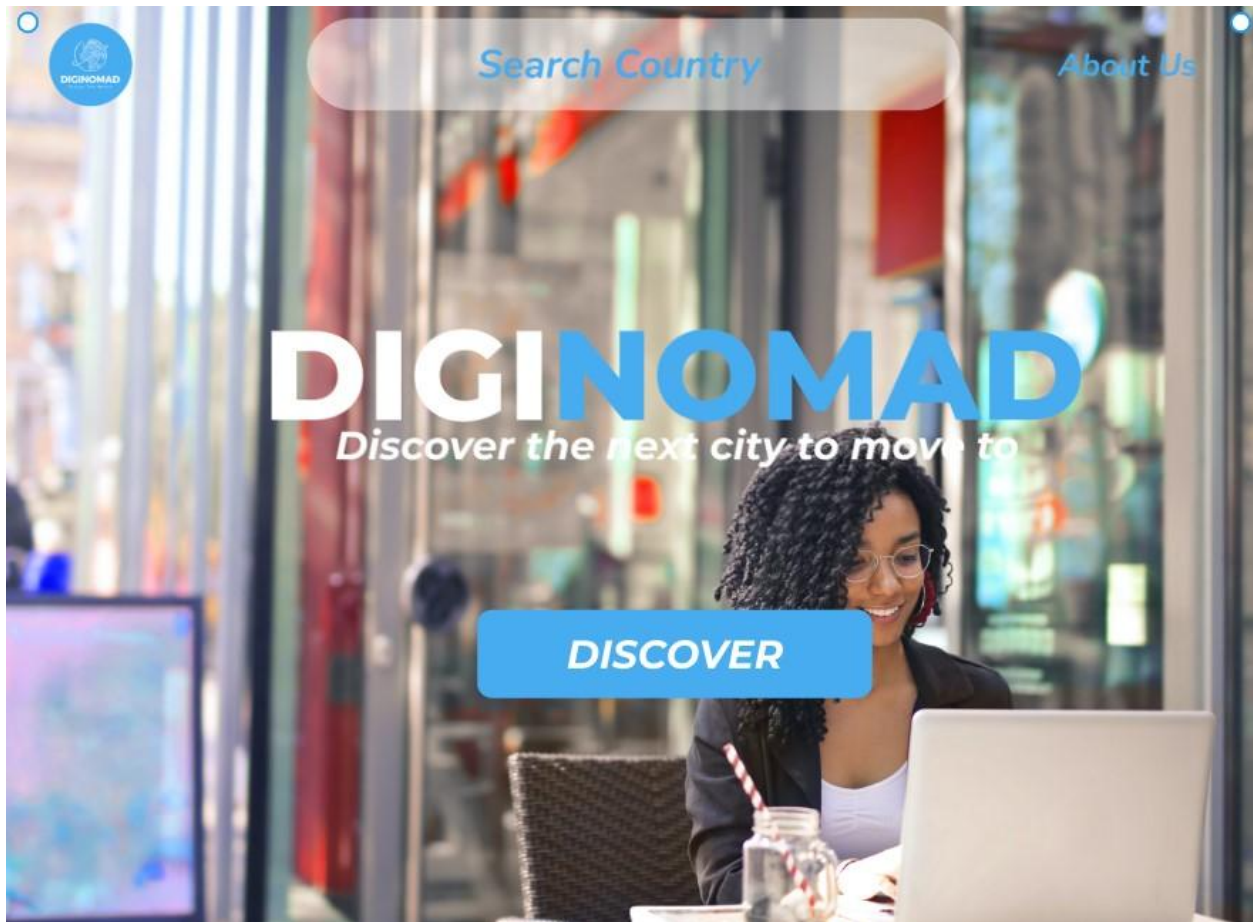


David Tendai Gondo
Student No: s2111013
WEB DESIGN TECHNOLOGIES
WEB REPORT
Coursework Two

Website Link : <https://stoichiometrical.github.io/diginomad/>



Discussion and justification of development approach you have taken

Marks(10) Excellent. Thorough and detailed discussion with critical evaluation evident.

Framework

I made the site primarily using Reactjs. Reactjs is the best option because of its component based approach and ease of integration with some essential libraries like recharts, react-scroll, react-router-dom and material ui.

API Integration

I used three APIs, one for maps, one for country facts and the other for getting prices of goods in countries. The main method for making api requests was the Fetch API. I however tested

Design Considerations

I used design thinking and user centered design to make a minimal website that provides the visitor with the most relevant information as fast as possible without going through a lot of unnecessary pages and texts. The user's main goal is to find the cost of living in a country, so I put search bar at the navigation so that that's the first thing they see on every page. On searching they are quickly shown all the details they need to know about the country.

Then at the bottom is an option to compare countries. I made this in the form of a table so that it's easy to see and compare categories. To add even better comparison I included an option to visualize the prices as a bar chart.

I only used two main colors, light blue and white which are bright and clean colors. I also included imagery of happy people to complement a happy experience the user will get. Over these approaches made sure that the site is intuitive, user friendly and effective in getting a user to achieve their goals.

Development process

The process was iterative. I used an agile approach to make as many components as possible .

Firstly, I focused on creating the full design of the site in figma before starting an coding. This was to make sure that once I get started with the code I am only focusing on the code. However , I later experienced a challenge where I had to make some tweaks to the code ,since my approach was agile. It meant changing the designs and the code but because I was using React components, it was just easy to just remove the component and replace it with a new one without affecting a large part of the design or adjacent components, so change management on design was smooth.

After setting up create-react-app, I then divided my site into two categories components and pages. Pages are the main navigation pieces of the site whereas components is where I put the smaller elements that could be reused in the different pages. Each page was to have minimal JSX and be mostly components to enhance readability and easy of navigation. In different work sprints, I would focus on certain set of components and in other I would focus on putting them together to make a concrete page. One major challenge in this was state management because I had to change the state of several components when an api request was triggered, because I am only familiar with using the standard React hook , this gave me a big challenge and I had to do further research to make the state flow work but finally figured it out using a combination of react hooks. After everything was in place I then went on to focus on making the site responsive. This wasn't a very effective approach , I should have done it during the creation of the components not after everything was done . I did research on a few libraries and media queries, the available options were media queries, bootstrap and MaterialUI , I ended up using a combination of MaterialUI and media queries.

After all the component integrations and API integrations were done, I then compiled my React app using `npm run build` to create a production version. I then deployed this version to a live site using github pages hosting.

The site is live on this link:

<https://stoichiometrical.github.io/diginomad/>

Explanation of how you have accessed data from external API

Marks(5) Excellent. Clear discussion of how data can be accessed from an external API Thorough and detailed discussion of different options available with critical evaluation evident

There are 4 main methods for accessing data from an API name, Fetch API , axios , XMLHttpRequest and jQuery.ajax .

In this project I used the Fetch API because it is easier to use, has less code and can take different data formats. It also provides asynchronous handling which is very helpful for our projects because we need to first get the data then wait before assigning it to different components that need it like the bar graph, map and the comparator tables.

GENERAL ACCESSING :

User types country name into input field. The value of the input field is assigned to a state using the React hook `useState`. When the user clicks search button, the api request is sent and the country name is set as a parameter to the api request. The API request used the Fetch API to send requests to several apis, depending on the context. The request had a header which contains options like request type, GET, it also api key and other necessary parameters. The api is sent through an async function. It returns a response object which is then converted into json and then set to an array which will be accessed by index to get its different attributes.

Here is an example using the Rest countries api

API Name	Usage	Extra
Rest countries	<p>I used this api to get facts about different countries in the world. I used it on the country page in my website. The user inputs a country name in the search bar and clicks the search icon.</p> <p>When search icon is click, it fires a <code>onClick</code> event listener which calls on a function for getting the data from the api. The function uses a javascript <code>fetch</code> method to send a GET request to rest countries. Rest countries api is open, so there is no need for api keys or any special header elements in the request. The request returns a response object which is first converted to json before being set to a variable in the component state. The different facts of the countries can then be accessed via indexing and are mapped onto suitable jsx components for display</p>	<p>The url below is the one I passed with to send a GET request. The <code>countryName</code> parameter depends on what country the user has put in the input field `https://restcountries.com/v2/name/\${countryName}`.</p> <p>The GET request is fired via a <code>useEffect</code> and set as an async function to make sure that all the data is retrieved asynchronous and that the request will respond to a change in the <code>countryName</code></p>

--	--	--

ALTERNATIVE METHODS

Method	Method of Operation
Axios	It uses an axios.get() method for sending an api request. It takes in a header with appropriate options and returns a promise which can be handled asynchronously. It is an excellent choice next to Fetch but I didn't use it because it was a little more complex, so I opted for the simpler Fetch api
jQuery.ajax	It sends asynchronous http requests to the server and returns a response object. This can then be accessed through indexing to get the specific data points. I however didn't use it because it's slower compared to fetch() as it has to be loaded from an external library. It also returns data through callbacks, which is not as effective as just promises like what fetch does. Thus I wouldn't have an effective error handling like fetch does
XMLHttpRequest	To get data, first create an instance of an XHR object then set the appropriate header options e.g. URL, api key. The request returns a response which can then be handled using a callback function. However I didn't use it because it requires a lot of boilerplate code to get started/make a simple request and does not support promises so makes handling the asynchronous requests more challenging

Description of the client-side scripting techniques you have used to present the data and provide interactivity

Marks(5) Excellent explanation of well constructed code.

The main client side scripting techniques used are :

Technique	Usage	Extra Comments
JSX	Used to mark up all React components in the website. JSX provides HTML-like tags that allow for nesting of components and different markup tags to describe the page layout	It allows you to easily render render dynamic data Improves code readability and maintainability
Reactjs	Used to create all the components of the website and the general website structure. It comes with react dom that allows for flexible and easy changes to the site. It also has many libraries that can work well with it to improve appearance and code dynamicity	It was the overall javascript library I used for the project Robust library for reusable components , easy debugging
SCSS	Used it for mainly styling the website.It is a preprocessor scripting language that extends and enhances the functionality of css . It reduces repetition, improves flexibility and comes with extra set of features for manipulating css	It is finally compiled to css for running and provides an organized css structure for our code
MaterialUI	Used Reactjs Material UI for flexibility with responsive components and icons.Mainly used it to make comparator table and some svg icons used throughout the website	It is provides user friendly and intuitive interfaces and scalable vector icons
React-scroll	Used for easy scrolling across long pages	

Discussion of alternative approaches you could have used (e.g. custom JavaScript written from scratch, jQuery, library such as React or Vue)

Marks(10) Excellent. Thorough and detailed discussion of different options available with critical evaluation evident

The alternative approaches I would have used was either using HTML,CSS and Vanilla Javascript or HTML,SCSS and jQuery

Using the first approach,(HTML,CSS and Vanilla Javascript).This approach would have been a good option because of its simplicity and light weight.The technologies dont require any extra configurations and involve less files that using React and therefore makes it easier to maintain for small sites.Because of the less integrations and libraries /modules involved these approaches would make a relatively faster website.It is also easier to set up the website using these technologies that using React.However the main disadvantage would be that code reuse, some sections of different pages are similar , these ones would have to be reused ie copying and pasting them to each page.With Reactjs there is no need for that as we just create one component for each element and just reference it where it needs to be in the page without writing the page.Some sections would need variables from other parts of the site or need slight changes , with React this is simply achieved by using props.

JSX used in React is also better than HTML because it allows javascript to be written inside html , this is particularly useful especially when making api calls and replacing text with data from html.It enhances the level of dynamicity in the code.

An alternative to the SCSS that I used is just pure CSS.Using CSS would purely works but usually results in very long files and because of my code is based on dividing the page into as many sections as possible and styling them , we would have ran out of meaningful class names to give our divs but with scss, the classnames can be nested and therefore be re-used in different contexts.SCSS also setting variables for commonly used css attributes like setting the color to a constant variable so that we dont have to type it everytime we want to use it in our css files .

jQuery.ajax and XHR(XMLHttpRequest) would have been good alternative approaches to making api requests.jQuery has inbuilt error handling and automatic data serialization.This would have been a great option that optimizes the way we way we get data from the api and also easier api debugging but the main trade off will be speed .On the other hand XHR is faster and allows for more control over the way the data is fetched from the api, this would have been helpful if we wanted more complex requests to the apis.The main disadvantage however is that the easy of customization benefit tradeoff with more boilerplate code to be written to perform a simple function.XHR is

also not compatible with other browsers so would limit the functionality for some users with newer browsers

Analysis of the advantages and disadvantages of using a bespoke approach to the development of this site rather than making use of Content Management System such as WordPress.

Marks(10) Excellent. Thorough and detailed discussion with critical evaluation evident.

A bespoke approach involves building a website from scratch by manually coding and making the design whereas using a content management system involves using pre-build templates and customer management systems to implement the website. Both approaches work but yield optimal results based on the context of usage.

The main advantage of using a bespoke approach is the flexibility for customization , by creating the code from scratch we have control over the type of design we want to use and what components we want to include in the design. It gives us greater control over how the final website will look like or even perform as we can choose technologies to optimize and make it work more efficiently and faster. Its also easy to scale the website as the number of users increase. With a bespoke approach we can start making the website with a growth mindset and we can design the architecture to accommodate increasing user base or number of requests so they would be no need to start making something anew. It is also very secure because we can customize the security protocols we include and how we integrate them into the website. However, customizability comes with the tradeoff of technical expertise. We need a technical person who is able to create the website using the given technologies , this can be very costly. Building from scratch also takes a long time this might not be ideal if we want to have a version of the website out as soon as possible. To fix bugs we would also need the technical person, to fix them or do any updates, this might not be ideal if we have a small budget or don't have inhouse IT support people working with us.

However using a CMS , can solve some of these challenges. It is easier to use and can be easily configured by a non-technical person. This makes it a better option when there isn't a large dedicated budget allocated for development. Using a CMS also takes shorter time because we have pre build interfaces or interfaces which require very little customization to set up. Using CMS ensures that a certain level of consistency and standardization is maintained through out the site. This is because templates of pages are reused and a design system can be put in place which will be easily enforced by people working on building the website. Using a CMS also benefits from regular updates from the provider of the CMS , so we don't need to be worried about having an inhouse tech support team. However , CMS have less customisation and security ..The low

security is because of prebuilt security protocols which hackers might have already found ways to exploit. There is less flexibility to build out an entirely new design because it has to follow what's provided through the CMS. It will also be difficult to scale the CMS architecture because some of them are not built to scale and scaling might be very difficult and even more costly.

Overall, a bespoke approach was the best option for our project as it allowed to customize the website based on our designs and we were able to fully implement our custom APIs and choose our hosting with greater flexibility