UTM
UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

SCSJ3483
WEB TECHNOLOGY
ASSIGNMENT 4

GROUP NAME: TYPO

GROUP MEMBER: PANG WEN JIE (A18CS0232)
TAN CHONG LIM (A18CS0255)
ONG SHI BING (A18CS0230)
YEOH KAI XIANG (A18CS0278)

SECTION: 02

LECTURER: MR NORIZAM KATMON

Table of Contents

1.0 Prerequisites **3**

2.0 Prepare the Environment **4**

2.1 MySQL Installation and Configuration 4

2.2 PHP Installation and Configuration (Including phpMyAdmin) 6

2.3 Node.js Installation and Configuration 10

3.0 Setup the Project in Production Mode (CRAS) **12**

3.1 Setup the Database (MySQL) 12

3.2 Setup the Backend (Node.js) 14

3.3 Setup the Frontend (Angular) 15

4.0 References **16**

1.0 Prerequisites

- Ubuntu 20.04 x64 server
- Nginx installed and running
- Domain name (and subdomain) pointing to the server's IP as follows

Type	Name	Content	TTL	Proxy status	
A	admin	134.209.102.65	Auto	Proxied	Edit
A	api	134.209.102.65	Auto	Proxied	Edit
A	cras	134.209.102.65	Auto	Proxied	Edit
A	typo.ninja	134.209.102.65	Auto	Proxied	Edit
A	www	134.209.102.65	Auto	Proxied	Edit

2.0 Prepare the Environment

In this manual, the server will be configured to handle multiple services under one server, with different subdomain name. Nginx will be used as reverse proxy to manage the subdomain. Table below summarize the details.

Domain/Subdomain	Service	Folder Path
typo.ninja www.typo.ninja	Default landing page	/var/www/html
admin.typo.ninja	phpMyAdmin	/var/www/admin.typo.ninja
api.typo.ninja	Backend for CRAS (node.js)	/var/www/api.typo.ninja
cras.typo.ninja	Frontend for CRAS (angular)	/var/www/cras.typo.ninja

2.1 MySQL Installation and Configuration

1. Run `sudo apt-get install mysql-server` to download mysql server

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfcgi-perl
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils
```

2. Run `mysql_secure_installation` to start the installation

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:
```

3. Choose `No` for the first question, then enter new password for MySQL root user.

```
Press y|Y for Yes, any other key for No: no
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.
```

4. Enter y for the following questions until the end.

```
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
root@ubuntu-s-1vcpu-1gb-sgp1-01:~#
```

5. Run sudo service mysql status to check MySQL status, make sure it is **active** and **running**.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo service mysql status
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2021-07-01 08:31:10 UTC; 4min 38s ago
   Main PID: 13750 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 1136)
     Memory: 329.5M
     CGroup: /system.slice/mysql.service
             └─13750 /usr/sbin/mysqld

Jul 01 08:31:09 ubuntu-s-1vcpu-1gb-sgp1-01 systemd[1]: Starting MySQL Community Server...
Jul 01 08:31:10 ubuntu-s-1vcpu-1gb-sgp1-01 systemd[1]: Started MySQL Community Server.
root@ubuntu-s-1vcpu-1gb-sgp1-01:~#
```

2.2 PHP Installation and Configuration (Including phpMyAdmin)

Note: phpMyAdmin will be used only for managing database in this project.

1. Run `sudo apt-get install php-fpm php-mysql` to install php processor.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo apt-get install php-fpm php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  php-common php7.4-cli php7.4-common php7.4-fpm php7.4-json php7.4-mysql php7.4-opcache php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  php-common php-fpm php-mysql php7.4-cli php7.4-common php7.4-fpm php7.4-json php7.4-mysql php7.4-opcache
  php7.4-readline
0 upgraded, 10 newly installed, 0 to remove and 4 not upgraded.
Need to get 4202 kB of archives.
After this operation, 18.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

2. Run `sudo nano /etc/php/7.4/fpm/php.ini` , then uncomment and change 1 to 0 at line `cgi.fix_pathinfo=0`.

```
796    ; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
797    ; http://php.net/cgi.fix-pathinfo
798    cgi.fix_pathinfo=0
```

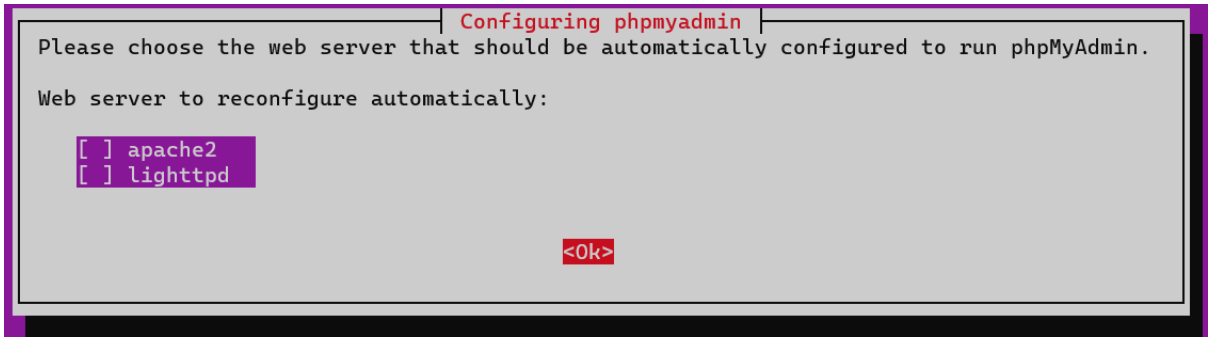
3. Save and restart php processor by running `sudo systemctl restart php7.4-fpm`
4. Run `sudo nano /etc/nginx/sites-available/default`. Remove all the default configuration and add the following lines of codes.

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;
        server_name _;

        location / {
                try_files $uri $uri/ =404;
        }
}

server {
        listen 80 ;
        listen [::]:80 ;

        root /var/www/admin.typo.ninja/phpmyadmin;
        index index.php index.html index.htm index.nginx-debian.html;
        server_name admin.typo.ninja;

        location / {
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    }

        location ~ /\.ht {
                        deny all;
        }
}
```

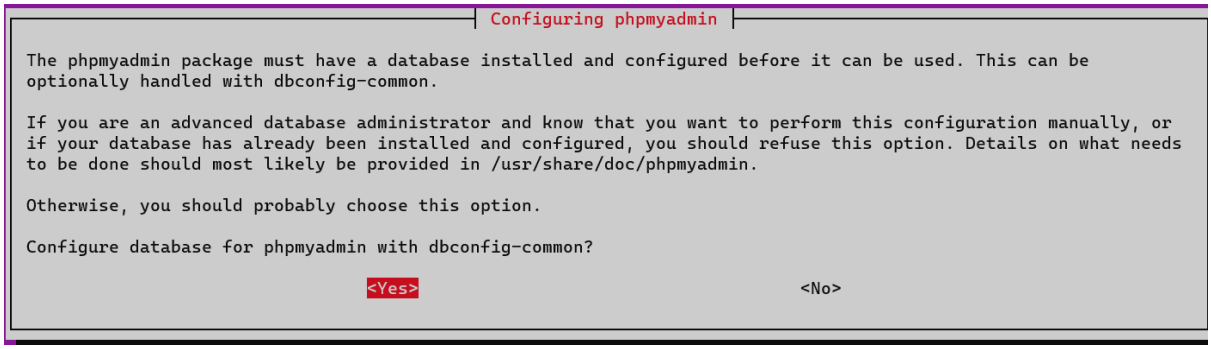
5. Make a new directory by typing `mkdir /var/www/admin.typo.ninja` .
6. Run `sudo systemctl restart nginx` to restart nginx server.
7. Run `sudo apt-get install phpmyadmin` to install phpMyAdmin

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql icc-profiles-free javascript-common libjs-jquery libjs-openlayers libjs-sphinxdoc
```

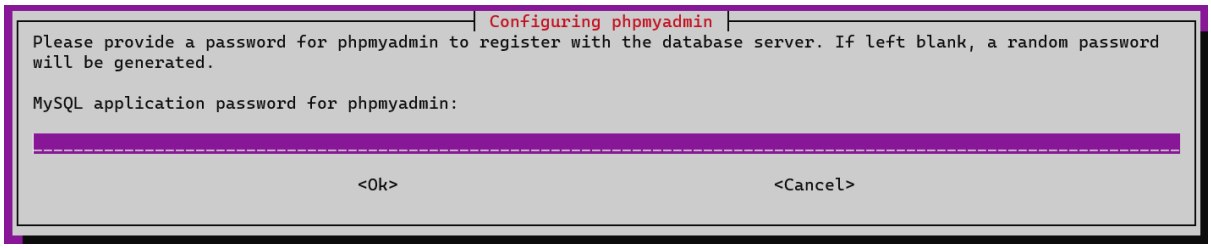
8. Press TAB to skip web server configuration



9. Choose YES to configure database.



10. Input new password for phpMyAdmin



11. Run `sudo ln -s /usr/share/phpmyadmin /var/www/admin.typo.ninja` to link the php files to nginx directory.
12. Run `sudo systemctl restart php7.4-fpm` to restart PHP processor.
13. Go to https://admin.typo.ninja . You should see the following results, meaning that phpMyAdmin server is up and running.

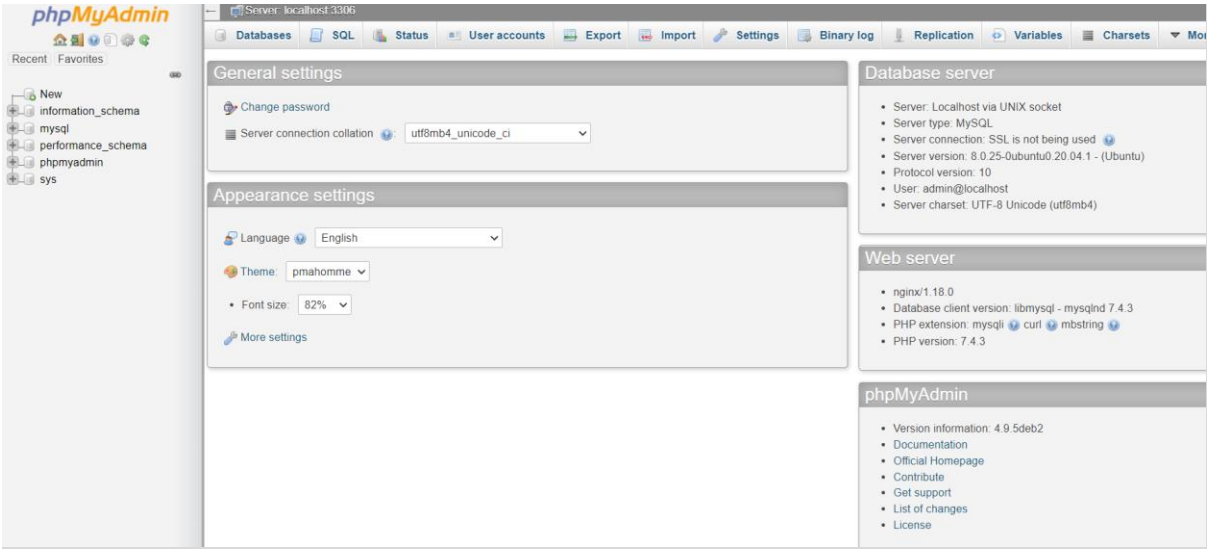
14. Create new superuser for phpMyAdmin.
15. Run sudo mysql -p -u root and type your password to login as root user.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo mysql -p -u root
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

16. Run the following SQL statement to create new user.

```
CREATE USER 'admin'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'new_password';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;
EXIT;
```

17. Login phpMyAdmin using admin and new_password. You should see the following result.


phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
sys
Server: localhost:3306
Databases
SQL
Status
User accounts
Export
Import
Settings
Binary log
Replication
Variables
Charsets
General settings
Change password
Server connection collation
utf8mb4_unicode_ci
Appearance settings
Language
English
Theme:
pmahomme
Font size:
82%
More settings
Database server
Server: Localhost via UNIX socket
Server type: MySQL
Server connection: SSL is not being used
Server version: 8.0.25-0ubuntu0.20.04.1 - (Ubuntu)
Protocol version: 10
User: admin@localhost
Server charset: UTF-8 Unicode (utf8mb4)
Web server
nginx/1.18.0
Database client version: libmysql - mysqlnd 7.4.3
PHP extension: mysqli curl mbstring
PHP version: 7.4.3
phpMyAdmin
Version information: 4.9.5deb2
Documentation
Official Homepage
Contribute
Get support
List of changes
License

2.3 Node.js Installation and Configuration

1. Run curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh to download version 14.x node setup file.
2. Then, run sudo bash nodesource_setup.sh.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# cd ~
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# nano nodesource_setup.sh
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo bash nodesource_setup.sh

## Installing the NodeSource Node.js 14.x repo...


## Populating apt-get cache...

+ apt-get update
Get:1 http://mirrors.digitalocean.com/ubuntu focal InRelease [265 kB]
Get:2 http://mirrors.digitalocean.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://mirrors.digitalocean.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://mirrors.digitalocean.com/ubuntu focal-updates/main amd64 Packages [1081 kB]
Get:5 http://mirrors.digitalocean.com/ubuntu focal-updates/universe amd64 Packages [831 kB]
Get:6 http://mirrors.digitalocean.com/ubuntu focal-updates/universe Translation-en [174 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 2679 kB in 1s (2008 kB/s)
Reading package lists... Done

## Confirming "focal" is supported...
```

3. Run sudo apt-get install nodejs to install nodejs.
4. Run sudo apt-get install build-essential to install other dependencies.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 24.9 MB of archives.
After this operation, 121 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_14.x focal/main amd64 nodejs amd64 14.17.1-deb-1nodesource1 [24.9 MB]
Fetched 24.9 MB in 1s (17.7 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 67985 files and directories currently installed.)
Preparing to unpack .../nodejs_14.17.1-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.17.1-deb-1nodesource1) ...
Setting up nodejs (14.17.1-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

5. Run node --version and npm --version to verify the installation.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# node --version
v14.17.1
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# npm --version
6.14.13
```

6. Run sudo npm i -g pm2 (process manager for Node.js applications)

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# sudo npm install -g pm2
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certain
 circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random for details.
/usr/bin/pm2-runtime -> /usr/lib/node_modules/pm2/bin/pm2-runtime
/usr/bin/pm2 -> /usr/lib/node_modules/pm2/bin/pm2
/usr/bin/pm2-dev -> /usr/lib/node_modules/pm2/bin/pm2-dev
/usr/bin/pm2-docker -> /usr/lib/node_modules/pm2/bin/pm2-docker
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules/pm2/node_modules/chokidar/node_modules/fse
vents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any
"} (current: {"os":"linux","arch":"x64"})

+ pm2@5.1.0
added 180 packages from 203 contributors in 12.207s
```

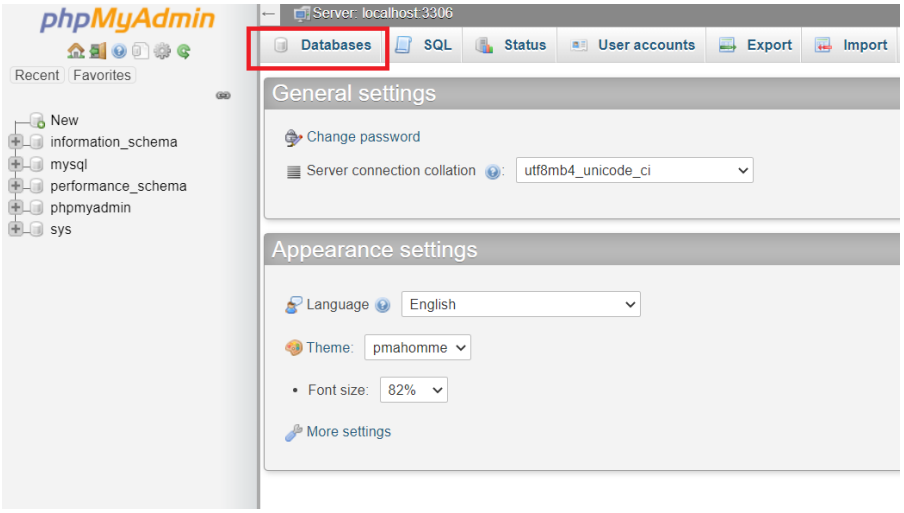
7. Run pm2 startup systemd to enable run on boot.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# pm2 startup systemd

                        -------------

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                      Runtime Edition

       PM2 is a Production Process Manager for Node.js applications
                    with a built-in Load Balancer.

              Start and Daemonize any application:
              $ pm2 start app.js

              Load Balance 4 instances of api.js:
              $ pm2 start api.js -i 4

              Monitor in production:
              $ pm2 monitor

              Make pm2 auto-boot at server restart:
              $ pm2 startup
```

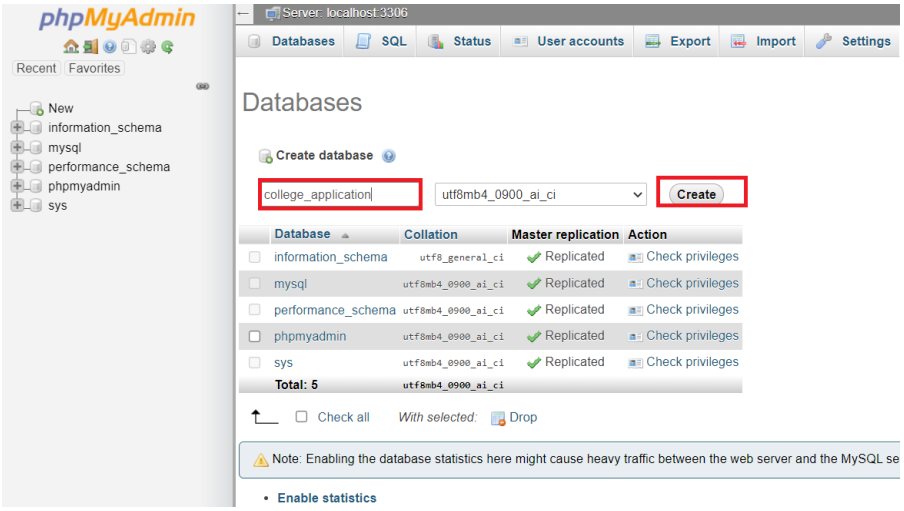
3.0 Setup the Project in Production Mode (CRAS)

3.1 Setup the Database (MySQL)

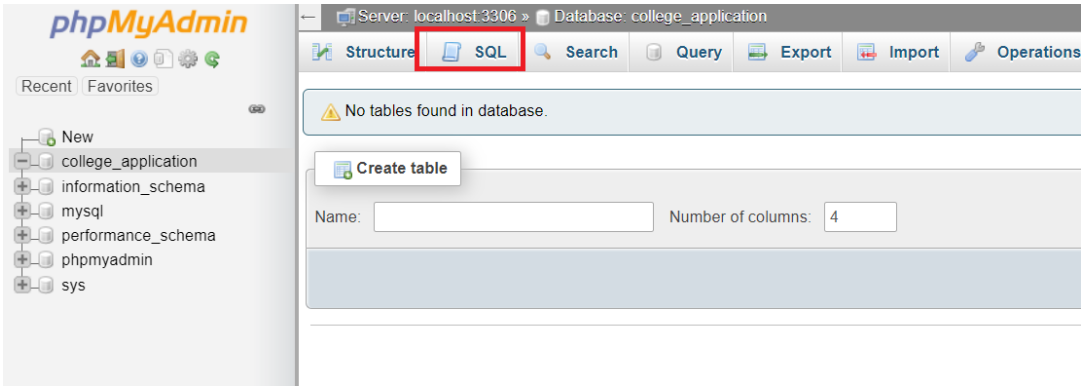
1. Go to https://admin.typo.ninja and login as admin.
2. Create new database by click on Database tab



3. Create new database named college_application

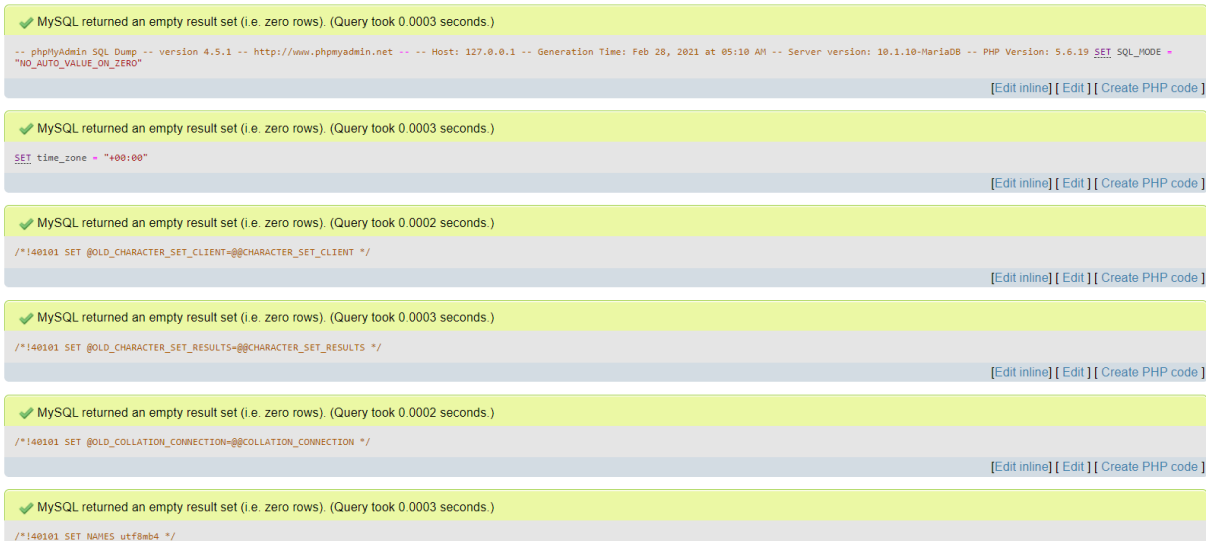


4. Press SQL tab



5. Copy and paste the SQL commands from college_application.sql into the field, and press Go

6. The results should be similar to this.



3.2 Setup the Backend (Node.js)

1. Download the repository: git clone https://github.com/tanchonglim/CRAS-Angular.git

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# git clone https://github.com/tanchonglim/CRAS-Angular.git
Cloning into 'CRAS-Angular'...
remote: Enumerating objects: 385, done.
remote: Counting objects: 100% (385/385), done.
remote: Compressing objects: 100% (231/231), done.
remote: Total 385 (delta 163), reused 344 (delta 122), pack-reused 0
Receiving objects: 100% (385/385), 441.87 KiB | 17.67 MiB/s, done.
Resolving deltas: 100% (163/163), done.
```

2. Run cp -r CRAS-Angular/backend/ /var/www/api.typo.ninja to copy the code to new directory.
3. Run cd /var/www/api.typo.ninja to change current working directory.
4. Run npm i to install the dependencies.
5. Set the environment variable for database username and password by running

   export DATABASE_USER=admin

   export DATABASE_PASSWORD=new_password

6. Run node index.js . If the response is as below, meaning that setup is correct.

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:/var/www/api.typo.ninja# node index.js
Server started at http://localhost:3000
Connected to database
```

7. Press ctrl+c to close the service and run pm2 start index.js to start the server as background process.

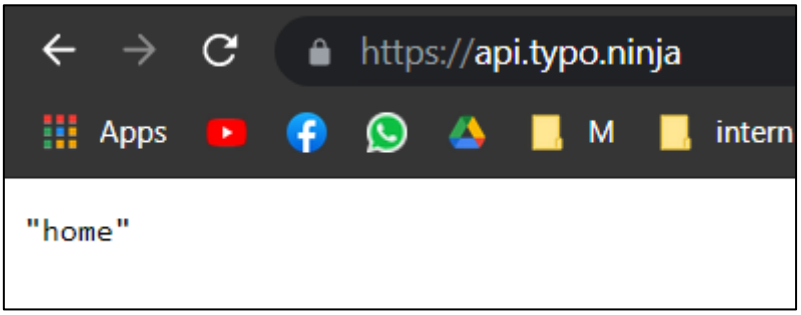
```
root@ubuntu-s-1vcpu-1gb-sgp1-01:/var/www/api.typo.ninja# pm2 start index.js
[PM2] Applying action restartProcessId on app [index](ids: [ 0 ])
[PM2] [index](0) ✓
[PM2] Process successfully started
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	index	default	1.0.0	fork	14169	0s	0	online	0%	20.4mb	root	disabled

8. Run sudo nano /etc/nginx/sites-available/default and add the following codes below the current code. Then, save and exit.

```
server {
        listen 80;
        listen [::]:80;

        root /var/www/api.typo.ninja;
        index index.html index.htm index.nginx-debian.html;
        server_name api.typo.ninja;

        location / {

                proxy_pass http://localhost:3000;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection 'upgrade';
                proxy_set_header Host $host;
                proxy_cache_bypass $http_upgrade;
        }
}
```

9. Run `systemctl restart nginx` to restart nginx server.
10. Go to https://api.typo.ninja/ . You should see the following output.



3.3 Setup the Frontend (Angular)

1. At your own PC, download the repository from https://github.com/tanchonglim/CRAS-Angular
2. Run `npm i` in the frontend/ directory
3. At /src/environments/environments.prod.ts , change the apiUrl to https://api.typo.ninja
4. Run `ng build` in frontend/ directory
5. Upload the files under /dist/cras/ to hosting server under directory /var/www/cras.typo.ninja by using FileZilla or any FTP client.
6. Run `sudo nano /etc/nginx/sites-available/default` and add the following codes below the current code. Then, save and exit.

```
server {
        listen 80;
        listen [::]:80;

        root /var/www/cras.typo.ninja;
        index index.html ;
        server_name cras.typo.ninja;

        location / {
                try_files $uri $uri/ /index.html?$args;
        }
}
```

11. Run `systemctl restart nginx` to restart nginx server.
12. Go to https://cras.typo.ninja/ . You should see the home page of CRAS.

4.0 References

https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-in-ubuntu-16-04

https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-with-nginx-on-ubuntu-16-04

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-16-04

https://nicknetvideos.com/blog/post/how-to-run-a-website-in-a-subdomain-in-digital-ocean