**Programming in PHP**

**Lab05**



**Spring 2025**

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”



Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Summeya Salahudin**

Month Day, Year (13 05, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**OBJECTIVES OF THE LAB:**

**This lab aims at the understanding of:**

* + **PHP Basics**
  + **PHP Data Types**
  + **PHP Expressions**
  + **PHP Operators**
  + **PHP Conditionals**
  + **PHP Loops**

**TASK 5.1:**

In contrast to settype(), there is another method that causes a variable’s value to be treated as

a specific type. It is known as type casting. Note that the variable itself remains unaffected

during type casting. Consider the following variable:

$test\_var = 8.23;

Type cast this variable’s value to integer, string, and Boolean and show result using echo.

**CODE:**//type-casting

<?php

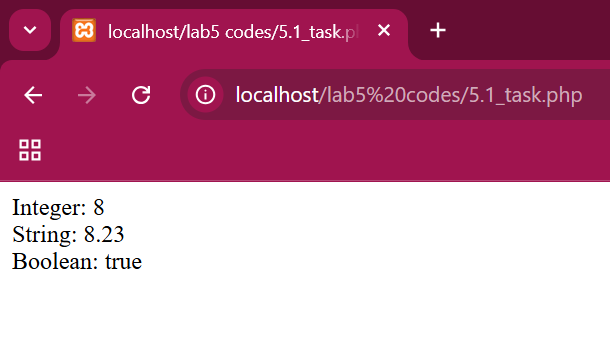
$test\_var = 8.23;

echo "Integer: " . (int)$test\_var . "<br>";

echo "String: " . (string)$test\_var . "<br>";

echo "Boolean: " . ((bool)$test\_var ? 'true' : 'false') . "<br>";

?>

**OUTPUT:  
**

**Task 5.2:**Use and Specify the purpose of following functions:

1) intval( value )

2) floatval( value )

3) strval( value )

**CODE:**<?php

// Purpose: Converts a value to an integer.

$x = "15.8";

echo "Integer: " . intval($x) . "<br>"; // Output: 15

// Purpose: Converts a value to a floating-point number (float).

$x = "15.8";

echo "Float: " . floatval($x) . "<br>"; // Output: 15.8

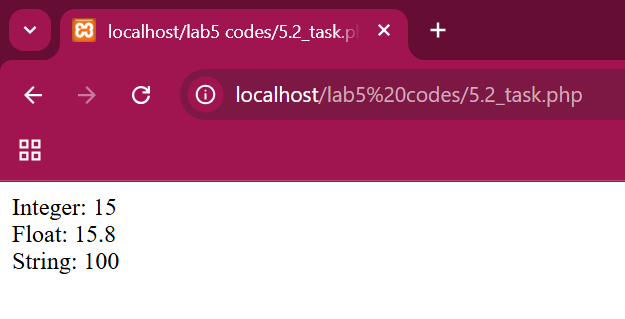
// Purpose: Converts a value to a string.

$x = 100;

echo "String: " . strval($x) . "<br>"; // Output: "100"

?>

**Output:**

****

**Task 5.3:**

Write PHP script that shows the division table displayed as in Table 5.2 using different loops. For each number, display whether that number is an odd or even number, and also display a message if the number is a prime number. Display this information within an HTML table.

**CODE:**<?php

function is\_prime($num) {

    if ($num < 2) return false;

    for ($i = 2; $i \* $i <= $num; $i++) {

        if ($num % $i == 0) return false;

    }

    return true;

}

echo "<table border='1' cellpadding='5' cellspacing='0'>";

echo "<tr><th></th>";

for ($col = 1; $col <= 10; $col++) {

    echo "<th>" . $col . "</th>";

}

echo "</tr>";

for ($row = 1; $row <= 10; $row++) {

    echo "<tr><th>" . $row . "</th>";

    for ($col = 1; $col <= 10; $col++) {

        $result = number\_format($row / $col, 3);

        $number\_type = ($row % 2 == 0) ? "Even" : "Odd";

        $prime\_info = is\_prime($row) ? "Prime" : "";

        echo "<td>" . $result . "<br>(" . $number\_type . ($prime\_info ? ", " . $prime\_info : "") . ")</td>";

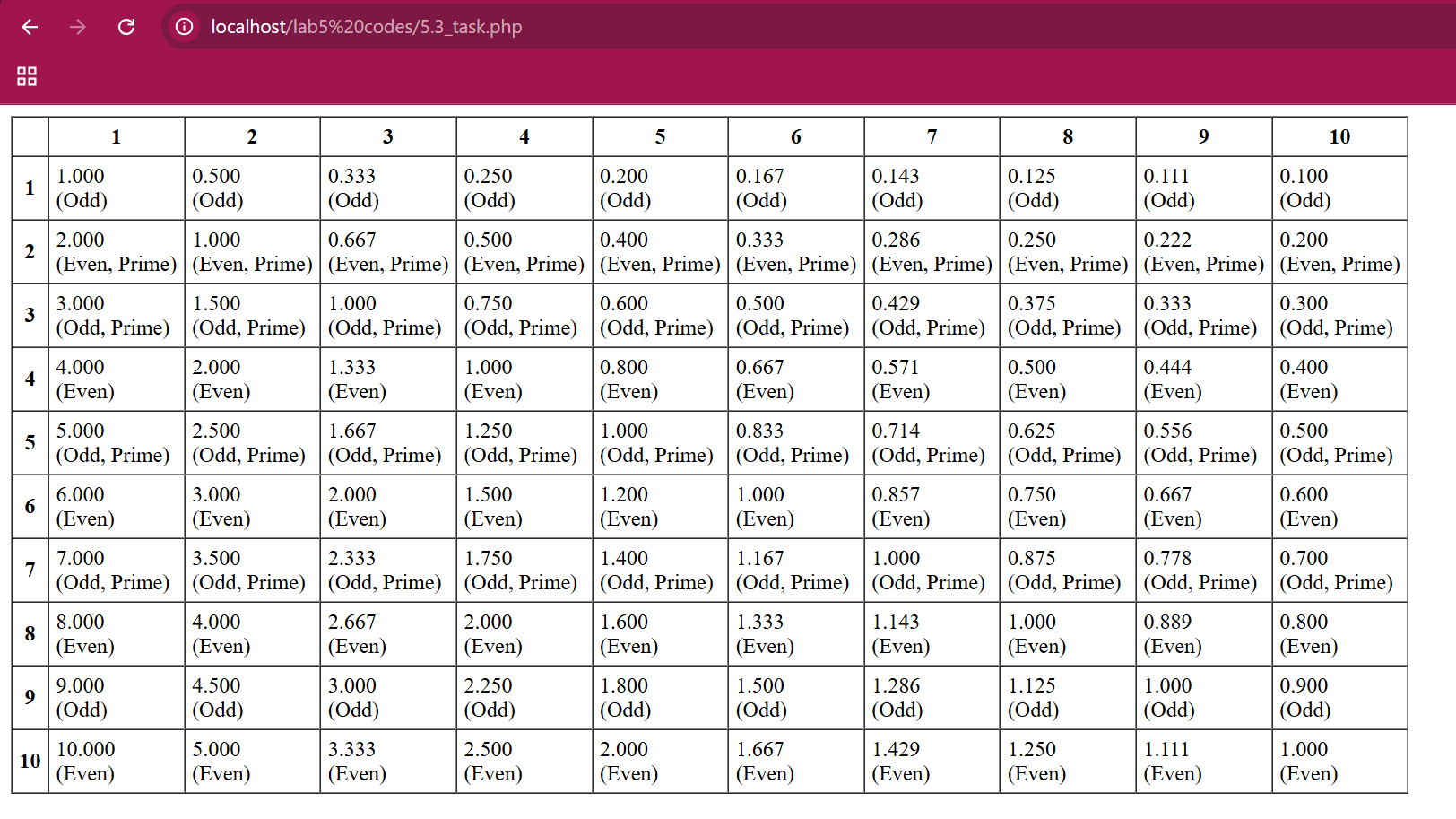
    }

    echo "</tr>";

}

echo "</table>";

?>

**OUTPUT:  
**

**Task 5.4:**Explore PHP Object Oriented using examples showing classes, objects, inheritance, and polymorphism.

**Classes and Objects:**

* A class is a blueprint for objects.
* An object is an instance of a class.

<?php

// Define a class named Car

class Car {

    public $brand;

    public $model;

    // Constructor to initialize properties

    public function \_\_construct($brand, $model) {

        $this->brand = $brand;

        $this->model = $model;

    }

    // Method to display car info

    public function displayInfo() {

        echo "Brand: " . $this->brand . ", Model: " . $this->model . "<br>";

    }

}

// Create objects (instances) of the Car class

$car1 = new Car("Toyota", "Camry");

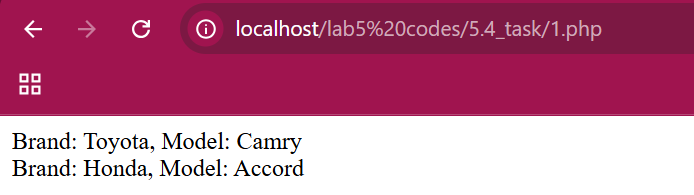
$car2 = new Car("Honda", "Accord");

// Call the method

$car1->displayInfo();

$car2->displayInfo();

?>

**OUTPUT:  
**

**Inheritance:**

* Inheritance allows a class to inherit properties and methods from another class.
* Use the extends keyword.

**CODE:**<?php

// Base class (Parent)

class Animal {

    public $name;

    public function \_\_construct($name) {

        $this->name = $name;

    }

    public function sound() {

        echo "Some generic sound<br>";

    }

}

// Derived class (Child) inheriting from Animal

class Dog extends Animal {

    // Overriding the sound method

    public function sound() {

        echo $this->name . " says: Woof!<br>";

    }

}

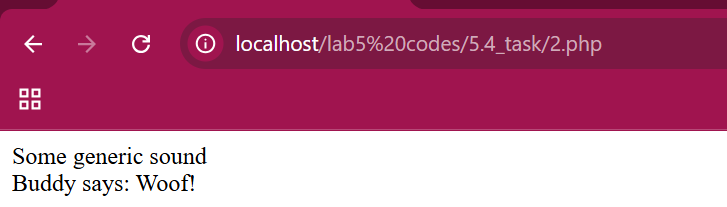
$genericAnimal = new Animal("Animal");

$genericAnimal->sound();

$dog = new Dog("Buddy");

$dog->sound();

?>

**OUTPUT:  
**

**Polymorphism:**

* Polymorphism allows methods to behave differently based on the object that is calling them.
* Achieved through method overriding.

**CODE:**

<?php

// Base class

class Shape {

    public function draw() {

        echo "Drawing a shape<br>";

    }

}

// Derived classes

class Circle extends Shape {

    public function draw() {

        echo "Drawing a circle<br>";

    }

}

class Square extends Shape {

    public function draw() {

        echo "Drawing a square<br>";

    }

}

// Polymorphism in action

function drawShape(Shape $shape) {

    $shape->draw();

}

$shape = new Shape();

$circle = new Circle();

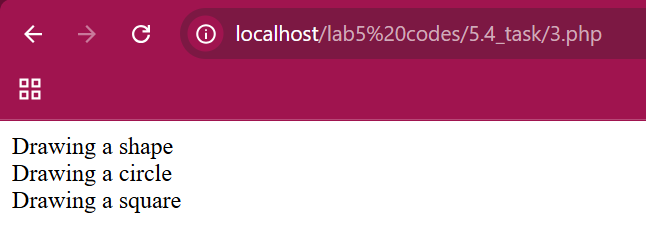
$square = new Square();

drawShape($shape);

drawShape($circle);

drawShape($square);

?>

**Output:  
**

**Task 5.5:**

Build an image gallery website using PHP without database. Your website must show all the images in a given directory. Use bootstrap/CSS in your website. Please note that when a given image is clicked, it enlarges it and shows in higher resolution. Also, your website should be flexible enough to show the images when number of images are increased or decreased from a given directory.

**CODE:**

<?php

function getImages($dir) {

    $images = array();

    if (is\_dir($dir)) {

        $files = scandir($dir);

        foreach ($files as $file) {

            if (in\_array(strtolower(pathinfo($file, PATHINFO\_EXTENSION)), ['jpg', 'jpeg', 'png', 'gif'])) {

                $images[] = $file;

            }

        }

    }

    return $images;

}

$imageDir = 'images';

$images = getImages($imageDir);

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Image Gallery</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        .gallery img { width: 100%; height: auto; border-radius: 8px; }

        .modal-img { width: 100%; height: auto; }

    </style>

</head>

<body>

<div class="container mt-4">

    <h2 class="text-center mb-4">Dynamic Image Gallery</h2>

    <div class="row gallery">

        <?php foreach ($images as $image): ?>

            <div class="col-md-3 mb-4">

                <img src="<?php echo $imageDir . '/' . $image; ?>" class="img-thumbnail" onclick="showImage('<?php echo $imageDir . '/' . $image; ?>')" />

            </div>

        <?php endforeach; ?>

    </div>

</div>

<!-- Modal to display enlarged image -->

<div id="imageModal" class="modal" tabindex="-1">

    <div class="modal-dialog modal-dialog-centered">

        <div class="modal-content">

            <div class="modal-body">

                <img id="modalImg" class="modal-img" />

            </div>

        </div>

    </div>

</div>

<script>

function showImage(src) {

    document.getElementById('modalImg').src = src;

    new bootstrap.Modal(document.getElementById('imageModal')).show();

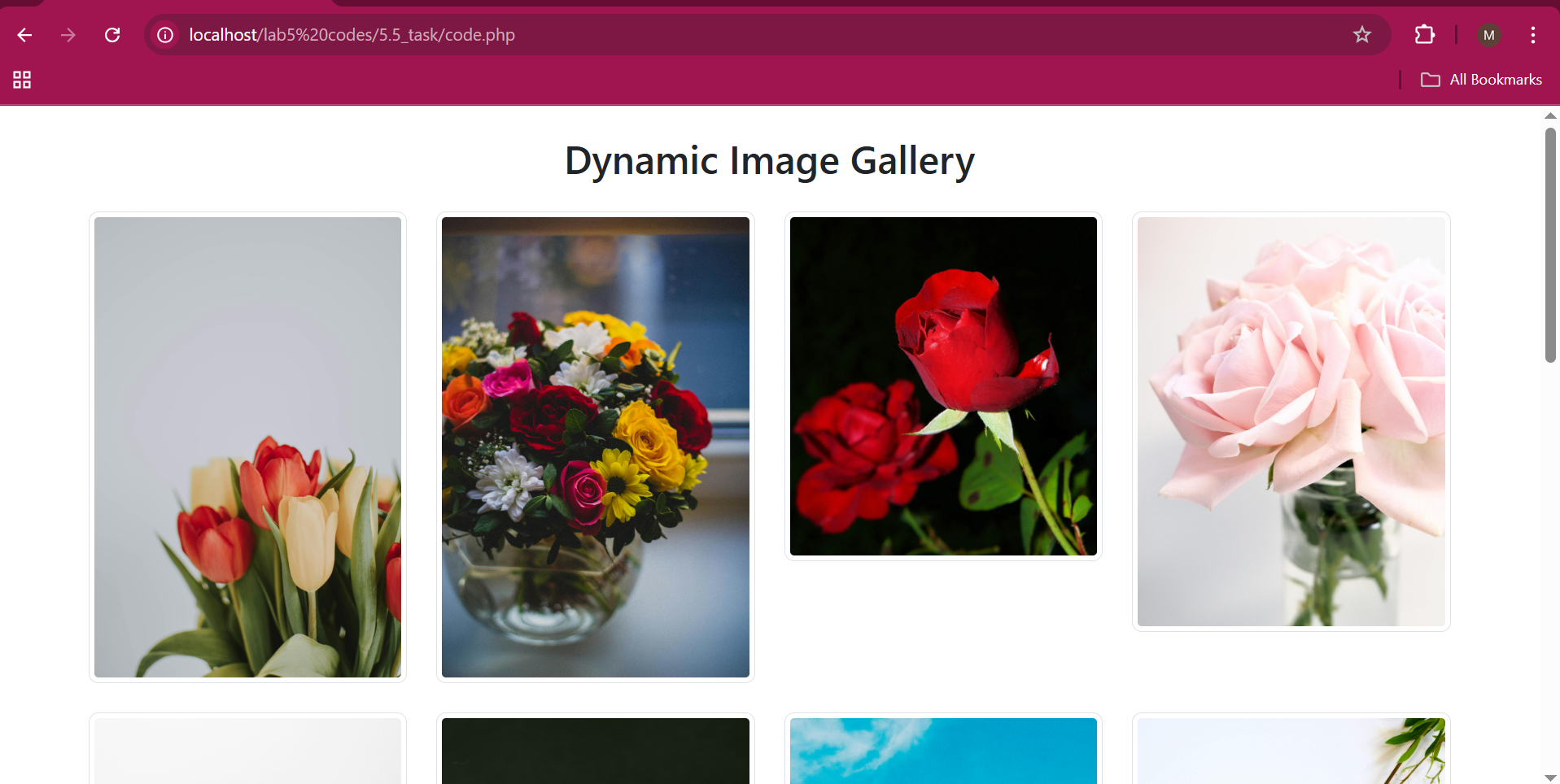
}

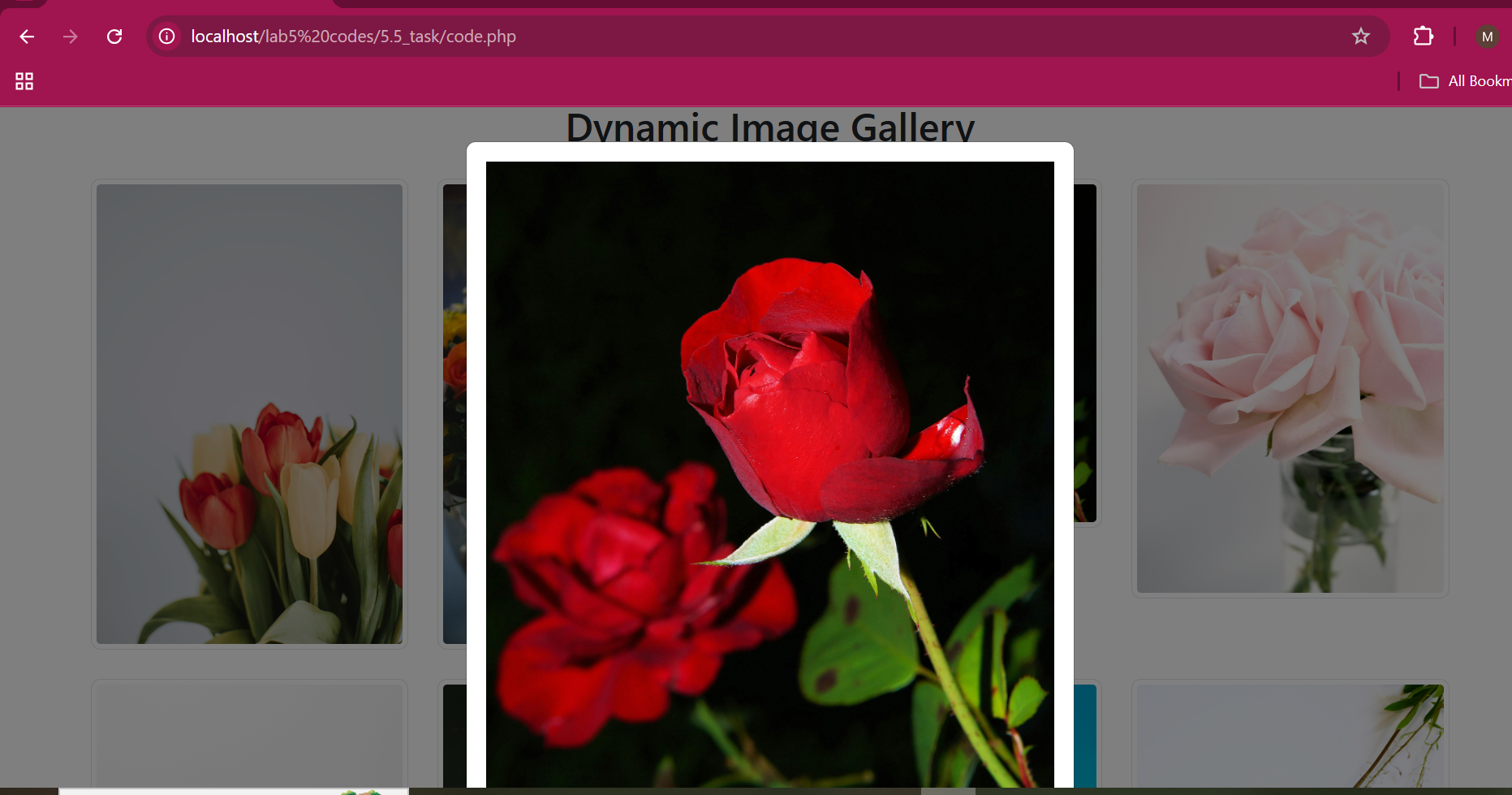
</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

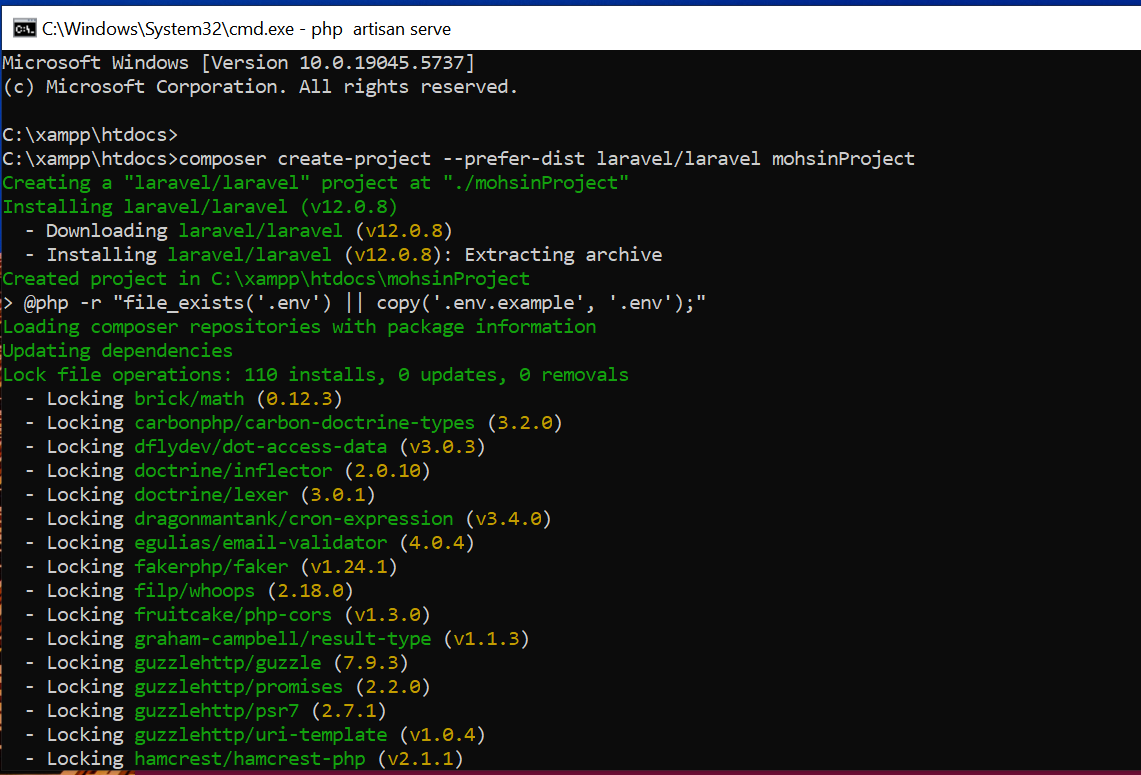
</body>

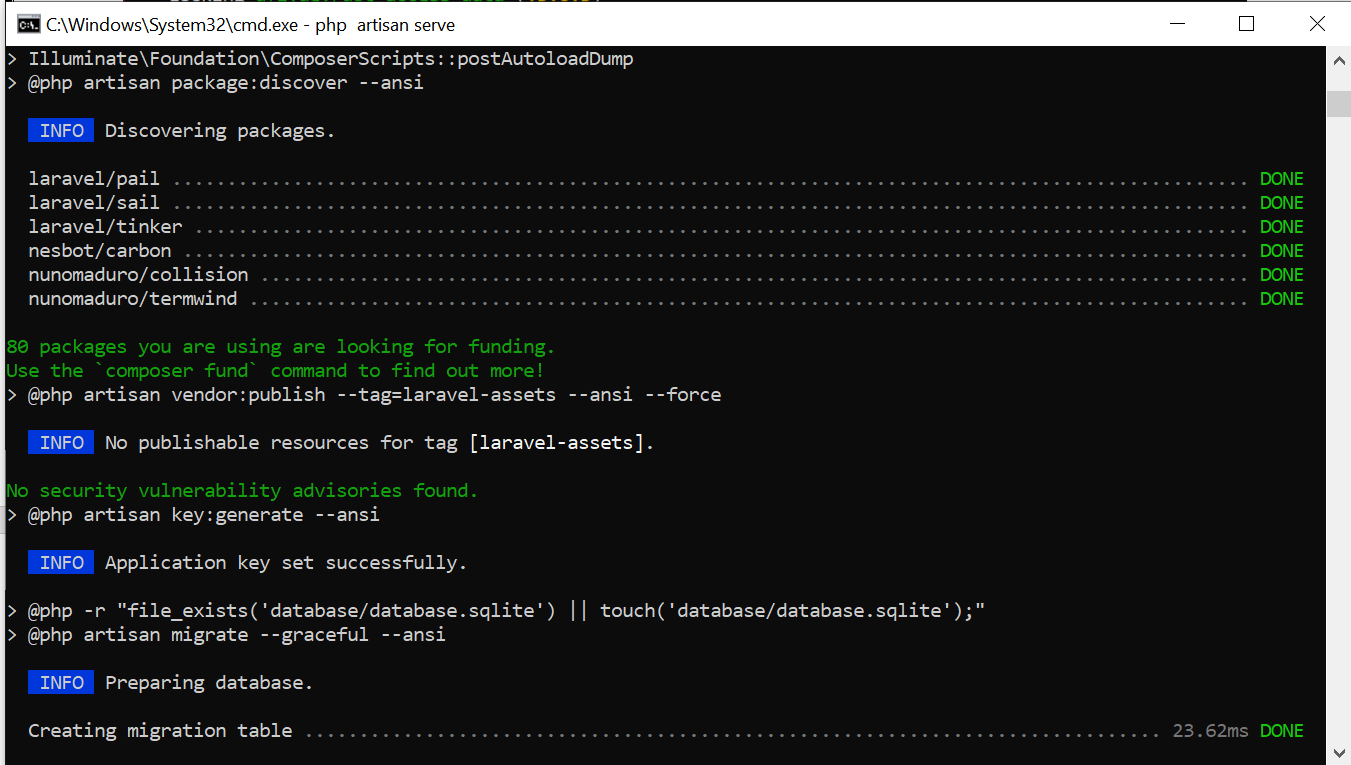
</html>

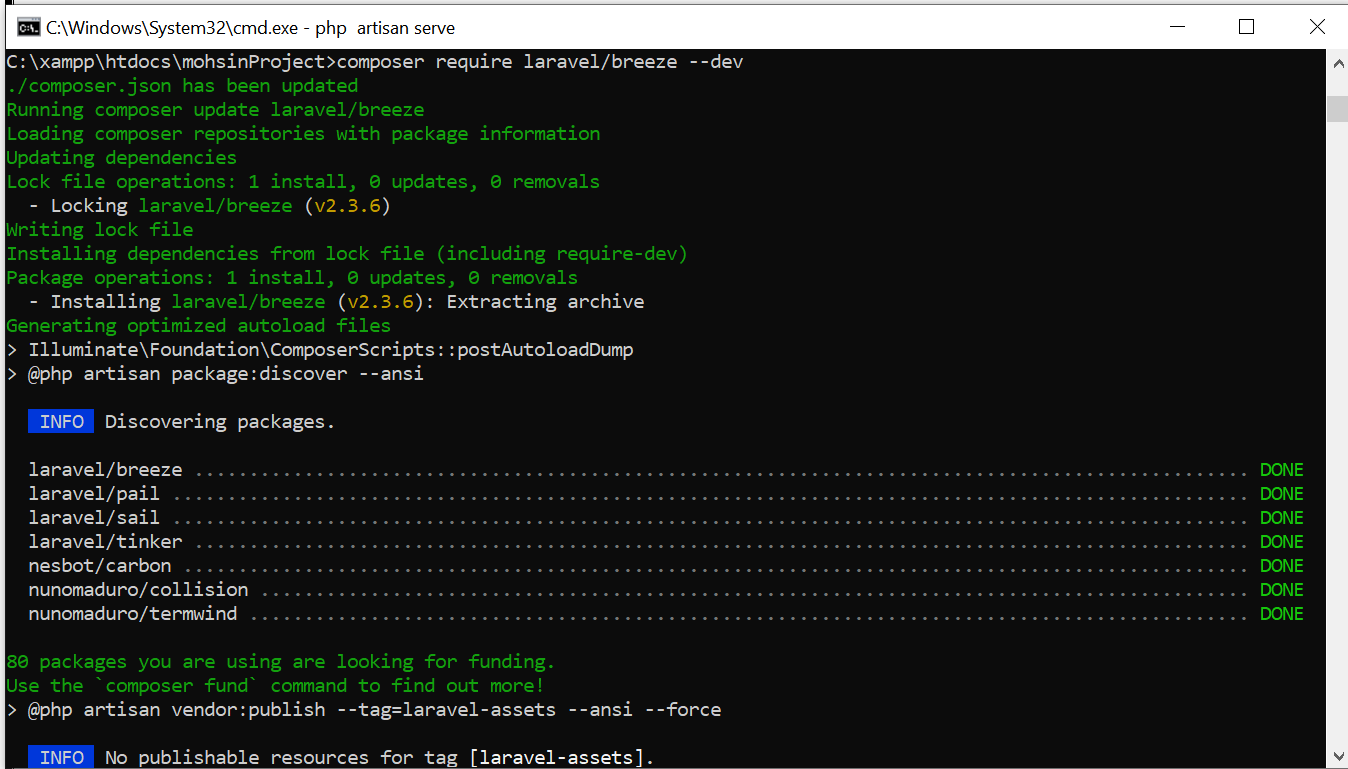
**OUTPUT:  
**

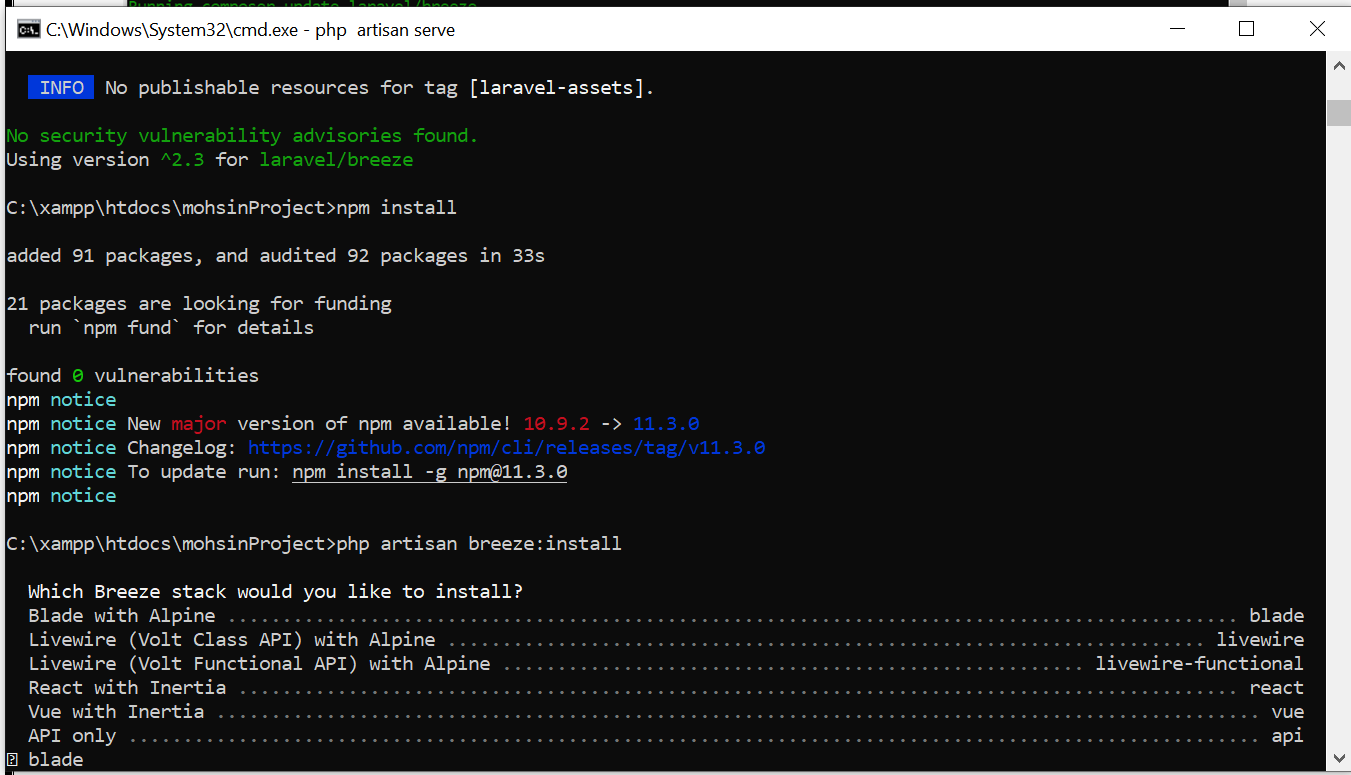
****

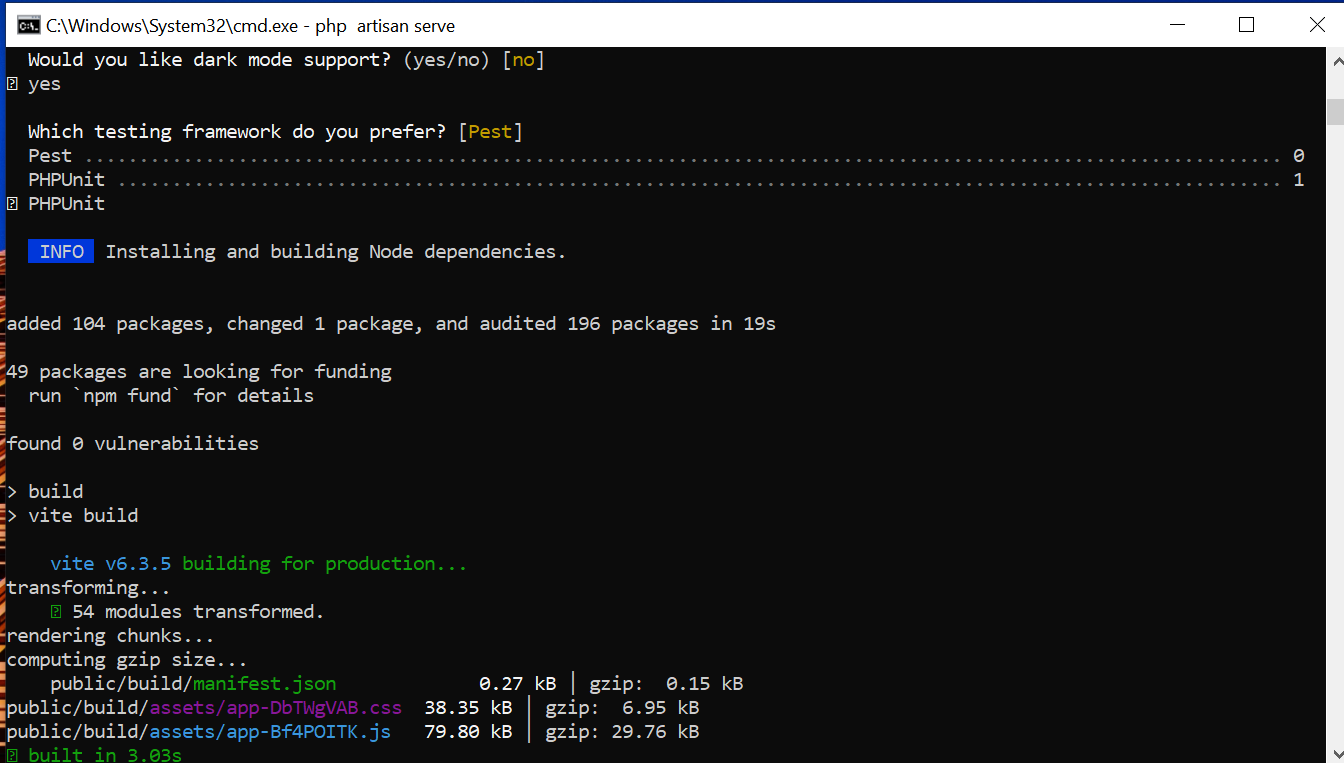
**Task 5.6:**

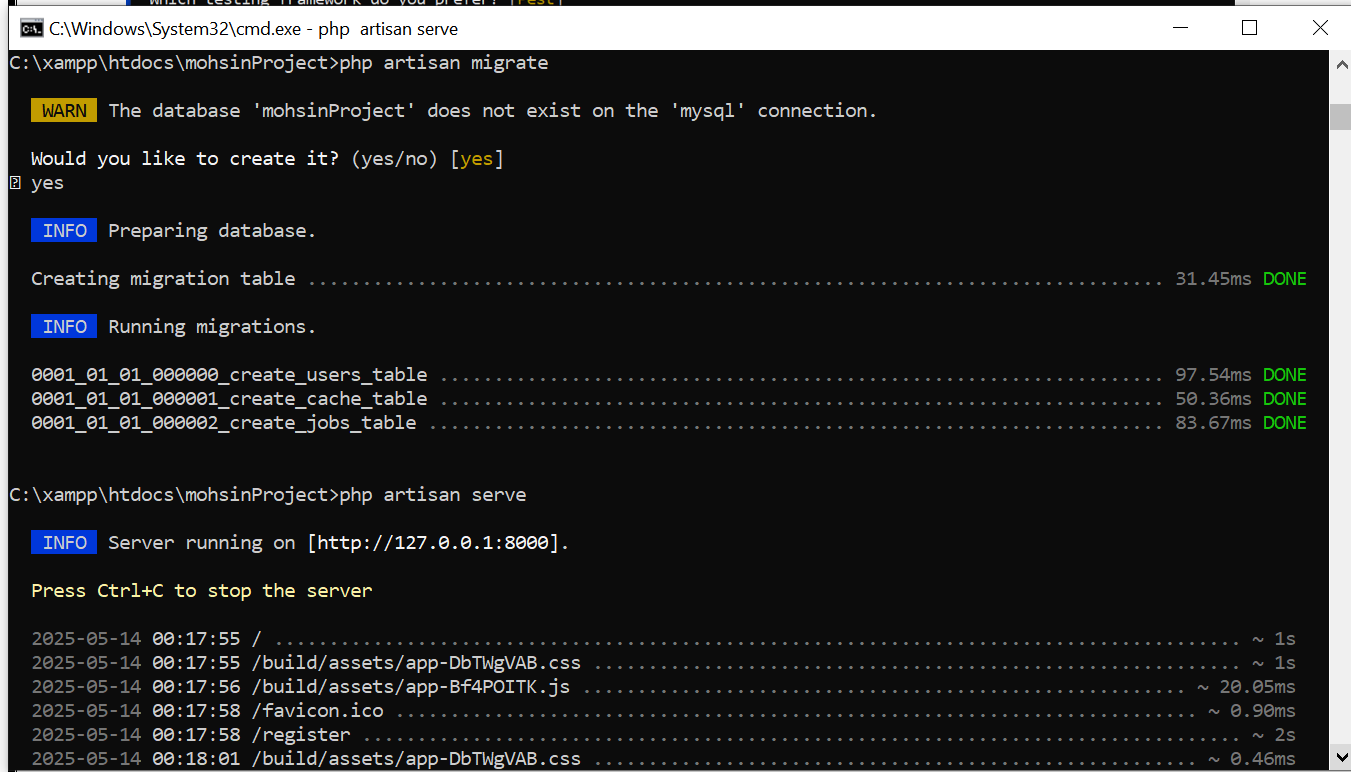
**LARAVEL  
  
**

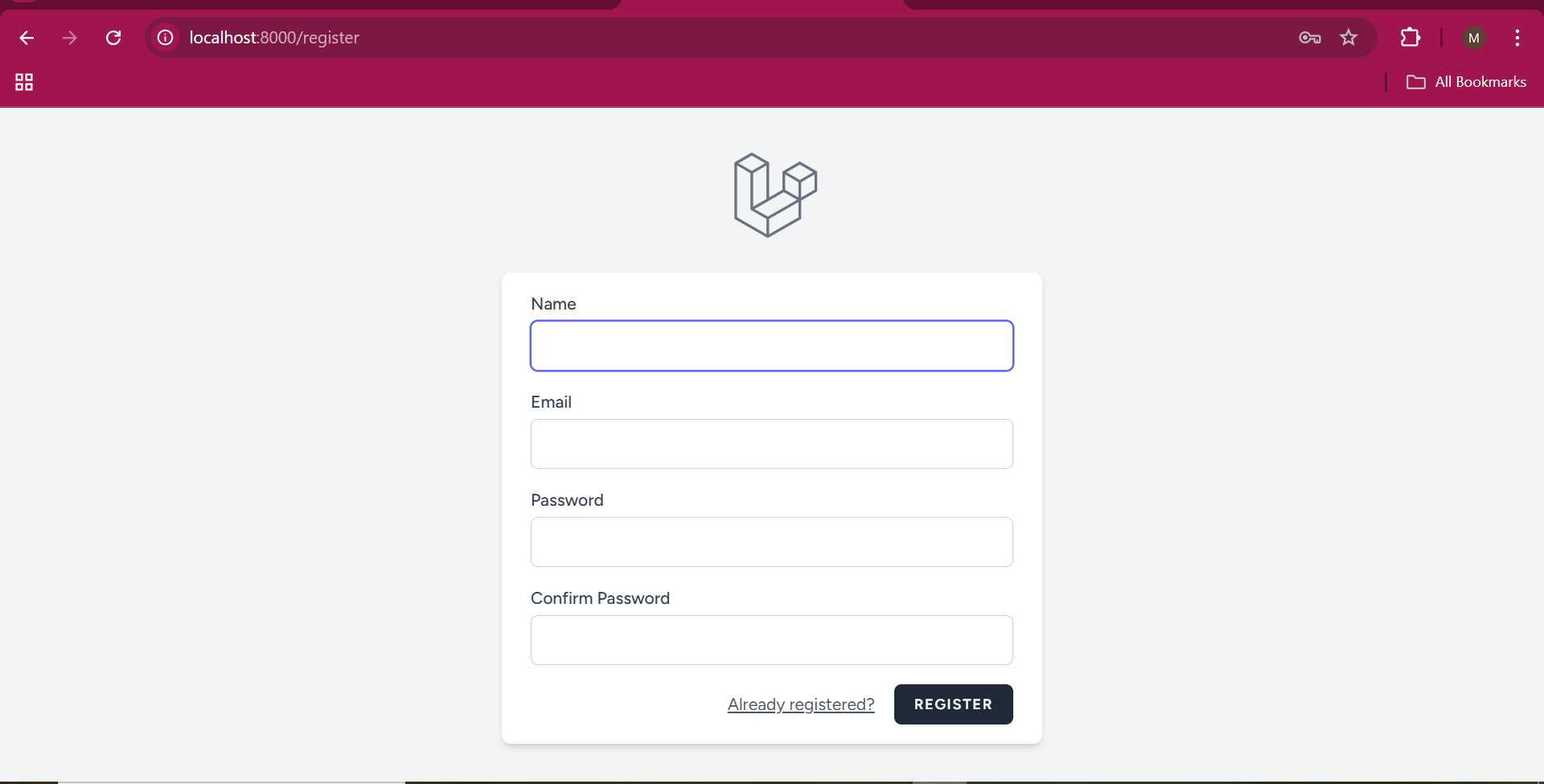
****

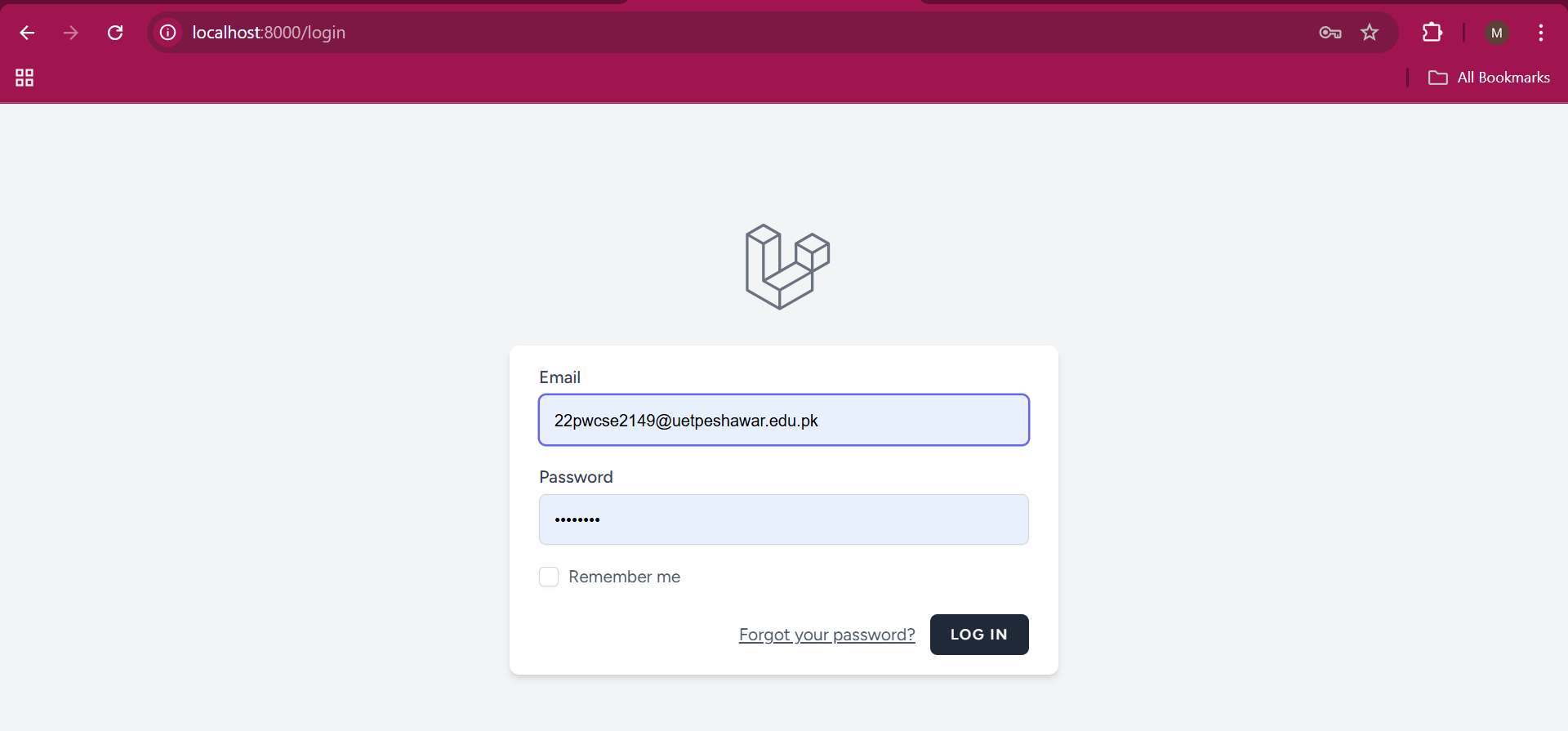
****

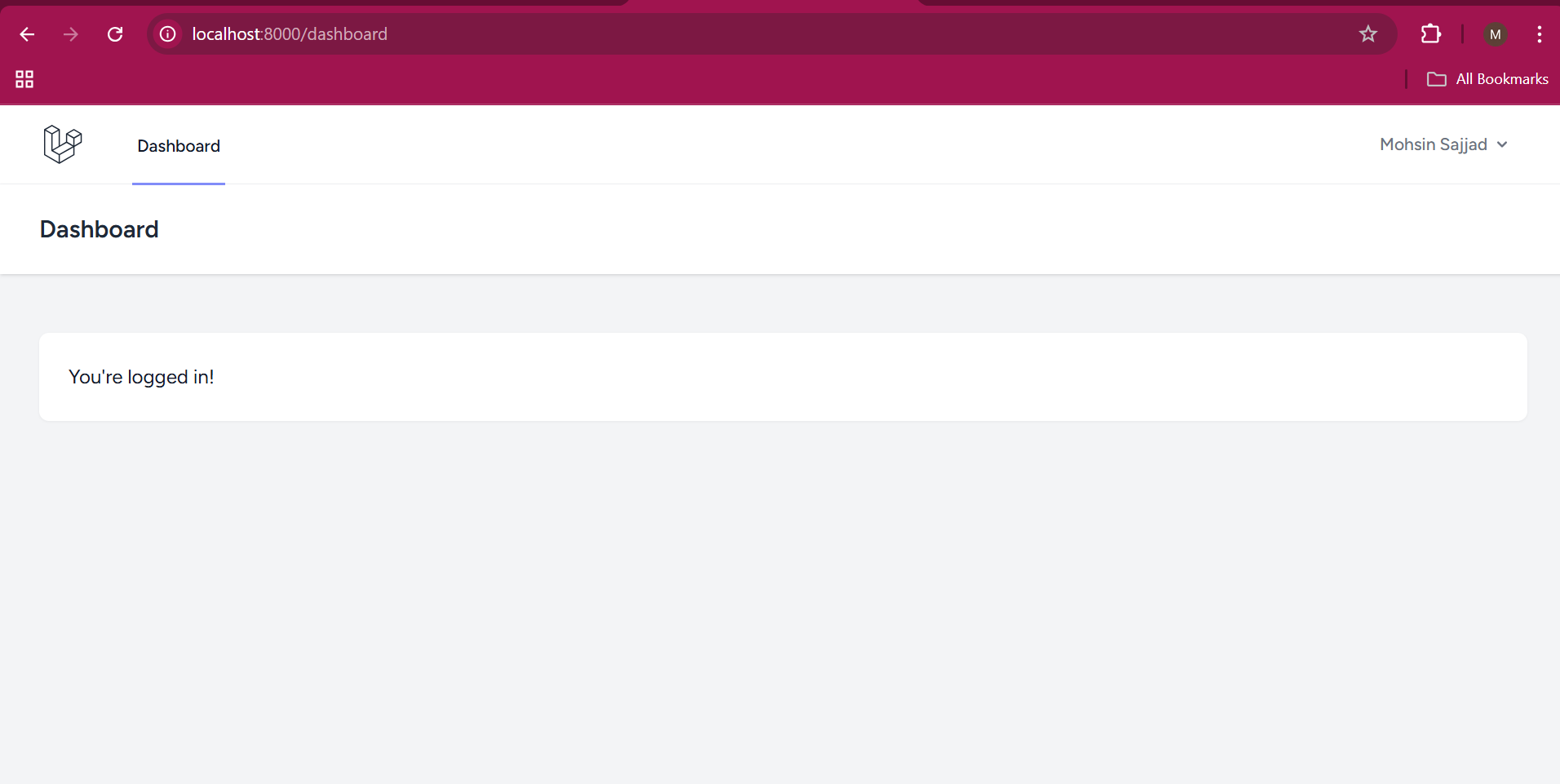
****

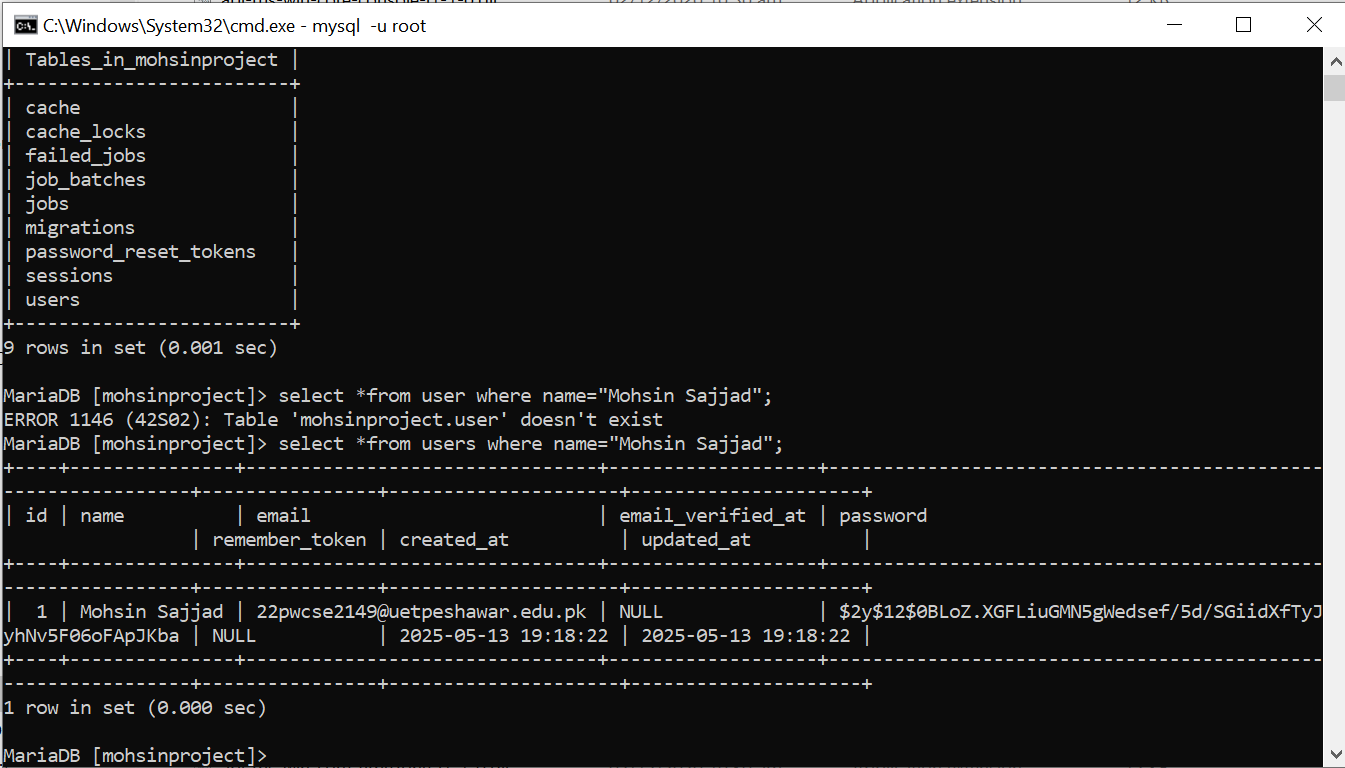
****

****

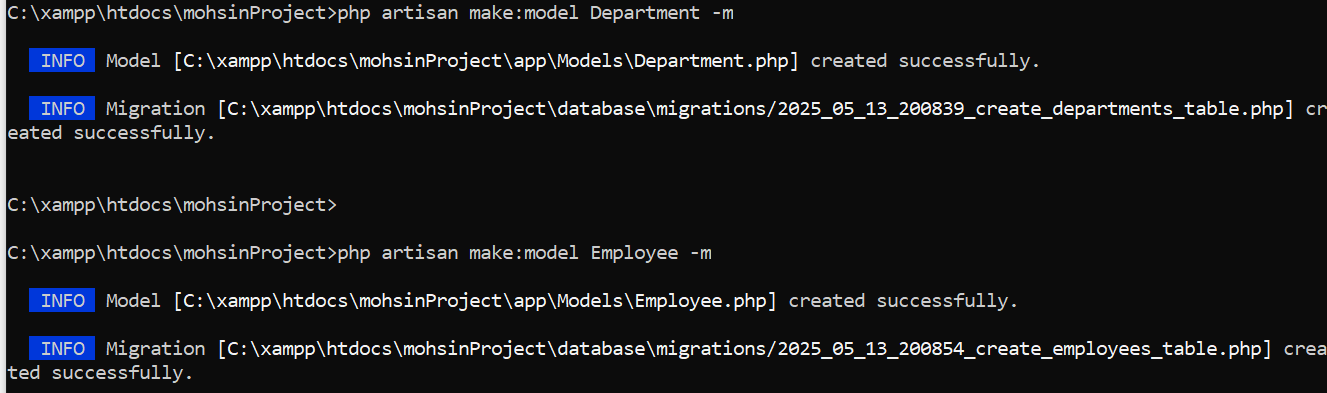
****

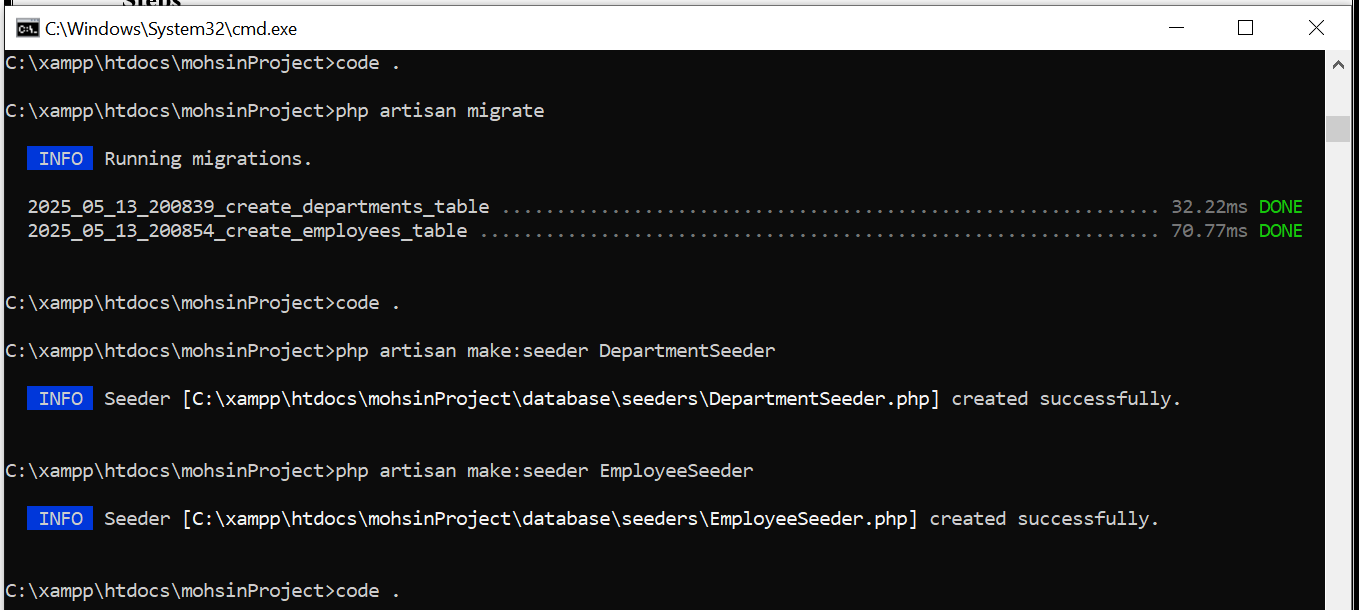
****

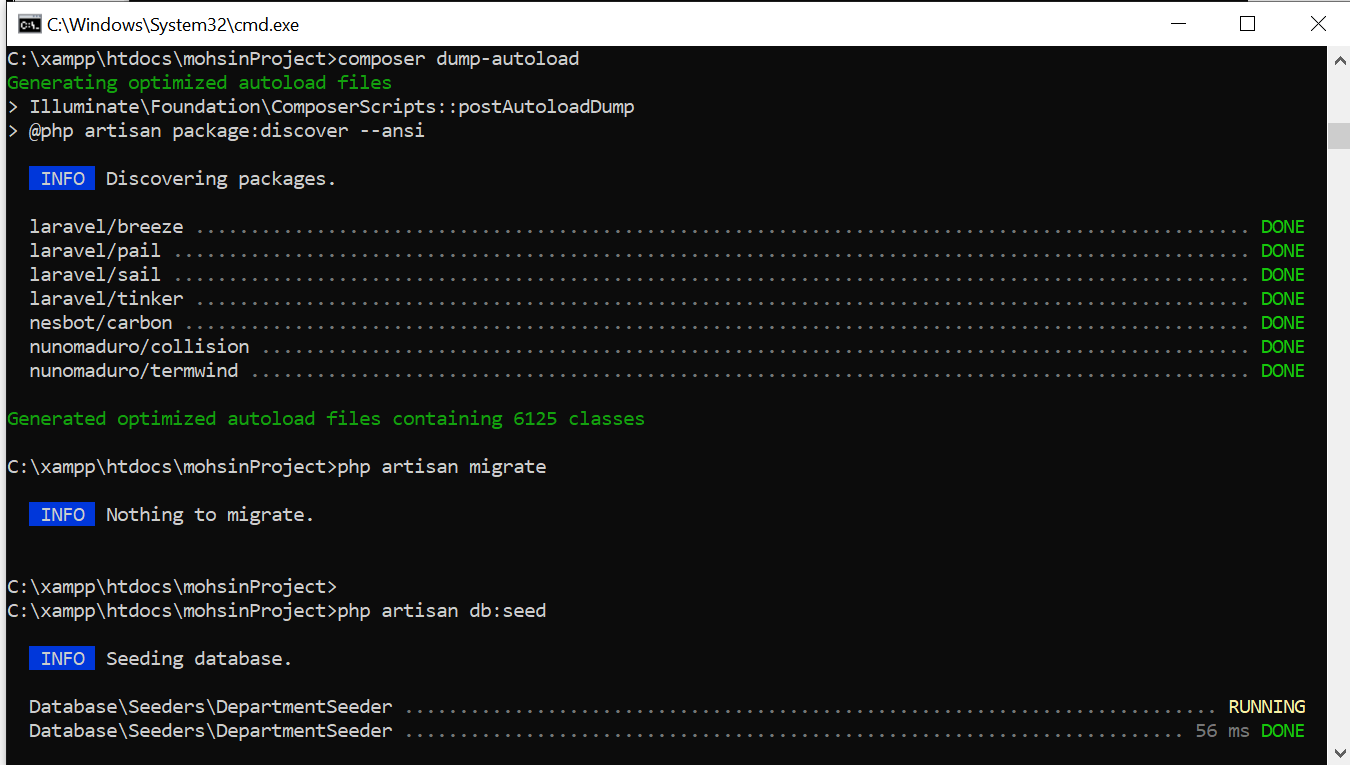
****

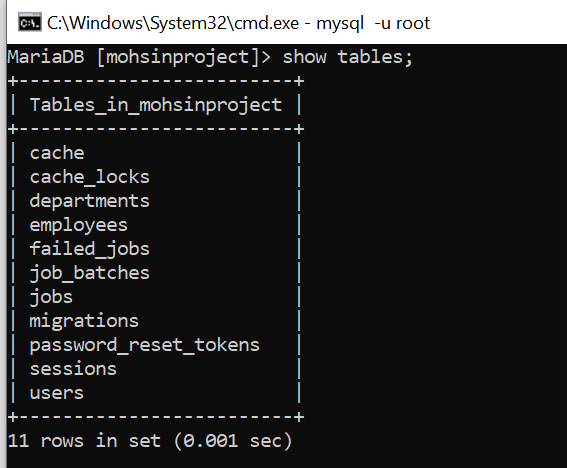
****

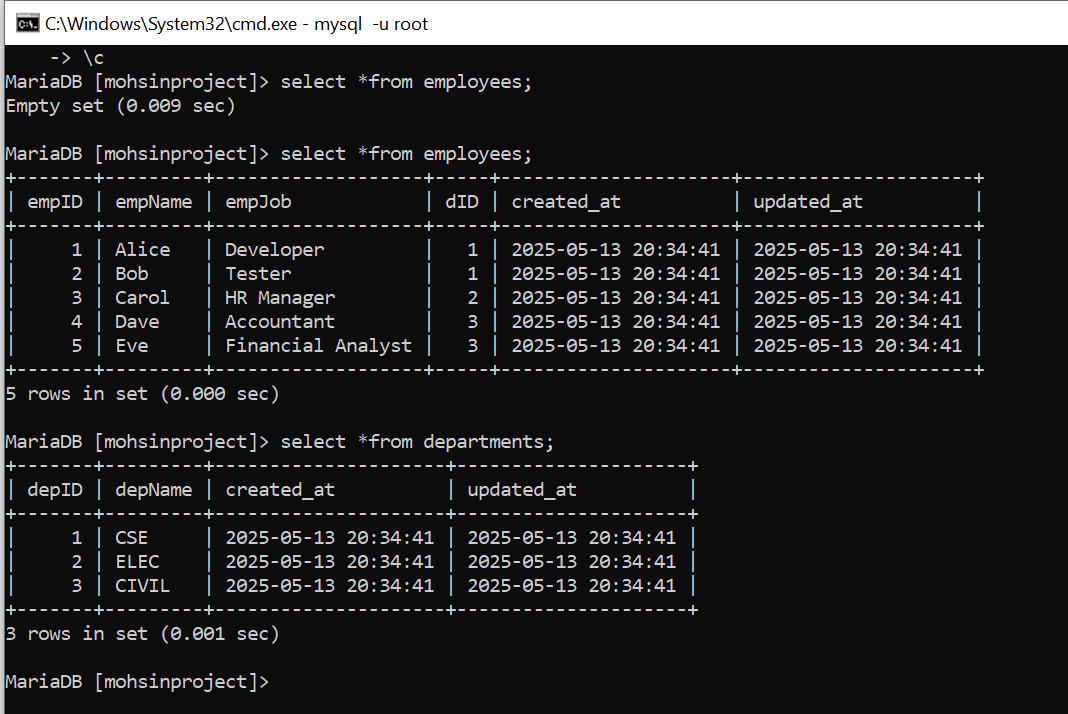
**Task 5.7:**Implement the following relationship in Laravel. Department: depID, depName Emplyee: empID, empName, empJob, dID Note that the department has at least one or many employees working in it; while employee works exactly in one department. Also, feed atleast five records in the created tables.

**Steps  
**

****

****

****

****