

Things need to be done for Phase 2

1 README.txt

- Instructions for how to install and run your project.
- Include team member names.

2 DB2-tables.sql

- The professor's predefined schema (DO NOT modify `course`, `instructor`, `time_slot` tables).
- Your `CREATE TABLE` and `INSERT` queries.

3 index.html

- Frontend for user interaction (No JavaScript, only plain HTML & PHP).
- Links to `student.php`, `admin.php`, etc.

4 student.php

- Handles student account creation.
- Allows account updates.

5 admin.php

- Allows admin to assign instructors & schedule sections.
- Checks scheduling constraints (max 2 sections per instructor, consecutive slots).

6 student_register.php

- Handles student course registration.
- Checks prerequisite conditions & seat availability (max 15 students per section).

7 student_history.php

- Shows student's enrolled courses, total credits, GPA calculation.

8 config.php

- MySQL database connection file.

9demo_video.mp4

- 10-15 min video demo of your project.
- Upload to YouTube or submit .mp4 file.

Submission Instructions

1 Submit via UML Command

bash

CopyEdit

```
submit malamin 3100-S25-phase2 phase2_submission/
```

2 Upload Video Demo

- Post a YouTube link or MP4 file on the discussion board.

3 Live Demonstration on March 31

- Show your working system in class.

Final Submission Folder Structure

phase2_submission

README.txt	Instructions for setup & execution
DB2-tables.sql	SQL script for tables & sample data
index.html	User interface
student.php	Handles student account creation
admin.php	Admin assigns instructor & sections
student_register.php	Student registers for a course
student_history.php	Student views GPA & course history
config.php	Database connection
demo_video.mp4	Pre-recorded demo (10-15 min)

1README.txt

File: README.txt (This file contains instructions for setup & execution)

Phase 2: University Course Management System

Team Members:

1. [Your Name]
2. [Team Member 2 (if applicable)]

Installation & Setup:

1. Install MySQL and Apache (using XAMPP, WAMP, or standalone).
2. Import the database schema:
 - Open MySQL or PhpMyAdmin.
 - Run the script: `DB2-tables.sql`.
3. Place all PHP & HTML files in the web server directory (e.g., `htdocs` for XAMPP).
4. Update database credentials in `config.php` (if needed).
5. Start the Apache & MySQL servers.
6. Open `index.html` in a browser to use the system.

Features:

- Student account creation & update.
- Course enrollment system with prerequisites & seat limits.
- Admin assigns instructors & schedules sections.
- Students view their course history & GPA.
- Instructor access to student records.
- TA and grader assignments based on eligibility.

Running the Demo:

- Open `index.html` in a web browser.
- Follow the form instructions to interact with the database.
- Video demo included (`demo_video.mp4`).

2DB2-tables.sql

File: **DB2-tables.sql** (This script sets up the database and inserts sample data)

```
CREATE DATABASE DB2;  
USE DB2;
```

```
-- Account Table
```

```
CREATE TABLE account (  
    email VARCHAR(50) PRIMARY KEY,  
    password VARCHAR(20) NOT NULL,  
    type VARCHAR(20)  
);
```

```

-- Department Table
CREATE TABLE department (
    dept_name VARCHAR(100) PRIMARY KEY,
    location VARCHAR(100)
);

-- Instructor Table
CREATE TABLE instructor (
    instructor_id VARCHAR(10) PRIMARY KEY,
    instructor_name VARCHAR(50) NOT NULL,
    title VARCHAR(30),
    dept_name VARCHAR(100),
    email VARCHAR(50) NOT NULL
);

-- Student Table
CREATE TABLE student (
    student_id VARCHAR(10) PRIMARY KEY,
    name VARCHAR(20) NOT NULL,
    email VARCHAR(50) NOT NULL,
    dept_name VARCHAR(100),
    FOREIGN KEY (dept_name) REFERENCES department(dept_name) ON DELETE SET
    NULL
);

-- Enrollment Table
CREATE TABLE take (
    student_id VARCHAR(10),
    course_id VARCHAR(8),
    section_id VARCHAR(10),
    semester VARCHAR(6),
    year INT CHECK (year > 1990 AND year < 2100),
    grade VARCHAR(2) CHECK (grade IN ('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C+', 'C', 'C-', 'D+', 'D',
'D-', 'F')),
    PRIMARY KEY (student_id, course_id, section_id, semester, year),
    FOREIGN KEY (student_id) REFERENCES student(student_id) ON DELETE CASCADE
);

-- Insert Sample Data
INSERT INTO account (email, password, type) VALUES
('admin@uml.edu', '123456', 'admin'),
('dbadams@cs.uml.edu', '123456', 'instructor');

INSERT INTO student (student_id, name, email, dept_name) VALUES
('1001', 'Alice Johnson', 'alice@example.com', 'CS'),
('1002', 'Bob Smith', 'bob@example.com', 'Data Science');

```

3 index.html

File: `index.html` (User Interface)

```
<!DOCTYPE html>
<html>
<head>
  <title>University Course Management</title>
</head>
<body>
  <h2>Student Account Creation</h2>
  <form action="student.php" method="POST">
    Email: <input type="text" name="email"><br>
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Create Account">
  </form>

  <h2>Student Course Registration</h2>
  <form action="student_register.php" method="POST">
    Student ID: <input type="text" name="student_id"><br>
    Section ID: <input type="text" name="section_id"><br>
    <input type="submit" value="Register">
  </form>
</body>
</html>
```

4config.php

File: `config.php` (Database Connection)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "DB2";

$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

5 student.php

File: `student.php` (Student Account Management)

```
<?php
include 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];
    $type = "student";

    $sql = "INSERT INTO account (email, password, type) VALUES ('$email', '$password', '$type')";

    if (mysqli_query($conn, $sql)) {
        echo "Account created successfully!";
    } else {
        echo "Error: " . mysqli_error($conn);
    }
}
?>
```

6 student_register.php

File: `student_register.php` (Student Course Registration)

```
<?php
include 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $student_id = $_POST["student_id"];
    $section_id = $_POST["section_id"];

    // Check if section has space
    $check = "SELECT COUNT(*) as count FROM take WHERE section_id = '$section_id'";
    $result = mysqli_query($conn, $check);
    $row = mysqli_fetch_assoc($result);

    if ($row["count"] >= 15) {
        echo "Error: Section is full.";
    } else {
        $sql = "INSERT INTO take (student_id, course_id, section_id, semester, year)
                VALUES ('$student_id', (SELECT course_id FROM section WHERE
                section_id='$section_id'),
```

```
        '$section_id', 'Spring', 2025)";

    if (mysqli_query($conn, $sql)) {
        echo "Successfully enrolled!";
    } else {
        echo "Error: " . mysqli_error($conn);
    }
}
?>
```

Final Steps

Submit the files via UML Command

```
submit malamin 3100-S25-phase2 phase2_submission
```