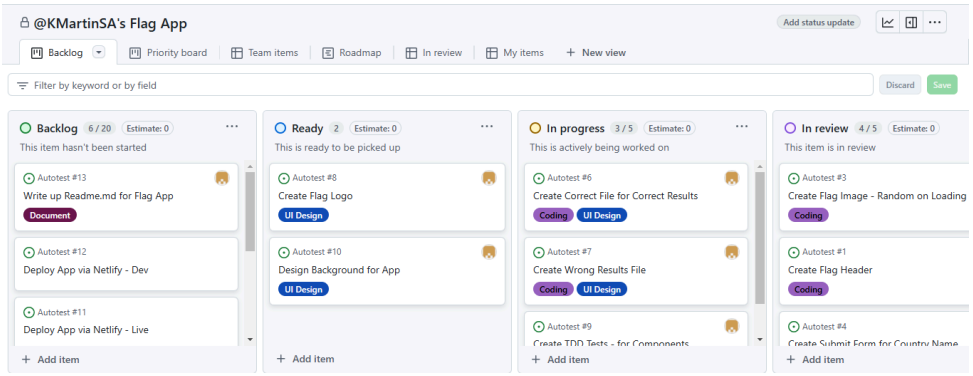
How well do you know your flags? This app created in React is a fun way to learn the flags of the world!! You will see as above an image of randomly picked flag and you just need to enter the matching country, then press submit and voila you will see if you are correct or not.

🎛️ Code Management: Git Project

To manage my tasks on this project I have been using Git Project, creating issues for each task, with an appropriate label, then assigning them to someone. Using a kanban board we can track the progress of those tickets by if they are ready, in progress, being reviewed or in the Done column.



Get Started

Prerequisite:

You would need to have Node.js installed please see link here: for both windows and Mac Then you need to have Git bash installed (if not already via another application):

Install an IDE like Visual Studio Code:

After you have those installed please clone above repo with `Git Clone <http:reponame>`

Run command: `npm start`

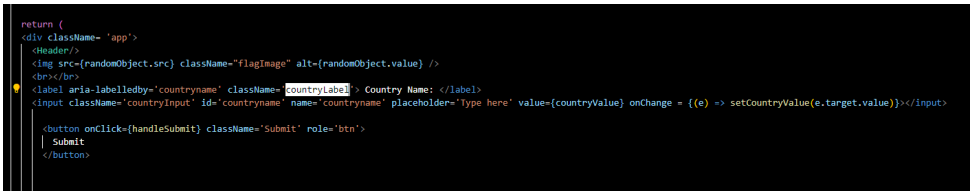
This runs the app in your local machine as below:
Open http://localhost:3000 to view it in your browser.

The page will reload when you make changes. You may also see any lint errors in the console.

You can also use other npm commands like test, build and eject please read further on for those.

💻 Creating the App

For this app I used standard coding practise of creating simplied function names, using all my classnames in the camelcase that is a better pratise when using Javascript. All the different section are broken down for a reader to clearly establish each function of a code partition and well indented for clarity. Please see image below as an example:



Testing via Jest

For testing we used an inbuilt testing functioanly of REACT called Jest that is included in our file App.test.js, you can run it in the VSC terminal at any time with the following command:

```
npm test
```

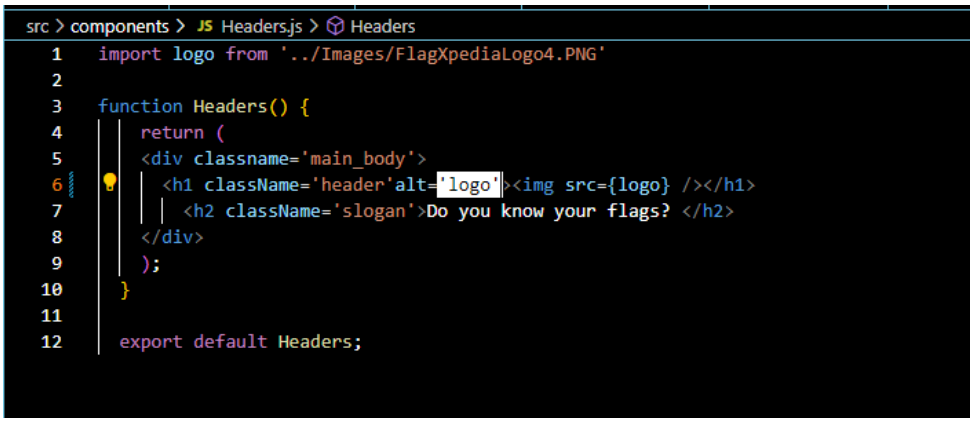
Launches the test runner in the interactive watch mode.
See the section about running tests for more information.

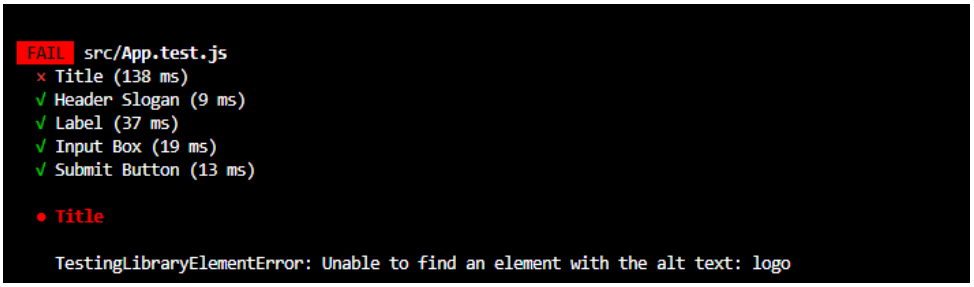
TDD Testing

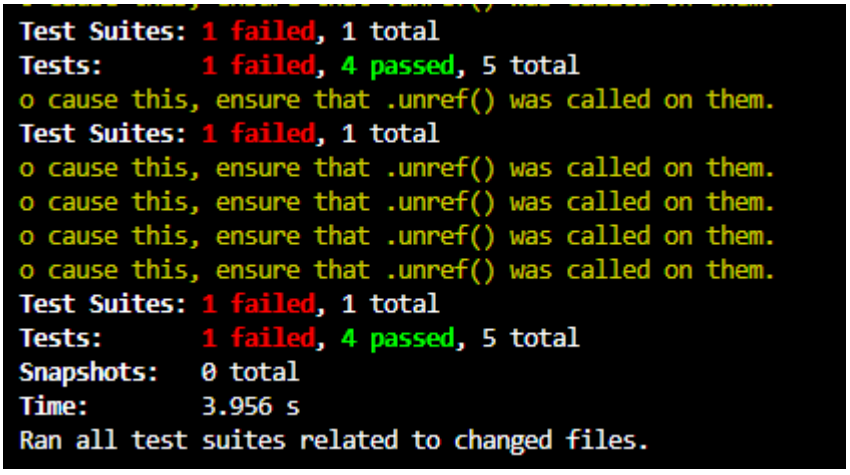
Testing in this manner mean that we could see the errors as we are building the application, that is a forward approach to testing. Please see some of the tests below:

Where are testing that you can see the title, slogan, Input label and submit boxes here. One test failed on...getting the new logo to be discovered by the

alt text at first as I was trying to add it it in the header 1 tag. As below:







We resolved that test by adding it to the actual image tag aand then it could discover the alt tag in this component. As featured below:

src > components > JS Headers.js > ...

```
1   import logo from '../Images/FlagXpediaLogo4.PNG'
2
3   function Headers() {
4       return (
5       <div classname='main_body'>
6         <h1 className='header'><img src={logo} alt='logo' /></h1>
7           <h2 className='slogan'>Do you know your flags? </h2>
8       </div>
9       );
10    }
11
12    export default Headers;
```

```
      at render (node_modules/@testing-library/react/dist/pure.js:246:10)
      at Object.<anonymous> (src/App.test.js:5:9)

 PASS  src/App.test.js
  √ Title (375 ms)
  √ Header Slogan (27 ms)
  √ Label (99 ms)
  √ Input Box (42 ms)
  √ Submit Button (66 ms)

A worker process has failed to exit gracefully and has been force exited.
o cause this, ensure that .unref() was called on them.
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        4.858 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

🖨️ Manual Testing via Excel

For manual testing I created a simple test case scenario with Excel that you can see below that just checks for most of the components in the app being displayed. Then a few to check functionality. We cover any manual testing for any scenario that the jest testing may not of been covered, giving the appp more test coverage. This is done to ensure less defects would be found in

later development or testing:

Test Case No	Test Case Name	Steps	Expected Result	Actual Result	Pass/Fail
1	Header Test	Go to UI/Localhost Check Header	FlagXpedia Logo is displayed	FlagXpedia Logo is displayed	Pass
2	Slogan Test	Go to UI/Localhost Check Slogan	Slogan displayed is "Do you know your flags"	Slogan displayed is "Do you know your flags"	Pass
3	Image Test	Go to UI/Localhost Check Image	Flag Image is Displayed	Flag Image is Displayed	Pass
4	Label Test	Go to UI/Localhost Check Label Name	Label displayed is "Country Name"	Label displayed is "Country Name"	Pass
5.1	Input Test	Go to UI/Localhost Check Input box is displayed	Input box is displayed	Input box is displayed	Pass
5.2	Input Test	Go to UI/Localhost Check Input box is displayed Type in Input Box	You can type in input box	You can type in input box	Pass
6	Submit Test	Go to UI/Localhost Type in Input Box Click Submit button	Submit button is displayed and Clickable	Submit button is displayed and Clickable	Pass
7.1	Returns Test	Go to UI/Localhost Type in Input Box Click Submit button Wait for Results to return	Input is returned	Input is returned	Pass
7.2	Results Test	Go to UI/Localhost Type in Input Box Click Submit button Wait for Results to return	Results shows the results of your guess	Results shows the results of your guess	Pass
7.3	Results Test - Correct	Go to UI/Localhost Type Correct Country Click Submit button Wait for Results to return	Shows a message that you are correct	Shows a messagae that you are correct	Pass
7.4	Results Test - Wrong	Go to UI/Localhost Type Correct Wrong Click Submit button Wait for Results to return	Shows a message that you are incorrect	Shows a message that you are incorrect	Pass

🖱 Deployment of App

`npm run build`

Builds the app for production to the `build` folder.
It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.
Your app is ready to be deployed!

See the section about deployment for more information.

Learn More

`npm run eject`

Note: this is a one-way operation. Once you `eject`, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

You can learn more in the Create React App documentation.

To learn React, check out the React documentation.

Code Splitting

This section has moved here: https://facebook.github.io/create-react-app/docs/code-splitting

Analyzing the Bundle Size

This section has moved here: https://facebook.github.io/create-react-app/docs/analyzing-the-bundle-size

Making a Progressive Web App

This section has moved here: https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

Advanced Configuration

This section has moved here: https://facebook.github.io/create-react-app/docs/advanced-configuration

Deployment

This section has moved here: https://facebook.github.io/create-react-app/docs/deployment

`npm run build` fails to minify

This section has moved here: https://facebook.github.io/create-react-app/docs/troubleshooting#npm-run-build-fails-to-minify