My approach to the Application:

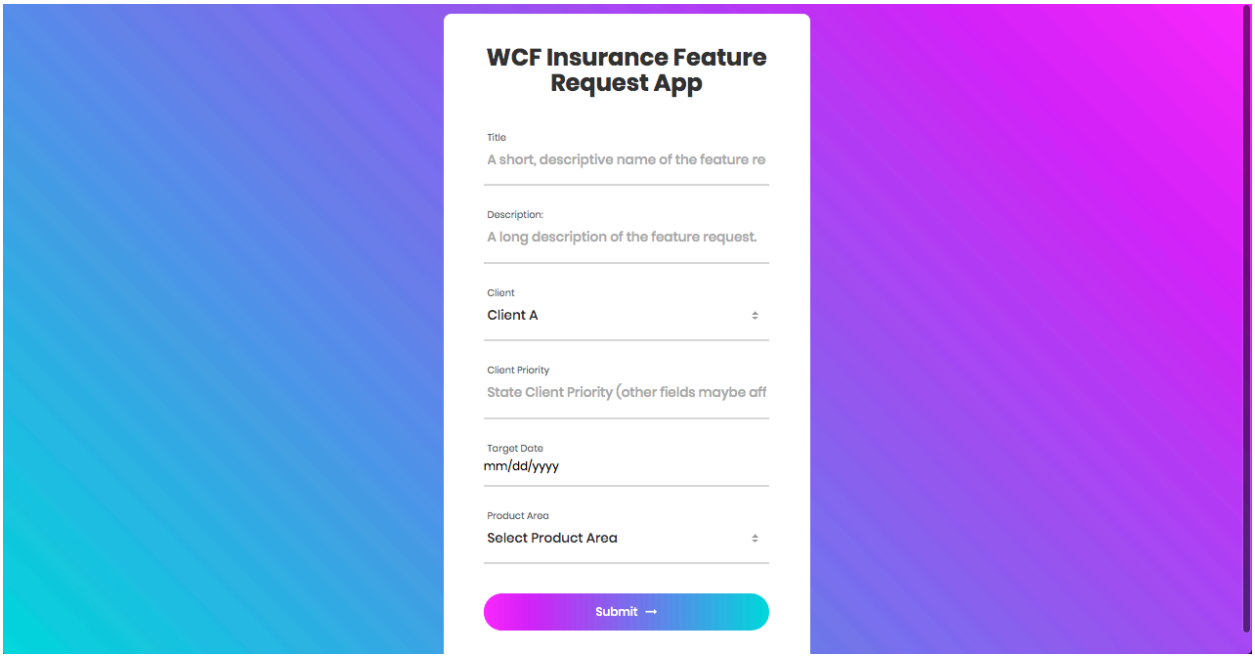
I built the API and the interaction layer using JavaScript. I extensively used NPM and specific packages such as Express, Mongoose, Express-Handlebars, Body-Parser and Nodemon. These are open source packages and tools that allowed the different segments to communicate, especially with the requests.

For the database, I used MondoDB which is another open source platform that allowed me to store requests and the data received from the form regarding the Feature Request App.

Finally, on the front end, I debated between using ReactJS or going with a HTML,CSS and Bootstrap model. In order to maximize my prior experience and producing the fastest solution, I chose to build the front end in HTML.

How I handled some particulars:

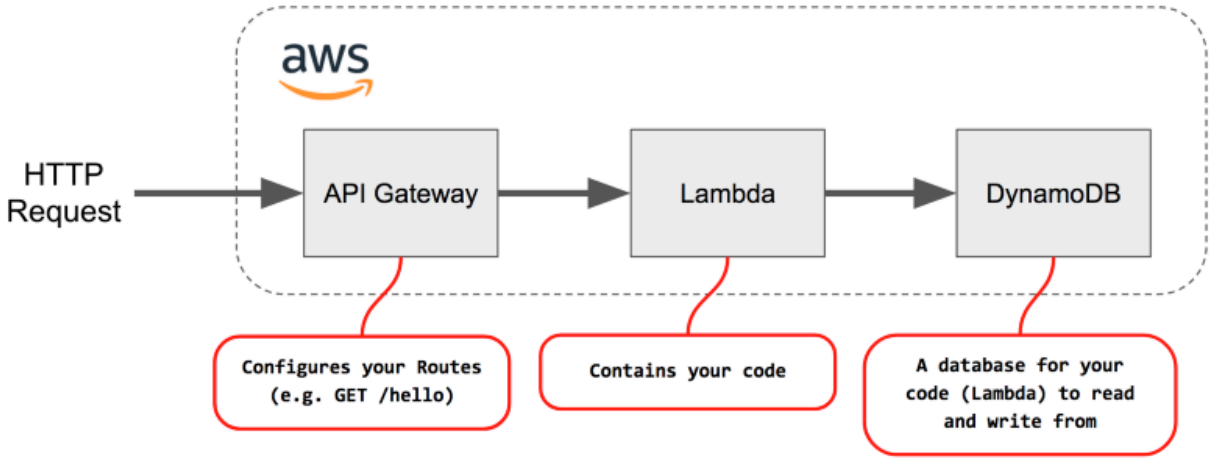
1. I used a JQuery function to ensure Client Priority was not repeated for the client and ensure that all the other fields were aligned alongside it.
2. Regarding the date, I used a small JS function to create a dropdown calendar option.
3. I used dropdown menus for Client and Product Area



Demo of the Application on my localhost

HOW I WOULD HOST THE APPLICATION:

Right now the app is running on a localhost on my machine and has a database in MongoDB I would take the following steps if I wanted to load it unto AWS:



Running it on a third party server, such as Azure or AWS would make it server-less

Steps to host it on AWS:

1. Clone the repository with all the files necessary
2. Evaluate the stack.yml file
3. Create and upload the CloudFormation Template to create stack and upload the stack.yml file.
4. Verify the Lambda that was created by the CloudFormation Template
5. Deploy the code on the Lamba previously stated; create and add S# support
6. Create and find the API Gateway
7. Test API Connection
8. Deploy API gateway

TESTING THE APPLICATION:

My plans for testing the Application:

On way to do this would be through Katalon Studio: it is a free automation testing tool that I previously used and it supports both SOAP and REST requests and is a reliable way to test applications. JMeter is also an alternative that can be also used for automated testing.

Since the app was built using Javascript, I would also consider testing it using Mocha, a Node.js framework for testing applications.

The test cases I would consider would be using different variations of the data involved in the form, leaving the boxes blank or putting unacceptable data based on the parameters selected. That would test the validation tests I already set up in the form. I would also ensure that the requests are sent and stored successfully on the database(MongoDb right now) and test issuing error messages. One key aspect I would definitely test would be the Client Priority aspect as it influences other fields as well.