# TITLE : OOP with PHP -1

# PRACTICAL -7

OOP with PHP -1

**Q-1. Create a HTML form which takes a bank account number, current balance, amount value for deposit or withdrawal operation. It is two button withdrawal and deposit. On click on button it performs a respective operation . Create a bank class having a property account number and amount and methods for deposit, withdrawal and for displaying the account number and current amount. On clicking on the withdrawal or deposit button it creates an object of the bank class and should call the respective method of a class. And display the account no and current available amount.**

Code:

index.html

<!Doctype html>

<html>

<head><title>From</title>

</head>

<body>

<form action="p1.php" method="post">

<label>Account Number:</label>

<input type="number" name="accountno"><br><br>

<label>Current Balance:</label>

<input type="number" name="currentB"><br><br>

<label>Amount Values:</label>

<input type="number" name="amount"><br><br>

<input type="submit" name="Deposite" value="Deposite">

<input type="submit" name="Withdraw" value="Withdraw">

</form>

</body>

</html>

P1.php

<?php

$account\_no=$\_POST['accountno'];

$amount=$\_POST['amount'];

$current\_amt=$\_POST['currentB'];

class bank{

public $account\_no;

public $amount;

public $current\_amt;

public function \_\_construct($account\_no,$amount,$current\_amt){

$this->account\_no=$account\_no;

$this->amount=$amount;

$this->current\_amt=$current\_amt;

}

public function deposit()

{

echo "Deposite: Account\_number::". $this->account\_no." "."Amount:". $this->amount." "."cureent\_balance:".$this->current\_amt."<br><br>";

echo $this->current\_amt + $this->amount." "."<br>";

}

public function withdraw() {

echo "withdraw: Account\_number::". $this->account\_no." "."Amount". $this->amount." "."cureent\_balance:".$this->current\_amt."<br><br>";

echo $this->current\_amt - $this->amount." "."<br>";

}}

$bank=new bank($account\_no,$amount,$current\_amt);

$bank2=new bank($account\_no,$amount,$current\_amt);

if(isset($\_POST["Deposite"]))

{

$bank->deposit();

}

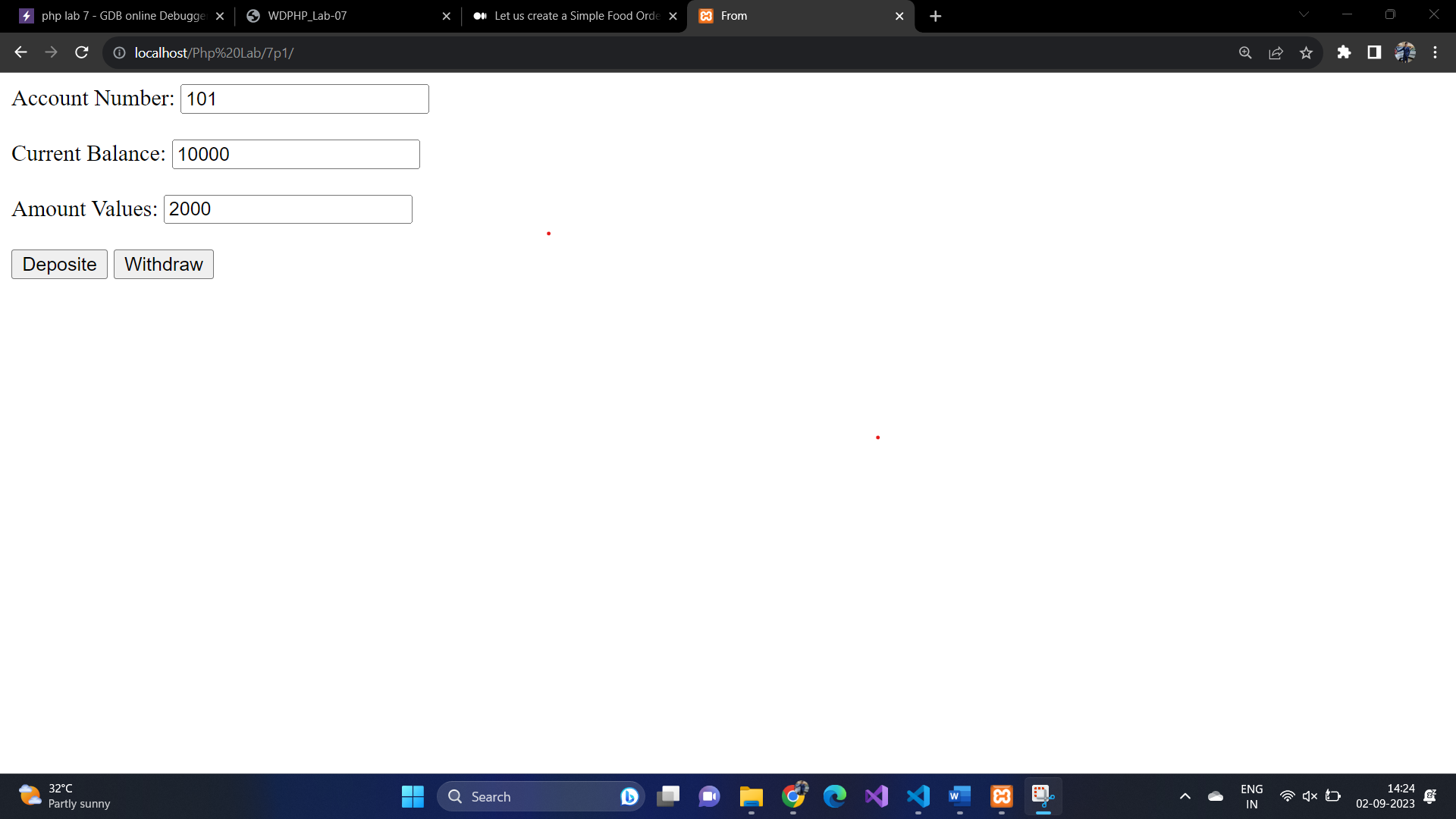
else{

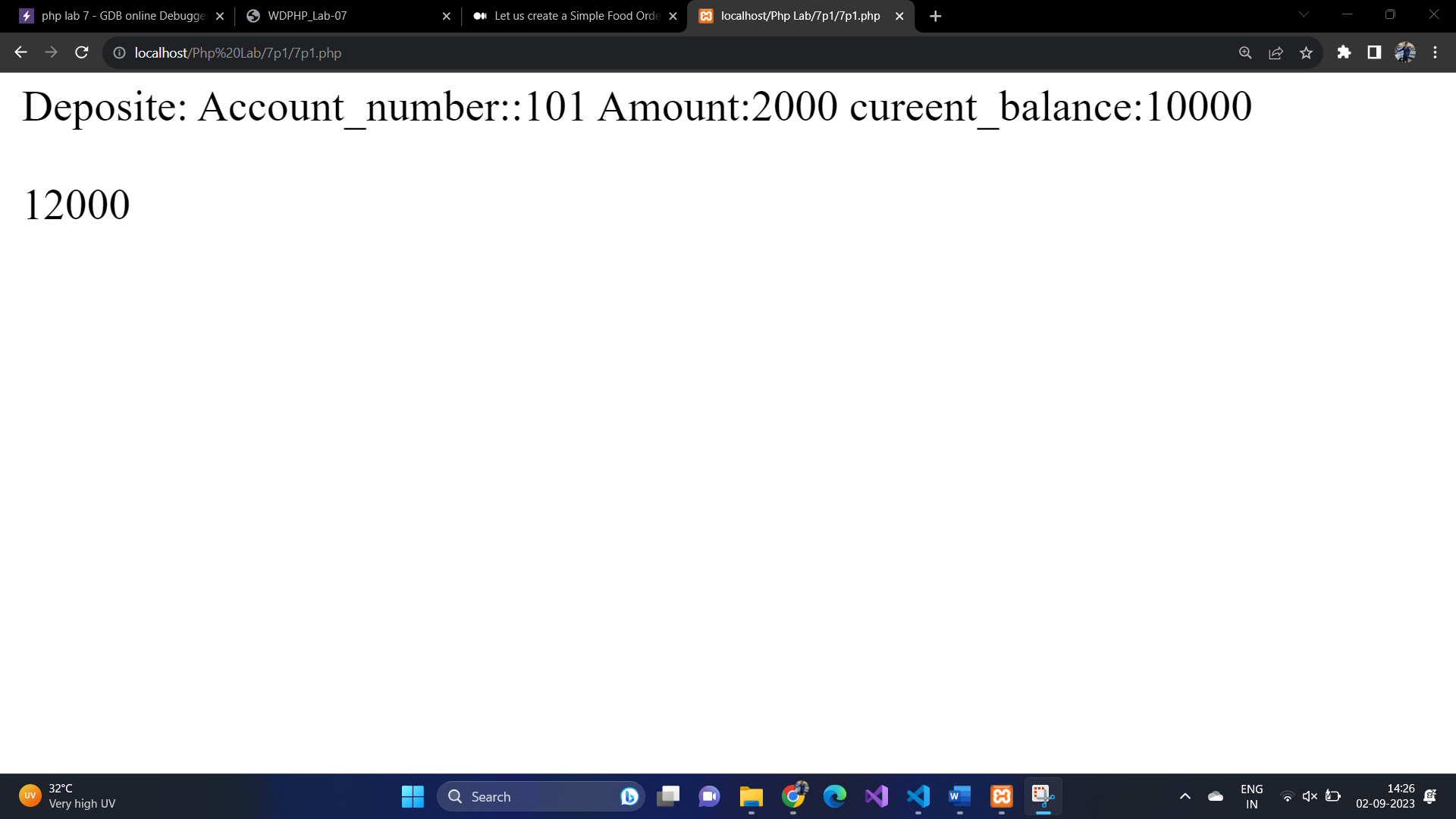
$bank->withdraw();

}

?>

OUTPUT:





Q-2) Create a class Food. Having a property name, category and price. Create a

constructor which assigns the values to the properties. Class should have a method

order () which takes the quantity and calculates the total price. Display () will

display all food details as well as total price of a food item. getTotalPrice () return

the total price of each food item (price\*qty) (\*qty and total price are not class

variables). Give all necessary input by the HTML form. And display output in

HTML form. Take the necessary input of three food items from HTML form. [ 3

objects of a class should be created]. Create a Php function which calculates the

final sum of all the object’s total price and display.

Code:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<title>Food order</title>

</head>

<body>

<form name="form1" action="7p2.php" method="post">

<table width="80%">

<tr>

<th>Food Item</th>

<th>Name</th>

<th>Category</th>

<th>Quantity</th>

<th>Price</th>

</tr>

<tr>

<td>Item 1:</td>

<td><input type="text" name="name1"></td>

<td><select name="category1">

<option value="">--Select Item--</option>

<option value="Vegetable">Vegetable</option>

<option value="Fruits">Fruits</option>

</select></td>

<td><input type="number" name="quantity1" value="0"></td>

<td><input type="text" name="price1"></td> </tr>

<td>Item 2:</td>

<td><input type="text" name="name2"></td>

<td><select name="category2">

<option value="">--Select Item--</option>

<option value="Vegetable">Vegetable</option>

<option value="Fruits">Fruits</option>

</select></td>

<td><input type="number" name="quantity2" value="0"></td>

<td><input type="text" name="price2"></td>

</tr>

<td>Item 3:</td>

<td><input type="text" name="name3"></td>

<td><select name="category3">

<option value="">--Select Item--</option>

<option value="Vegetable">Vegetable</option>

<option value="Fruits">Fruits</option>

</select></td>

<td><input type="number" name="quantity3" value="0"></td>

<td><input type="text" name="price3"></td>

</tr>

<tr>

<td></td>

<td></td>

<td><input type="submit" class="button" value="Create Bill"></td>

<td><input type="reset" class="button" value="Clear"></td>

</tr>

</table>

</form>

</body>

</html>

Data.php

<?php

$name1 = $\_POST["name1"];

$price1 = $\_POST["price1"];

$quantity1 = $\_POST["quantity1"];

$name2 = $\_POST["name2"];

$price2 = $\_POST["price2"];

$quantity2 = $\_POST["quantity2"];

$name3 = $\_POST["name3"];

$price3 = $\_POST["price3"];

$quantity3 = $\_POST["quantity3"];

class Food {

public $name;

public $price;

public $quantity;

public function \_\_construct($name, $price, $quantity) {

$this->name = $name;

$this->price = $price;

$this->quantity = $quantity;

}

public function getTotalPrice($quantity) {

return $this->price \* $quantity;

}

public function order($quantity) {

return $this->getTotalPrice($quantity);

}

public function displayOrder($quantity) {

echo "Name: " . $this->name . "<br>";

echo "Price: " . $this->price . "<br>";

echo "Quantity: " . $quantity . "<br>";

echo "Total Price: " . $this->order($quantity) . "<br>";

}

}

function calculateFinalSum($food1, $food2, $food3, $quantity1, $quantity2, $quantity3) {

return ($food1->getTotalPrice($quantity1) +

$food2->getTotalPrice($quantity2) +

$food3->getTotalPrice($quantity3));

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$food1 = new Food($name1, $price1, $quantity1);

$food2 = new Food($name2, $price2, $quantity2);

$food3 = new Food($name3, $price3, $quantity3);

$finalSum = calculateFinalSum($food1, $food2, $food3, $quantity1, $quantity2, $quantity3);

echo "<h1>Order Summary</h1>";

echo "<h2>Food 1</h2>";

$food1->displayOrder($quantity1);

echo "<h2>Food 2</h2>";

$food2->displayOrder($quantity2);

echo "<h2>Food 3</h2>";

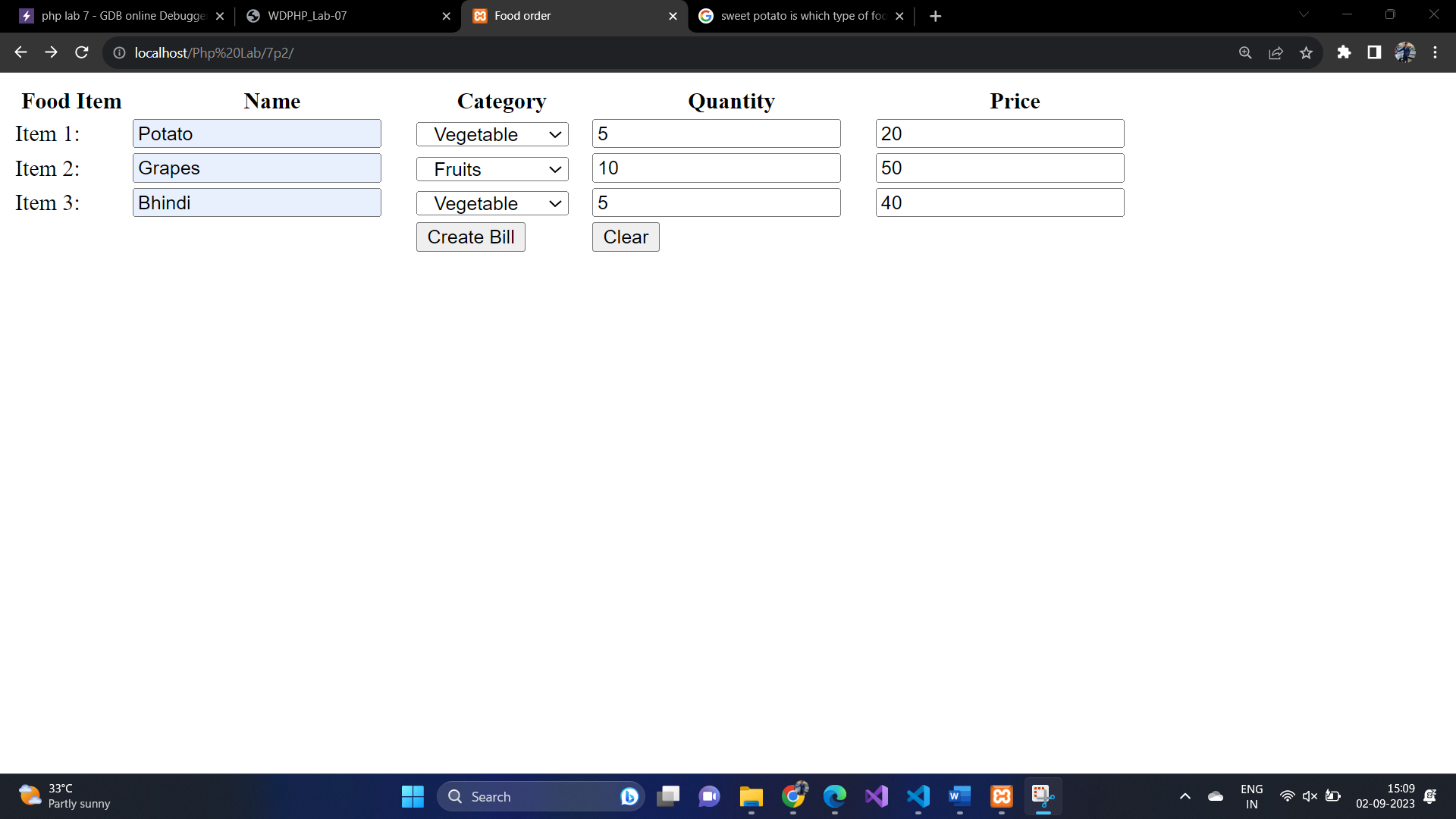
$food3->displayOrder($quantity3);

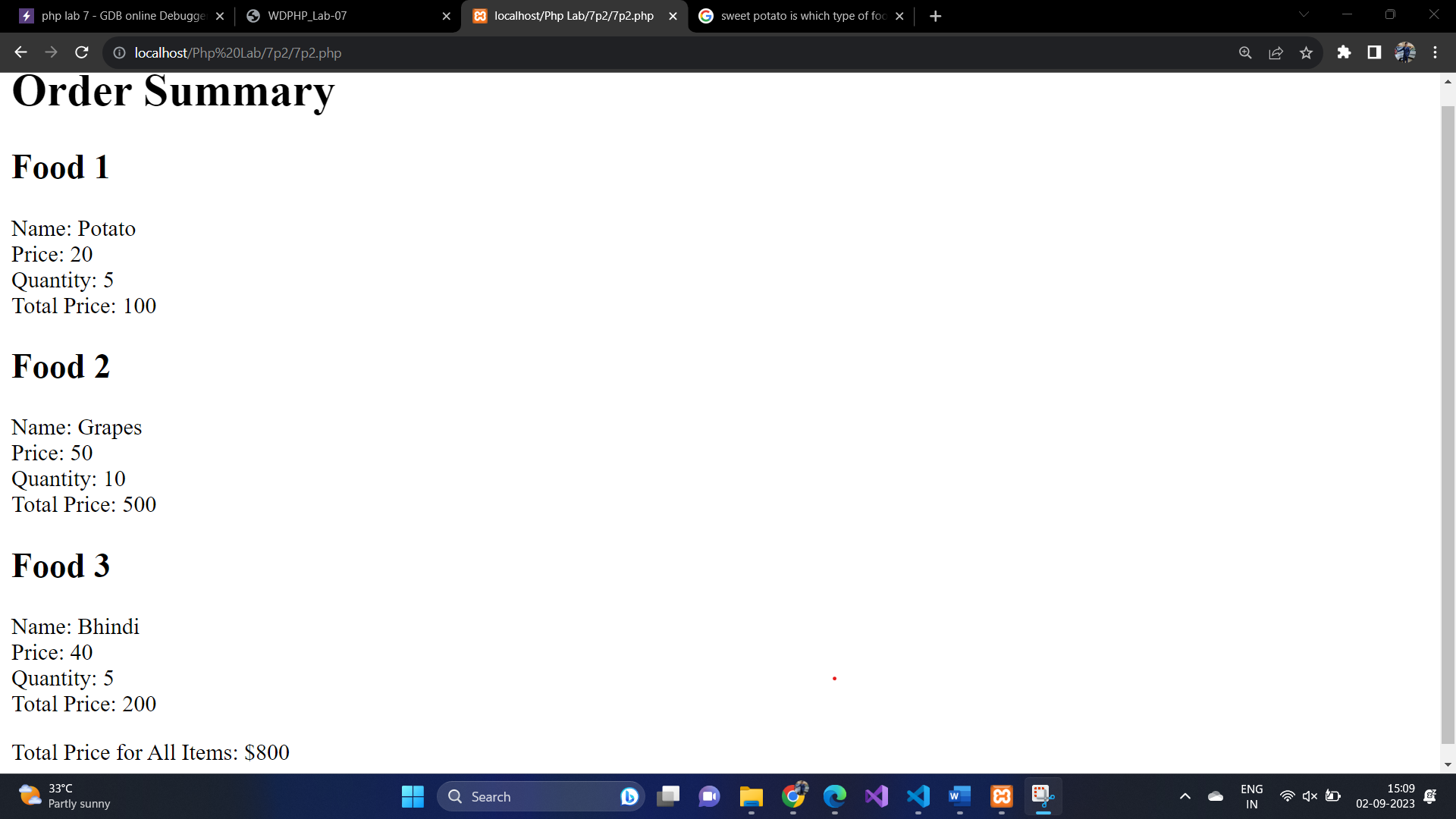
echo "<p>Total Price for All Items: $" . $finalSum . "</p>";

}

?>

OUTPUT





Q-3) Create a class course having properties course name, noofyear. Create setter and

getter public methods for each property. Create a student class which inherits the

course class. Student class should have name, pass out year, result class properties.

Student class contain public method set value () which set all the property’s value.

And display () which display all the values. Create four different objects of student

class and display details of each object.

Code: