

# Assignment #2 (Docker)

Group member Name:

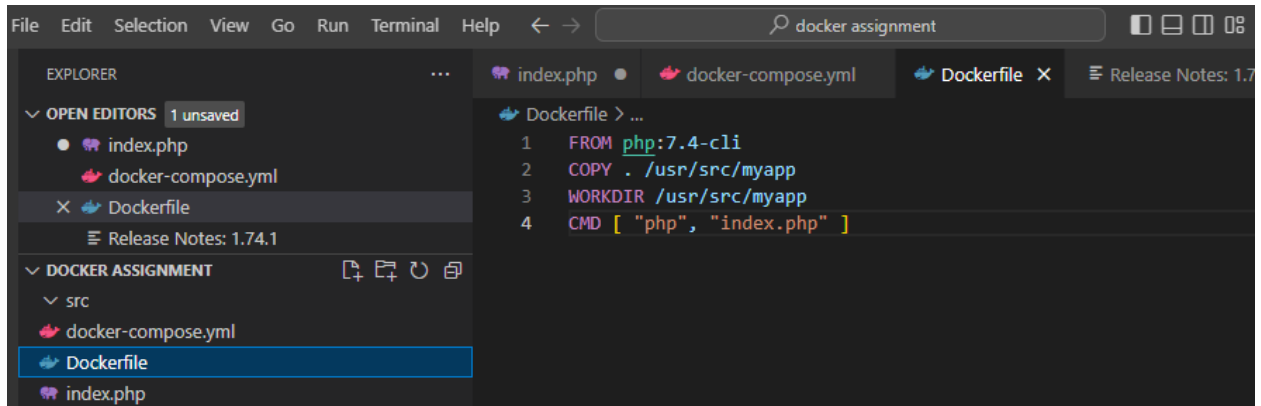
P19-1652 Hifza Majeed

P19-1664 Noman

P19-1672 Ahmad

Create a folder (docker assignment)

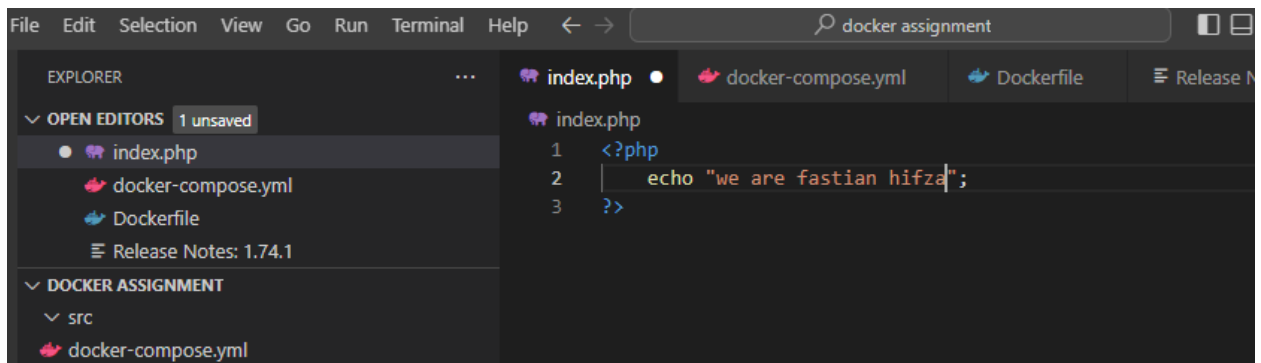
create a Dockerfile



The screenshot shows the Visual Studio Code interface with a project named 'docker assignment'. The Explorer sidebar on the left shows the file structure: 'OPEN EDITORS' (1 unsaved) containing 'index.php', 'docker-compose.yml', and 'Dockerfile'; and 'DOCKER ASSIGNMENT' containing a 'src' folder with 'docker-compose.yml' and 'Dockerfile'. The Dockerfile is selected and its content is visible in the editor: 

```
1 FROM php:7.4-cli
2 COPY . /usr/src/myapp
3 WORKDIR /usr/src/myapp
4 CMD [ "php", "index.php" ]
```

Create a index.php file



The screenshot shows the Visual Studio Code interface with the same project. The Explorer sidebar now shows 'index.php' as an additional file in the 'OPEN EDITORS' list. The 'index.php' file is selected and its content is visible in the editor: 

```
1 <?php
2 |     echo "we are fastian hifza";
3 ?>
```

## Run this command to build and run the Docker image

```
C:\Users\92325\Desktop\docker assignment>docker build -t my-php-app .
[+] Building 2.9s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 31B                                              0.0s
=> [internal] load .dockerignore                                                0.1s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/php:7.4-cli                  2.3s
=> [internal] load build context                                                0.1s
=> => transferring context: 59B                                                 0.0s
=> [1/3] FROM docker.io/library/php:7.4-cli@sha256:620a6b9f4d4feef2210026172570465e9d0c1de79766418d3affd09190a7f 0.0s
=> CACHED [2/3] COPY . /usr/src/myapp                                           0.0s
=> CACHED [3/3] WORKDIR /usr/src/myapp                                          0.0s
=> exporting to image                                                         0.1s
=> => exporting layers                                                         0.0s
=> => writing image sha256:66c9fa99d75568c04cd5ca0f278b4b7a7ca923402fa1ac8a510ed115f5164f6a 0.0s
=> => naming to docker.io/library/my-php-app                                   0.0s












Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

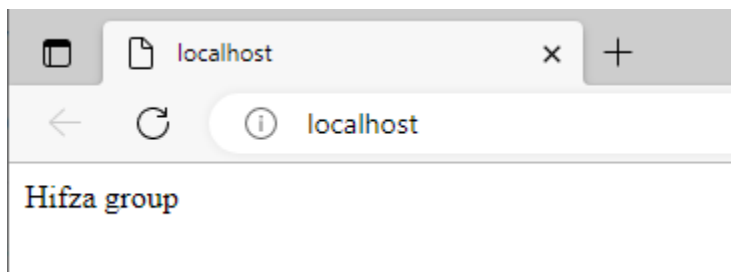
```
C:\Users\92325\Desktop\docker assignment>docker run -it --rm --name my-running-app my-php-app
we are fastian
```

## Apache without a Dockerfile

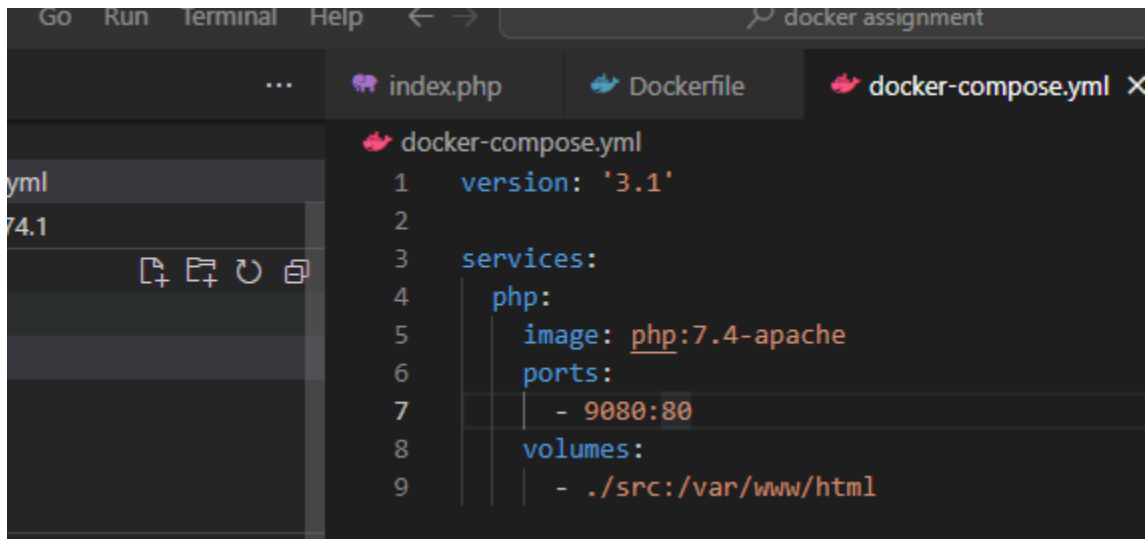
```
C:\Users\92325\Desktop\docker assignment>docker run -d -p 80:80 --name my-apache-php-app -v "C:\Users\92325\Desktop\docker assignment":/var/www/html php:7.2-apache
a58e405f612e4a168296d432d65373c5377d21be844c23dc6acfb0ccc3522ace
```

## Docker images

<input type="checkbox"/>	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	 <b>my-running-app</b> 279fe6c4c56b 	<a href="#">my-php-app:latest</a>	Exited			  
<input type="checkbox"/>	 <b>my-apache-php-app</b> 10983ee02bd8 	<a href="#">php:7.2-apache</a>	Running	<a href="#">80:80</a> 	45 seconds ago	  

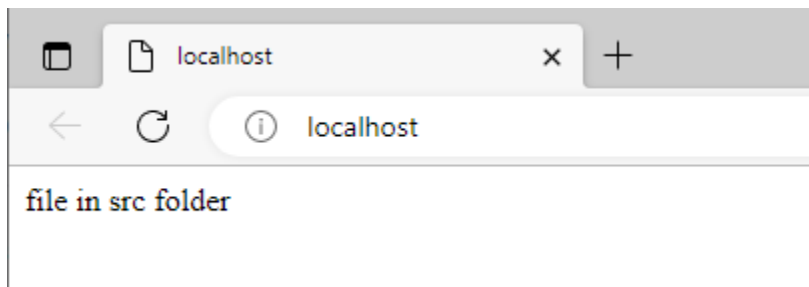


Create a docker-compose.yml file

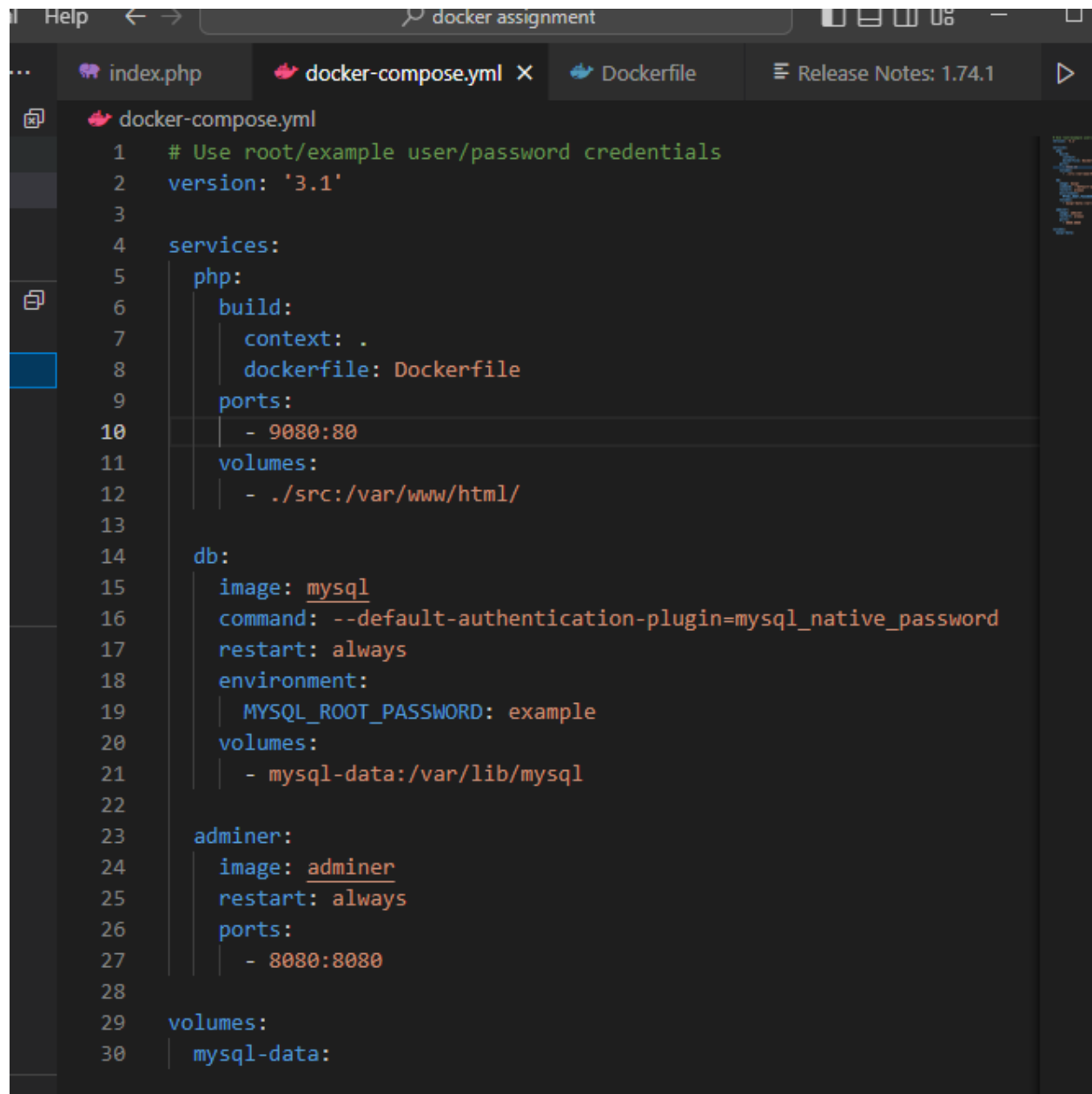


The screenshot shows a code editor with three tabs: 'index.php', 'Dockerfile', and 'docker-compose.yml'. The 'docker-compose.yml' tab is active, displaying the following content:

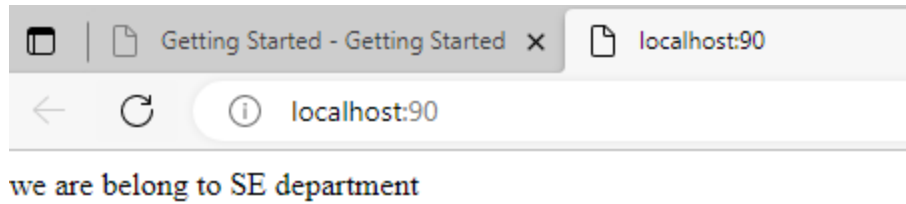
```
1  version: '3.1'
2
3  services:
4    php:
5      image: php:7.4-apache
6      ports:
7        - 9080:80
8      volumes:
9        - ./src:/var/www/html
```



Put the whole code in the docker-compose.yml file for database and php file running







```
1 # Use root/example user/password credentials
2 version: '3.1'
3
4 services:
5   php:
6     build:
7       context: .
8       dockerfile: Dockerfile
9     ports:
10      - 9080:80
11     volumes:
12      - ./src:/var/www/html/
13
14   db:
15     image: mysql
16     command: --default-authentication-plugin=mysql_native_password
17     restart: always
18     environment:
19       MYSQL_ROOT_PASSWORD: example
20     volumes:
21       - mysql-data:/var/lib/mysql
22
23   adminer:
24     image: adminer
25     restart: always
26     ports:
27       - 8080:8080
28
29 volumes:
30   mysql-data:
```



Run the command to create a container

```
C:\Users\92325\Desktop\docker assignment>  
C:\Users\92325\Desktop\docker assignment>docker-compose up -d
```

<input type="checkbox"/>		dockeraassignment	-	Running (3/3)			
<input type="checkbox"/>		php-1 f14fa0077548	<a href="#">php:7.4-apache</a>	Running	<a href="#">9080:80</a>	16 minutes ago	
<input type="checkbox"/>		adminer-1 77bad17a5b49	<a href="#">adminer:latest</a>	Running	<a href="#">8080:8080</a>	16 minutes ago	
<input type="checkbox"/>		db-1 ed04e627975f	<a href="#">mysql:latest</a>	Running		16 minutes ago	

Password is example

A screenshot of the Adminer 4.8.1 login page. The page has a light blue header with 'Adminer 4.8.1' and a 'Login' button. Below the header, there is a 'Language' dropdown set to 'English'. The main content area contains a login form with fields for 'System' (MySQL), 'Server' (db), 'Username' (root), 'Password' (masked with dots), and 'Database'. There is also a 'Permanent login' checkbox and a 'Login' button.

Language: English

**Adminer 4.8.1**

(MySQL) root@db - NAH  
(MySQL) root@db

System	MySQL
Server	db
Username	root
Password	.....
Database	

☐ Permanent login

localhost localhost:9080 Select database - db - Adminer

localhost:8080/?server=db&username=root

Language: English MySQL » db

**Adminer 4.8.1**

DB:

[SQL command](#) [Import](#)  
[Export](#)

**Select database**

[Create database](#) [Privileges](#) [Process list](#) [Variables](#) [Status](#)

MySQL version: **8.0.31** through PHP extension **MySQLi**

Logged as: **root@172.18.0.3**

	Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/>	<a href="#">information_schema</a>	utf8mb3_general_ci	?	?
<input type="checkbox"/>	<a href="#">mysql</a>	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	<a href="#">performance_schema</a>	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	<a href="#">sys</a>	utf8mb4_0900_ai_ci	?	?

Selected (0)

## Create a database

localhost localhost:9080 Database: NomiHifzaAhmad - db

localhost:8080/?server=db&username=root&db=NomiHifzaAhmad

Language: English MySQL » db » Database: NomiHifzaAhmad

**Adminer 4.8.1**

DB:

[SQL command](#) [Import](#)  
[Export](#) [Create table](#)

**Database: NomiHifzaAhmad**

Database has been created. 15:35:48 [SQL command](#)

[Alter database](#) [Database schema](#) [Privileges](#)

**Tables and views**

No tables.

[Create table](#) [Create view](#)

**Routines**

[Create procedure](#) [Create function](#)

**Events**

## Create a table

localhost

localhost:9080

Table: std\_info - db - Adminer

localhost:8080/?server=db&username=root&db=NomiHifzaAhmad&table=std\_info

Language: English

MySQL » db » NomiHifzaAhmad » Table: std\_info

Adminer 4.8.1

DB: NomiHifzaAhmad

[SQL command](#) [Import](#)  
[Export](#) [Create table](#)

[select std\\_info](#)

Table: std\_info

Table has been created. 15:38:53 [SQL command](#)

[Select data](#) **Show structure** [Alter table](#) [New item](#)

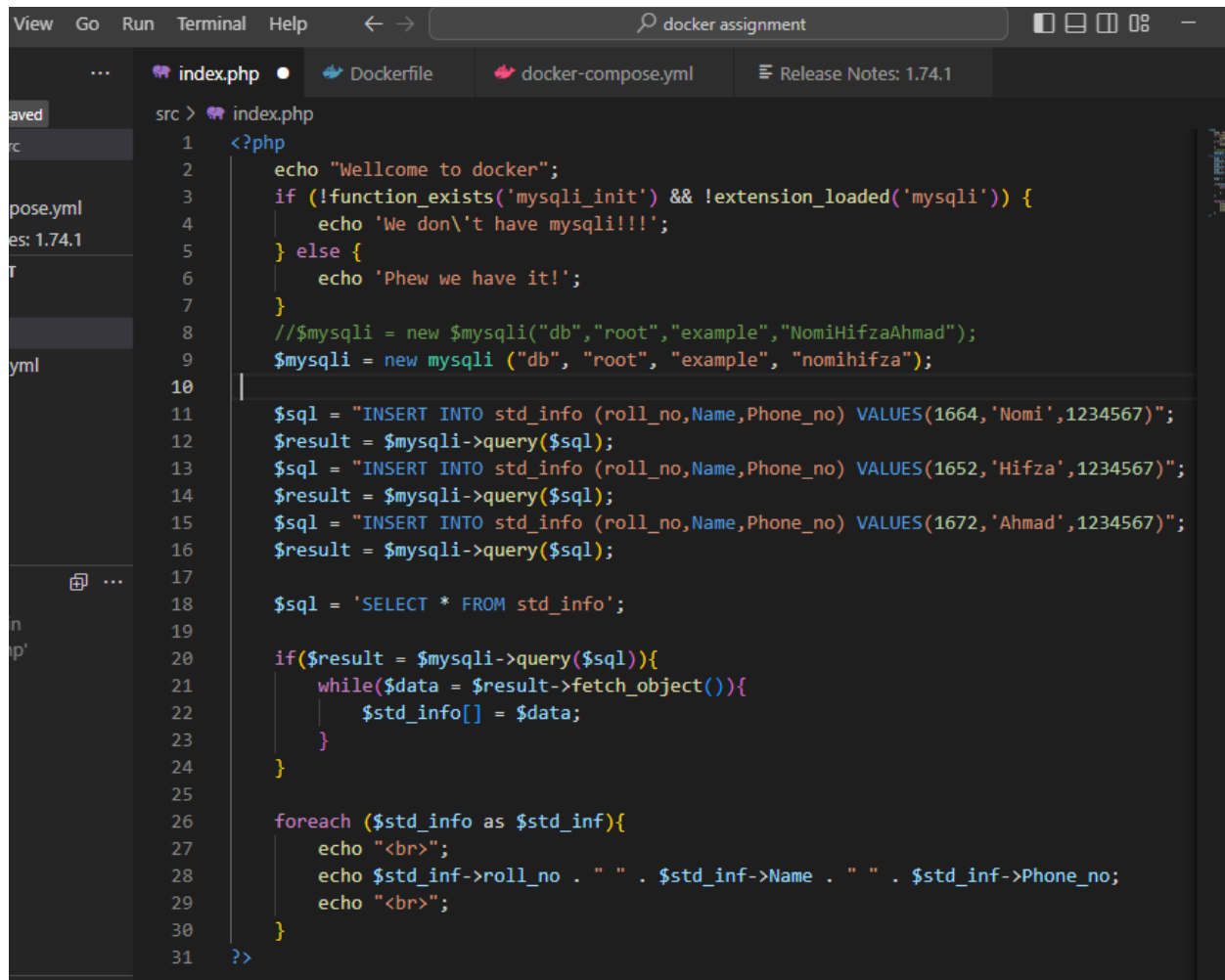
Column	Type	Comment
roll no	int	
Name	varchar(225)	
phone_no	int	

Indexes

[Alter indexes](#)

Foreign keys

Fill the data in to table and connect the php to db



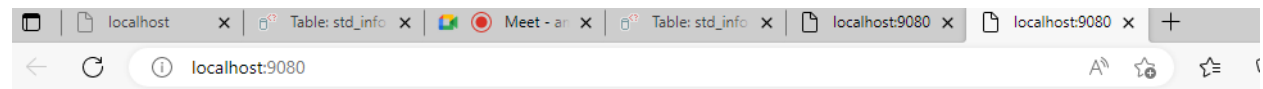
```
View Go Run Terminal Help docker assignment
... index.php Dockerfile docker-compose.yml Release Notes: 1.74.1
src > index.php
1 <?php
2 echo "Wellcome to docker";
3 if (!function_exists('mysqli_init') && !extension_loaded('mysqli')) {
4     echo 'We don\'t have mysqli!!!';
5 } else {
6     echo 'Phew we have it!';
7 }
8 // $mysqli = new mysqli("db","root","example","NomiHifzaAhmad");
9 $mysqli = new mysqli ("db", "root", "example", "nomihifza");
10
11 $sql = "INSERT INTO std_info (roll_no,Name,Phone_no) VALUES(1664,'Nomi',1234567)";
12 $result = $mysqli->query($sql);
13 $sql = "INSERT INTO std_info (roll_no,Name,Phone_no) VALUES(1652,'Hifza',1234567)";
14 $result = $mysqli->query($sql);
15 $sql = "INSERT INTO std_info (roll_no,Name,Phone_no) VALUES(1672,'Ahmad',1234567)";
16 $result = $mysqli->query($sql);
17
18 $sql = 'SELECT * FROM std_info';
19
20 if($result = $mysqli->query($sql)){
21     while($data = $result->fetch_object()){
22         $std_info[] = $data;
23     }
24 }
25
26 foreach ($std_info as $std_inf){
27     echo "<br>";
28     echo $std_inf->roll_no . " " . $std_inf->Name . " " . $std_inf->Phone_no;
29     echo "<br>";
30 }
31 ?>
```

Go to the container and go to the php terminal and run this command

docker-php-ext-install mysqli

docker-php-ext-enable mysqli





Wellcome to dockerPhew we have it!

1664 Nomi

1652 Hifza

1672 Ahmad