**Project and Server Setup**

Contents

[**1.** **Summary** 1](#_Toc156828199)

[**2.** **Server Setup** 1](#_Toc156828200)

[**3.** **Project installation** 4](#_Toc156828201)

# **Summary**

Please follow these steps to setup the ticket support system on a fresh installation of **Ubuntu 22.0.4**.

Please note that this project was setup and tested on Ubuntu 22.0.4 LTS using windows sub-system for Linux (WSL).

The projects composer requirements are as follows:

"require": {

"php": "^8.1",

"guzzlehttp/guzzle": "^7.2",

"laravel/framework": "^10.10",

"laravel/sanctum": "^3.3",

"laravel/tinker": "^2.8"

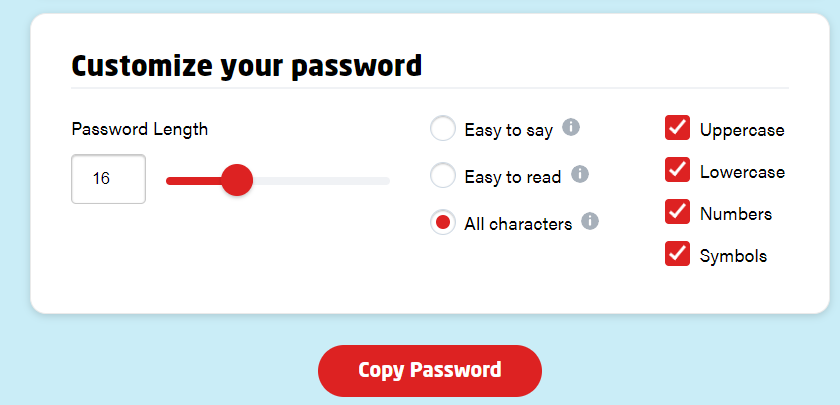
},

Commands that are to be inputted into the Ubuntu shell are highlighted in **$*grey***

# **Server Setup**

* Login to server as root user and change directory to the root home folder:
  + **$cd ~**
* We will now update the Ubuntu repositories and download the latest packages for the system:
  + ***$sudo apt update***
* Next we will install our dependencies to run our project on an Ubuntu server:
  + We will install MySQL and configure it using the official mysql\_secure\_installation script
    - ***$sudo apt install mysql-server***
  + The MySQL database software is installed, we can now configure it by running the following command
    - ***$sudo mysql\_secure\_installation***
    - You will be prompted to use the VALIDATE PASSWORD PLUGIN….for security purposes it is advisable to select “Y” for yes at this step  
      A black screen with white text

      Description automatically generated
    - The next prompt is to select the strength for validating the passwords. Select “1” for medium strength passwords  
      A screen shot of a black screen

      Description automatically generated
    - The next prompt is to enter the root password for MySQL. Follow the rules set on creating a medium strength password. It is advisable to use a password generator such as (<https://www.lastpass.com/features/password-generator>). Some sample settings for this would be:  
      
    - Save this password in a password repository such as KeePass
    - Once the password has been set the next prompt is to remove anonymous users. Select “Y” at this step  
      
    - The next prompt is to prevent root login outside of the server. For security purposes it is advisable to select “Y” for this  
      A black screen with white text

      Description automatically generated
    - The next prompt is to remove the test database. We will not be using it so select “Y” for this step
    - Then finally we are prompted to reload the database privileges. Select “Y” for this step  
      A black background with white text

      Description automatically generated
    - The MySQL database is now setup
  + We will now install PHP 8 and the necessary Laravel PHP modules using the following commands
    - ***$ sudo apt install software-properties-common apt-transport-https -y***
    - ***$ sudo apt install -y curl wget gnupg2 ca-certificates lsb-release apt-transport-https***
    - ***$ sudo add-apt-repository ppa:ondrej/php -y***
    - ***$ sudo apt update***
    - ***$ sudo apt install php-fpm php-common php-mysql php-xml php-xmlrpc php-curl php-gd php-imagick php-cli php-dev php-imap php-mbstring php-opcache php-soap php-zip php-intl php-bcmath***
    - Select “Y” on all prompts
    - PHP 8 and necessary Laravel PHP modules are now installed. We can confirm the version of PHP by typing:
    - **$php -v**
  + We will now install composer the PHP package manager which is used to install and update Laravel projects. And we will also install GIT which will allow us to pull the project code.
    - **$cd ~**
    - ***$ sudo apt install git***
    - ***$sudo apt install curl***
    - ***$sudo apt install php-curl***
    - ***$sudo curl -sS*** [***https://getcomposer.org/installer -o composer-setup.php***](https://getcomposer.org/installer%20-o%20composer-setup.php)
    - ***$sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer***
    - ***$sudo composer self-update***

# **Project installation**

* Our server is now setup. We are ready to setup our Laravel support ticket system
  + As we will be serving this project using “php artisan serve” we did not install Apache. As such we do not have a WWW folder and should create one:
    - ***$sudo mkdir /var/www/***
    - ***$cd /var/www/***
  + We can now clone the project from GitHub
    - ***$ sudo git clone*** ***https://github.com/Abigail96-github/support\_tickets.git***
    - ***$cd support\_tickets***
  + We are now going to run the shell script to setup our projects database and its associated user
    - ***$cd project\_install***
    - ***$ sudo chmod +x database\_setup.sh***
    - ***$ sudo ./database\_setup.sh***
  + If everything ran successfully you should get the message: The 'supportTickets' database has been created along with its user 'laravel\_support\_tickets'
  + We can now change directory back to the root directory of the project
    - ***$cd ../***
  + We are now going to use composer to set the project up
    - ***$sudo composer update***
    - Because we are running the command as root user we will be prompted to confirm running composer as root. Respond with “yes”  
      
  + We will now setup our .env file
    - A screenshot of a computer

      Description automatically generated
  + The next step is to migrate and seed the database.
    - Run Migration:
      * Run the command ***php artisan migrate*** to create a database table.
      * Additionally, run the command ***php artisan make:migration create\_tickets\_table***
      * Then run the above created table with this command ***php artisan migrate --path=/database/migrations/2020\_04\_01\_064006\_create\_tickets\_table.php***
    - Seeder to populate the users table
      * I have already created the user seeder.
      * Run the command ***composer dump-autoload*** to regenerate the seeder files.
      * Run the command ***php artisan db:seed --class=UserSeeder*** to execute that single seeder.
  + We can now serve the project
    - I suggest to serve the project on an obscure port such as 9090 eg: ***“php artisan serve –port=9090”***
    - Because I had set this project up in WSL2 I was able to browse the project on my Windows 11 PC using the URL (http://localhost:9090)