## Debugging the Application Locally

To debug the web application locally, we need to install the [PHP Debug](https://marketplace.visualstudio.com/items?itemName=xdebug.php-debug) VS Code extension. Afterward, we can open the file index.php in VS Code, click Debug and Run, and select the PHP Debugger. However, doing so results in an error message. In the debug console, we can see the following error:

Authentication Bypass

PHP Fatal error: Uncaught mysqli\_sql\_exception: Connection refused in src/config.php:8 Stack trace: #0 src/config.php(8): mysqli\_connect('127.0.0.1', 'db', Object(SensitiveParameterValue), 'db') #1 src/index.php(2): require\_once('...') #2 {main} thrown in src/config.php on line 8  
  
Fatal error: Uncaught mysqli\_sql\_exception: Connection refused in src/config.php:8 Stack trace: #0 src/config.php(8): mysqli\_connect('127.0.0.1', 'db', Object(SensitiveParameterValue), 'db') #1 src/index.php(2): require\_once('...') #2 {main} thrown in src/config.php on line 8

Looking at the file config.php referenced in the error message it contains the following code:

Code: php

<?php  
  
$servername="127.0.0.1";  
$dbusername="db";  
$password="db-password";  
$dBName="db";  
  
$conn = mysqli\_connect($servername, $dbusername, $password, $dBName);

The web application attempts to connect to a MySQL instance on localhost, which is currently not running. Instead of installing a MySQL server on our local machine, we can use a [MySQL Docker](https://hub.docker.com/_/mysql) container. To match the parameters provided in config.php, we can start the docker container using the following parameters:

Authentication Bypass

yovecio@htb[/htb]$ docker run -p 3306:3306 -e MYSQL\_USER='db' -e MYSQL\_PASSWORD='db-password' -e MYSQL\_DATABASE='db' -e MYSQL\_ROOT\_PASSWORD='db' mysql

This creates a new MySQL server with the credentials given in config.php. However, the database is empty. So, let us create a users table with a dummy user. To do so, we need to create a file called db.sql with the following contents:

Code: sql

CREATE TABLE `users` (  
 `id` int(11) NOT NULL,  
 `username` varchar(256) NOT NULL,  
 `password` varchar(256) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
#htb-stdnt:Academy\_student!  
INSERT INTO `users` (`id`, `username`, `password`) VALUES  
(1, "htb-stdnt", "44891a5fc2dad49eab817badff4cb98adec418e43e6c6cb39984f8d090c6b0c4");

Afterward, kill the docker container we started previously and start a new one with the following command from the directory containing the db.sql file:

Authentication Bypass

yovecio@htb[/htb]$ docker run -p 3306:3306 -e MYSQL\_USER='db' -e MYSQL\_PASSWORD='db-password' -e MYSQL\_DATABASE='db' -e MYSQL\_ROOT\_PASSWORD='db' --mount type=bind,source="$(pwd)/db.sql",target=/docker-entrypoint-initdb.d/db.sql mysql

Afterward, we can run the web application using PHP's built-in web server by clicking on Create new launch.json and selecting the Launch Built-in web server debugger in the drop-down menu on the left. Then, we can access the web application at the URL printed in the debug console.

Note: Keep in mind that the behavior of type juggling differs depending on the PHP version. Thus, we need to ensure that our local PHP version matches the PHP version used by the target web server.