



## General Assembly - Software Engineering Immersive

### BITCOIN INFORMATION 💰

#### Overview

This was my second project in the General Assembly Software Engineering Immersive course. It followed a week of learning about React and so it was a real test of how quickly I had picked up the language. The project was completed in a pair in 48 hours, and then presented to the rest of the class. You can launch the app on GitHub pages [here](#).

#### Brief

The app had to:

- Consume a public API – this could have held any data but it had to work in a project.
- Have several components, at least one classical and one functional.
- The app should include a router and at least two pages.
- Include wireframes that were designed before building the app.
- Have semantically clean HTML that makes structural sense, rather than thinking about how it might look, which is the job of CSS.
- Be deployed online and accessible to the public.

#### Technologies Used

- HTML
- CSS
- JavaScript (ES6)
- React
- APIs
- Insomnia
- Git and GitHub

#### Approach

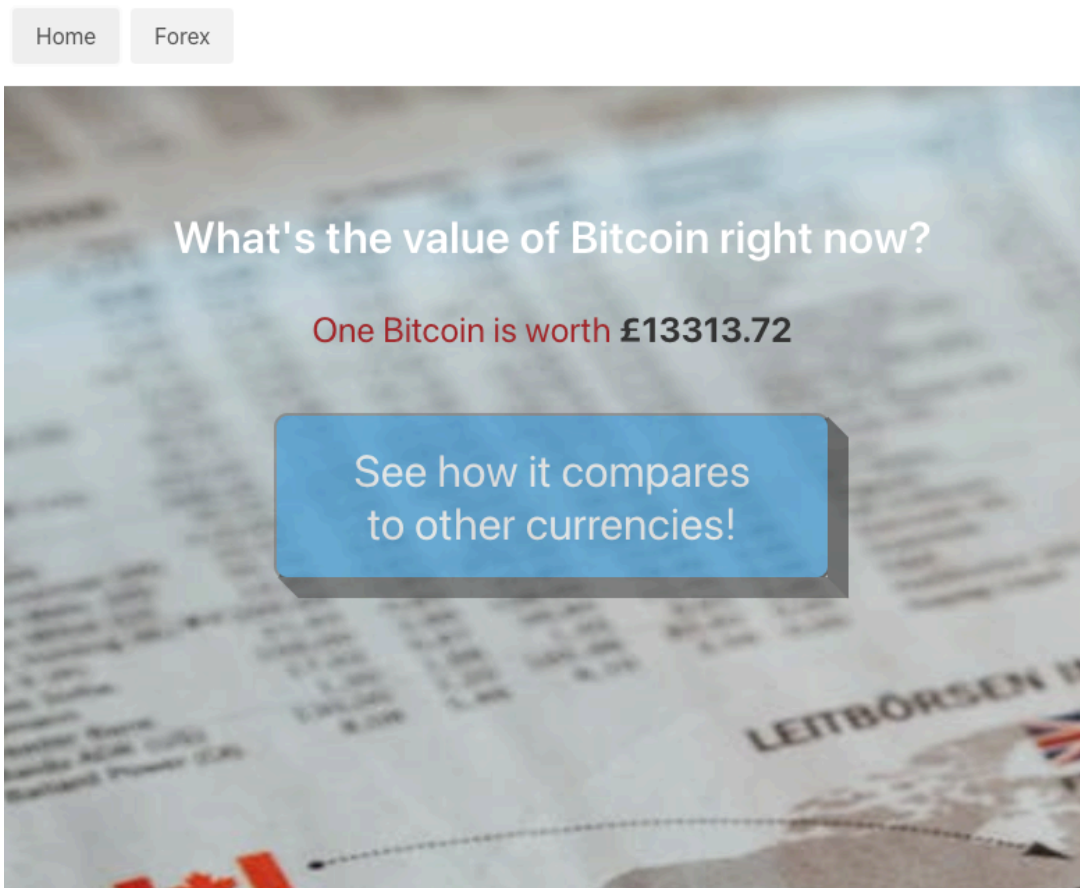
We looked at the various APIs that were available and decided that with a wealth of APIs that focus on the Bitcoin cryptocurrency, we could build an app that gave up to date Bitcoin prices compared to some of the key international currencies. The app uses APIs from Coindesk, Blockchain, Coingecko and Deezer.

The first page is relatively simple - we display the value of a Bitcoin compared to a British pound. On the second page we added other currencies to give a broader level of information.

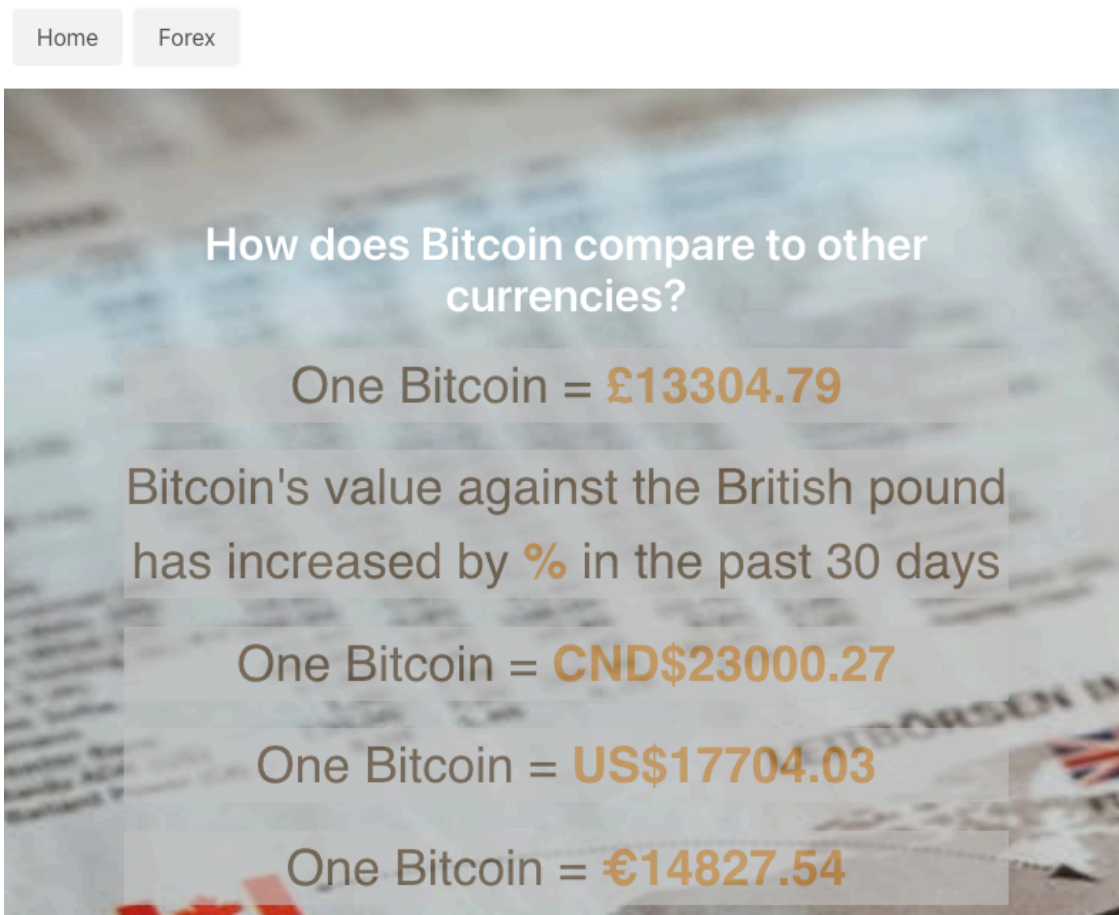
And just for fun, when the second page plays the theme song from the film Wall Street. For this we used the Deezer API.

## Screenshots

This is the Home page, which displays key Bitcoin currency information:



And this is the second page, which also plays the theme from the Wall Street film:



## Bugs

We were unable to add a comma to the numbers, so for example it will say “£10500.00” rather than “£10,500.00”. This makes it harder for the user to read.

There is also a delay of up to five seconds from when the second page loads before the music begins to play, which was frustrating when it came to presenting the project to the class.

## Lessons Learned

Many! I learnt a lot about using APIs, in particular extracting the correct data. I also learnt a lot about React Router and how to set that up correctly. There was some trial and error in the coding and that allowed me to understand how to make the music play.

I also used the console log in a lot more detail than I had before, which was particularly useful to understand what was working in the early stages of the code writing process, and to see if the data was being imported from the API successfully.

We also spent a fair amount of time getting to grips with Bulma, which wasn't as easy to work with as we had thought it would be. It takes some trial and error to get the result you want, but with some time and effort we got there.

It was also the first time I used React Audio Player and that was a success, although it required careful reading of the React literature. The code below looks relatively straightforward, but having spent just a week learning React it took a few hours to get it right:

```
const Music = () => {
  const [music, newMusic] = useState([])
  useEffect(() => {
    fetch('https://cors-anywhere.herokuapp.com/https://api.deezer.com/track/3614317')
      .then(resp => resp.json())
      .then((data) => {
        newMusic(data.preview)
      })
  }, [])
  console.log(music)
  return (
    <div>
      <ReactAudioPlayer src={music} autoPlay />
    </div>
  )
}
```

## Potential Future Features

These are endless to be honest. Some graphs to show how the value of Bitcoin has changed in certain time periods (one day, one month, six months, etc.) could add real value to the user.

## Credits

Thanks to Sagal Osman, with whom I worked on this project.