CRSchedules

By Cormac Raftery G00348802

Abstract

This dissertation is based on the project 'CRSchedules', which was completed as part of the module 'Applied Project and Minor Dissertation' by Cormac Raftery. This project is a dynamic web-application that connects employers with employees. It is a 3-tier application, using MongoDB for the Data Tier, Node js as the Logic Tier and Angular 9 for the Presentation Tier. These technologies are very commonly known as the MEAN stack and have been very widely used for the past decade. An objective of this project was to gain an understanding of the complexities behind the development of a MEAN stack dynamic web application. An agile approach was taken to develop a system that uses a MongoDB database that can be consumed by an Angular web application. Throughout this investigation I found that a nodemon server combined with Angular 9 and MongoDB function very well together to create a full stack project.

This project has been developed fully by Cormac Raftery with ID number G00348802. The project source code may be found on the Github Repository:

https://github.com/CormacRaftery/4thYearProject

Project Objectives

1. Gain a better understanding of full stack development.

2. Create a system capable of hosting a scheduling website for employers.

3. Create a user-friendly interface.

4. Integrate a feature to allow an employer to add/edit/delete employees.

5. Authentication that's simple to use but secure.

6. Allow employees to easily view their schedule.

7. Allow employers only to edit the schedule.

8. Develop skills working with agile methodology.

Methodology

This project was developed using the agile methodology. Before each feature was added there was a planning and design phase, the feature was then tested before moving to the next feature.

Database Design

Once the base idea for the project was selected; a database could be designed. It was looked at what information would need to be saved in the database that could potentially apply to an employee.

Once this simple database wireframe was set up, it was possible to write queries against the mock data to ensure that the database server was working as expected and that the data could be interacted with.

With a back-end wireframe up and running, and that it could perform CRUD operations on very simple data using all of the required technologies, work could begin on a single page angular web-application. This involved creating methods that could perform the HTTP GET and HTTP POST transactions with the data in the database.

Now that it was verified that a front to back end wireframe was in place and working correcting it was felt that sufficient research and experimentation had been conducted and now sprints to begin building on this wireframe could begin.

Hosting and Web Navigation

The website was hosted on an AWS EC2 Virtual Machine. The reason for this is that it allows specific port allocation for forwarding and incoming requests. Connection to the front-end application was done using port 4200, with access to the back-end and database done on 27017. The web navigation was done by providing a route to each component in 'member.routes.js'. This allowed routing to each component of the project by including a route in the URL.

Front End Configuration

To begin front-end development, the view of the employer was taken which included the list employee and the create employee pages which included CRUD operations on each employee.

Front End and Back End Connectivity

With the initial skeleton of the web application in place and tested, it was time to get started connecting the web app, to the RESTful service backend.

This meant it was time to think about the behind the scenes architecture of the web application. It was decided that the server should be hosted with nodemon then and accessed with the mongod daemon.

When this was all working, it was possible to then do a MEAN stack data transfer test by querying some data on the database from the web app and displaying it in the browser console.

Styling

Bootstrap and HTML were used heavily to achieve the attractive, simple user interface.

Testing

A sample test cases were run on the project using the built-in testing technologies in Angular, Karma and Jasmine. These revealed that if used as intended the application works fine, however if incorrect values are used for certain inputs the page will throw an error. For example, the age parameter must be an integer and if a string of characters are entered the user will not be submitted. As the user is not submitted it does not cause any harm to the database.

Technology Review

This section is comprised of an in-depth view of all the technologies used throughout the project.

Data Tier

MongoDB

MongoDB is a cross-platform document-oriented database that provides high performance and is easily scalable. MongoDB is perfect for applications that involve working with very large amounts of data, or where the database needs to be easily scalable. One of the main issues with relational databases (such as MySQL), is that it becomes complicated beyond practicality when the database needs to be scaled. This is because many relational databases were designed to run on a single server as opposed to MongoDB which has been built to scale horizontally. This is a major bottleneck in the world of computing in the age of nearly ginormous data. Big data projects naturally require the data layer to have the ability to scale up and scale out as per demands. This is why NoSQL solutions really seem to be on the rise. MongoDB is one of the most popular and easy to use NoSQL databases. The main reason that MongoDB was chosen as the database technology for this project is because it was believed that MongoDB has and will become increasingly popular and influential in the coming years. Another reason MongoDB was chosen, is because of the fact that prior to the development of this project, the developer had little experience with MongoDB, and thought that it would be crucial to be able to use this technology with ease when working with industry projects.

Afterthoughts

MongoDB proved to be very easy to get set up with and use. The syntax used to read and write MongoDB is called BSON. BSON is also known as “binary JSON” and is very human readable (Looks very similar to JSON syntax). MongoDB worked well with this project, as all it involved was having a different collection for each of the different categories of data in the application. Another reason it fit really well with this project, is because of the fact that this project utilized JSON syntax for transferring the data over HTTP. This meant that the data looked almost the same in transition as it did in the database, making manual inspection of the web service much easier.

Logic Tier

Nodejs

Node js is a platform built on Google Chrome’s Javascript runtime for easy building fast and scalable network applications. Node js uses an event-driven, non-blocking I/0 model that makes it lightweight, efficient and perfect for data-intesive real-time web applications.

Reasoning

The developer wished to gain more knowledge using the MEAN stack and so Nodejs is an integral part of that stack. Using Nodejs made the application scalable and reliable.

Mongoose

Mongoose is an Object Data Modeling library for MongoDB and Node.js. Mongoose manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB. Mongoose is designed to work in an asynchronous environment and drastically reduces the amount of ‘boilerplate’ coding. Mongoose

provides a straight-forward, schema-based solution to modeling your application data and includes built-in type casting, validation and query building.

Reasoning

Mongoose was developed specifically for communicating with MongoDB and Node.js and as such I thought it best to include it to link the two technologies together.

Presentation Tier

Angular 9

Angular is one of the most used platforms for building web applications. Introduced by Google in 2012, Angular is often described as “Superheroic” amongst JavaScript frameworks. The angular framework consists of individual components. An example of a component might be a Create Client page. The component would have a TypeScript file, a HTML file, and a CSS/SCSS file. This completely decouples an application’s logic from DOM manipulation.

Angular is being constantly maintained by Google and is changing on a yearly basis with the first release coming in 2009 being dubbed AngularJS. Since the original release the “JS” was dropped to avoid confusion among developers as Angular2 was released in 2016, followed by Angular4 by the end of 2016. Angular5 followed in 2017 with Angular6 and Angular7 coming along in 2018. Angular8 was released in May 2019 followed shortly by Angular9 in February 2020. This is a technology that you can count on advancing continuously for the foreseeable future.

Reasoning

The reasoning behind choosing Angular for the development of this project was simple; It was the most popular technology in industry that the developer previously did not have much experience with. Angular also uses tools that the developer was already comfortable with, for example HTML and TypeScript. The developer also had experience using JavaScript, which TypeScript is a syntactical super-set of. It also became known to the team, prior to development, that Angular provides tools to help with testing the application such as Jasmine and Protractor. There is also a large community of active angular developers. This meant that, should any problems arise while using Angular, there would be an abundance of resources to find more information about such problems.

Afterthoughts

Angular was an extremely intuitive piece of technology. The decoupling of logic and view elements of the application made it useful to create web applications with. Any situations that arose could be searched online and there was an abundance of developers online with similar problems trying to make the switch from Ruby or other languages. The project was originally started using Angular8 but after encountering a multitude of errors such as CORS was switched to Angular9 which solved a lot of problems.

Bootstrap

Bootstrap is an open-source framework (originally created by Twitter) that one can use as a basis for creating web sites or web applications. Bootstrap contains a combination of JavaScript, HTML, and CSS designed to help build user-interface components for responsive web applications. The reasoning

behind using Bootstrap in this project was the fact that it is extremely easy to use and there is an abundance of documentation and resources available for it. Bootstrap goes hand-in-hand with Angular, so it made sense to use it.

Afterthoughts

As expected, Bootstrap was a very easy technology to integrate with an Angular front-end application. The documentation was very explanatory and easy for anyone, regardless of whether they would be considered a beginner, intermediate or advanced in software development, to follow. The developer strongly recommends including Bootstrap into any Angular-based project.

Languages

Typescript

TypeScript is an open-source programming language developed and maintained by Microsoft. TypeScript was originally created to bridge the gap between JavaScript and the growing demand for JavaScript to grow into a fully-fledged server-side technology. JavaScript's failure to embrace the features of "Object-Orientation", strong type checking and compile-time error checks prevented JavaScript from succeeding at enterprise level. TypeScript is a strongly typed, object-oriented, compiled language. TypeScript is the syntactical super-set of JavaScript, meaning all JavaScript syntax can also be considered TypeScript. TypeScript gets compiled to JavaScript. The main reason TypeScript was used for this project is because it works very nicely with Angular. Typescript has also been growing rapidly in popularity over the past number of years.

Development tools

Git/Github

Git is an open-source version control system that was started by the same man who created Linux; Linus Torvalds. Git is considered, by many, to be the most preferred version control system on the market. A Git "Repository" is simply a directory on a machine that is being tracked by Git. Any changes made to any of the files in this directory can be "added" and then "committed" to the repository. The users have control over what files get added, committed or even tracked by Git within that repository. GitHub is a web-based hosting service for version control using Git. Users of Git can create remote versions of their Git repository and "push" changes made in their local repository to the remote repository on the Github servers. Github has multiple useful features including a nice graphical user interface, in contrast to the Git command-line tool. Git is also one of the most popular version control systems in industry. Git is used in nearly every module in the software development course and it is obvious why.

Angular CLI

Angular Command Line Interface is a command line tool for creating and updating angular apps. Manually creating components and services in angular requires a lot of in-depth code which is automatically generated through the use of Angular CLI. When creating component files. Angular CLI also generates the necessary testing files using Jasmine, Karma and Protractor.

Karma/Jasmine

When creating an Angular application using the Angular-CLI tool, Jasmine and Karma are already resolved and configured for the app. Jasmine is the framework used to create the tests. Karma is a task

runner for the Jasmine tests. It uses a configuration file in order to set the startup file, the reporters, the testing framework and the browser among other things. The default browser used by Karma is Chrome.

NPM

Node Packet Manager is a software registry that contains over 350,000 code packages. NPM is open-source and free to use, where developers download all NPM public software packages without any registration or login. NPM is the default packet manager for the JavaScript run-time environment 'Node.js'. NPM is very commonly used to handle any dependencies involved in creating Angular applications. The reason NPM was used as a packet manager in this project, was because NPM is a crucial tool when building Angular applications. Any Angular dependencies that needed to be installed for the application were handled by NPM on the command line interface.

Development environment

Visual Studio Code

Visual Studio Code is a very popular IDE developed by Microsoft. Visual Studio Code provides support for debugging, embedded Git control, syntax highlighting, intelligent code completion, and more. Visual Studio Code was selected as the source code editor for the project.

AWS

The dynamic website application was hosted on AWS EC2 Virtual Machine. An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. Amazon provides a variety of types of instances with different configurations of CPU, memory, storage, and networking resources to suit user needs. Each type is also available in two different sizes to address workload requirements. The AWS Platform was chosen for its scalability, flexibility, and reliability.

Conclusion

To conclude, the developer was relatively content with the outcome of the project. While the project did not reach all its original goals, it still provided basic functionality for a scheduling system. The developer believes that they have failed massively in the security aspect and would be interested expanding their knowledge in web app security. The developer also believes that working in a team would prove much more beneficial in terms of learning how to communicate and use methodologies more efficiently. The developer believes that they succeeded in reaching the following goals:

1. Gain a better understanding of full stack development.

2. Create a system capable of hosting a scheduling website for employers.

3. Create a user-friendly interface.

4. Integrate a feature to allow an employer to add/edit/delete employees.

5. Allow employees to easily view their schedule.

6. Allow employers only to edit the schedule.

With the above in mind, it can be concluded that most of the initial objectives for this project were met in some capacity or another.

Learning Outcomes

The developer senses that experience with MEAN stack development will help them in future industry projects, in particular working with full stack development projects. Prior to this project, the developer had a basic knowledge of the individual components of full stack development. This knowledge has been refined as the team got more practical exposure to the complexities behind creating a full stack application. The developer had the opportunity to improve their skills in multiple domains of development, such as database management, server configuration, client-side development and system architecture. This experience also exposed the gaps in the developer's knowledge such as system security. Going forward, the developer intends to pursue a more in-depth knowledge about this topic. All aspects of this final year project have challenged the developer in terms of research, development, and self-discipline.

Appendix

Source code link to github - https://github.com/CormacRaftery/4thYearProject

Link to video demonstrating the project - https://www.youtube.com/watch?v=GNMCsNTCyoY&feature=youtu.be

Bibliography

https://samirbehara.com/2018/01/16/the-rise-of-nosql-databases/

https://nodejs.org/en/about/

https://mongoosejs.com/

https://devopedia.org/typescript

https://www.codeinwp.com/blog/angular-vs-vue-vs-react/

https://getbootstrap.com/docs/4.4/about/overview/

https://www.typescriptlang.org/docs/handbook/declaration-files/deep-dive.html

https://codecraft.tv/courses/angular/unit-testing/jasmine-and-karma/

https://docs.npmjs.com/about-npm/

https://www.otava.com/reference/aws-vs-azure-key-differences/

https://www.sciencedirect.com/topics/computer-science/database-design

https://en.wikipedia.org/wiki/Angular_(web_framework)

https://zenkit.com/en/blog/agile-methodology-an-overview/