

Assignment 7:

Github Link: https://github.com/Lumineskz/IT2-Assignments/tree/main/Unit6_DatabasePHP

1.



```
● ○ ●
1  <?php
2      class Vehicle{
3          private $color;
4          private $model;
5
6          public function __construct($color, $model){
7              $this->color = $color;
8              $this->model = $model;
9          }
10
11         public function getColor(){
12             return $this->color;
13         }
14
15         public function getModel(){
16             return $this->model;
17         }
18     }
19
20     class Car extends Vehicle{
21         private $numDoors;
22
23         public function __construct($color, $model, $numDoors){
24             parent::__construct($color, $model);
25             $this->numDoors = $numDoors;
26         }
27     }
28
29     class Motorcycle extends Vehicle{
30         private $hasSidecar;
31
32         public function __construct($color,$model,$hasSidecar){
33             parent::__construct($color,$model);
34             $this->hasSidecar = $hasSidecar;
35         }
36     }
37
38     $car = new Car("red", "Toyota", 4);
39     echo "Car Model: " . $car->getModel() . ", Color: " . $car->getColor() . "<br>";
40
41     $motorcycle = new Motorcycle("blue", "Honda", false);
42     echo "Motorcycle Model: " . $motorcycle->getModel() . ", Color: " . $motorcycle->getColor() . "<br>";
43 ?>
```

Output:

The screenshot shows a web browser window with two tabs open. The left tab is titled "localhost/unit7/Abstraction/interfa" and the right tab is titled "localhost/unit7Assignment/Vehicle". The main content area displays the output of a PHP script named "VehicleMgmtSys.php". The output consists of two lines of text: "Car Model: Toyota, Color: red" and "Motorcycle Model: Honda, Color: blue".

```
Car Model: Toyota, Color: red
Motorcycle Model: Honda, Color: blue
```

2.

```
1  <?php
2      class BankAccount{
3          private $accountNumber;
4          private $balance;
5          private $accountHolder;
6          private $pin;
7
8          public function __construct($accountHolder, $accountNumber, $pin){
9              $this->accountHolder = $accountHolder;
10             $this->accountNumber = $accountNumber;
11             $this->balance = 0;
12             $this->pin = $pin;
13         }
14
15         public function deposit($amount){
16             if($amount > 0){
17                 $this->balance += $amount;
18                 return "Deposited: Rs." . $amount . "<br>";
19             } else {
20                 return "Deposit amount must be positive.<br>";
21             }
22         }
23
24         public function withdraw($amount,$inputPin){
25             if($inputPin!==$this->pin){
26                 return "Invalid PIN.<br>";
27             }else{
28                 if($amount > 0 && $amount <= $this->balance){
29                     $this->balance -= $amount;
30                     return "Withdrew: Rs." . $amount . "<br>";
31                 } else {
32                     return "Insufficient balance or invalid amount.<br>";
33                 }
34             }
35         }
36
37         public function getBalance($inputPin){
38             if($inputPin === $this->pin){
39                 return "Current Balance: Rs." . $this->balance . "<br>";
40             } else {
41                 return "Invalid PIN.<br>";
42             }
43         }
44
45         public function changePin($oldPin, $newPin){
46             if($oldPin === $this->pin){
47                 $this->pin = $newPin;
48                 return "PIN changed successfully.<br>";
49             } else {
50                 return "Invalid old PIN.<br>";
51             }
52         }
53     }
54
55     $ba = new BankAccount("John Doe", "123456789", "4321");
56     echo $ba->deposit(500);
57     echo $ba->getBalance("4321");
58     echo $ba->withdraw(200, "4321");
59     echo $ba->getBalance("4321");
60     echo $ba->changePin("4321", "1234");
61 ?>
```

Output:

Deposited: Rs.500
Current Balance: Rs.500
Withdrew: Rs.200
Current Balance: Rs.300
PIN changed successfully.

3.

```
<?php

// Base abstract Person class

abstract class Person {

    protected $name;

    protected $email;

    protected $phone;

    public function __construct($name, $email, $phone) {

        $this->name = $name;

        $this->email = $email;

        $this->phone = $phone;

    }

    abstract public function getRole();

    public function getName() { return $this->name; }

    public function getEmail() { return $this->email; }

    public function getPhone() { return $this->phone; }

}

// Student class

class Student extends Person {

    private $studentId;

    private $course;

    private $semester;
```

```
private $marks = [];  
  
public function __construct($name, $email, $phone, $studentId, $course, $semester) {  
    parent::__construct($name, $email, $phone);  
    $this->studentId = $studentId;  
    $this->course = $course;  
    $this->semester = $semester;  
}  
  
public function addMarks($subject, $mark) {  
    $this->marks[$subject] = $mark;  
}  
  
public function calculateGPA() {  
    if (empty($this->marks)) return 0;  
    $total = array_sum($this->marks);  
    return round($total / count($this->marks), 2);  
}  
  
public function getRole() {  
    return "Student";  
}  
  
public function getStudentId() { return $this->studentId; }  
public function getCourse() { return $this->course; }  
public function getSemester() { return $this->semester; }
```

```
public function getMarks() { return $this->marks; }

}

// Teacher class

class Teacher extends Person {

    private $teacherId;
    private $department;
    private $subjects = [];

    public function __construct($name, $email, $phone, $teacherId, $department) {
        parent::__construct($name, $email, $phone);
        $this->teacherId = $teacherId;
        $this->department = $department;
    }

    public function addSubject($subject) {
        $this->subjects[] = $subject;
    }

    public function getRole() {
        return "Teacher";
    }

    public function getTeacherId() { return $this->teacherId; }
    public function getDepartment() { return $this->department; }
    public function getSubjects() { return $this->subjects; }
}
```

```
}

// Management System class

class ManagementSystem {

    private $students = [];

    private $teachers = [];


    public function addStudent($student) {

        $this->students[] = $student;

    }

    public function addTeacher($teacher) {

        $this->teachers[] = $teacher;

    }

    public function displayStudents() {

        echo "<h2>Student List</h2>";

        echo "<table border='1' cellpadding='10'>";

        echo "<tr><th>ID</th><th>Name</th><th>Email</th><th>Phone</th><th>Course</th><th>Semester</th><th>GPA</th></tr>";

        foreach ($this->students as $student) {

            echo "<tr>";

            echo "<td>" . $student->getStudentId() . "</td>";

            echo "<td>" . $student->getName() . "</td>";

            echo "<td>" . $student->getEmail() . "</td>";

            echo "<td>" . $student->getPhone() . "</td>";


        }

    }

}
```

```

echo "<td>" . $student->getCourse() . "</td>";
echo "<td>" . $student->getSemester() . "</td>";
echo "<td>" . $student->calculateGPA() . "</td>";
echo "</tr>";
}

echo "</table><br>";
}

public function displayTeachers() {
    echo "<h2>Teacher List</h2>";
    echo "<table border='1' cellpadding='10'>";
    echo
"<tr><th>ID</th><th>Name</th><th>Email</th><th>Phone</th><th>Department</th><th>
Subjects</th></tr>";
    foreach ($this->teachers as $teacher) {
        echo "<tr>";
        echo "<td>" . $teacher->getTeacherId() . "</td>";
        echo "<td>" . $teacher->getName() . "</td>";
        echo "<td>" . $teacher->getEmail() . "</td>";
        echo "<td>" . $teacher->getPhone() . "</td>";
        echo "<td>" . $teacher->getDepartment() . "</td>";
        echo "<td>" . implode(", ", $teacher->getSubjects()) . "</td>";
        echo "</tr>";
    }
    echo "</table><br>";
}

```

```
$system = new ManagementSystem();

$student1 = new Student("John Doe", "john@example.com", "1234567890", "S001", "BCA",
1);

$student1->addMarks("PHP", 85);

$student1->addMarks("Database", 90);

$student1->addMarks("HTML", 88);

$student2 = new Student("Jane Smith", "jane@example.com", "9876543210", "S002",
"BCSIT", 2);

$student2->addMarks("PHP", 92);

$student2->addMarks("Database", 88);

$student2->addMarks("HTML", 95);

$system->addStudent($student1);

$system->addStudent($student2);

$teacher1 = new Teacher("Mr. Ahmed", "ahmed@example.com", "5555555555", "T001",
"BIT");

$teacher1->addSubject("PHP");

$teacher1->addSubject("Database");

$teacher2 = new Teacher("Ms. Sarah", "sarah@example.com", "6666666666", "T002",
"BBA");

$teacher2->addSubject("HTML");

$teacher2->addSubject("CSS");
```

```
$system->addTeacher($teacher1);  
$system->addTeacher($teacher2);
```

```
$system->displayStudents();  
$system->displayTeachers();
```

```
?>
```

Output:

← ⌂ ⓘ localhost/unit7Assignment/miniStudentMgmtSystem.php

Student List

ID	Name	Email	Phone	Course	Semester	GPA
S001	John Doe	john@example.com	1234567890	BCA	1	87.67
S002	Jane Smith	jane@example.com	9876543210	BCSIT	2	91.67

Teacher List

ID	Name	Email	Phone	Department	Subjects
T001	Mr. Ahmed	ahmed@example.com	5555555555	BIT	PHP, Database
T002	Ms. Sarah	sarah@example.com	6666666666	BBA	HTML, CSS