Define the project

1**.Create an empty project directory.**

Name the directory something easy and in our case, we name it **‘ ICT\_Project\_Final.’** This directory is the context for your application image. The directory should only contain resources to build that image.

2. **Change into your project directory.**

For example,

$ cd ICT\_Project\_Final/

3. **Create a docker-compose.yml file**

version: '3.8'  
services:  
 php:  
 build:  
 context:  
 dockerfile: Dockerfile  
 ports:  
 - 81:80  
 volumes:  
 - ./src:/var/www/html/  
 env\_file:  
 - .env  
 networks:  
 - lan  
 db: *# mysql database server*  
image: mysql:latest  
 restart: always *#this restarts the service upon any configuration changes*  
environment: *# to use and access MySQL server,we neet to set up an authentication environment*  
MYSQL\_ROOT\_PASSWORD: password  
 MYSQL\_DATABASE: wordpress  
 MYSQL\_USER: wordpress  
 MYSQL\_PASSWORD: wordpress

ports:  
 - "6034:3306" *#6034 in the local machine and 3306 in the container*  
volumes: *# it persists updates made to the database*  
 *# - db\_data:/var/lib/mysql*  
- ./db:/var/lib/mysql # this file was created to enable easy sharing of db  
 networks:  
 - lan  
  
 phpmyadmin: *#another service to manage database server*  
depends\_on:  
 - db  
 image: phpmyadmin/phpmyadmin  
 restart: always  
 ports:  
 - '82:80'  
 environment:  
 PMA\_HOST: db  
 networks:  
 - lan  
  
 *# Wordpress*  
wordpress:  
 depends\_on:  
 - db  
 image: wordpress:latest  
 ports:  
 - '83:80'  
 restart: always  
 volumes:  
 - ./wordpress:/var/www/html  
 environment:  
 WORDPRESS\_DB\_HOST: db:3306  
 WORDPRESS\_DB\_USER: wordpress  
 WORDPRESS\_DB\_PASSWORD: wordpress  
 MYSQL\_DATABASE: wordpress

networks:

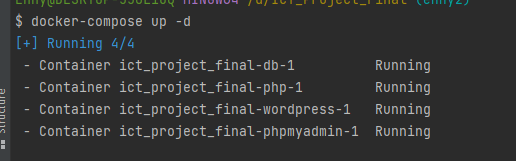
- lan  
volumes:  
 dbdata:

networks:  
 lan:

4.**Build the project**

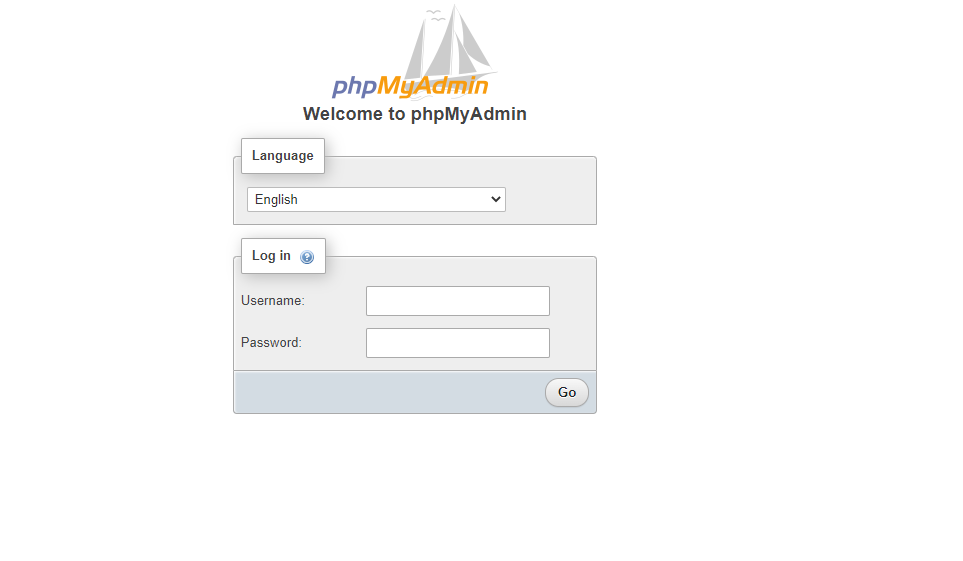
Run, docker-compose up –d from the project directory.

This runs docker-compose up in detached mode, pulls the needed Docker images, and starts the wordpress and database containers as requested.



**5. Log into PhpMyAdmin**

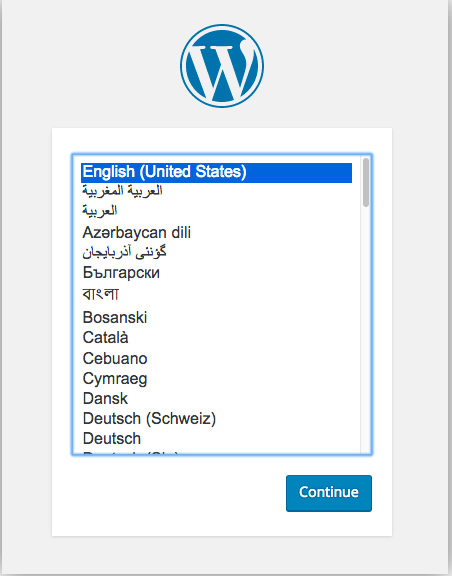
Check yml file for details

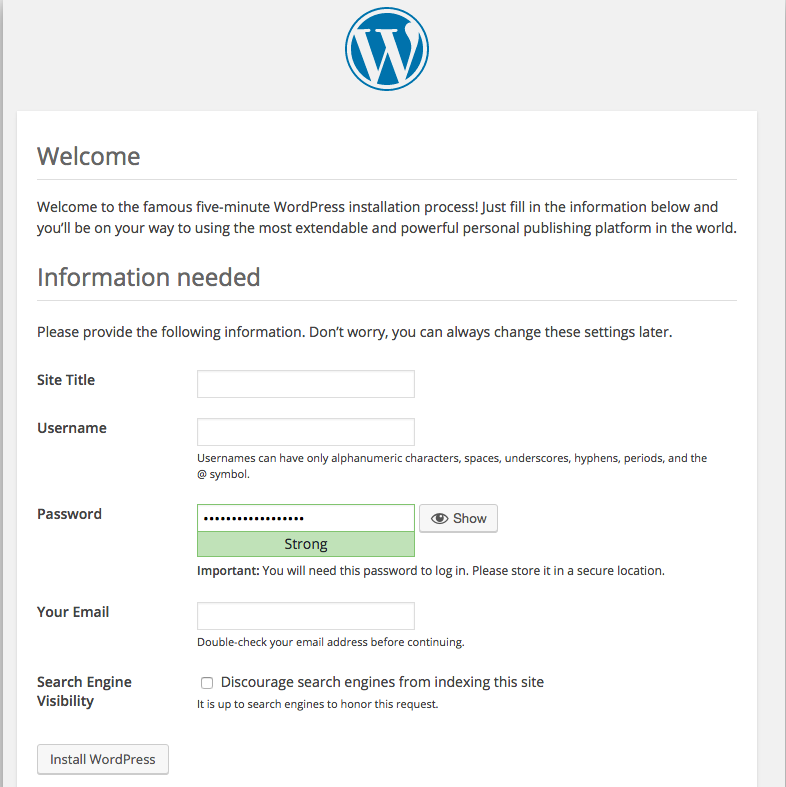


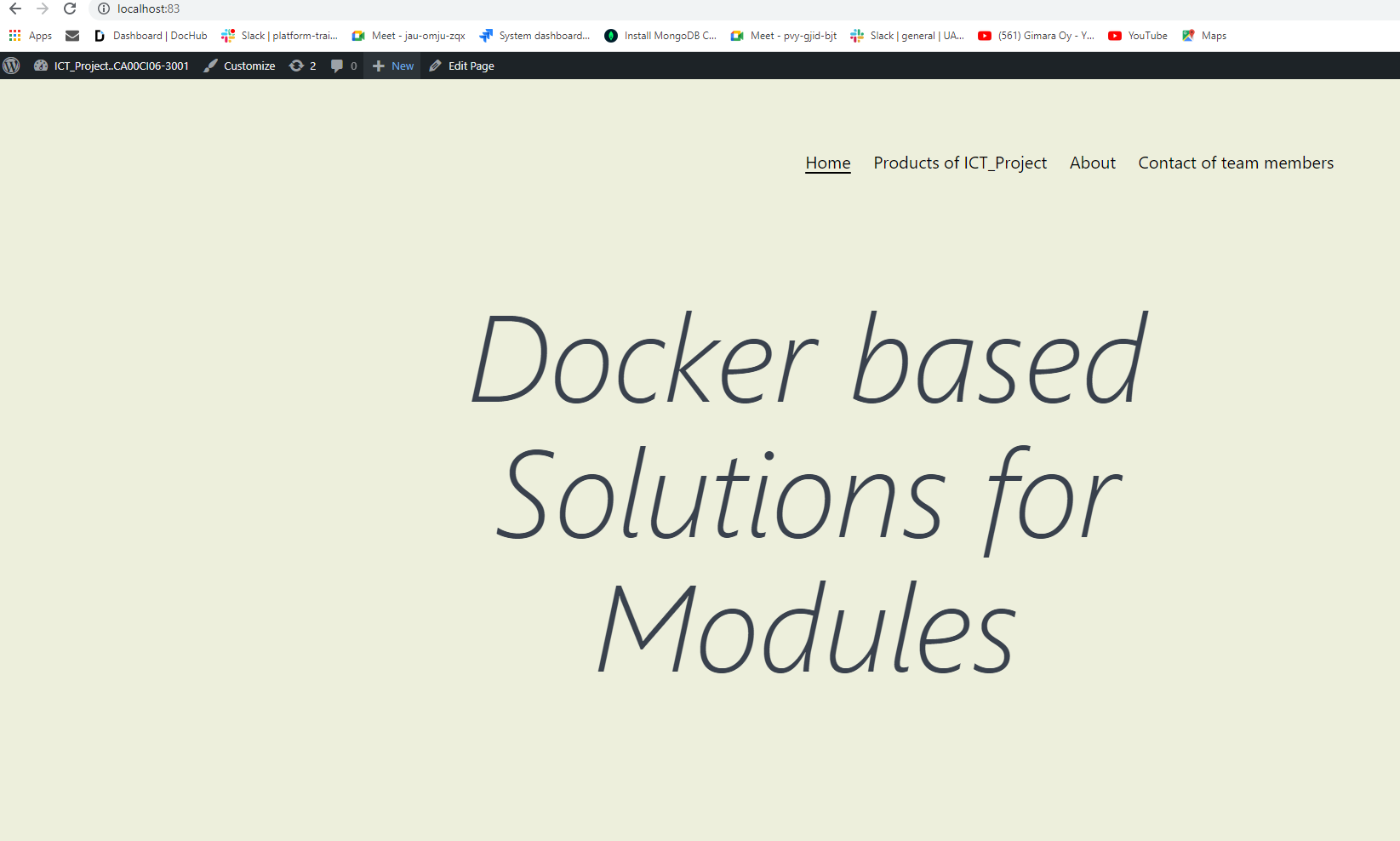
### 6. **Bring up WordPress in a web browser**

The project is running on port 83 of our docker host, so let's complete the installation of wordPress.

If Docker Desktop for Mac or Docker Desktop for Windows is in use, you can use http://localhost as the IP address, and open http:// http://localhost:83/ in a web browser.





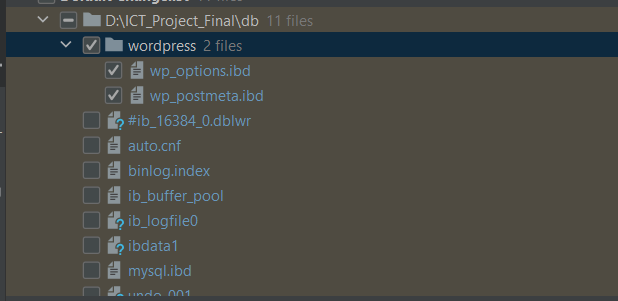


7. Create Repository on GitHub.

* + 1. In the upper-right corner of any page, use the drop-down menu, and select **a new repository**.
    2. Type a short, memorable name for your repository. For example, "ICT\_Project\_Final ".
    3. Choose repository visibility. For more information, see "[About repositories](https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories#about-repository-visibility)."
    4. Select **Initialize this repository with a README**.
    5. Click **Create repository**.

8. Design Wordpress

* + Make changes on wordpress Page to reflect what you have in mind.
  + Save it on Code editor, commit and push with comment to reflect changes.
  + Push to GitHub branch the wordpress files **only**



* + Other team members update the project.
  + Merge the incoming files into the current branch in the code editor.
  + Shut down docker with **docker-compose down.**
  + Restart docker with **docker-compose up –d.**
  + Check browser site for update.

Reference

<https://docs.docker.com/samples/wordpress/#define-the-project>

https://docs.github.com/en/get-started/quickstart/create-a-repo