CSC301 A1

Sang Won Kim

1003063622

1.

Introduction

I've chosen to do a webapp due to previous experience and to prepare for the group project webapp. For the web-app I've went with the standard html/CSS/JavaScript application.

Frontend

Out of the three popular choices for JavaScript frameworks, I've went with React mainly due to its popularity and wide range of additional resources (such as NextJS) I might benefit from down the line. I've had previous experience with VUE and Angular2, but using Angular wasn't intuitive for me, and VUE, while easy to learn, became more and more unmanageable working with a team as the project became bigger. I've also used NextJS, a React framework, for its built-in server-side rendering for React.

Backend

Since the project was very small in scale, I opted in for the serverless backend solution provided by mongoDB-Realm/Atlas (backend/database). I initially wanted to do Express/NodeJS solution, since it's a fast and flexible, but I realized I didn't need any special configurations on APIs for just a simple checkout calculator. As mentioned before, I've used NextJS, which also meant many of the features (server rendering) of NodeJS solution wasn't needed. MongoDB is a non-relational database, which I thought was appropriate since there wasn't any complex queries needed.

Deployment

I initially wanted to do Heroku, since it allowed automatic updates to the deployment using git. I then found mongoDB-Realm, which did the same thing, but had the other benefit of being more closely integrated with mongoDB database. However, both options had trouble configuring with NextJS React apps (its server interaction features got in the way in weird ways). So, I opted for Vercel, which allowed me to use NextJS with almost no configurations, and it also allowed automatic updates through git.

2.

The deployed app can be found here: https://assignment-1-31-samkim10.vercel.app/

Instructions on how to use app

1. Click the link above to access. May take a couple of seconds to load up the data.
2. You will see 3 sections. From top to bottom, the first section is the 'Item List', where it displays items (from database) as boxes that can be put into 'your Cart'. You can click the individual boxes to add the item.
3. The second section, 'your cart', has all the items you have added so far. The added items are displayed as boxes of that item. You can click them to remove the item from 'your cart'.
4. The third section simply shows the total price of all the items in 'your cart'. The value will change depending on the items in 'your cart'.

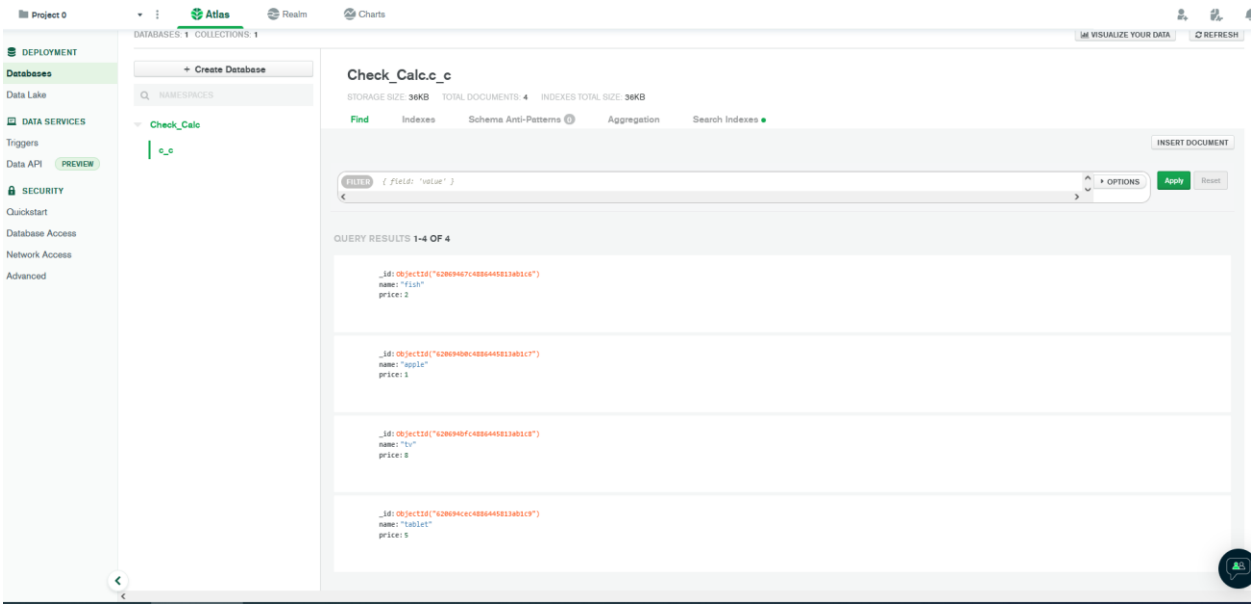
Checking my code

If need to check my code, you should look at two places:

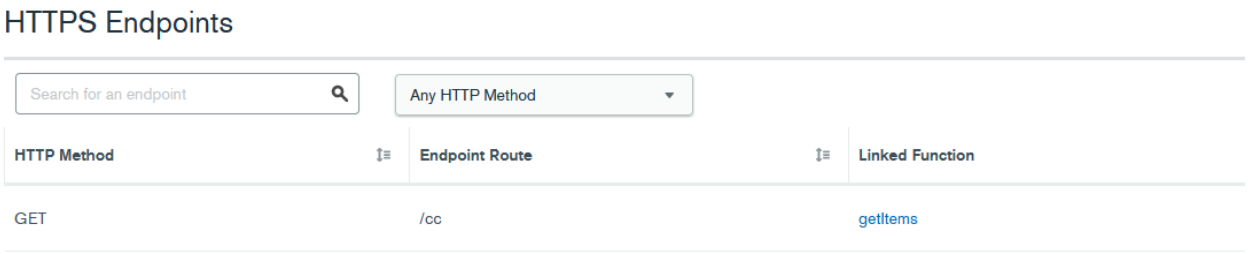
1. Calculator.js : has all the Front end code.
2. pages/index.js: interacts with the database.

Serverless Solution

I can provide with an invitation (requires email) to the serverless project solution on MongoDB-Realm if needed.



MongoDB Cluster containing the database I've used



Endpoint (example below) on MongoDB-Realm (Serverless). I can add any HTTP methods from here.

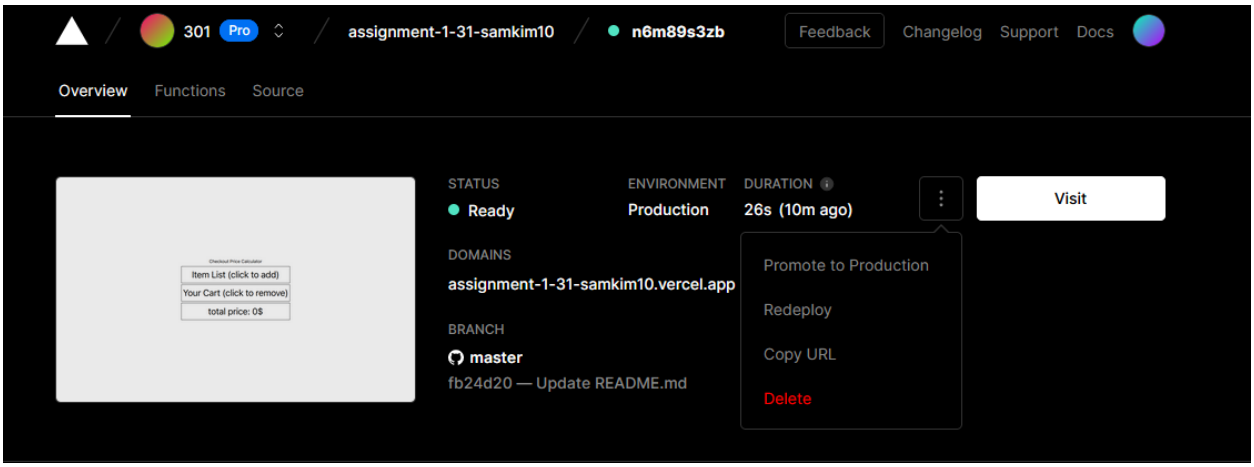
Example get call

Example endpoint testing

https://us-east-1.aws.data.mongodb-api.com/app/checkout_calculator-qawwr/endpoint/cc

Use the above endpoint to do a 'Get' call. It will return an array of all the items you can buy in the checkout calculator. (I've personally used postman to test).

Deployment method



Vercel page, where you can see 'Branch' linking to my master branch on github. Any Push to the github repo will automatically rebuild and redeploy the app.