**ASSIGNMENT 2 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | Unit 9: Cloud Computing | | |
| **Submission date** |  | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Nguyen Quoc Anh | **Student ID** | GCH18888 |
| **Class** | GCH0718 | **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:** | | |

A. INTRODUCTION

B) DESIGN

I. Overview Function

1. Solution

2. Use-case diagram

3. Website Screen Shots

II. Code implement and deploy process

1. Tools and framework

1.1 IDE

1.2 Framework

1.3 Source Code Manager

1.4 Database server

1.5 Cloud computing module

2. Deploy

2.1 Config framework express on the env

2.2 Config and connect with mongodb

2.3 Config git and upload file to github

2.4 Deploy code on Heroku server

3. Code implement

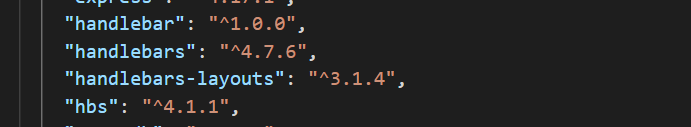
4. Source code and website

III. Difficulties

There are many difficulties when i develop and implement this project. But i did research and learn how to fix it. In this part, i will list some difficulties which i have to encountered and how i can overcome them.

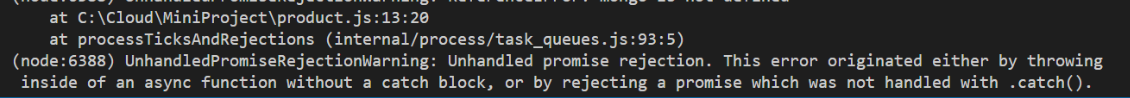
1. Programming

a) Difficulties

Other than that this is my first time using it, I made quite a few mistakes during development. Nodejs has too many different frameworks like Express, NestJs, SailJs, KoaJs. Each framework has some different pros and cons. This makes me very confused in choosing the right framework for the project. After choosing a suitable framework, we have to choose the suitable view engine ... Because Nodejs is an open source platform, so it has the advantage of abundant community-resources not like other closed platform like asp, javaspring, ... This leads to inconsistency in naming libraries. These libraries all have similarities and differences, so it can be difficult to distinguish when searching for tutorial articles. During the first use, when i dont remember the exactly name of handlebars and only install handlebar library but it's still work till i have some bugs and need to uninstall and reinstall handlebars. Each library has its own specific syntax, so I confused a lot when looking up their functions.

Next, I ran into a problem working with the view engine. Different from writing server side and client-side code together in one file. Source Code is structured in a model that is almost equivalent to MVC. Hence transferring the data between my view and routes in the early stages encountered some difficulties.

Beside, when working with data, I had many errors about asynchronous processing. Unlike some other server-side languages, javascript strictly handles asynchronous handling. It is easy to make an error if we use the inappropriate callback functions.



b) Solution

Faced with the above problems, the first thing I did was to thoroughly re-learn how nodejs works. By understanding the basic keywords as well as operating principles, I analyzed pros, cons and chose express as framework for the project. Then I chose hbs view engine because its syntax and keywords are easy to remember. In addition, hbs has a lot of users, so it's easy to look up information. because nodejs is inconsistency in naming libraries. so I'll just read the articles and documentation on the nodejs homepage. Also, when consulting the tutorials on forums, I need to look up exactly which library I am using. For example, with handlebars viewengine I have 2 libraries: hbs and handlebars. They have a lot of similarities such as supporting 1 view engine, compile, register helper, .... However, I will only select 1 library for the whole article to ensure consistency.

For asynchronous handling I have chosen to use async function and await instead of the callback function. Async / await is a special syntax that makes working with Promises easier. When using async / await, the structure of an asynchronous handler will be more similar to a synchronous handler. If handled this way, even if I make more requests, the program structure is still very clear and coherent.

2. Debugging

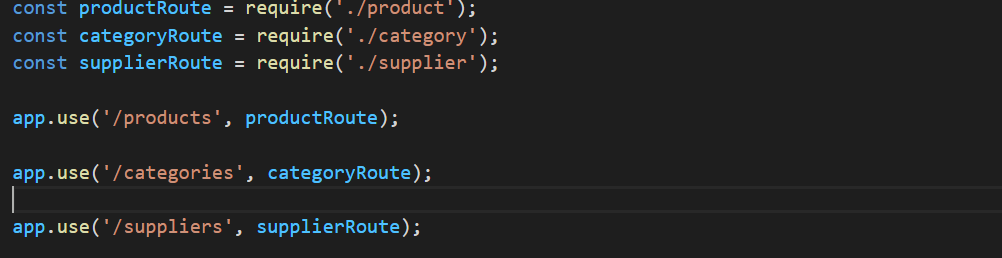
a) Difficulties

when i start to write code, all code are written in the one file is server.js. it make debug become more hard but easier to get error. When the software is too functional, route naming will easily be duplicated. When a bug occurs, it's very hard to find the module which is related to the bug. In addition, passing data between the views and routes also sometimes leads to errors and makes it difficult to detect the cause. Because, the cause may be database or form.

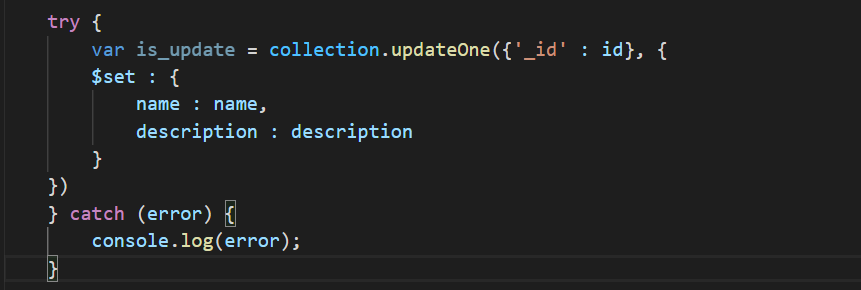
b) Solution

To resolve the problem, i have use router function of express. Which router function, i can divide the app.js to many other route with different features. The names of the routes in each router may overlap. It makes finding bugs easier, when a function occur errors, i can find exactly which router and route relate to it.



In figure below, I have divided app to 3 main router. Product router has all the function relate to the product like update, delete , add ,… It’s the same for category and supplier router. 

Beside, i have to use try catch and console.log in debugging. It helps me catch all the possible exeception. I can also check the data if it can send from form.

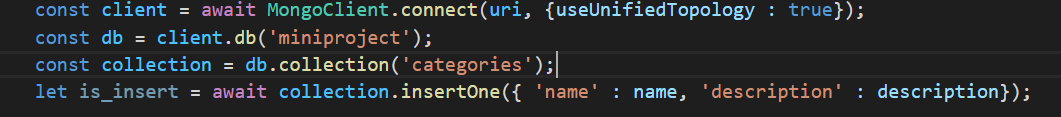


3) Deploying

a) Database: I spent a lot of time connecting to the database. In the first time i tried to connect to database, it got a lot of errors and I don't know what it means. There is many other guide for that problem. I have tried many of them but are not successful.

D:\131683173_2973026899585153_1422831879248269584_n.png

Solution: I have connected according to documentation in the mongodb homepage, in addition I use asynchronous handling in database connection to ensure stability.



b) Git: When using git, I ran into problems when the github GUI repositories did not sync with repositories on the web. Using git in the console was quite difficult for me as I did not remember all the commands.

Solution: I searched for information about that problem on forums like stackoverflow and found the cause. The Git GUI will only show the repositories in the Github directory in the document. So when I create a new repository I will create in that directory. Also I have tried to learn the command commands and it was not too difficult. Now i have switched to using command instead of GUI.

C. SECURITY

I. Some issue of Cloud computing platform

1. Public Cloud

a) Issue

Loss of control. When you outsource to the public cloud, literally out of reach. Any configuration and other aspects of IT management are left to a group that are not directly involved in daily operations Low security level, because data often shares common location, not as secure as other Cloud models.

b) Solution

Choose from reputable providers who have a clear policy and commitment to ensuring the security of your information and your interests.

2. Private Cloud

a) Issue

Private clouds have strict restrictions on access.. when a company’s employees were working outside the office, they have some difficulties to connect to the private cloud lead to securely access any applications, data or files they needed from the company servers.

b) Solution

Use VPN, VPN has two main advatange in this case

Cost-effectively provide employees, wherever they are, with a secure connection to both the company’s cloud and data center-based applications and data.

Authenticate users and ensure comprehensive, consistent security without having to purchase expensive hardware or networking equipment, or add IT complexity.

3. Hybrid Cloud

a) Issue

Hybrid cloud is the combination of public cloud and private cloud. Therefore, it has all the issue of two kind cloud. In addition, the issue of technology and optimization also needs to be carefully considered.

b) Solution

Careful analysis of the size and usage of the system is required to come up with an appropriate model. Depending on the size and scale of the system, we need to calculated about the security and technical aspects of the system is required in order to avoid technical errors and security mishaps.

II. Security issues and solution in cloud computing enviroment.

**1. Data Loss/Leakage**

Cloud-based environments make it easy to share the data stored within them. These environments are accessible directly from the public Internet and include the ability to share data easily with other parties via direct email invitations or by sharing a public link to the data.

Although this is one of the main advantages of cloud, the ease of data sharing in the cloud raises significant concerns about data loss or leakage. In fact, as their main cloud security issue, 69 percent of organizations point to this. Data exchange using shared connections or setting up a public cloud-based archive makes it open to those with link information, and there are unique resources for searching for these unsecured cloud installations on the Internet.

**Solution:**

**2. Physical Security Risk**

The security measures are useless if an attacker can connect directly to the system. That way, they'll be able to bypass the majority of security systems and third-party security services like cloudflare. Besides, criminals can directly harm the physical servers of the cloud.

**Solution:** The physical location of the cloud data center must be secured by the CSP in order to prevent unauthorized on-site access of CSC data. In addition, there should be security policies such as system guards or supervisors. There are security systems such as fingerprint scanning, keycard access and biometric scans to restrict access to sensitive locations in the data center.

**3. Organization**

If a Cloud Service Provider (CSP) leaves business or is purchased by another entity. There may be a place The threat of the organization's malicious insiders who might harm using the data generated by their CSCs(Customer/Consumer of Cloud Service).

Solution: The risk of malicious workers being hired by a CSP can be mitigated by Putting strict legal restrictions on contracts when recruiting employees. A thorough review of the CSP As well as a comprehensive security breach notification process. On the user side, it is important to work closely with the new distributor's representative to clarify the contractual terms. Besides, users also need to prepare forms to secure data.

**4. Risks of Compliance and Audit**

There are law-related threats. That is, risks associated with lack of jurisdictional information, changes in jurisdictional information, Jurisdiction, the contract's unlawful provisions and pending legal conflicts. Depending on venue, for instance, Any CSPs, if requested by the government, may be required by law to turn over confidential information.

Solution: This area deals mainly with legal problems and, as such, both Cloud Service Providers (CSPs) and CSCs need to consider legal and regulatory responsibilities and ensure the CSCs (Cloud Service Customer/Consumer) That these duties be met by any contracts made. The CSP should also ensure that its capabilities for exploration are Security and data protection are not affected. Having seen several approaches used in the next four fields to avoid safety lapses, in the next one We shall look at a subsection of some of the primary techniques used for ensuring data security (Worlanyo, (2015)

**5. Technology security risks**

These risks are the failures associated with the hardware, technologies and services provided by the CSP (Cloud Service Provider) In the public cloud, with its multi tenancy features, these include resource sharing isolation problems, and risks related to changing CSPs, i.e. portability. Regular maintenance and audit of infrastructure by CSP is recommended (Worlanyo, (2015).

Solution: Virtualized trust protection focused on defence and integrity - CSP may use the following Structure: a DHT-based overlay network hierarchy, with particular tasks to be performed by each layer. The lowest layer deals with the accumulation of reputations and colluders being probed. The top stratum deals with Similar attacks. The aggregation of credibility here is linked to the use of multiple sources to check those connections and probing colluders refer to testing if known malignant sources are associated with any sources.

Secure virtualization - CSP can use an Advanced Cloud Protection system (ACPS) to ensure the security of guest virtual machines and of distributed computing middleware. Behavior of cloud components can also be monitored by logging and periodic checking of executable system files. Trust model for interoperability and security. There should be separate domains for providers and users, each with a special trust agent. An independent entity that gathers security information used to gather security information is a confidence agent. Check for an end point. For service providers and clients, there should also be various trust strategies (Worlanyo, (2015).

III. How an organisation should protect their data

1. Using reputable and quality Cloud Server service

Cloud technology is developing day by day, security is also more complete. To ensure the safety of Cloud Server, the system has advanced security in many layers, ensuring safety for data in and out of the system, ... However, hacker attacks are also more and more sophisticated. ITs and businesses need to update the most modern security technology and software to ensure data safety on the Cloud. Use reputable and quality Cloud Server services so that vulnerabilities are discovered and patched in a timely manner, ensuring data safety. For example: when a new updated version, there are so many hackers will try to break its security system. Therefore, the software will always need to have updated patches.

2. Choose a reputable provider

Choosing a reputable provider is considered a top priority when using Cloud Server. Because new reputable providers ensure your data is secure and secure. The IT team of the enterprise also needs to keep up with and fully update security methods to prevent network threats aimed at Cloud Server.

3. Data encrypt