ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 9: Cloud Computing		
Submission date	December 27, 2020	**Date Received 1st submission**	December 27, 2020
Re-submission Date	January 2, 2020	**Date Received 2nd submission**	January 2, 2020
Student Name	Bạch Tuấn Anh	**Student ID**	GCH18768
Class		**Assessor name**	
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P5	P6	P7	P8	M3	M4	D2	D3

❒ **Summative Feedback:**	❒ **Resubmission Feedback:**	
Grade:	**Assessor Signature:**	**Date:**
Internal Verifier's Comments:		
Signature & Date:		

Table of Contents

Table Of Figure

A. Introduction

Below is this report that will address a specific issue. In this report, I will come up with a specific scenario and do that using the Node.js web platform - namely the agile framework, then I'll cover the features included in the chapter. To send. Also, I'll use MongoDB for data storage, and eventually, I'll use git and Heroku services to deploy that web to the internet.

B. Design

I. Overview Function

1. Solution

Through technical analysis and evaluation, we find that the application of the cloud computing model is completely suitable for the company. The implementation will be detailed in this presentation.

2. Use case diagram

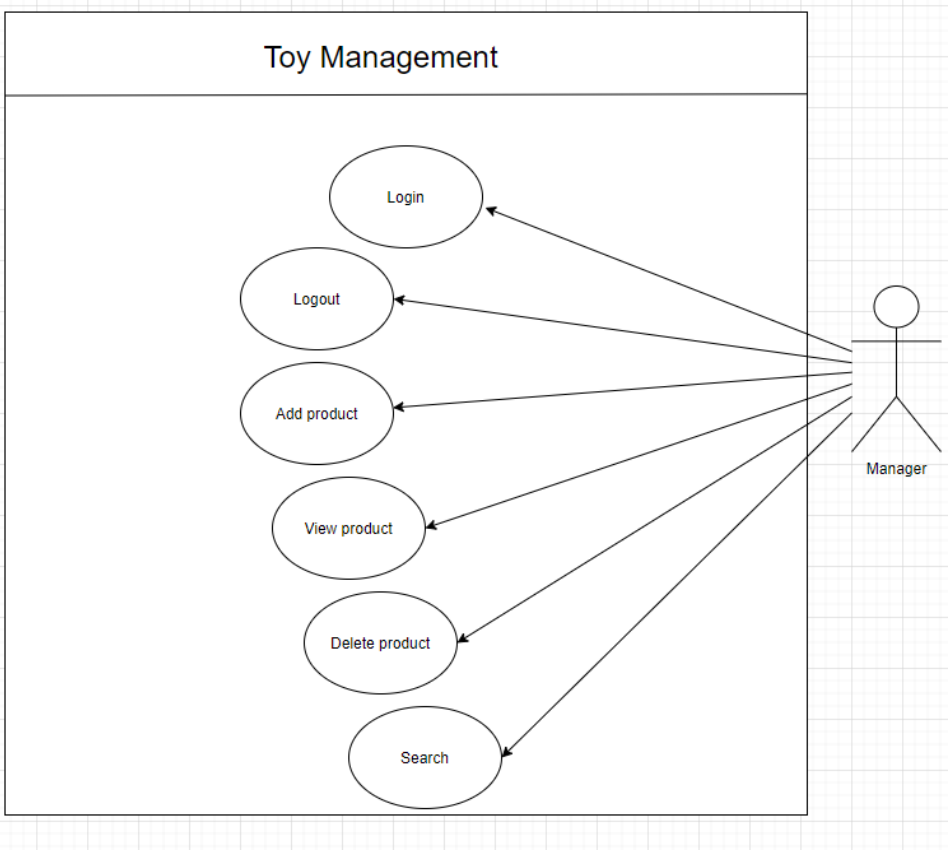


Figure 1. Use case diagram

Through the use case diagram we see the book management program includes 5 main functions:

- First you can login and logout
- Add books, this is one of the main functions that book managers have, they can add book information
- View all book
- Search book, the manager can search for books
- Delete book of the book manager
- Update book, the manager can update books continuously

3. Website screenshots

3.1. Login page

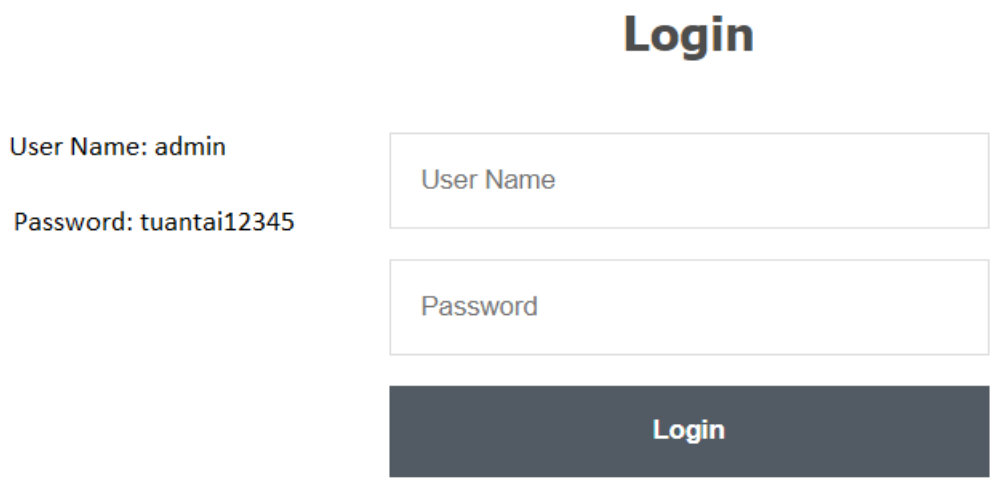


Figure 2. Login page

First, we need to log in to view the product and only the administrator can access because only the administrator has the account and password to login.

3.2. View page

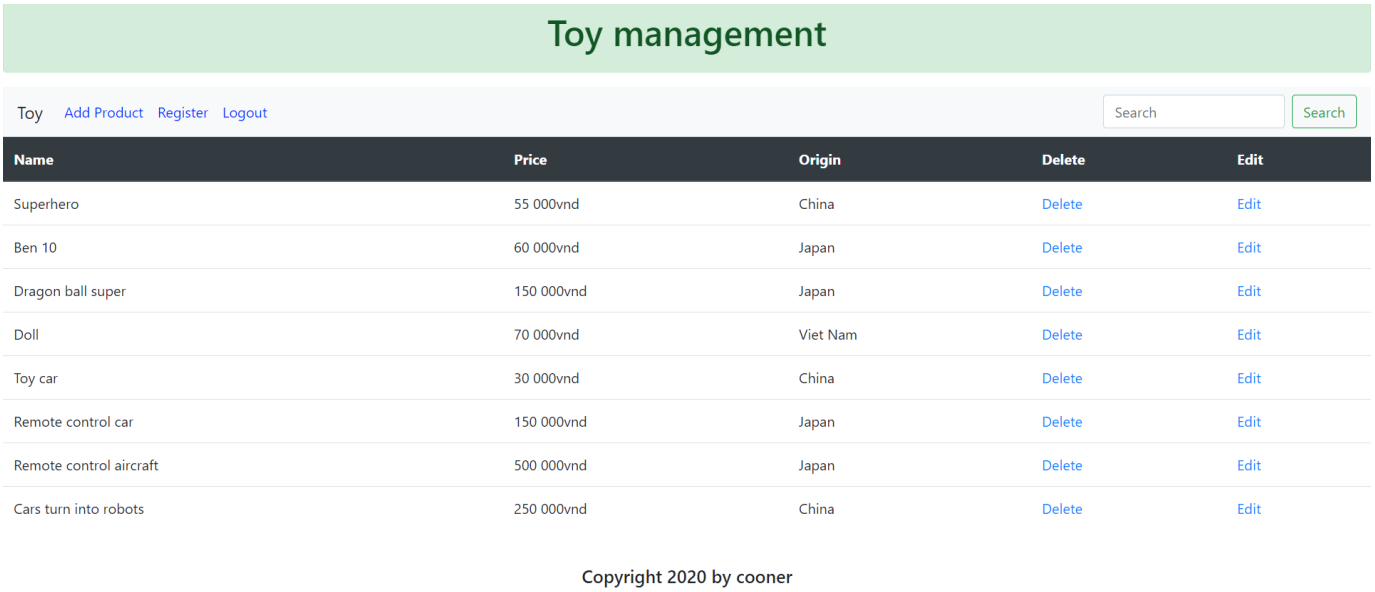


Figure 3. View page

After the manager logs in, they will see the interface of the entire product under management, users can also sign up for additional accounts if they want.

3.3. Add page

After filling the information, the manager clicks the add button, the system will add the product information to the database and redirect the page to the view page.

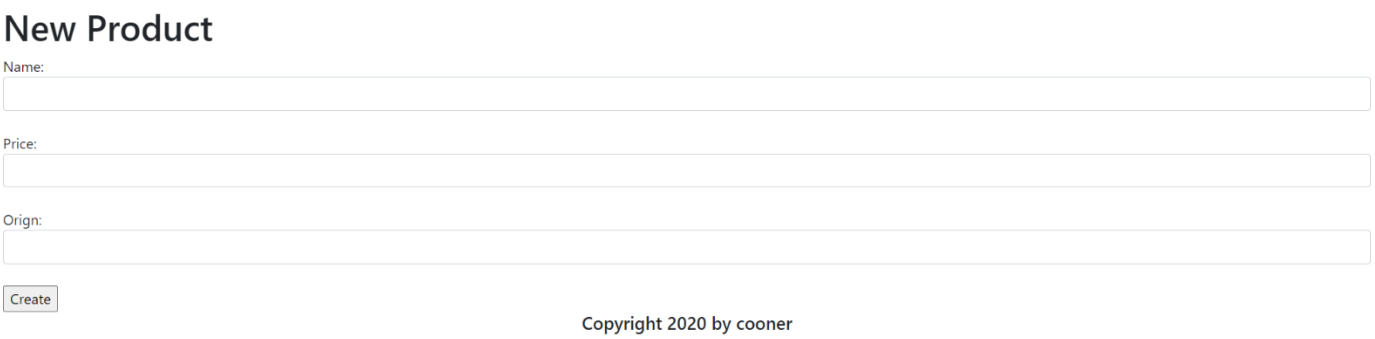


Figure 4. Add page

3.4. Update page

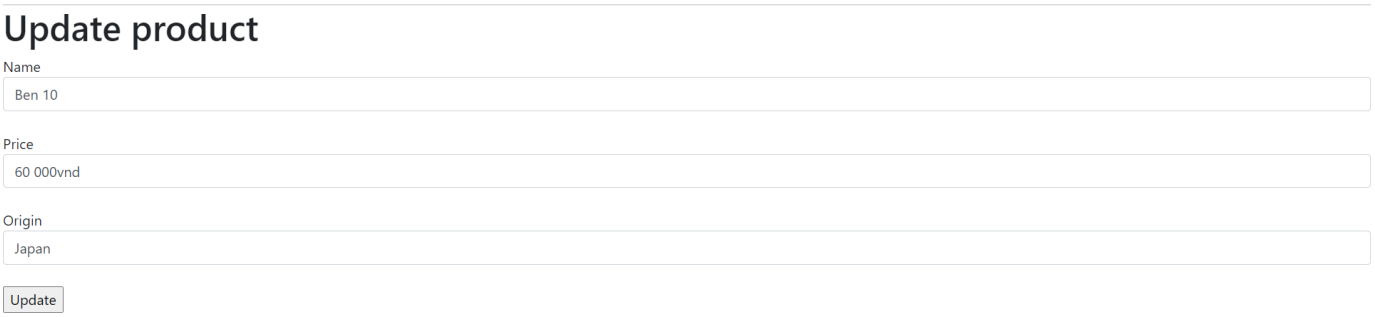


Figure 5. Update page

After clicking the update button of a product, all information on that product will be displayed on the page update. Here the manager will update the bill's information and click the update button to save the changes.

3.5. Delete product

When the manager clicks delete button to delete any product, the system will display a message to manager confirm. If the user clicks OK, the system will delete the product and redirect to the page view, if the manager wants to cancel, then click cancel.

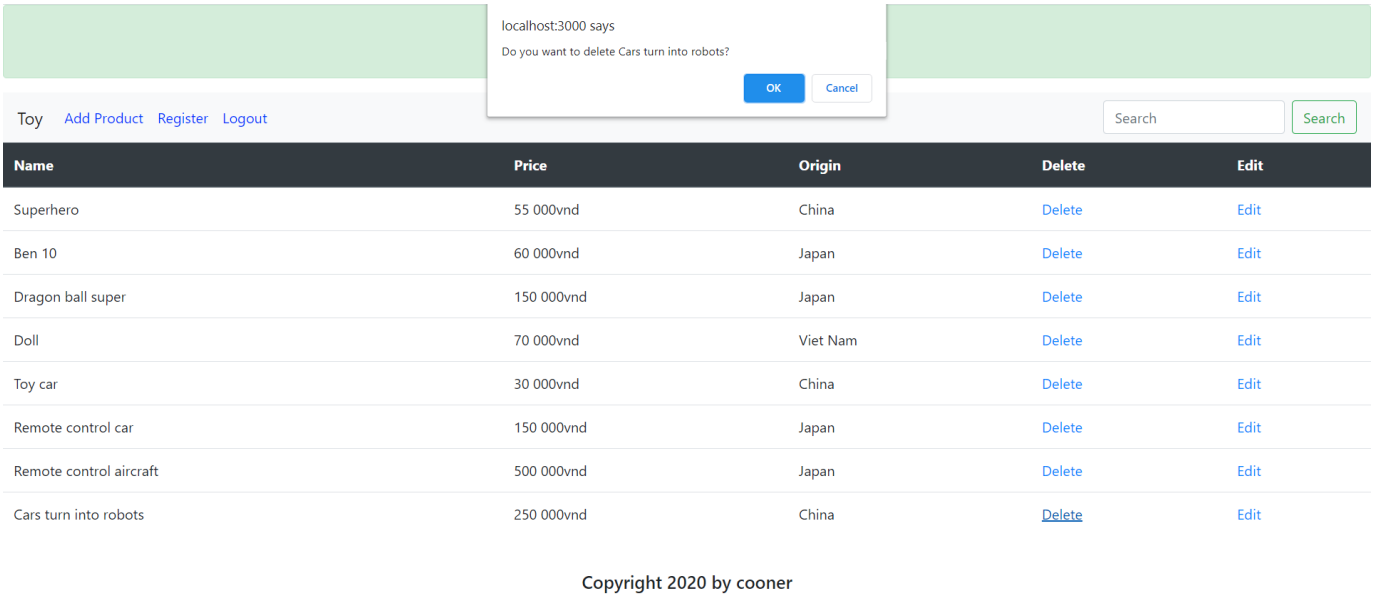


Figure 6. Delete product

II. Code implement and deploy process

1. Source code manager

Use Github to manage source code because:

(BRADFORD, 2020)

- GitHub provides developers and researchers with a dynamic and collaborative environment, commonly known as a social coding platform, that supports peer reviews, comments, and discussions.
- GitHub is a very versatile tool to use. It's ideal for working on projects of any size, and it's a great tool for web workflows.
- Essentially, you can use it to publish your work, as a version control system and as a collaboration tool.
- It is ideal for creating a backup of your work instead of saving the code on your desktop or on a corporate server.

Use to manage source code. By committing, pulling, pushing source code changes from one repository. Developers can easily manage project versions.

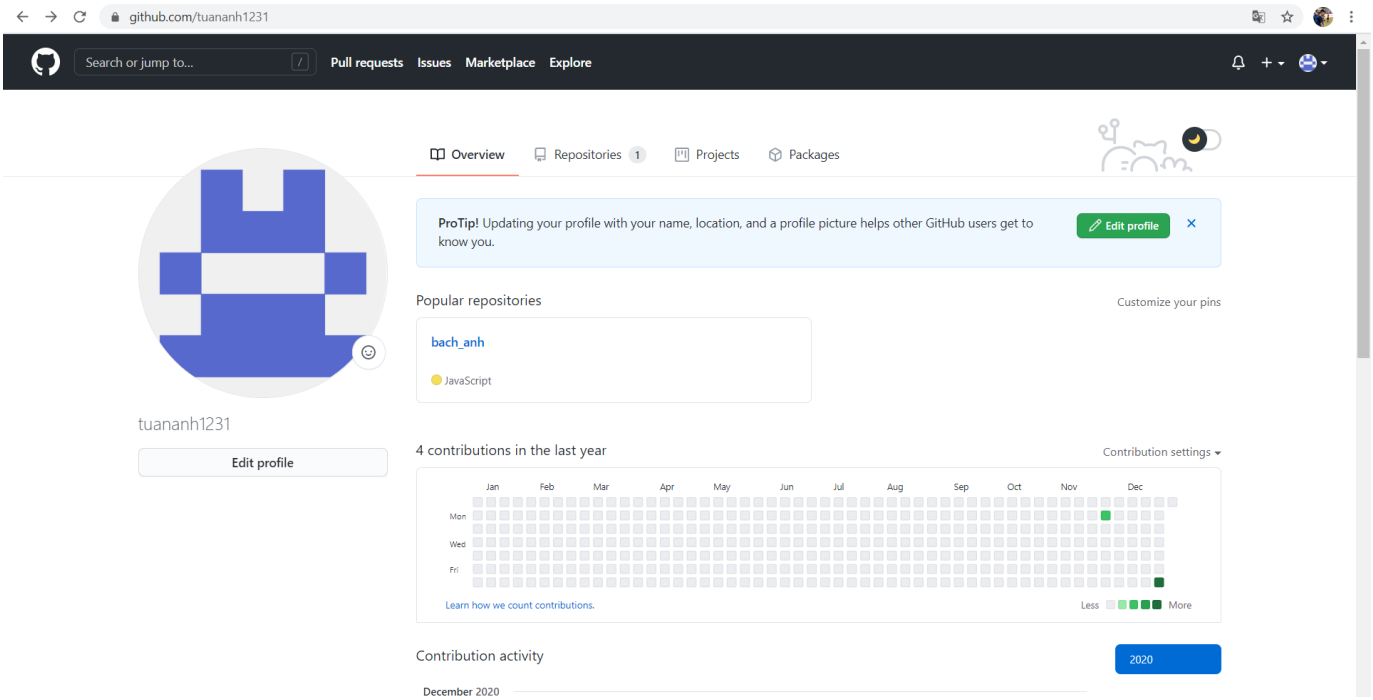


Figure 7. Github

2. Database server

Use MongDB because:

(Anon., 2020)

- Document-driven archiving - The data is stored as JSON-style documents.
- Index on any property
- Replication and high availability
- Automatic protection
- Rich queries
- Quick update in place
- Professional support by MongoDB

In this picture, I use the MongoDB atlas to store data on the internet. Thereby, I can access this database anywhere that is permitted.

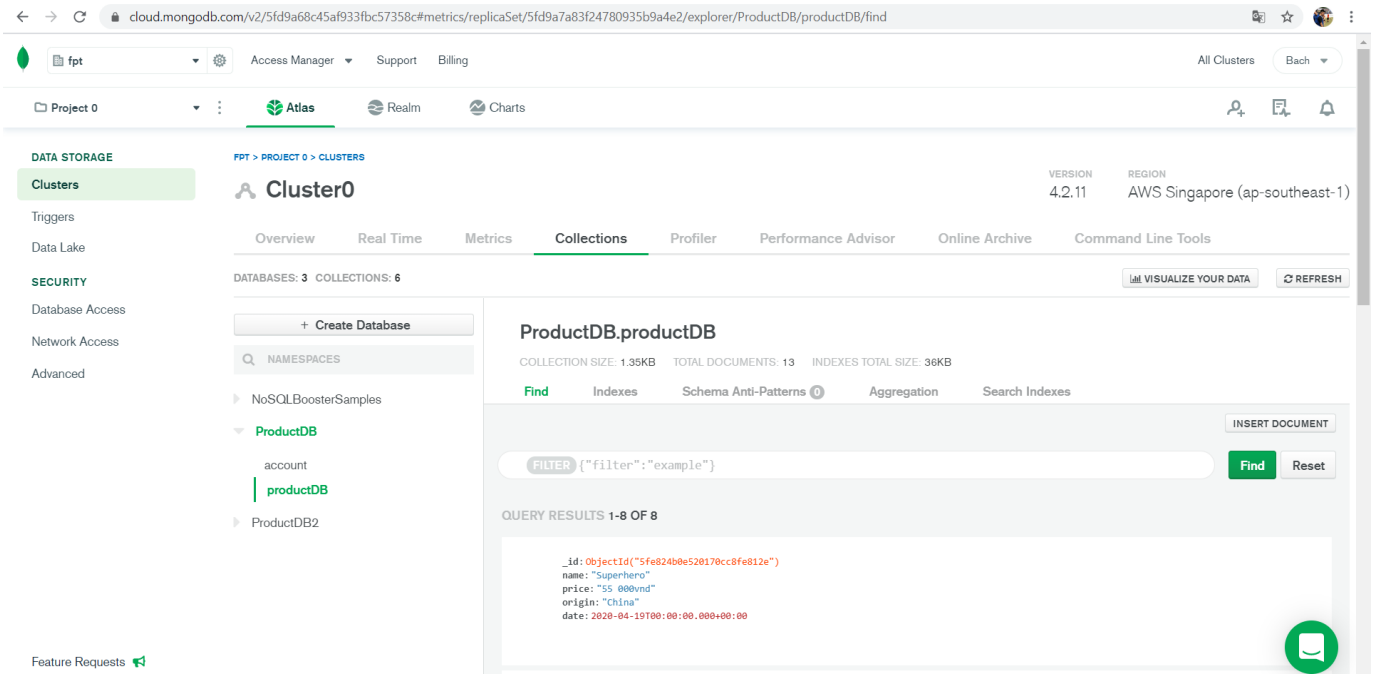


Figure 8. MongoDB atlas

3. Cloud computing module

Finally, deploy the web application on the internet. I use the heroku cloud server because it allows us, as a relatively small team, to offer competitive hosting plans powered by the largest, most secure cloud platforms and most stable in the world. And it allows our customers to sleep well at night knowing that our team of infrastructure experts is on hand to keep a close eye on everything.

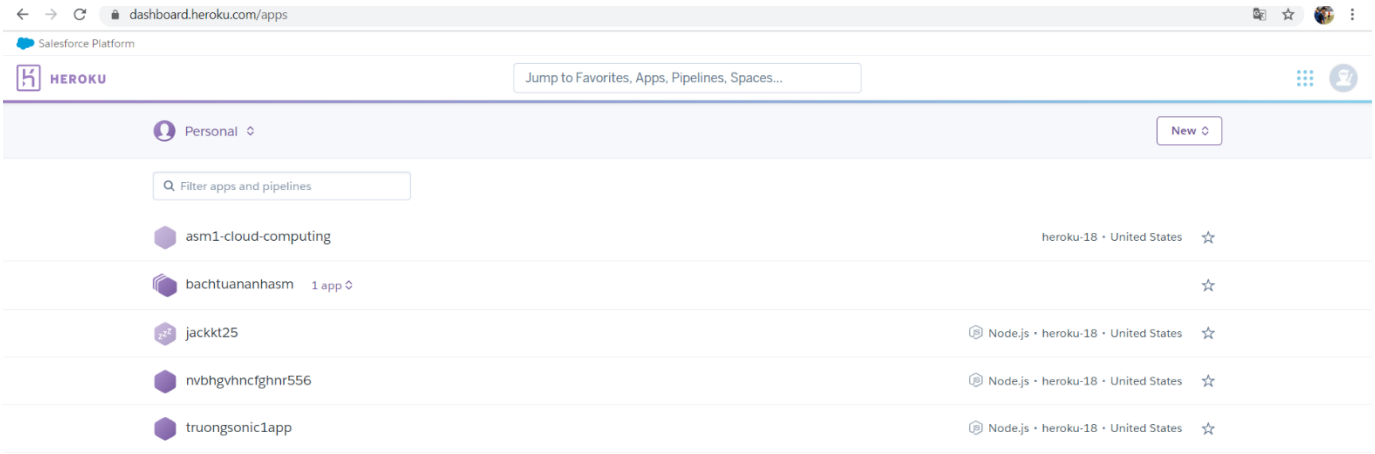


Figure 9. Heroku

4. Deploy

a. Config framework express on the env

To use this framework, firstly I need to download node.js from the website.

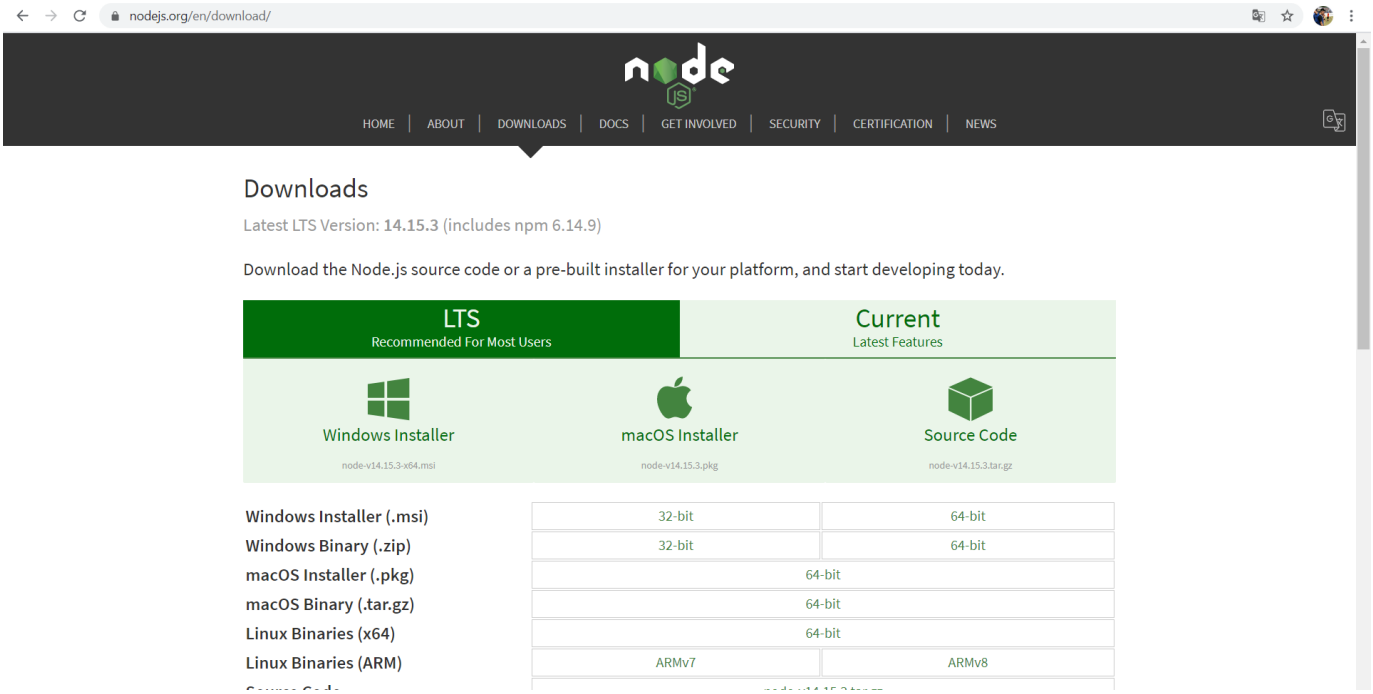


Figure 11. Download node js

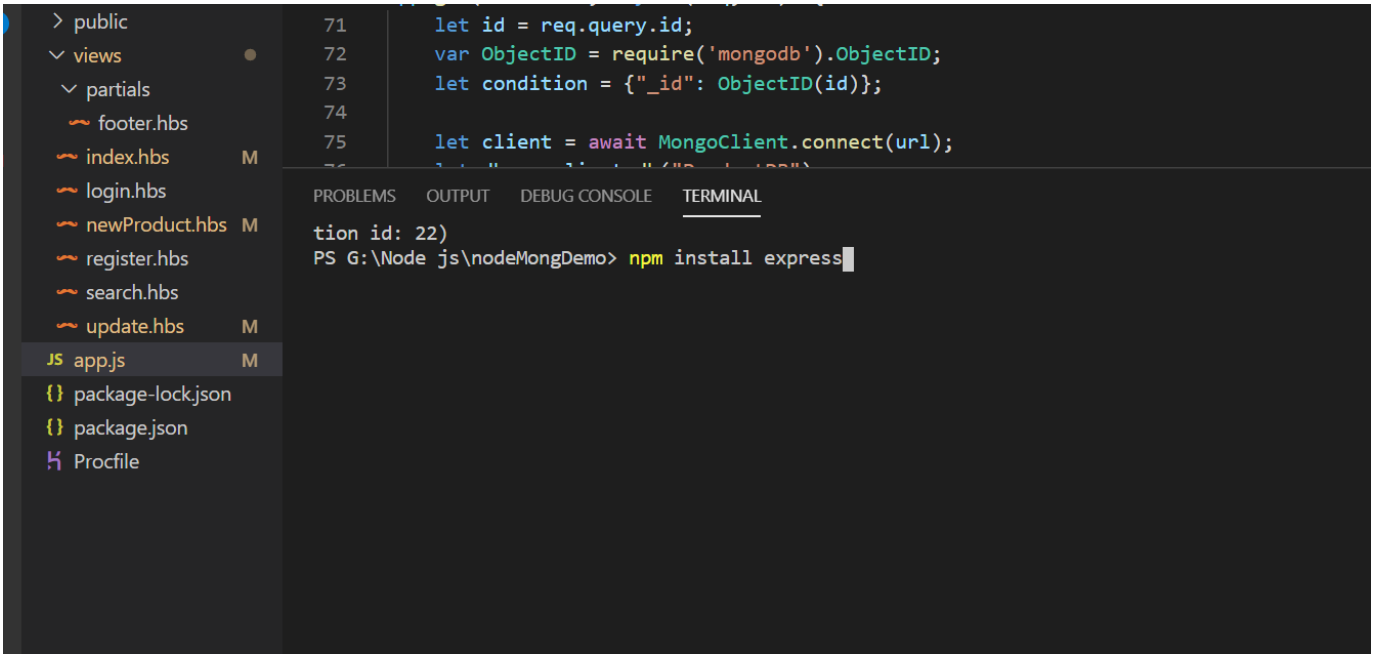


Figure 10. Confix express

After downloading and installing node.js and the desktop configuration environment, create a new project in visual studio and open the terminal. To use the express framework, we need to install it from the Node package manager using the command: npm install express. So the instance framework is ready to be used in this project.

b. Config and connect with MongoDB

Next, I will connect to MongoDB to complete the login, add, update, and delete functions of the web application.

After successfully registering an account on MongoDB. Access to the MongoDB atlas.

- **Step 1:** Click to name the newly created data storage (Cluster0)
- **Step 2**: Click connect
- **Step 3**: Select "Connect using MongoDB Compass"

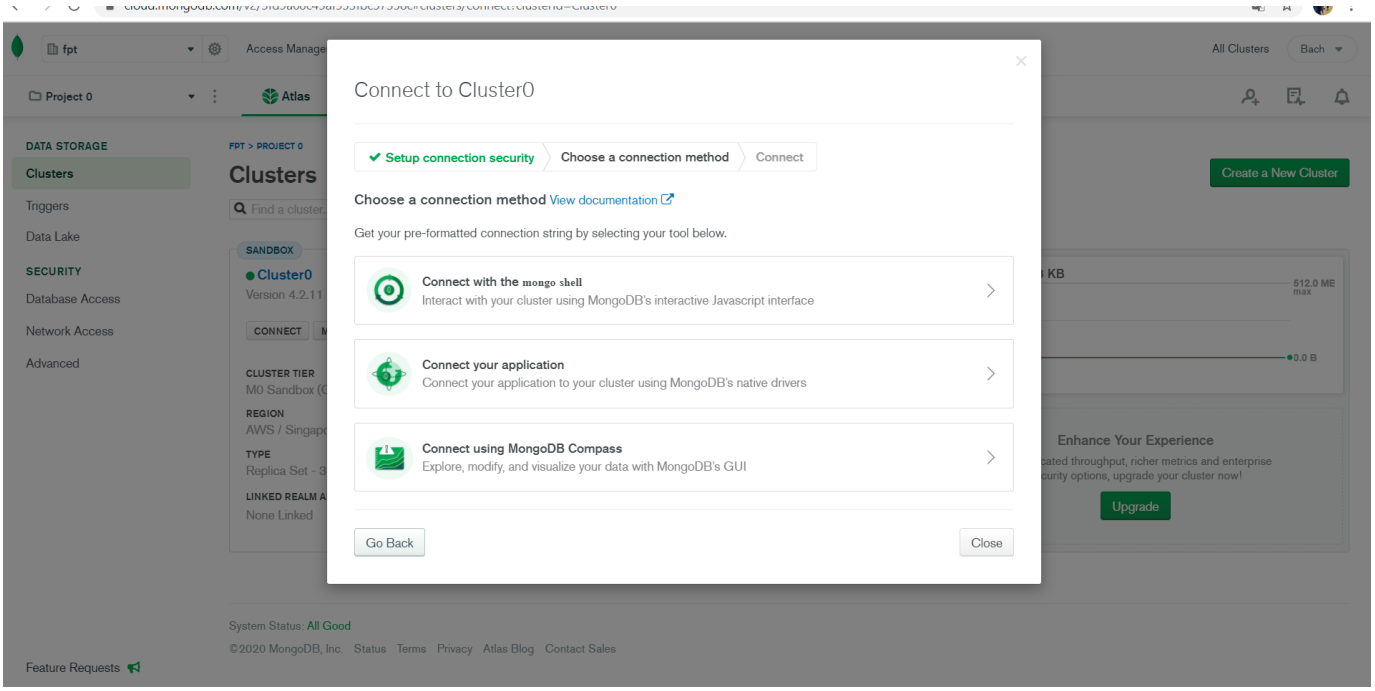


Figure 12. Connect MongoDB

- **Step 4**: Click copy to copy the connection string

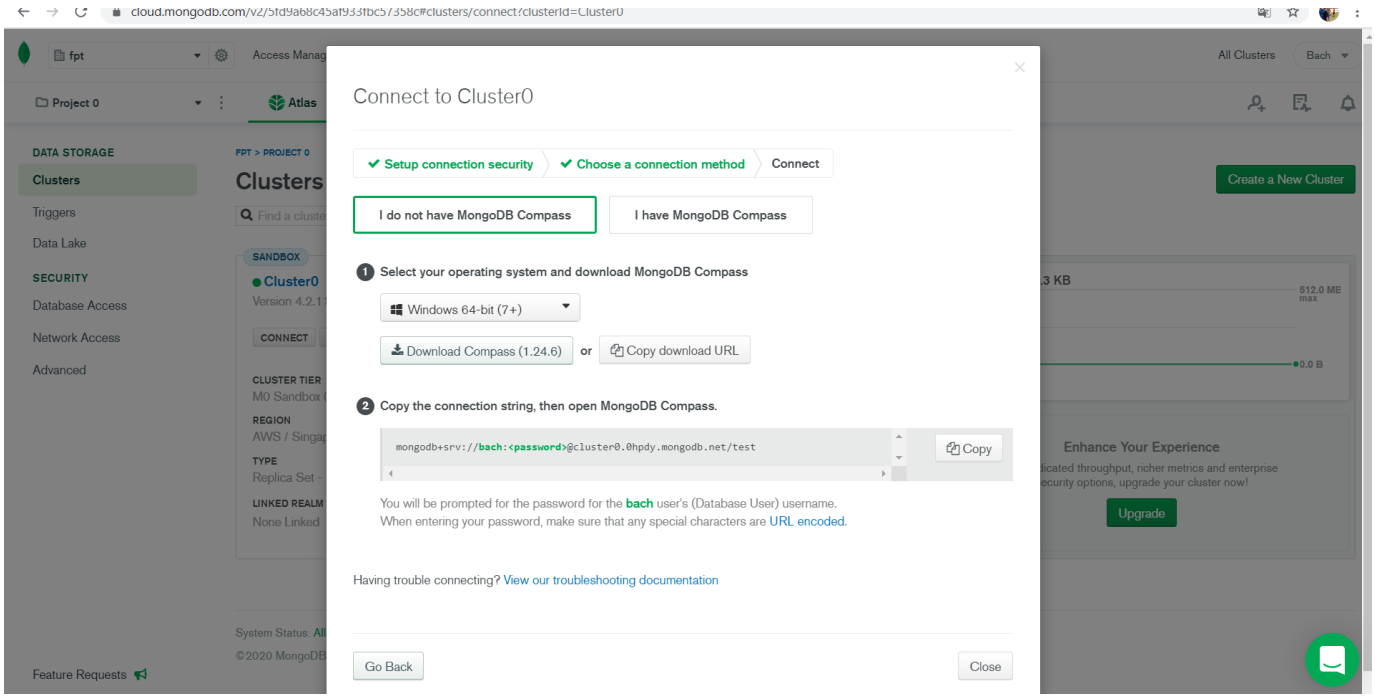


Figure 13. Connect MongoDB

- **Step 5:** Open nosqlbooster to mongodb on computer and paste the connection string in the connection of this MongoDB Compass. After that, changing username and password that you register to use this database.

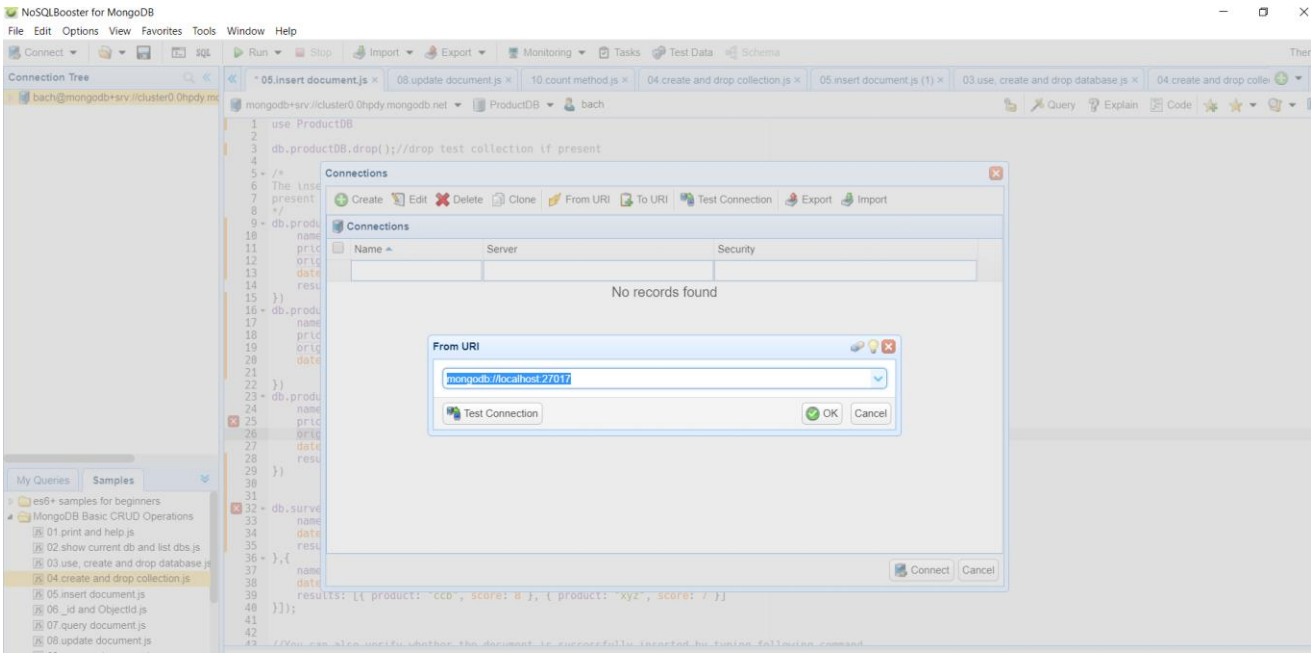


Figure 14. Connect mongoDB

Step 6: Click CREATE DATABASE to create database and collections for this project.

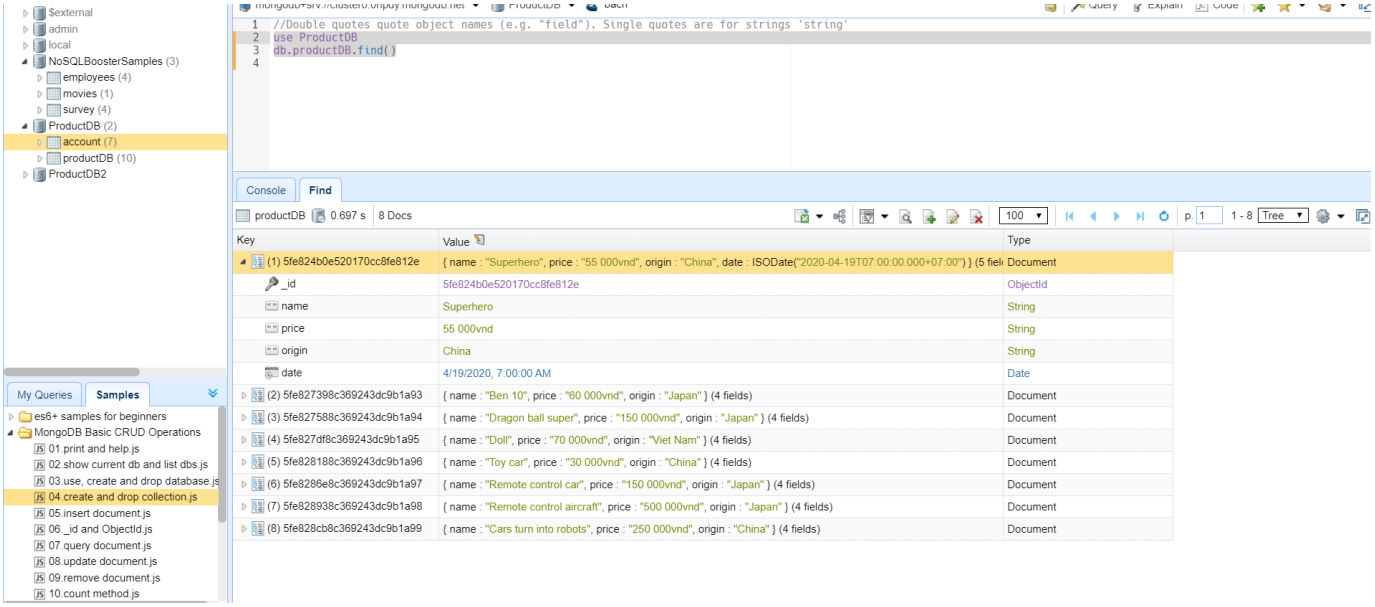


Figure 15. Connect mongoDB

c. Config git and upload file to github

After complete web application in visual studio code, I need upload this code to GitHub to deploy it on Heroku server.

After creating an account on the GitHub, I created a repository named Demo.

Next, Access to the directory containing the web application, open git bash there and execute the commands one after the other.

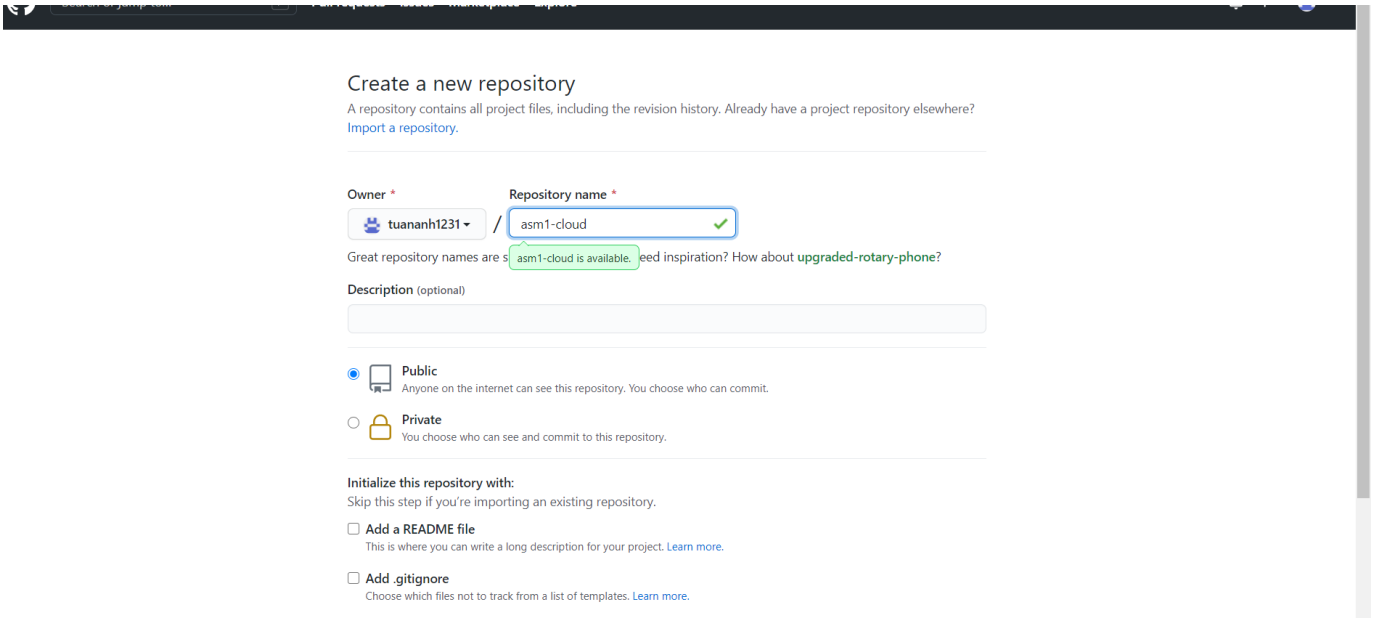


Figure 16. Creating repository

Then I copy the source code and go to the computer screen right click and select "git bash here"

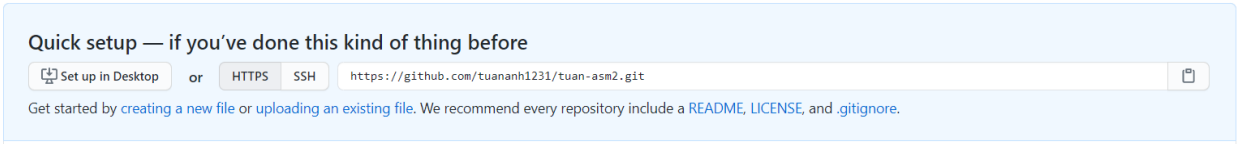


Figure 18. Copy link github

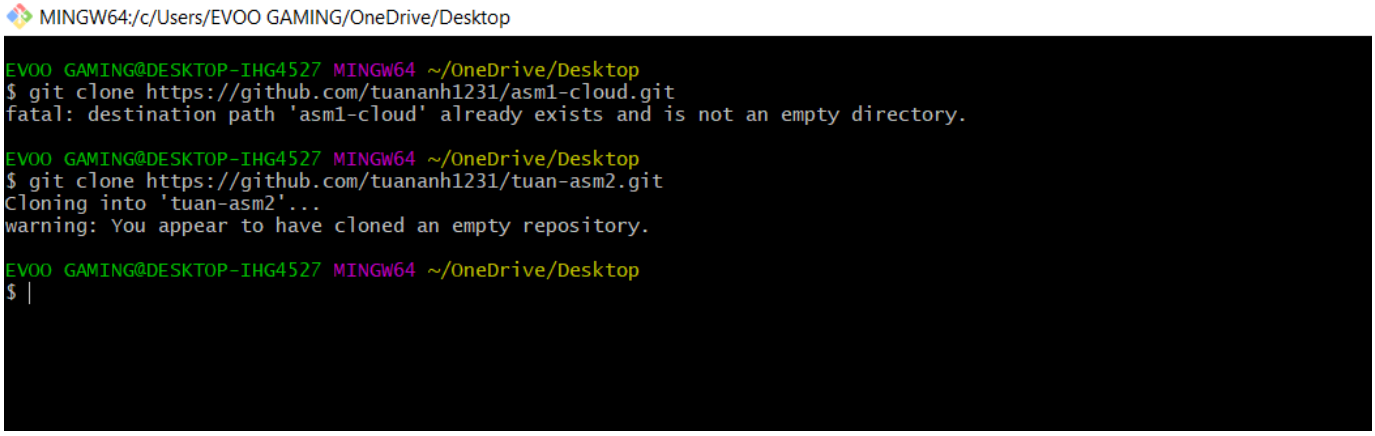


Figure 17. Git bash here

Then I copy the old file into the new one I created with git bash here

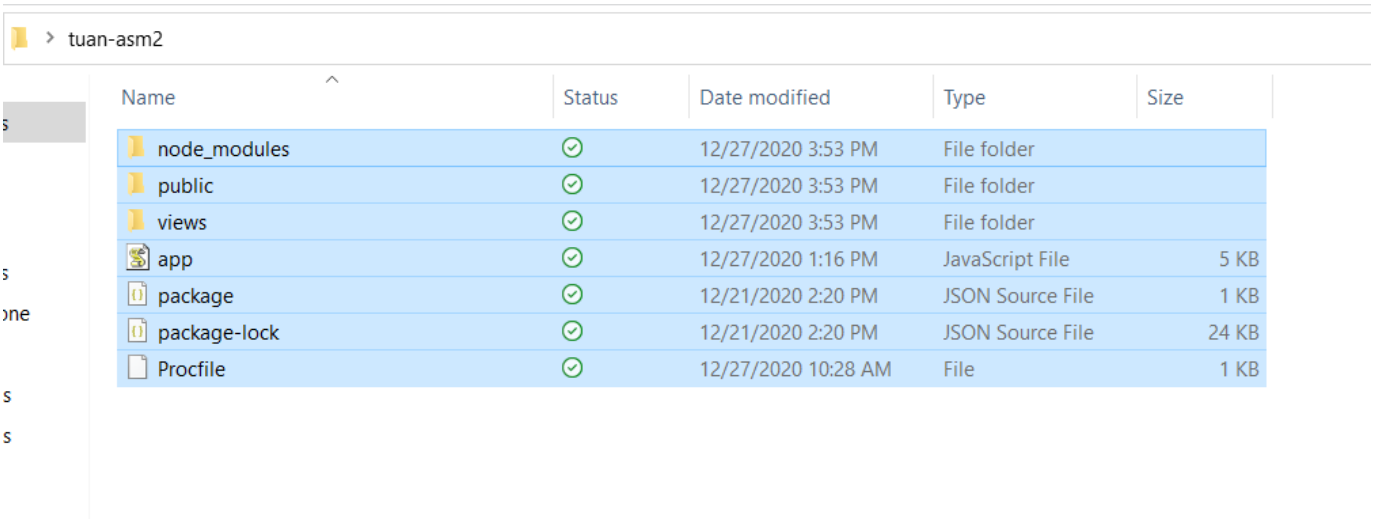


Figure 19. Copy new file create by git bash here

Then I copy the commands below into git bash and push to github

```
git commit -m "first commit"
```

```
 create mode 100644 views/register.hbs
 create mode 100644 views/search.hbs
 create mode 100644 views/update.hbs

EVOO GAMING@DESKTOP-IHG4527 MINGW64 ~/OneDrive/Desktop/tuan-asm2 (master)
$ git push
Enumerating objects: 1236, done.
Counting objects: 100% (1236/1236), done.
Delta compression using up to 12 threads
Compressing objects: 100% (1199/1199), done.
Writing objects: 100% (1236/1236), 2.89 MiB | 591.00 KiB/s, done.
Total 1236 (delta 174), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (174/174), done.
To https://github.com/tuananh1231/tuan-asm2.git
 * [new branch]      master -> master

EVOO GAMING@DESKTOP-IHG4527 MINGW64 ~/OneDrive/Desktop/tuan-asm2 (master)
$
```

Figure 20. Push github

After upload successful, repository of this project looks like in the following image

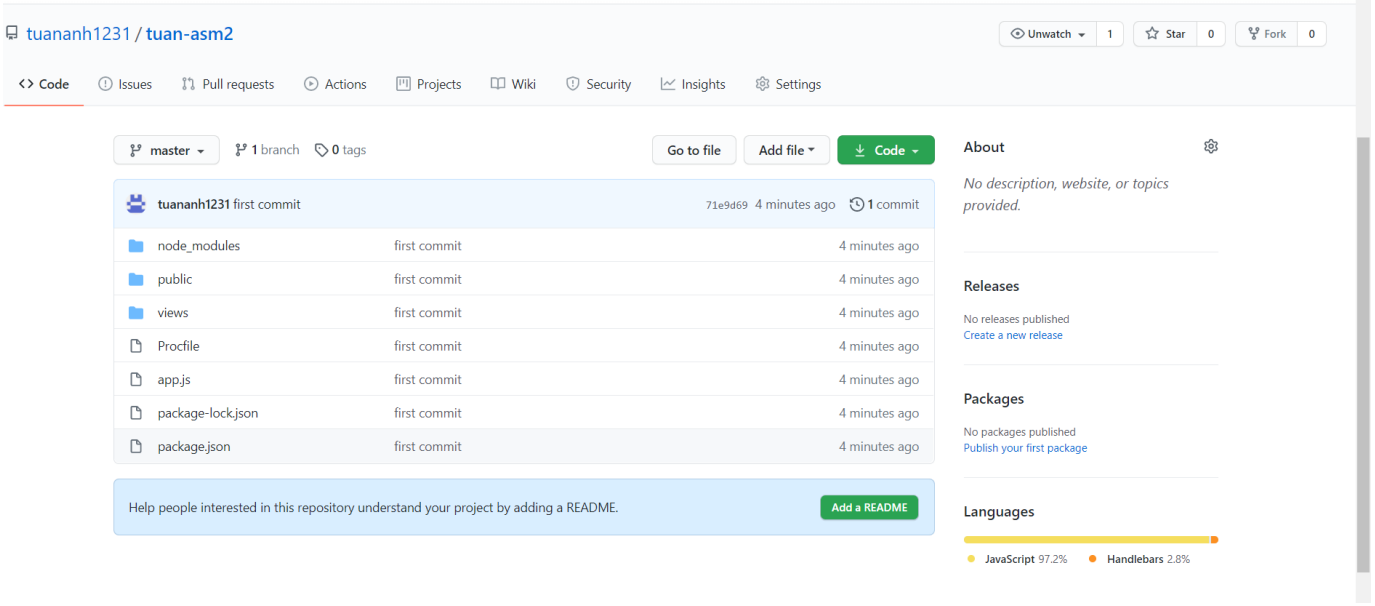


Figure 21. Github online

d. Deploy code on Heroku server

Before deploy web application on Heroku server. I need to config something in code.

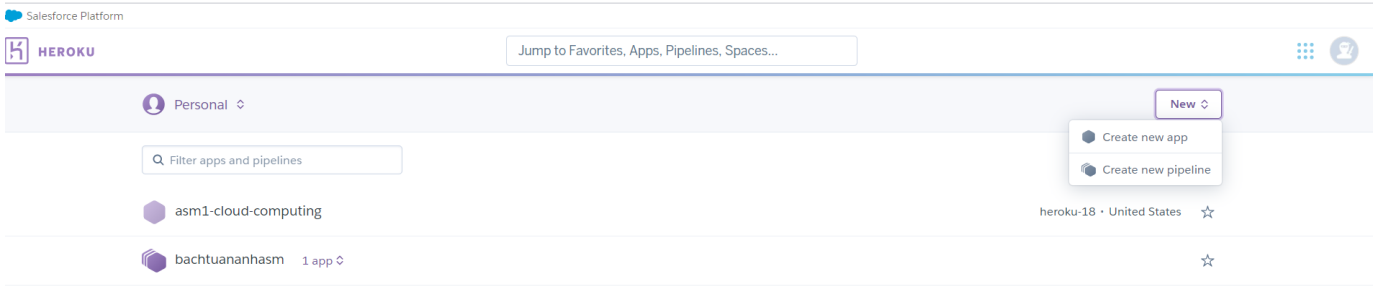
Firstly, running command “npm init” to create file package.json.

File package.jon is used to:

- Specify the project version and project description (project name, author, copyright ...)
- List the packages that the project depends on (the libraries that the project uses)
- Easily share projects among developers, making projects reusable as a library.

Figure 22. File package.json

Next, I created a Heroku account to deploy web application. After successful account creation Click "Create new app" to create the app run on server.



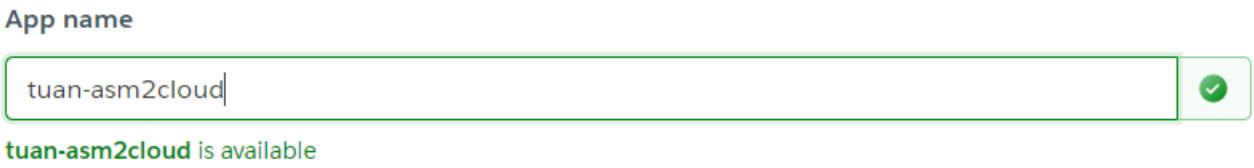


Figure 23. Creating app on Heroku server

After creating the app, go to the deploy interface

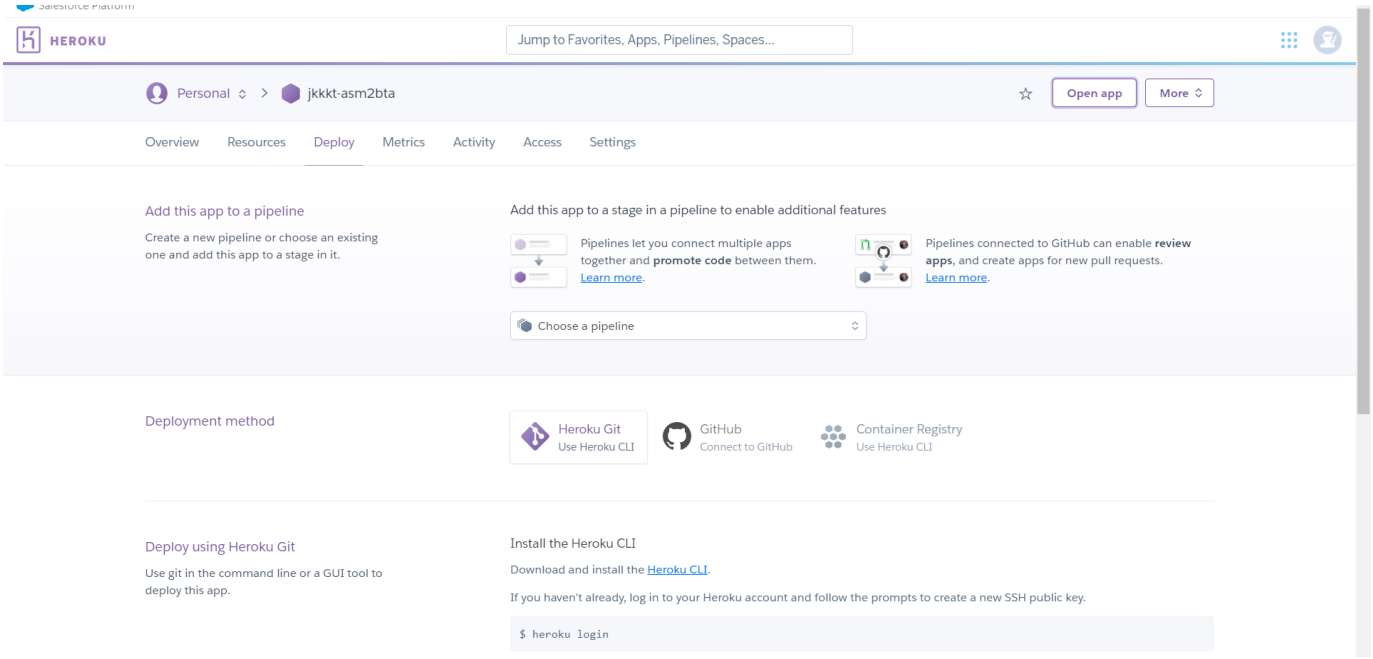


Figure 24. Deploy page

Next, I will copy this code into the app and to build the app

```
$ heroku login
```

Clone the repository

Use Git to clone jkkkt-asm2bta's source code to your local mac

```
$ heroku git:clone -a jkkkt-asm2bta
$ cd jkkkt-asm2bta
```

Deploy your changes

Make some changes to the code you just cloned and deploy th

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

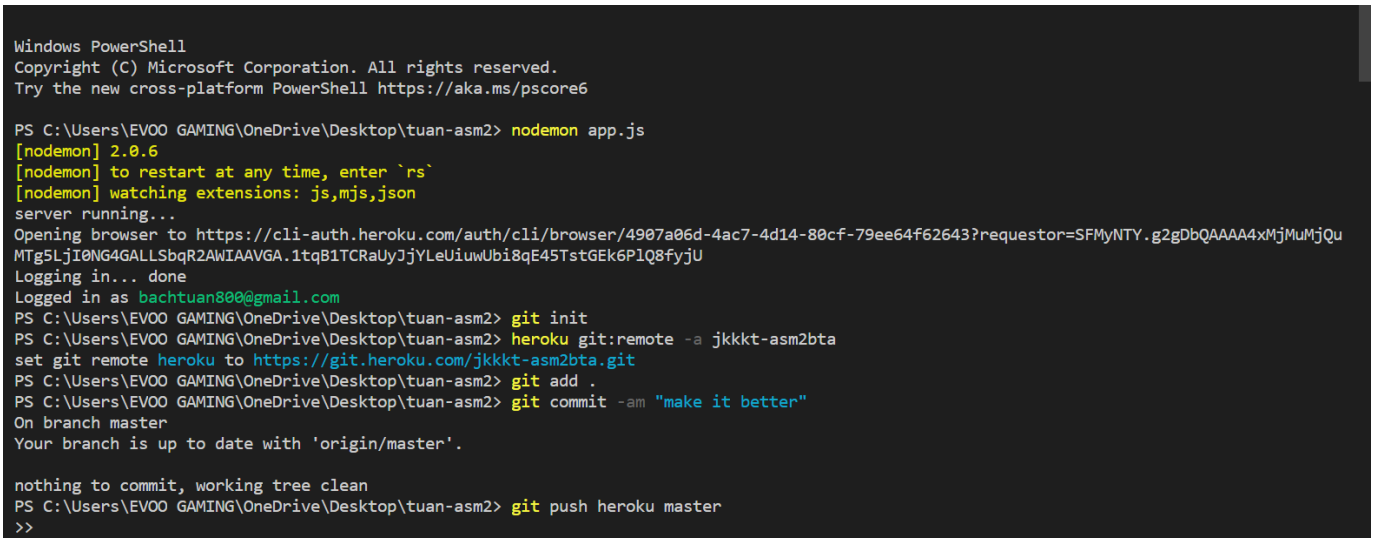


Figure 25. Build app Heroku

During the process I got a small error and missing an app file after adding and rebuilding the app was pushed up and the website was running fine.

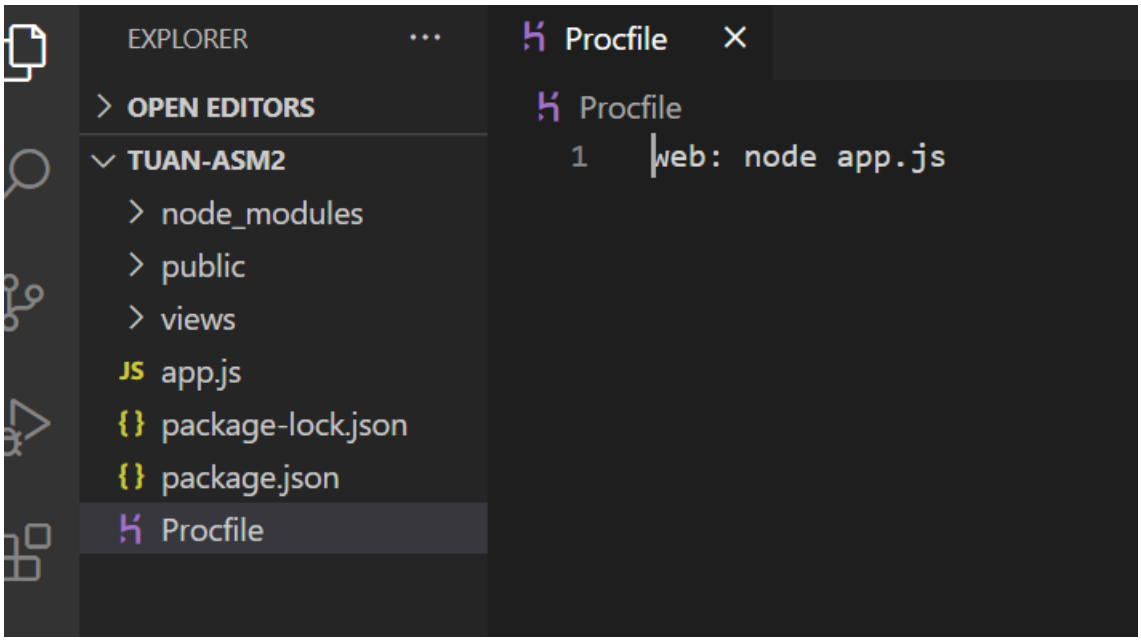


Figure 26. Procfile in visual

After the push is finished we press open the app and view the website after the code is pushed to heroku and perform the functions of the administrator.

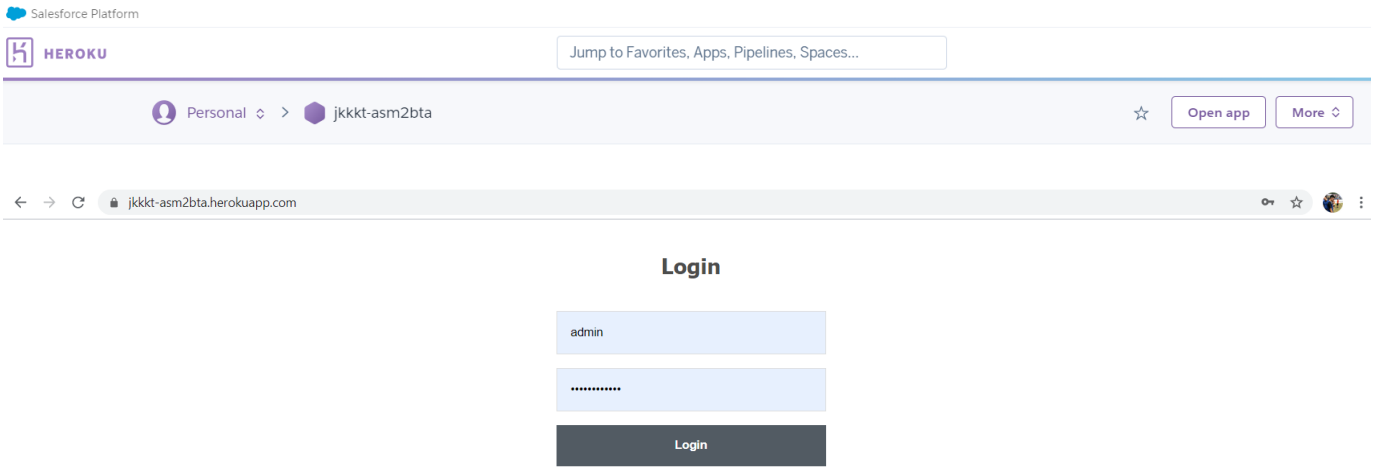


Figure 27. Heroku app

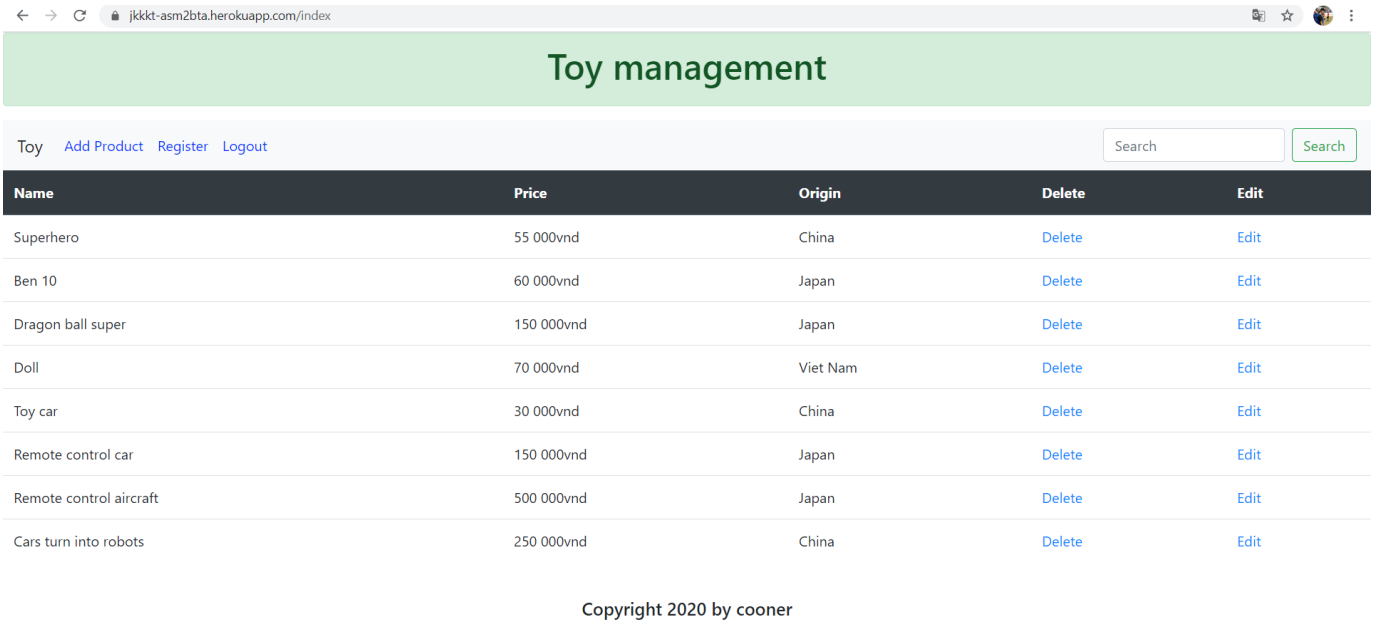


Figure 28. Heroku app

5. Source code and website

Source code: https://github.com/tuananh1231/tuan-asm2

Webside: https://jkkkt-asm2bta.herokuapp.com/

III. Difficult issues

In the process of implementing this project I encountered many difficulties. And below, I'll give you some of the challenges and how I overcome them.

First, for the versioning framework of the node.js. platform. Since this is my first time using express to build the web, it is still difficult to perfect the website. Especially when applying Ajax to this project, there are really a lot of errors (event not receiving, server not receiving data, client not transmitting data ...), there were times when I felt annoyed. This part, it took me a lot of time.

Next up is the MongoDB database. When I connect the MongoDB compass to the MongoDB atlat on the server, I keep getting disconnected. I have been unable to connect to the database for many hours. Sometimes it's the wrong account and password, sometimes it's the carrier. Also, in the process of setting up the IP address to access the database, I do not leave it everywhere, this is also a difficulty that I encountered.

Next is the git part. This part seems less difficult to me and it is also easier. At first, when I created the repository, I didn't know how to connect the repository to the local project. Also when I have trouble removing one remote to add another.

Finally, deploy the web application to the Heroku server. This is a new knowledge, so I also encountered some difficulties here such as needing to set Port env for app.listen before deploying them to Heroku, or creating a package.json file before deploying web applications to Heroku.

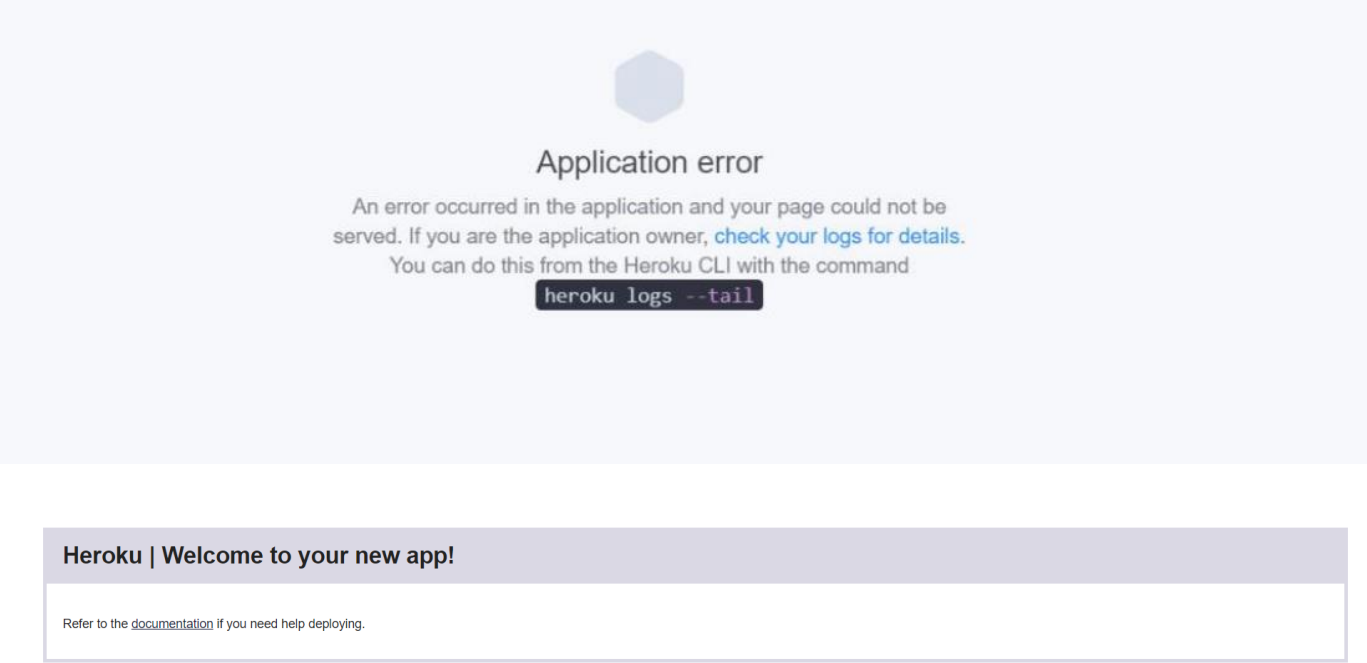


Figure 29. Heroku error when build

This is one of the bugs that I am having difficulty deploying web applications. I had to create a new repository, a new github to upload the source code, I even had to rewrite the code to be able to fix this.

- So how to overcome them:

To overcome those difficulties - including mental problems (tired, wanting to give up), I had to work a lot, spend more time figuring out the answers. And "google.com" and "stackoverflow.com" were the most important friends that helped me get through all of that.

There are also friends and teachers who have helped me a lot. The syntax code has many errors, you and the teachers help a lot.

C. Security

I. Some issues of Cloud Computing platform

1. Deployment Model

(Anon., 2020)

a. Public cloud

- **Issues:** There is one most problem in the Public Model. That is the low security of all data. The right to manage and store data is not used by the customer but by the provider. Disadvantage for large companies that want to store information internally.
- **Solution:** Make commitments between users and providers. The user must know what kind of data the provider stores. In addition, the hosting providers are committed to keeping information such as buyer / visitor / user details confidential. Customers should commit to the seller having access, what kind of information will be sent to the supplier.

b. Private Cloud

- **Issues:** Only internal users have access to the product, and outsiders cannot access the product without a license.

  Although resources and all information are provided and used by the users themselves, it also means that the user must manage the technology themselves. Expensive maintenance and other upgrades work.
- **Solution**: Use some open source or some storage like OpenStack, StorPool, StorageOS, StoreVirtual, etc.

c. Hybrid Cloud

- **Issues**: The hybrid model is a model that can be combined between public and private models. But that is both the advantage and the disadvantage of this series. Because hybrid companies need to work on the same project on both models and find a way to optimize that application.
- **Solution**: Suggest 2 vendors to meet your needs at lower prices than other companies are Joyent and Liquid Web.

  Because: Joyent is a pioneer in creating hybrid cars, so they can be considered as the ones who understand the pros / cons of this model. They made many modifications to suit the needs of their customers to increase their application scalability; Liquid Web will custom create a package that is right for you. Can be used alone between Joyent and Liquid Web or combined to create a highly secure and scalable Hybrid Cloud.

II. Security issues and solution in cloud computing environment

1. Security issues in cloud computing environment

a. Organizational security risks

According to (Tiwari, 2020) If a CSP is out of business or acquired by another organization, we need to negotiate with that organization first, otherwise. There could be a threat from malicious insiders within the organization who could cause harm using the data provided by their CSC.

b. Physical Security Risks

According to (Jain, 2013) The physical location of the cloud datacenter must be secured by the CSP to prevent on-site unauthorized access to CSC data. All customer information must be strictly kept confidential by CSP. Even the firewall and encryption cannot protect against physical data theft

c. Data Security Risks

The three main attributes that we need to ensure are (Jain, 2013)

- Data integrity: The ability to ensure that a system and its data are not tampered with. Integrity protection not only protects data, but also protects the operating system, applications and hardware from being altered by unauthorized individuals.
- Confidentiality: Ensuring that data exchanged is not accessible to unauthorized users. Users can be applications, processes, other systems and / or people. When designing a system, appropriate control mechanisms are needed to enforce security, as well as policies that dictate what authorized users can and cannot do with data. The more sensitive the data, the higher the security level.
- Availability: ensuring that systems, applications and data are available to users when they need them. The most common attack affecting availability is denial of service, in which an attacker interrupts access to information, systems, devices, or other network resources.

d. Technology security risks

According to (Jain, 2013) These risks relate to hardware, technology and services provided by CSP

In the public cloud, with many rental features, these include resource sharing isolation issues and the risks associated with changing CSP, i.e. mobility. CSP's infrastructure needs regular maintenance and inspection.

e. Compliance and Audit Risks

According to (Jain, 2013) These are risks related to the law

That is, the risk involves a lack of information about jurisdiction, changing jurisdictions, illegal contractual terms, and ongoing legal disputes.

For example, depending on the location, some CSPs may be required by law to transfer sensitive information if required by an authorized person.

2. Solution

a. Data security properties

According to (Jain, 2013) Privacy: Privacy ensures that CSC's personal information and identity are not disclosed to unauthorized users.

Confidentiality: This is related to data privacy as it is the property that guarantees that data belonging to a CSC is not disclosed to any unauthorized party.

Integrity: Refers to the confidence that data stored in the cloud has not been altered by unauthorized parties in any way when it is accessed.

Availability: Ensure that CSC has access to their data and is not denied access by mistake or due to malicious attacks of any organization

b. Data stages

According to (Jain, 2013) Data in transit: This is when data in the process is transferred to a cloud infrastructure or computing device used by CSC. Encoding is often used here

Data at rest: This is when the data is already being stored in cloud infrastructure. The main problem of this stage for the CSC is that they lose control of the data

Data in use: This is when data is being processed into information. Here, the problem may lie with the data being corrupted while it is being processed

c. Methods of ensuring data security

According to (Jain, 2013) Authentication in the cloud

- CPC validation can be done by the CSP or outsourced to third-party expert examples

Cloud encryption technology

- Caesar cipher
- S-DES
- RSA
- Secure port layer

d. Caesar cipher

According to (Jain, 2013) It is a classic alternative cipher. A simple example of such a cipher replaces the character in the alphabet with a character that precedes it three steps, eg "ZULU" will be converted to "CXOX".

Hence, only 25 viable key and password options can be easily enforced

e. S-DES

According to (Jain, 2013) The simple data encryption standard has a key generation process instead of using a key for encryption and decryption, S-DES's key generation process generates two sub-keys after the initial 10-bit input processing.

It is no longer widely used as the power of computation has caught up with its disruptions.

f. RSA

According to (Jain, 2013) A cryptographic algorithm whose encryption key is public and different from the decryption key kept secret

It is one of the more commonly used encryption algorithms today

g. Secure Sockets Layer (SSL)

According to (Jain, 2013) It is 128 bit encrypted. This is a protocol commonly used to manage the security of transmitting messages over the Internet and it uses public and private key encryption.

III. ATN Company Issues

As I mentioned above, the cloud has many advantages and disadvantages. With ATN's company problem, I think that the most recognizable risk here is the risk of technology security. To get around this, I used visual studio code, MongoDB, Heroku, and GitHub. In my opinion, this risk is due to CSP. In the website I developed, I created a website with some functions such as logging in, displaying product information, deleting products. These functions have been developed and perfected. But the website's interface is not eye-catching. During development, the server sometimes encountered connectivity errors. In my opinion, the server must be guaranteed a stable connection with all CSCs.

D. Conclusion

In short, through the evaluation and analysis of common problems of the cloud computing model gave me an overview of the possible risks to the system. From there, help me give out specific solutions to apply in the system during operation. Avoid the expected risks.

Besides, security issues are also discussed and suitable solutions are proposed. From there, the system's data safety is absolutely protected.

References

Anon., 2020. *Cloud_Service_and_Deployment_Models.pdf.* [Online]
Available at:
https://cloudcomputing.ieee.org/images/files/education/studygroup/Cloud_Service_and_Deployment_Models.pdf
[Accessed 27 12 2020].

Anon., 2020. *why-use-mongodb.* [Online]
Available at: https://www.mongodb.com/why-use-mongodb
[Accessed 27 12 2020].

BRADFORD, L., 2020. *what-is-github-and-why-should-i-use-it-2071946#:~:text=GitHub%20is%20an%20open%2Dsource,changes%20made%20to%20every%20iteration..* [Online]
Available at: https://www.thebalancecareers.com/what-is-github-and-why-should-i-use-it-2071946#:~:text=GitHub%20is%20an%20open%2Dsource,changes%20made%20to%20every%20iteration.
[Accessed 27 12 2020].

Jain, P. R., 2013. *www.cse.wustl.edu.* [Online]
Available at: https://www.cse.wustl.edu/~jain/cse570-15/index.html
[Accessed 31 12 2020].

Lambert, J., 2018. *77b34ddcd183.* [Online]
Available at: https://morioh.com/p/77b34ddcd183
[Accessed 27 12 2020].

Tiwari, P., 2020. *316922625_Security_Issues_and_Solutions_in_Cloud_Computing.* [Online]
Available at:
https://www.researchgate.net/publication/316922625_Security_Issues_and_Solutions_in_Cloud_Computing
[Accessed 27 12 2020].