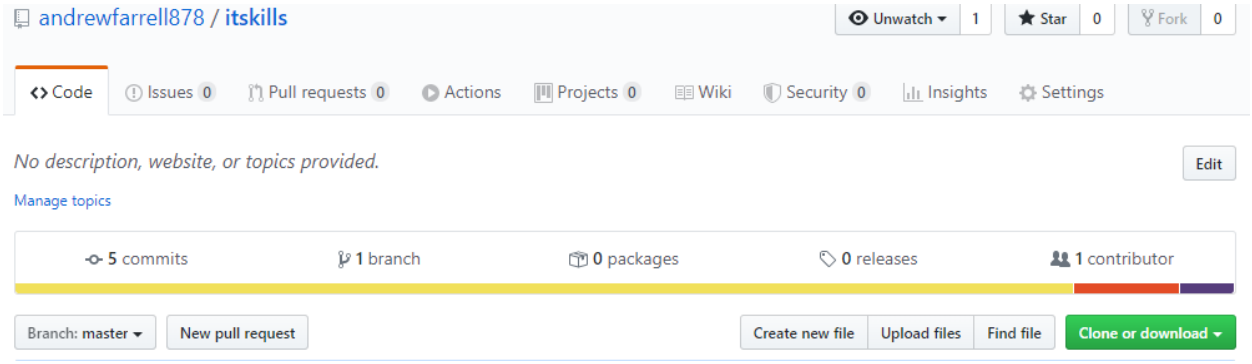
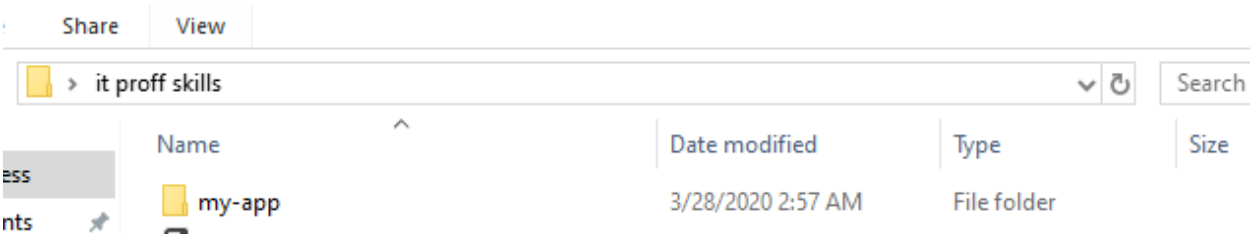
# Professional Practice In IT Andrew Farrell G00360243

This is a document showing my project in Professional Practice in It on how it works and how to run it. You need npm node installed, visual studio and git installed to run my app.

Firstly to run my app git clone my repository from my github




Then copy the link, create a new folder on your desktop and type cmd into that folder




Then type git clone followed by my repository link

```
C:\Users\andre\Pictures\testing>git clone https://github.com/andrewfarrell878/itskills.git
Cloning into 'itskills'...
remote: Enumerating objects: 70, done.
remote: Counting objects: 100% (70/70), done.
remote: Compressing objects: 100% (49/49), done.
Unpacking objects:  92% (65/70), 385.20 KiB | 42.00 KiB/s
```

Then cd into itskills. To view code you must have visual studio code installed then type code . You can view my code from here

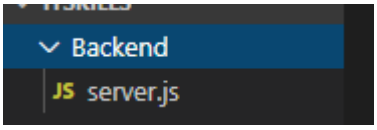
```
C:\Users\andre\Pictures\testing>cd itskills

C:\Users\andre\Pictures\testing\itskills>code .

C:\Users\andre\Pictures\testing\itskills>
```

To run my app type npm start

```
C:\Users\andre\Pictures\testing\itskills>npm start
```

This will load the main component of my app on the localhost

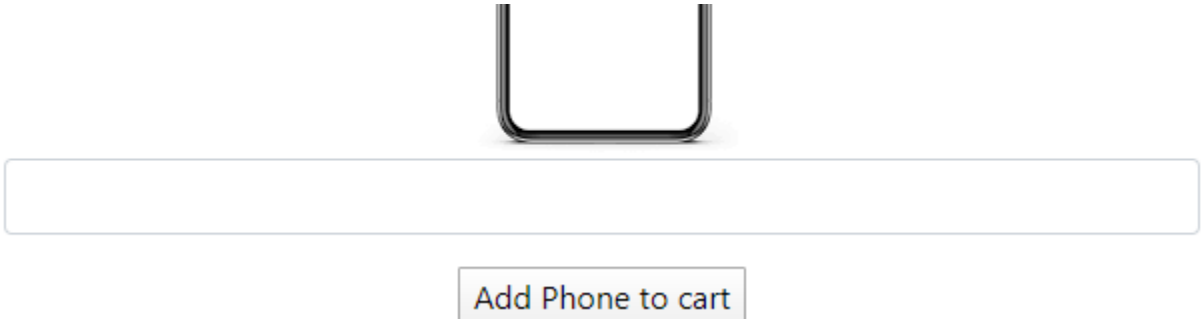
To run the server side of this we must navigate to the Backend folder and then go into server.js




Mongoose must be installed in server.js. Right click on server.js and click open in terminal. We use this terminal to install mongoDB into the server by typing install mongoose. Once mongodb is installed We run this by then typing node server.js this prompt should occur.

```
PS C:\Users\andre\Desktop\it proff skills\my-app\Backend> node server.js
Example app listening on port 4000!
(node:18956) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
```

We can now type in desired phone and storage type




We can check when data is uploaded when we get this prompt in the terminal

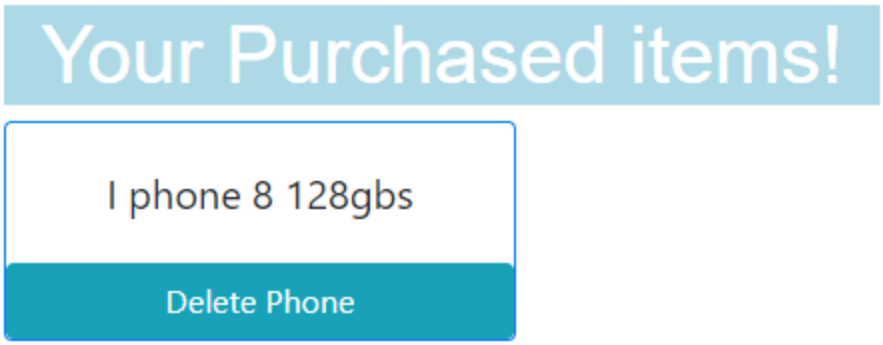
```
5e9f1ee1d60d854d5ce38306
Post request Successful
i phone 128gbs
```

This will show what you typed in.

This text should appear in cart page where you can delete it or keep it.

This app or website is a basic phone shop where the user can add phone to the cart by typing in desired model and phone type in the form underneath the picture. The text where the user entered should appear in the cart folder. I used monogoDB database as the backend as I was familiar with is from mobile applications last semester. I discuss this in the screencast I submitted


This should then be uploaded to the cart your desired phone




For my project requirements I wanted a functional website or app that would allow a user to browse a number of phones similar to how an actual website would work and purchase a item by typing a phone model and a storage size as seen above to view it in the cart page of my app.

For the methodologies I used I used react and mongodb as server or the backend. I used these two methodologies as I was familiar with these two technologies as I used these before. I wanted to go with something simple as I was doing this project by my self and I thought react and mongodb where great assets to use.

I had many limitations when trying to design or develop my app , originally I wanted to upload a local image to the cart as I first intended. But there where a few problems I occurred with this. I first thought I had this component working as when I had one variable for example my first variable was called Iphone that was being called into the server it would upload the image of the Iphone to the cart in the on form submit method which was working as intended , but when I added a second variable and a second image being a picture of a Samsung phone it would cause problems. I would create a new form on submit method and make a new variable being passed into the server, but when  a user would click an just an iphone the second image of the Samsung would appear too which was not intended to happen. Two images where being uploaded to the cart instead of one , with one button click and I couldn't find I way to fix this so I just used  a form to submit data to the cart using user input.

I had many conclusions with this project some being limitations other being problems with creating my react app. As when I tried creating a new react app it would give me a whole host of errors I hadn't seen before as it took me many weeks to find out what the problem was. I later found out that it was newer version of react that was conflicting with my laptop. I came to the conclusion that downgrading to the older version would be more beneficial.  I also came to the conclusion that uploading local images was not working so I just used user input in form.