Lab Report No 05

**Programming in PHP**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**403L-Database Management Systems**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Sumayyea Salahuddin**

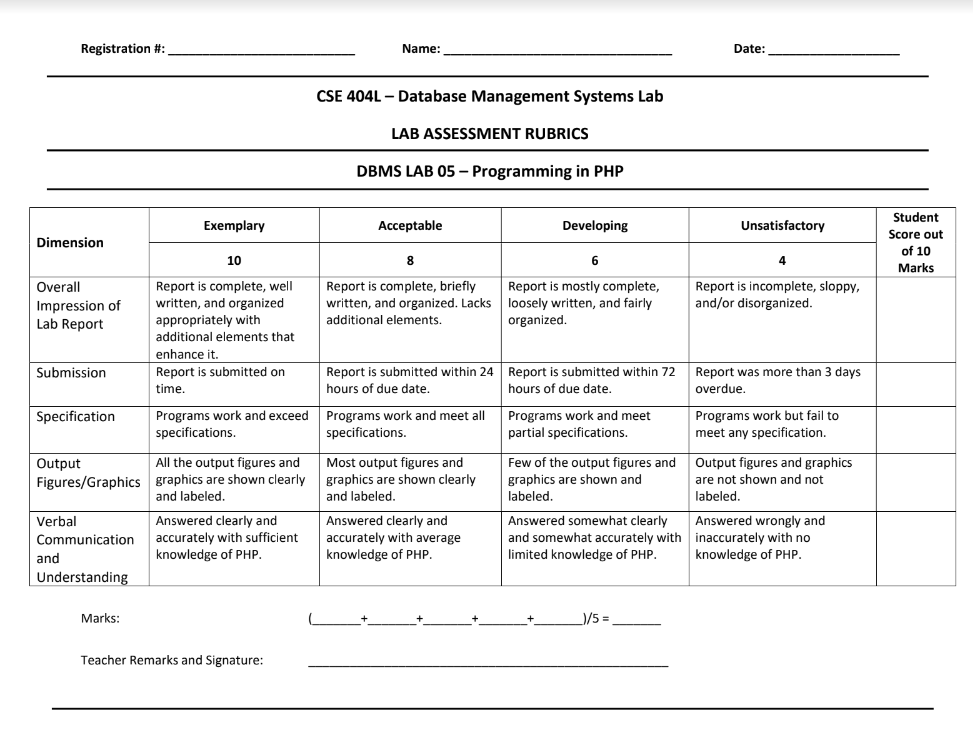
Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**



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



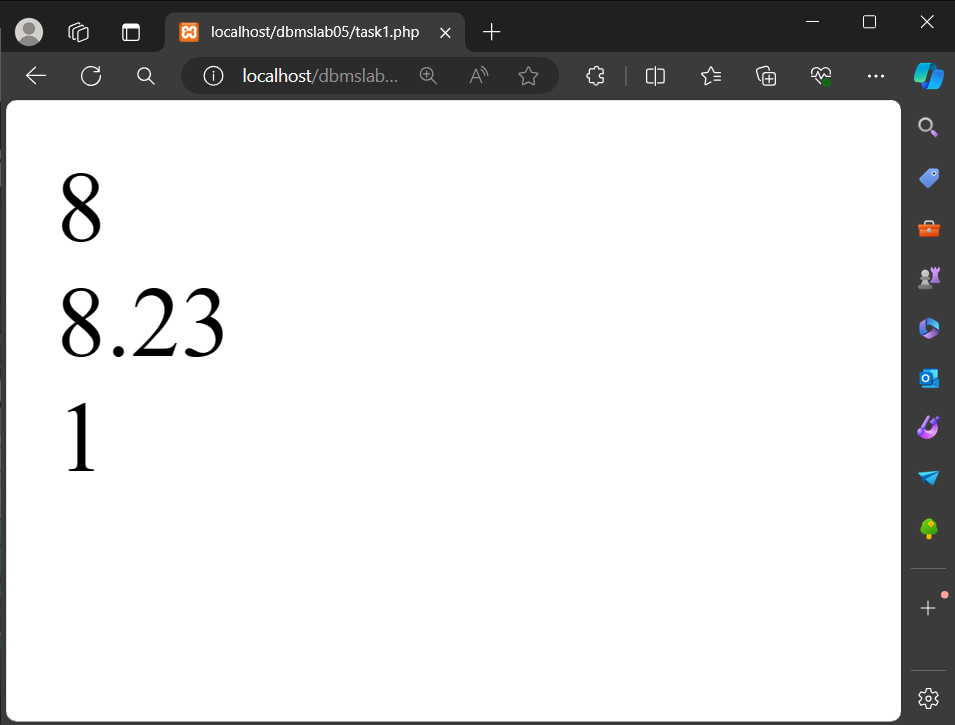
21pwcse1997 Muhammad Saad17/5/2024

**Lab # 05: Programming in PHP**

**Objective:**

This lab aims at the understanding of:

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

**Tasks 5.1**

**Q1: 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.**

**A:**

<?php

*$test\_var* = 8.23;

// Type cast to integer

*$int\_cast* = (int)*$test\_var*;

echo *$int\_cast*; // Output: 8

// Type cast to string

*$str\_cast* = (string)*$test\_var*;

echo *$str\_cast*; // Output: "8.23"

// Type cast to boolean

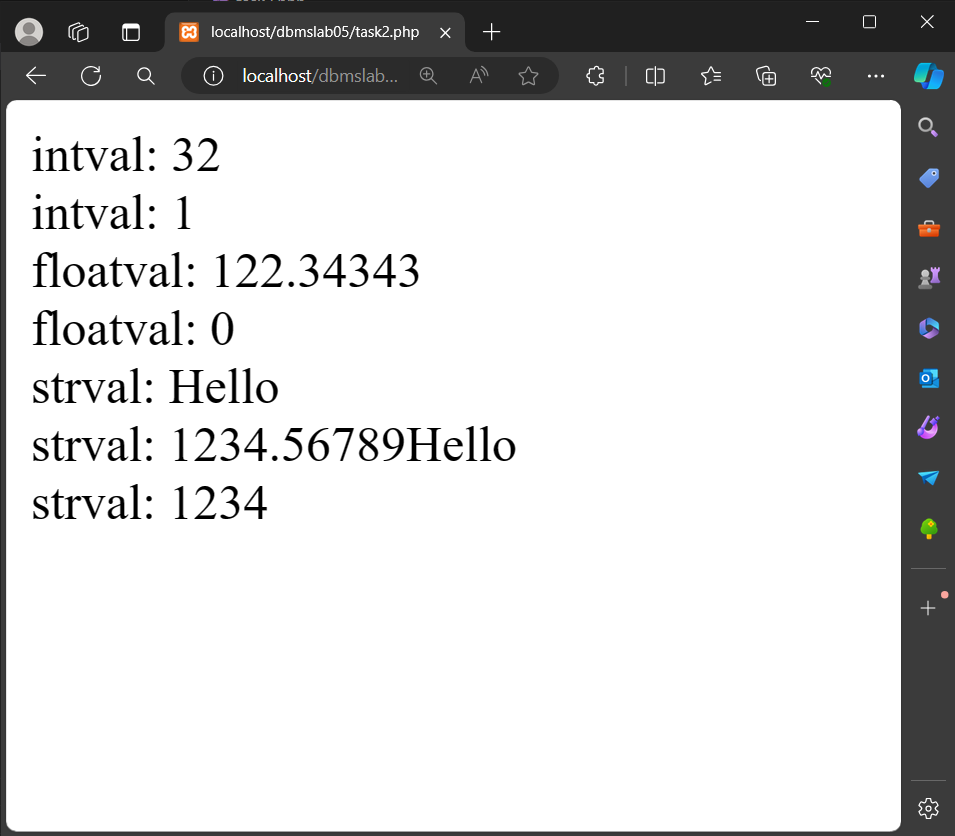
*$bool\_cast* = (bool)*$test\_var*;

echo *$bool\_cast*; // Output: 1 (true)

?>

Type casting of double to int, string and bool

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Tasks 5.2**

**Q: Use and Specify the purpose of following functions:**

**1) intval( value ):** Converts and extracts int value from any data type.

**2) floatval( value ):**Converts and extracts float value from any data type

**3) strval( value ):** Converts any data type into string type.

**Ans:**

<?php

*$a* = 32;

echo "intval: ".*intval*(*$a*)."<br>"; // Output: 32

*$e* = array("red", "green", "blue");

echo "intval: ".*intval*(*$e*)."<br>"; // Output: 1

*$var* = "122.34343The";

*$float\_value* = *floatval*(*$var*);

echo "floatval: ".*$float\_value*."<br>"; // Output: 122.34343

*$var* = "The122.34343";

*$float\_value* = *floatval*(*$var*);

echo "floatval: ".*$float\_value*."<br>"; // Output: 0

*$a* = "Hello";

echo "strval: ".*strval*(*$a*)."<br>"; // Output: "Hello"

*$c* = "1234.56789Hello";

echo "strval: ".*strval*(*$c*)."<br>"; // Output: "1234.56789Hello"

*$e* = 1234;

echo "strval: ".*strval*(*$e*)."<br>"; // Output: "1234"

?>

Typecasting and extracting the data accordingly

**Tasks 5.3**

**Q: 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.**

**Ans**

<?php

*$rows* = 10;

*$columns* = 10;

echo '<table border="1">';

for (*$i* = 1; *$i* <= *$rows*; *$i*++) {

echo '<tr>';

for (*$j* = 1; *$j* <= *$columns*; *$j*++) {

*$number* = *round*(*$i* / *$j*, 2);

*$isOdd* = (*$number* % 2 === 1);

*$isPrime* = is\_prime(*$number*);

echo '<td>';

echo "$number<br>";

echo *$isOdd* ? '<b>Odd</b><br>' : '<b>Even</b><br>';

echo *$isPrime* ? '<span style="color: green;">Prime</span>' : '';

echo '</td>';

}

echo '</tr>';

}

echo '</table>';

function is\_prime(*$num*) {

if (*$num* <= 1) {

return false;

}

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

if (*$num* % *$i* === 0) {

return false;

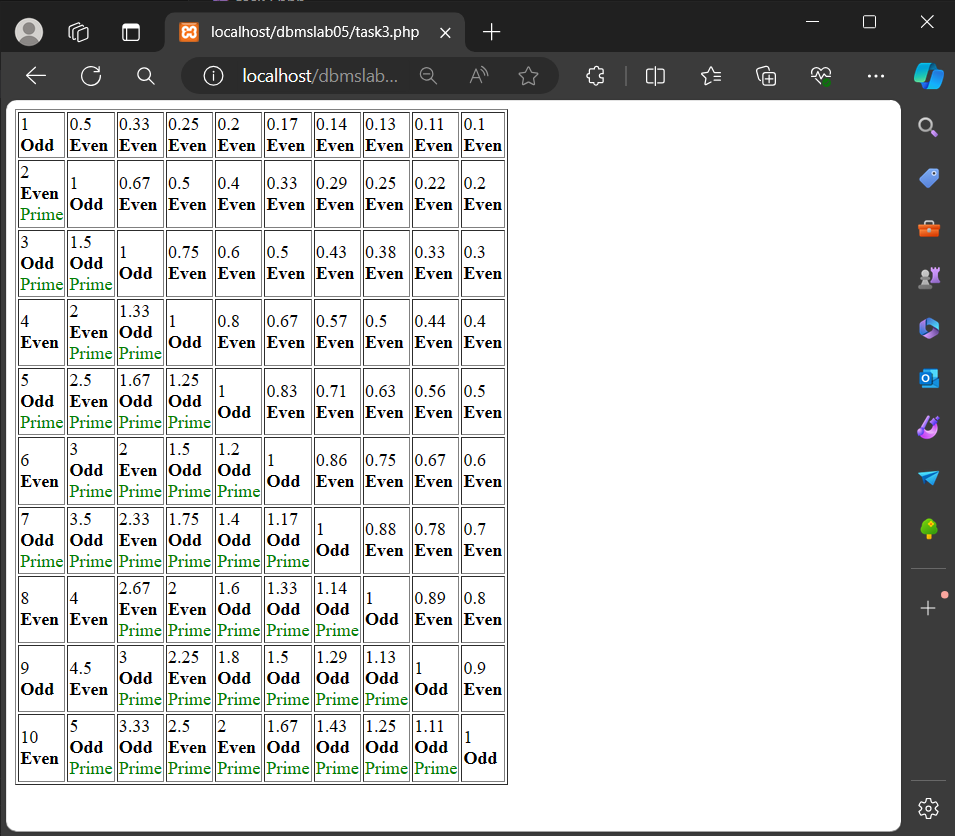
}

}

return true;

}

?>



Creating the table given in the task in PHP

**Tasks 5.4**

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

**A:**

<?php

class Animal {

public function makeSound() {

echo "The animal makes a sound.\n<br>";

}

}

class Dog extends Animal { //Inheritancea: Dog inherits Animal

public function makeSound() {

echo "The dog barks.\n<br>";

}

}

class Cat extends Animal {

public function makeSound() {

echo "The cat meows.\n<br>";

}

}

interface CanFly {

public function fly();

}

class Bird extends Animal implements CanFly {//Polymorphism: Defines an abstract function of CanFly class

public function makeSound() {

echo "The bird chirps.\n<br>";

}

public function fly() {

echo "The bird is flying.\n<br>";

}

}

*$animal* = new Animal(); //Defining object of Animal Class

*$animal*->makeSound();

*$dog* = new Dog(); //Defining object of Dog Class and Inherting Animal Class

*$dog*->makeSound();

*$cat* = new Cat(); //Defining object of Animal Class

*$cat*->makeSound();

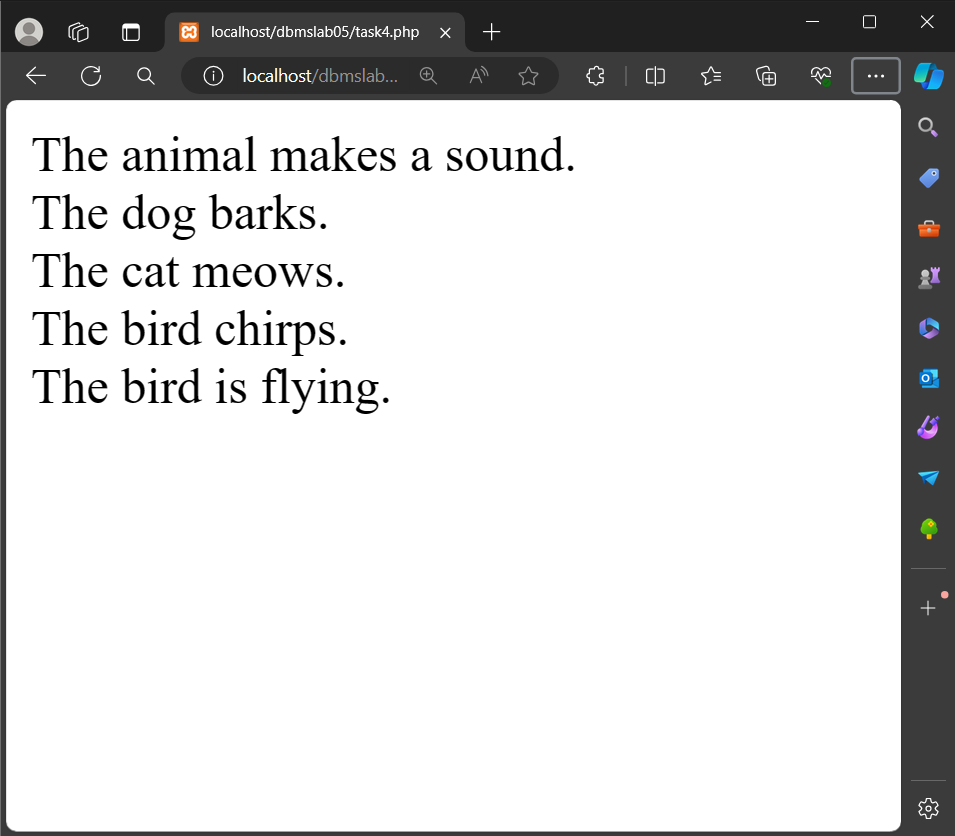
*$bird* = new Bird(); //Defining object of Animal Class

*$bird*->makeSound();

*$bird*->fly();

?>

Controller for table ‘product’



Example of Objects, Classes, Inheritance and Polymorphism

**Tasks 5.5**

**Q: Gallery using HTML, CSS and PHP.**

**A:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Document</title>

<?php

*$servername* = "localhost";

*$username* = "root";

*$password* = "";

*$database* = "myDatabase";

*$conn* = new mysqli(*$servername*, *$username*, *$password*, *$database*);

?>

</head>

<body>

<?php

if (*$conn*->connect\_error) {

die("Connection failed: " . *$conn*->connect\_error);

}

echo "<h4>Connected successfully</h4>";

*$insertQuery* = "INSERT INTO employee (empID, name, depID)

VALUES ('21345', 'Kamran', '1')";

*$selectQuery* = "SELECT \* FROM employee";

*$conn*->query(*$insertQuery*);

*$result* = *$conn*->query(*$selectQuery*);

if (*$result*->num\_rows > 0) {

while (*$row* = *$result*->fetch\_assoc()) {

echo "ID: " . *$row*["empID"] . " - Name: " . "<b>{*$row*["name"]}</b>". "<br>";

}

} else {

echo "No records found.";

}

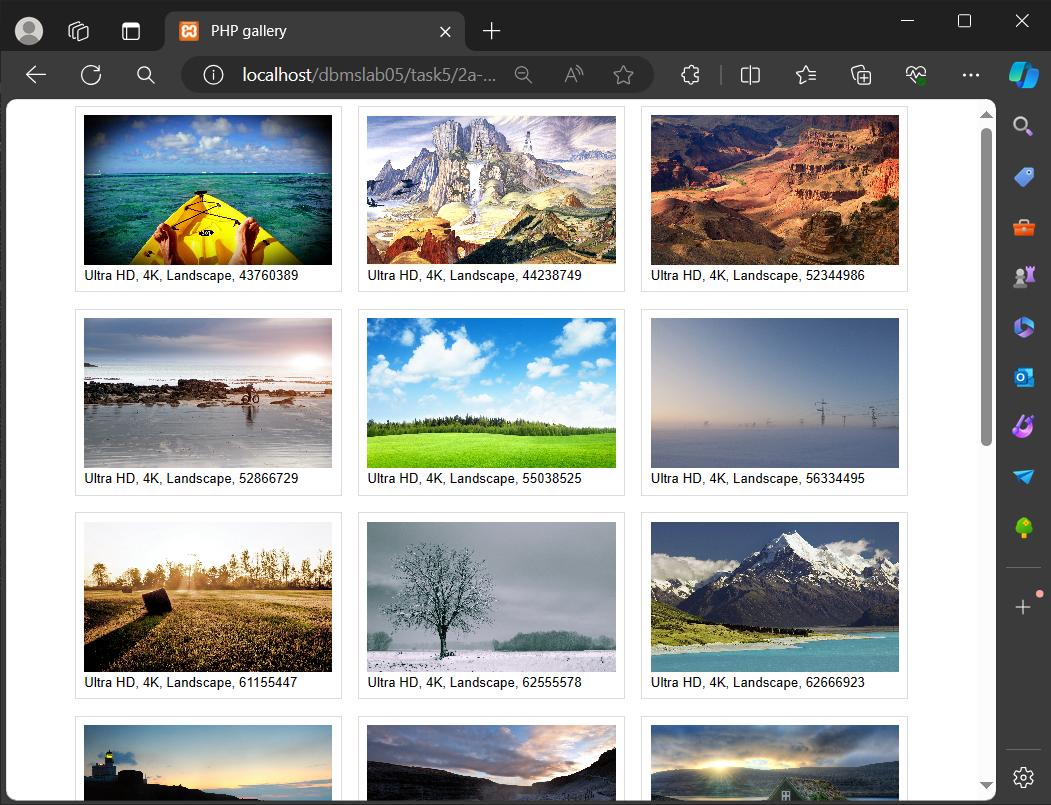
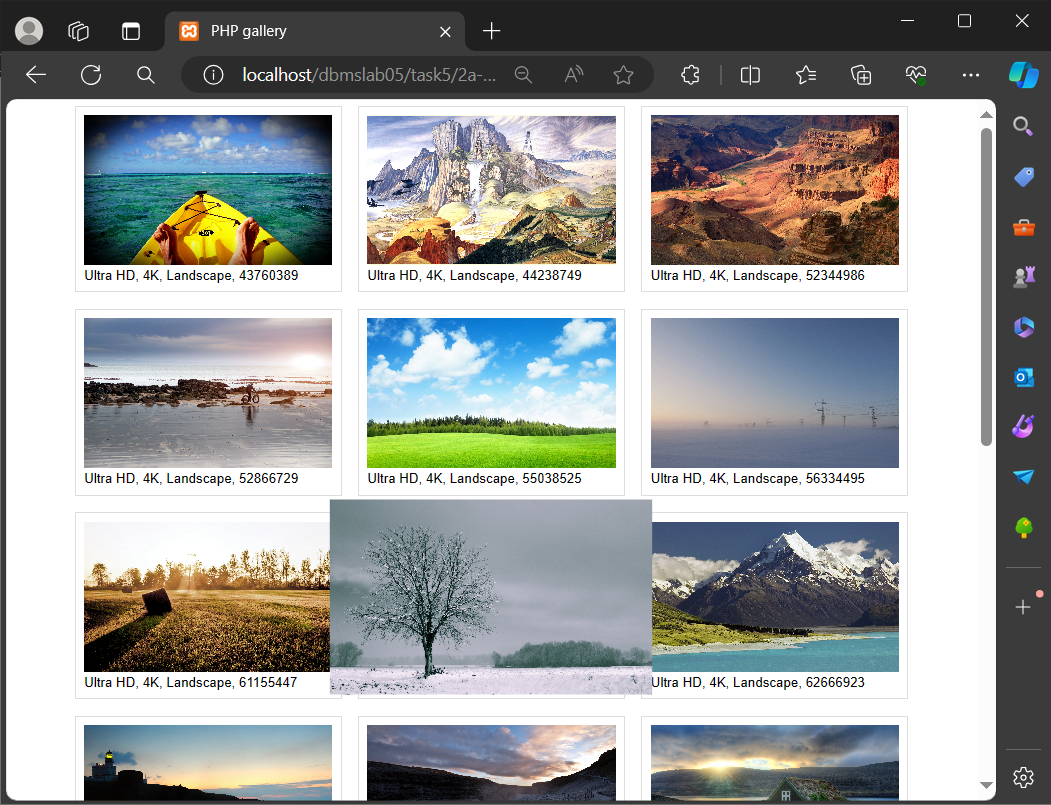
*$result*->free\_result();

*$conn*->close();

?>

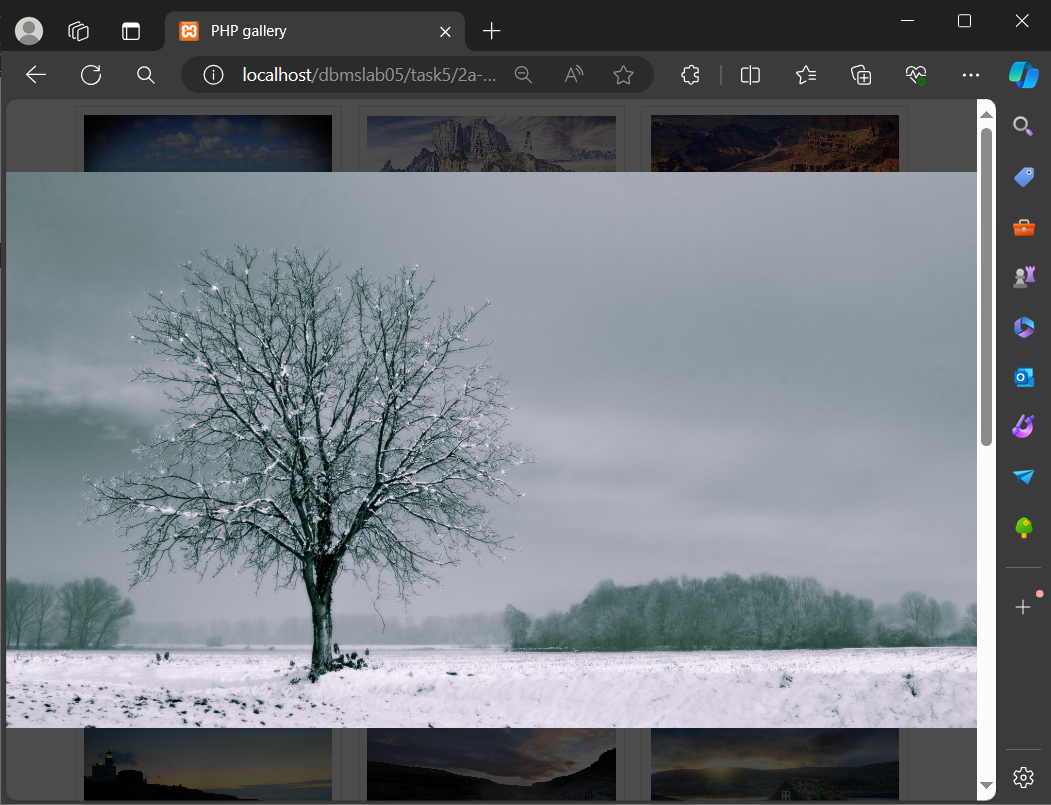
</body>

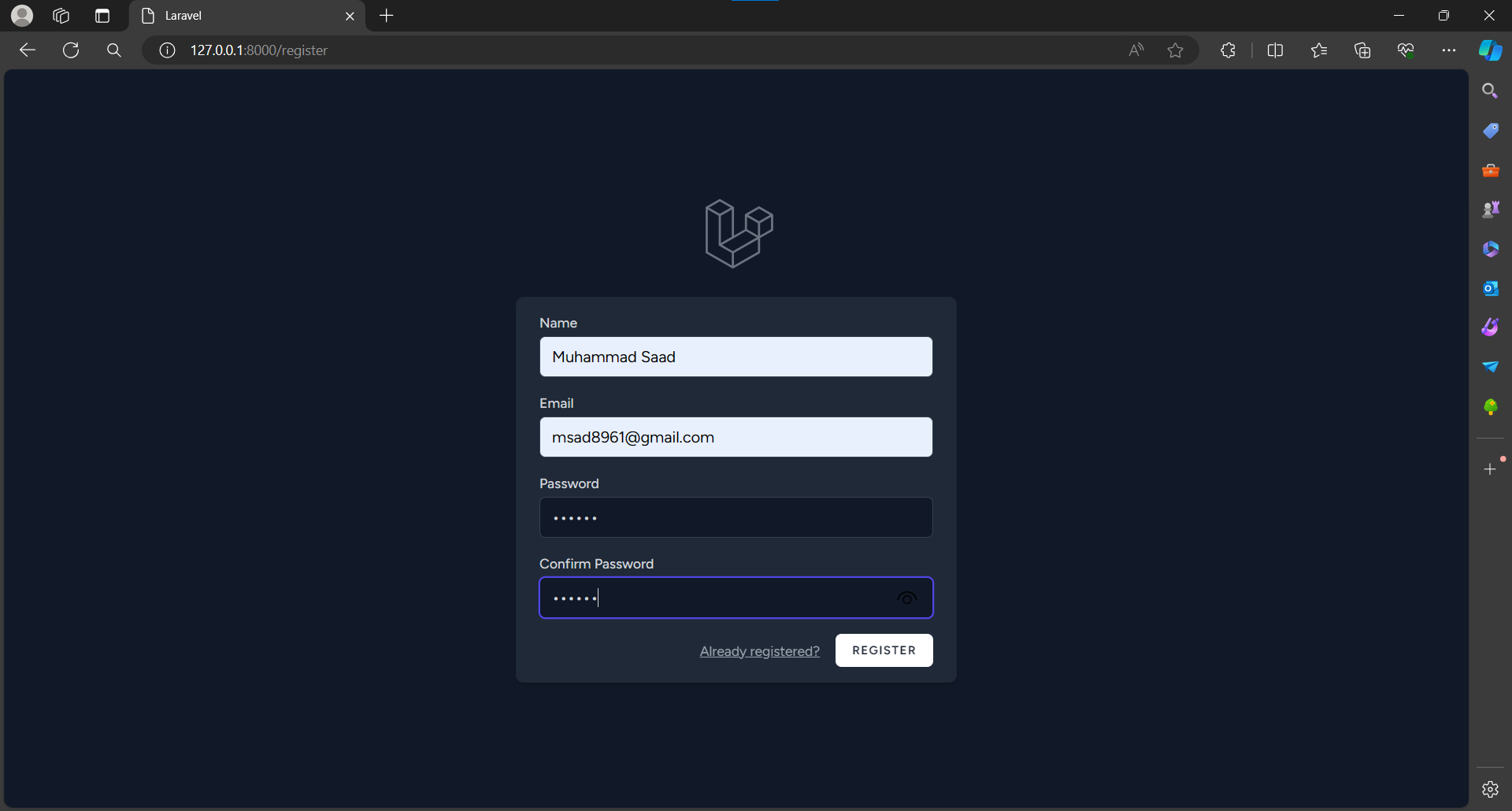
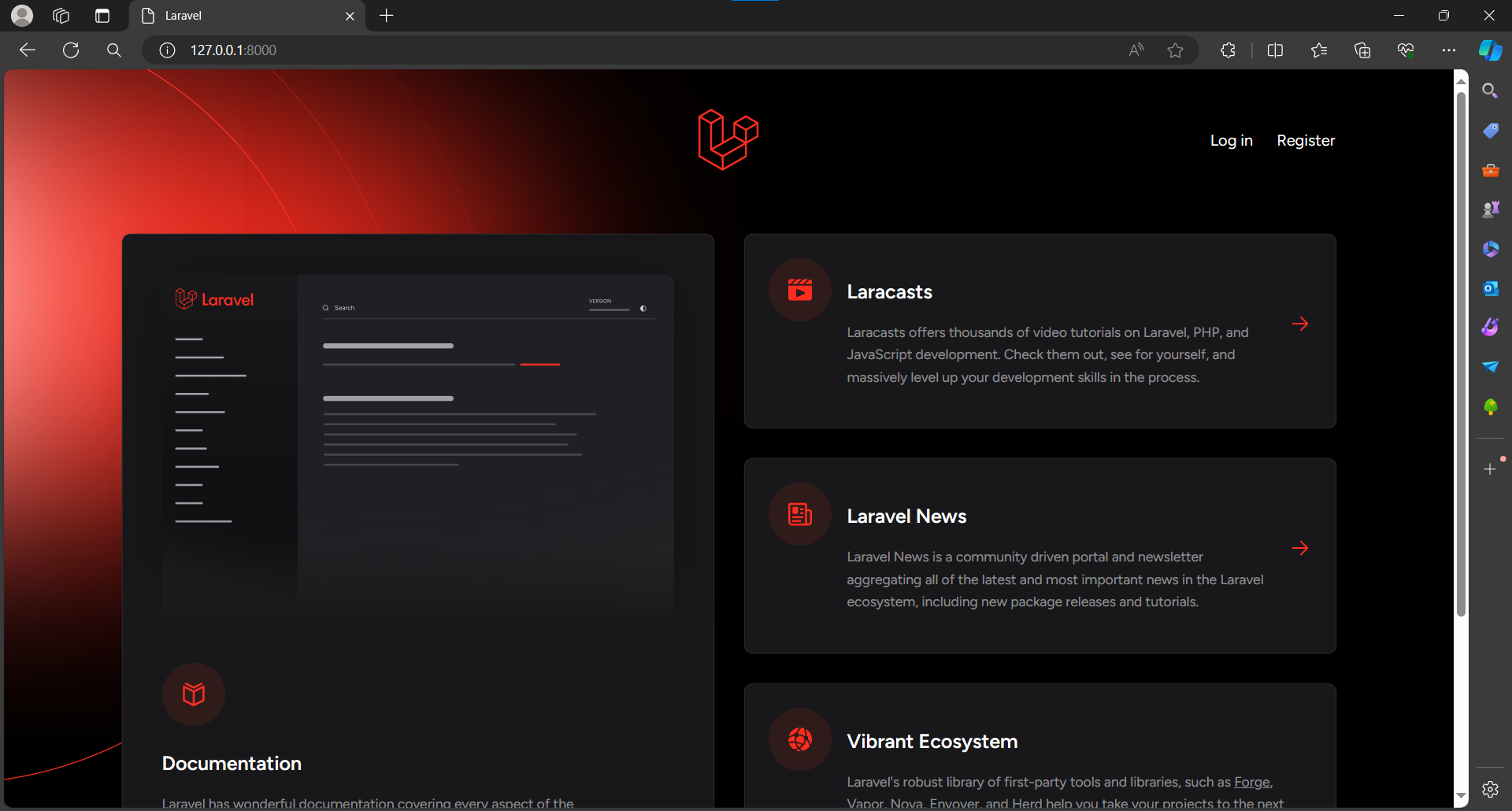
</html>



Gallery in PHP

Gallery PHP: Selecting photo

Gallery PHP: Selecting photo

**Tasks 5.6**

**Q: Write PHP-MYSQLI Object-Oriented interface script that perform insert, update and delete.**

**Ans:**

composer create-project laravel/laravel project-name

composer require laravel/ui –dev

php artisan ui vue –auth

composer require laravel/breeze --dev

php artisan breeze:install

php artisan make:model Department

php artisan make:migration create\_employees\_table

Command Used

**Tasks 5.7**

**Q: Implement the following relationship in Laravel.**

**Department: depID, depName**

**Emplyee: empID, empName, empJob, dID**

**Ans:**

j<?php

namespace App\Http\Controllers;

use App\Models\Employee;

use Illuminate\Http\Request;

class EmployeeController extends Controller

{

public function index(){

*$employee* = Employee::with('department')->get();

return view('dashboard', ['employee' => *$employee*]);

}

}

**Employee Controller**

<?php

namespace App\Http\Controllers;

use App\Models\Department;

use Illuminate\Http\Request;

class DepartmentController extends Controller

{

public function index(){

*$department* = Department::all();

return view('dashboard', ['department' => *$department*]);

}

}

**Department Controller**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Department extends Model

{

public *$table* = 'department';

use HasFactory;

protected *$fillable* = [

'depID',

'name',

];

}

**Department Model**

class Employee extends Model

{

public *$table* = 'employee';

public function department()

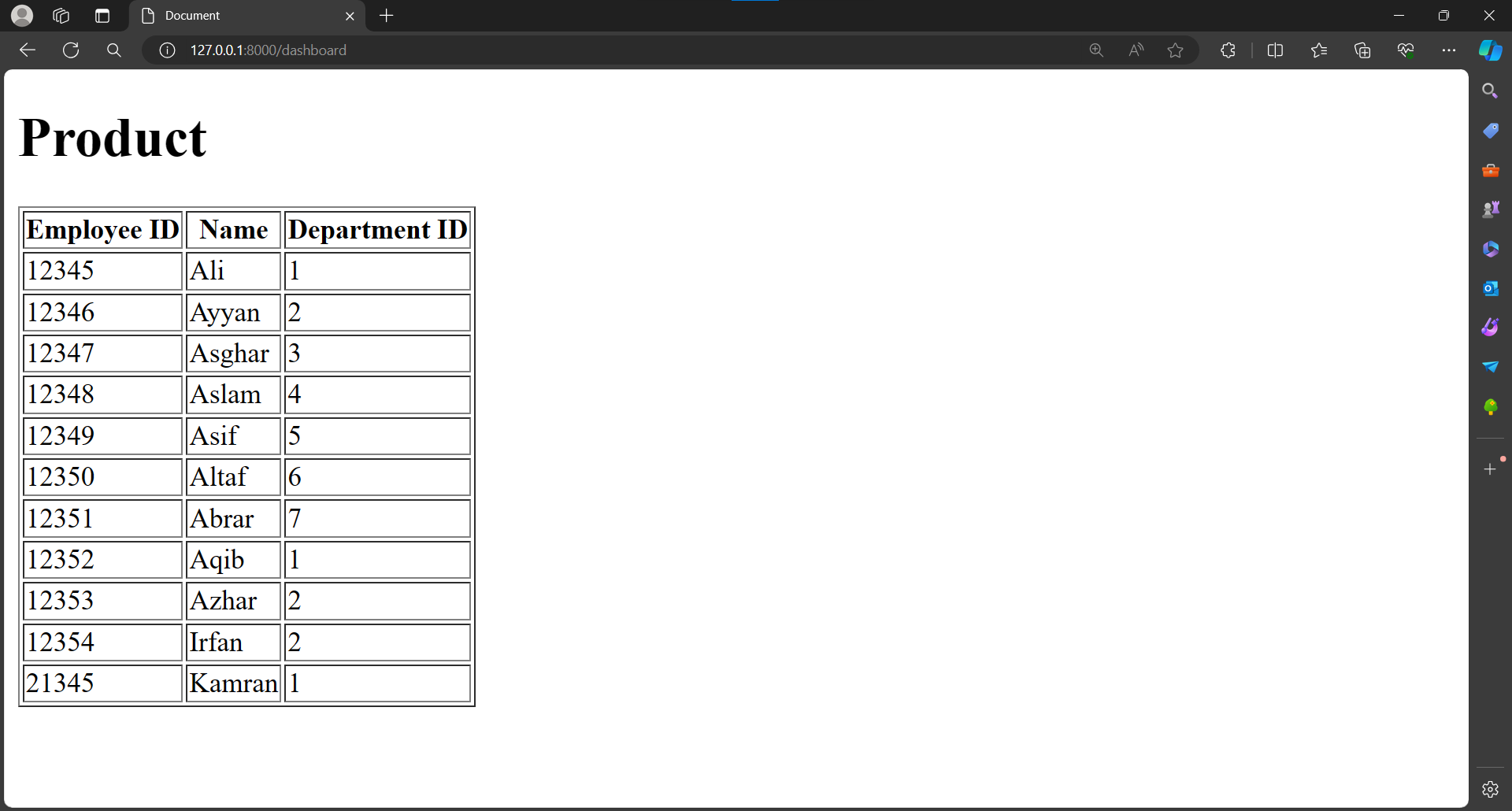
{

return *$this*->belongsTo(Department::class);

}

}

**Employee Model**

****

<!DOCTYPE html>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<meta http-equiv="X-UA-Compatible" *content*="ie=edge">

<title>Document</title>

</head>

<body>

<h1>Product</h1>

<div>

<table *border*=1>

<tr><th>Employee ID</th>

<th>Name</th>

<th>Department ID</th>

</tr>

@foreach (*$employee* as *$emp*)

<tr><td>{{*$emp*->empID}}</td>

<td>{{*$emp*->name}}</td>

<td>{{*$emp*->depID}}</td>

</tr>

@endforeach

</table>

</div>

</body>

</html>

**View for table**

Output

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_