immersive map experience, such as a 'Street View' option, possibly with directions added in.

Focusing on the green theme, it would be interesting to include cycle routes and walking routes - two features that also focus on the mapping aspect of the site.

Credits

Thanks to the rest of the GreenWorld team: Dec Burns, Baltasar Romero and Florian Wilisch.