CSC301
Assignment 1 Report
Pair 39
Chantal Sorias, Ya-Tzu Wang

Repo for Mobile:
https://github.com/csc301-fall-2020/assignment-1-39-ytwwww-chantalsorias-mobile

Repo for Web:
https://github.com/orgs/csc301-fall-2020/teams/39-ytwwww-chantalsorias-web

Note: Instructions for running each app is located in their respective readme files. A video for testing the mobile app is in the repo for mobile.

Contents

Web

Summary

We compared React, Angular, Vue.js for frontend. For the backend, Node, Python, and Go are considered. In terms of CI/CD tools, we considered GitHub Actions and CircleCI. To deploy the web app, Netlify and Heroku were strong candidates. We ended up using the MERN stack (MongoDB, Express, React, Node) for the web app.

Frontend

We consider three popular frontend frameworks for our web app: React, Angular, and Vue.js. Angular is developed by Google and is the largest library of the three with the most support for handling and validating form as well as http client. Vue is open-source and developed by its community. Vue is viewed as the easiest to learn out of the three. React offers the least functionalities of the three but excel at rendering performance and making light-weight applications. We decided to go with React because it is easy to make and reuse components.

In order to make this app responsive to different screen sizes for phone, tablet, and desktop, we chose one of React UI frameworks, Material UI’s Grid and Appbar for layout.

Continuous Integration & Continuous Deployment

For continuous integration and continuous deployment, we consider both GitHub Actions and CircleCI. GitHub Actions is easy to set up since the app is stored on GitHub already. GitHub Actions also provide clear and comprehensive documentation as well as example templates to guide first-time users using this tool. CircleCI is another popular tool for CI/CD used by many companies, it offers better UI with more detailed reports, statistics, and options than GitHub Actions. Due to the above-mentioned advantages, I chose CircleCI as the tool for CI/CD.

For deployment of the website, we try to set up both Netlify and Heroku at first and find Heroku a breeze to set up and use. One advantage of Netlify is that its app never goes to sleep, meaning it will not take a long time to load after some time of inactivity. Although apps hosted on Heroku will go to sleep after some time of inactivity and will take a bit more time to load after such inactivity, Heroku is more flexible and offers great support for backend and database. Additionally, Heroku works well with git. It takes only a few commands to deploy the app for the first time.

After the initial setup, it takes only one command of “git push Heroku master” to deploy any new changes.

Testing

There are many potential testing libraries available such as Jest and React Testing Library. Both are powerful tools for testing JavaScript code while React Testing Library has some advantages for testing React components. React Testing Library comes with create-react-app and allows testers to render a specific component easily and check whether something such as a message shows up correctly. Testing this using Jest is more complex and requires more code. Since ease of testing various React components is important for our web app, we chose React Testing Library over Jest.

Mobile

Summary

We compared React Native, Xamarin, and Flutter for frontend development. And compared NodeJs, Python, and Golang for backend development. Our solution for the mobile application was a React Native frontend and NodeJs backend.

Frontend

The mobile application we built is compatible on both Android and iOS. We thought it was important to have it work on both platforms so more people can have access to it. This is why we wanted to work with a cross-platform app development framework so we can write once and deploy to both. Then came the decision of which framework to use. The most popular options are React Native, Xamarin, and Flutter.

React Native is free and reliable because it is developed and supported by Facebook. It uses Javascript and it implements native UI components which allows apps to look like native apps. It offers a vast library of components which allows for faster development time. It has hot reload so developers can apply changes right away. A lot of very popular apps have been created with React Native such as Facebook, Instagram, Uber Eats, Skype, Pinterest, Discord, and more. A developer does not need to know native code in order to use it. Some disadvantages include navigation which is not seamless, it struggles to build complex animations, and issues rendering large datasets as it is dependent on JavaScript bridge.

Xamarin has good performance. It also has a complete development ecosystem; all you need are C#, .Net framework and Microsoft Visual Studio with Xamarin to build an app. It allows developers to reuse around 96% of the code to create iOS and Android apps. Some apps built using Xamarin are Skulls of the Shogun, SuperGiant Games, Storyo, and more. However, it is only free for small teams. And this free version has a limited feature set and limited resources. It has heavy graphics and building complex applications will require developers to write in native code. It also has large app sizes due to translating C# into native, making them larger than native apps.

Flutter is reliable as it is developed by Google. It is customizable because you can create your own widgets or customize pre-existing ones. It also has hot reload to allow developers to fix bugs faster. However, it has large app sizes as well and it does not expose several native APIs. It uses the language Dart which would require us to learn a new language in order to develop. Some apps built using Flutter are Google AdWords, Google Greentea, Alibaba and more.

After doing research, we discovered that all three cross-platform frameworks are good to work with. We decided to go with React Native because it uses JavaScript, which we already know, and is a free open-source platform. It also allows the app to look like native apps without having to learn native code. Out of the three, it is the most popular and has the best community support. Lastly, because it is very popular, it is easy to find online tutorials to help us learn how to use it.

Continuous Integration & Continuous Deployment

For CI/CD, GitHub Actions resources were easier to find and follow. You can also create any number of YAML files in the the directory while you can only have one for CircleCI. Using GitHub Actions, we created two actions, one for running tests after every push and another for deploying the mobile application to Expo if pushed to the master branch (these are found in the github/workflows directory). Then the spending limit was reached and then we decided to use CircleCi. Tutorials were not as easy to find, however, which required us to do more research. We then created jobs for testing on every push and deploying to Expo for dev and prod. Unfortunately, we could not test deployment as no more credits were available.

Testing

Jest is popular for testing React Native applications. Facebook, who created React Native, uses Jest to test, therefore it must be reliable. Jest is a JavaScript testing framework designed to ensure correctness of any JavaScript codebase. A snapshot test, enabled by Jest, is a test which first takes a textual “snapshot” representation of a component. If the component changes later on then it will result in failing the test. We created one such test along with unit tests for testing individual functions. These tests can be found in the __tests__ directory.

Both Web and Mobile

Backend and Database

For the backend, we decided to compare Nodejs, Python, and Go. Nodejs uses JavaScript, so once you are comfortable with Javascript, then it is easy to learn. Picking React Native for frontend, we can write both client-side and server-side using JavaScript. The libraries are managed by NPM (Node Package Manager) which has one of the biggest repositories and it is easy to use. NodeJs can do both frontend and backend. It offers scalability as it is built into the runtime environment. It cannot maintain CPU-intensive tasks. It is said that it is not the best option for large development projects.

Python has a big advantage because its syntax allows developers to write fewer lines of code. It is very popular in data science and machine learning. It is a backend and frontend programming language. Packages and libraries are handled by pip. Python also has a large community. It lacks proper scalability as the runtime interpretation of Python code makes it slower and it also does not support multithreading. It is also not the best for mobile computing.

For Golang (Go), the new programming language has a great combination of the performance and security benefits of C/C++ and speed of Python. It is good for its garbage collection handling, memory safety, and dynamic interfaces. It has high raw performance. Go is more functional and works with parallel threads. On the other hand, developers are required to learn the language which has fewer resources online. We would have to do a lot of research in order to get a grasp on it. It is also only used for backend development. Another disadvantage is that you must implement explicit error checking. The compile-time and run-time errors are handled differently than the norm which raises problems for developers.

We decided to go with Nodejs because it is faster compared to Python, although similar real-life performance compared to Go. NPM for Nodejs has far more packages than pip for Python. We are also comfortable with the common throw-catch approach that Nodejs offers compared to Go's error handling mechanism. Another reason we chose Nodejs is because we are comfortable with JavaScript while we would have to learn a new language for Go. It is also very popular and has a large community. It is easy to find tutorials online.

Another reason why we chose Nodejs is because the MERN (MongoDB, Express, React, Node) stack is very popular. MongoDB is a document database, Express is a Node web framework, React is a client-side JavaScript framework, and Node is the premier JavaScript web server. Finding resources to construct this stack is easy to find. For the mobile development case we substituted React Native, instead of React. For this reason, MongoDB was used as a database. MongoDB is easy to set up and has a cloud service called MongoDB Atlas that works well with our web app deployed using Heroku.