

Initial Thoughts

The assignment allows selection between 3 options; jQuery Mobile, Angular and React. The assignment is relatively lightweight and simple. All three options are more than capable of completing the assignment (a conclusion supported by the option to use any of the three).

Meta Analysis

jQuery Mobile is the most familiar, being used in the first assignment and more lessons and workshops.

Angular was the choice for previous years of the module, suggesting a particular suitability, but is no longer the default choice.

React is a new focus for the module, implying that the DSA see advantages in focussing on React.

jQuery Mobile's greater familiarity would make it easier to produce a better app, and more practice with any given framework is more useful than a little bit of practice in many.

However, in a structured and assessed learning environment, proficiency in and exposure to more frameworks is a positive. Once the basics are learnt in a guided way, development and improvement is easier to conduct on a personal level.

On a Meta-Analysis level, attempting to use the newer React library or Angular Framework seems to offer the better learning opportunities. For a production App, it might be better to go with the more developed jQuery skillset, but in this case, learning is the target, not a production App. Angular suffers a little from being neither the newest focus, or the most familiar.

Selection Model

YouTube channel techsith have a comparison of JavaScript libraries/frameworks that breaks the consideration down into elements titled: Popularity, Job Market, Learning Curve, Documentation and Future. Even when consulting other resources in investigating libraries/frameworks, this model acts as a good metric to evaluate the candidates. The evaluation will vary from techsith's as the job market in Silicon Valley differs considerably from the South East of Ireland.

Evaluation

The Meta-Analysis made React an early favourite.

The candidates were evaluated against the above selection model. React proves to be a more popular option for GitHub hosted projects, and is more popular than JQM or Angular when searching Job Sites for opportunities in Waterford.

However, React is noted as having a steeper learning curve than Angular, especially if one is used to Angular 1 and updating to Angular 5. Both are rated as having good Documentation and Future prospects due to the support of large industry players.

After dismissing jQuery Mobile in favour of learning something new, this leaves a roughly even split between Angular and React (1 meta and 2 metric "points" for React versus 3 metric points for

Angular). React was ultimately chosen due to potential confusion between Angular versions in online resources, and the apparent performance advantages offered by ReactDOM, as well as React's focus on JSX and breaking applications down into loosely coupled components combining logic and markup for each concern.

In a roughly even breakdown of two new libraries/frameworks, React offered the more interesting features to develop with experience, as well as being closer to the Object Oriented Java/C++ paradigm the author is more comfortable and experienced with.

Conclusion

React offers an opportunity to learn a popular, new, well supported and well documented library for App development. It is in demand in the workplace, and once mastered offers interesting and powerful features in a, in the authors opinion, clean and well-structured design philosophy.

A relatively small design scope and limited time to explore Reacts features limit the scope of what is possible in the assignment. However, with clear documentation and tutorials online the project is clearly feasible. An unusual feature was an expressed preference for inline style elements. Further investigation will be required to define what exactly "preferred" means in this context, although styling through separate css files resulted in some unexpected results. Although previous web development experience suggests styling is best done in separate files, and create-react-app generates separate css files, using inline styling does appear to follow the paradigm of separating by concern rather than technology. More experimentation will be required before making a snap judgement on future approaches.

Expanding the Application into a more useful live application will require more experience and familiarity with React and some experience in combining with the likes of phonegap. Based on initial experiences and research into React, no significant hurdles appear likely.