# Department of Computing

**CS344: Web Engineering**

**Class: BESE – 13AB**

# Lab 11: Attendance Recorder (Laravel)

**Date: 28 Nov 2024**

# Time: 09:00AM – 11:50 AM/ 02:00 AM - 04:50 PM

# Instructor: Dr. Ayesha Hakim

# Lab Engineer: Ms. Ayesha Asif

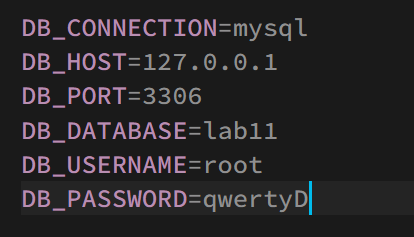
## ****Overview****

[**https://github.com/amunim/WE-semester-5-Labs**](https://github.com/amunim/WE-semester-5-Labs)

This document provides a detailed explanation of the Attendance Management Application implemented in PHP Laravel, designed to manage student attendance at NUST. The project fulfills the specified requirements by enabling teachers to mark attendance and view session records while allowing students to view only their attendance. The application adopts a master view design and enforces role-based authentication for secure access.

To run the project, first make sure to install php, node, composer. Then run the following commands.

Go to .env and make sure to update DB\_\* variables:



```

Composer install

Npm install

Php artisan migrate

Php artisan key:generate

Npm run build

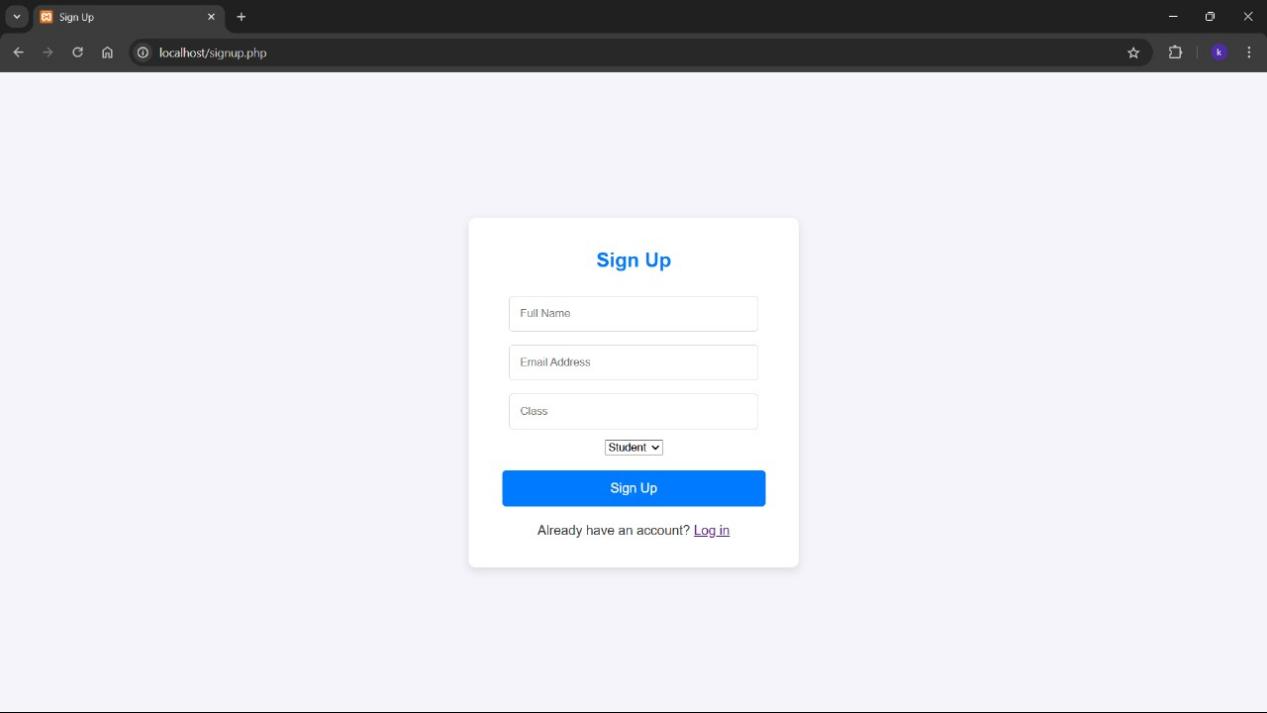
Php artisan serve

```

Now you can visit the login page, check user seed.php to get data of login

## ****Core Features****

### ****Authentication****

1. 

* **Implementation**:
  + Users log in with valid credentials. The application verifies the user’s role (teacher or student) using session variables.
  + Based on the role, users are redirected:
    - **Teachers**: Directed to a dashboard where they can manage attendance records.
    - **Students**: Directed to their attendance view.
* **Code Reference**:
* Route::**middleware**('auth')->**group**(function () {
* Route::**get**('/student-dashboard', [StudentDashboardController::class, 'index'])
* ->**name**('student.dashboard')
* ->**middleware**('role:student');
* Route::**get**('/teacher-dashboard', [TeacherDashboardController::class, 'index'])
* ->**name**('teacher.dashboard')
* ->**middleware**('role:teacher');
* Route::**get**('/mark-attendance', [AttendanceController::class, 'showForm'])->**name**('mark.attendance');
* Route::**post**('/mark-attendance', [AttendanceController::class, 'markAttendance'])->**name**('mark.attendance.submit');
* Route::**get**('/add-class', [ClassController::class, 'showForm'])->**name**('add.class');
* Route::**post**('/add-class', [ClassController::class, 'storeClass'])->**name**('add.class.submit');
* });
  + This ensures only authenticated users access the system.

### 2. ****Master View Design****

* **Implementation**:
  + The application follows a single master layout for consistency across all pages.
  + Teachers and students share the same basic structure, with views tailored to their roles.
* **App blade.php Reference**:
* <!DOCTYPE html>
* <html lang="en">
* <head>
* <meta charset="UTF-8">
* <meta name="viewport" content="width=device-width, initial-scale=1.0">
* <title>Laravel Project</title>
* <link rel="stylesheet" href="{{ **asset**('css/styles.css') }}">
* </head>
* <body>
* @yield('content')
* </body>
* </html>
  + The header is unified and dynamically highlights the active view.

### 3. ****Teacher Features****

#### a. ****View Sessions**** WhatsApp Image 2024-11-26 at 21.54.21_63497e5d

* **Implementation**:
  + Teachers can view:
    - **Current Sessions**: Ongoing sessions.
    - **Past Sessions**: Completed sessions with attendance records.
    - **Upcoming Sessions**: Scheduled sessions.
* **SQL Query**:

SELECT id, starttime, endtime

FROM class

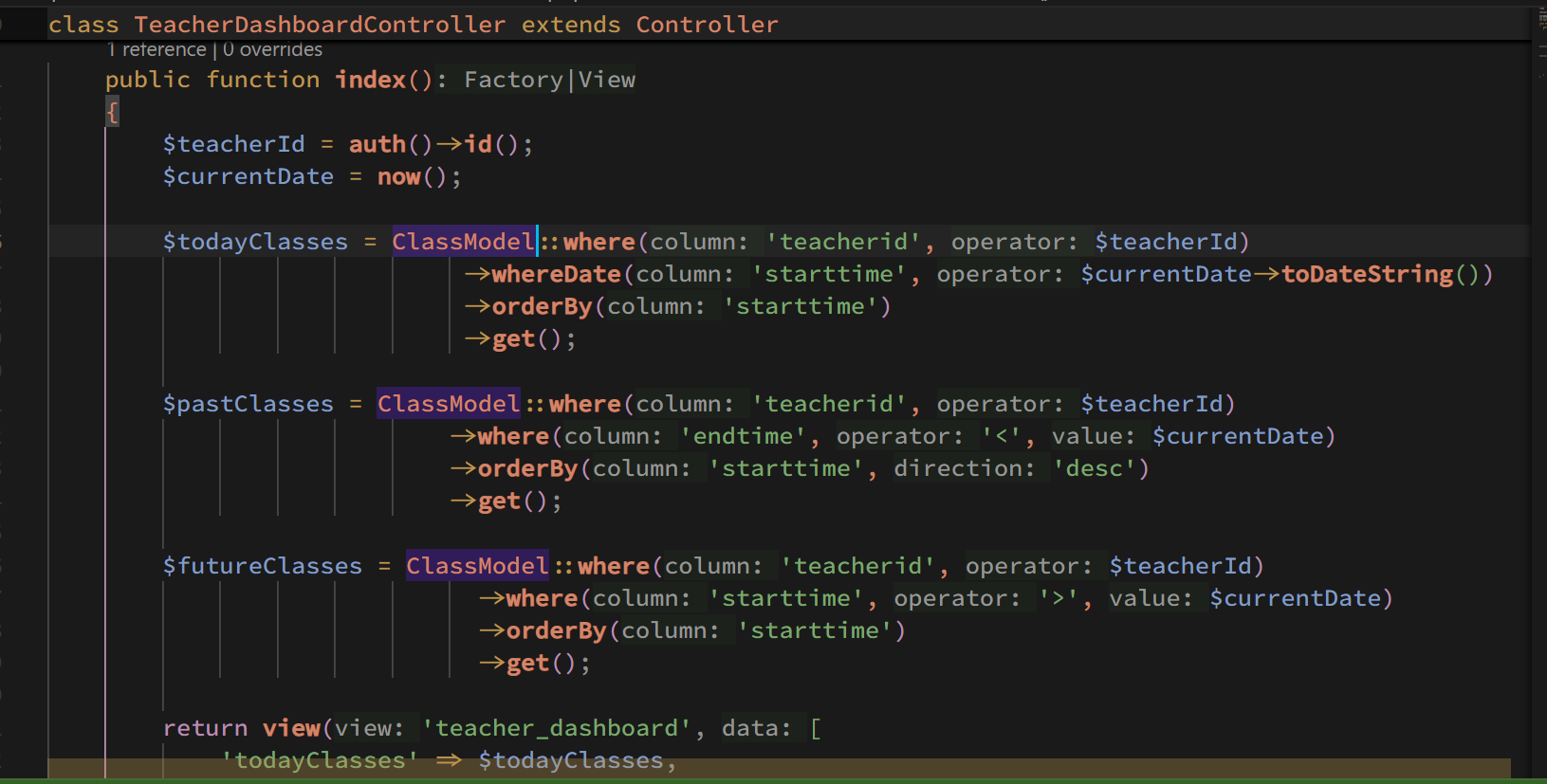
WHERE teacherid = ?

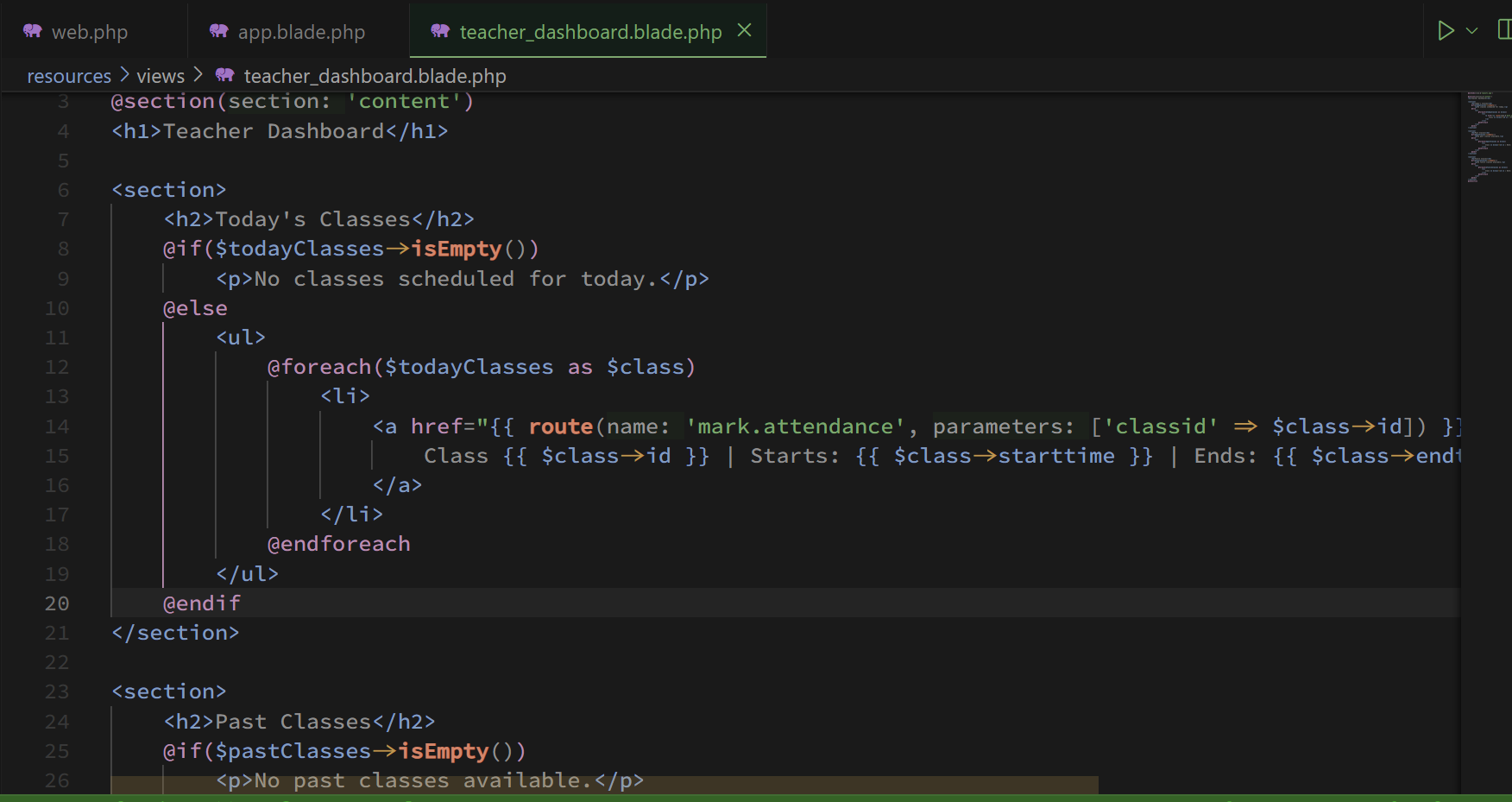
AND (

starttime <= NOW() OR starttime > NOW()

);

In Laravel with ORM becomes





* + Sessions are categorized based on starttime.

#### b. ****Mark Attendance**** WhatsApp Image 2024-11-26 at 21.34.56_72372097

* **Implementation**:

A screen shot of a computer program

Description automatically generated

* + Teachers can mark or update attendance for any student in any session.
  + Attendance is recorded with isPresent and optional comments.
* **PHP Code**:

$stmt\_insert = $conn->prepare("

INSERT INTO attendance (classid, studentid, isPresent, comments)

VALUES (?, ?, ?, ?)

");

$stmt\_insert->bind\_param("iiis", $classFilter, $studentId, $isPresent, $comment);

$stmt\_insert->execute();

* + This ensures attendance records are either inserted or updated based on existing data.

#### c. ****View Attendance Records** Screenshot_2024-11-26_201947[1]**

* **Implementation**:
  + Teachers can view attendance for any session with details of student participation.
* **Code Reference**:

$stmt\_attendance = $conn->prepare("

SELECT a.studentid, u.fullname, a.isPresent, a.marked\_at

FROM attendance a

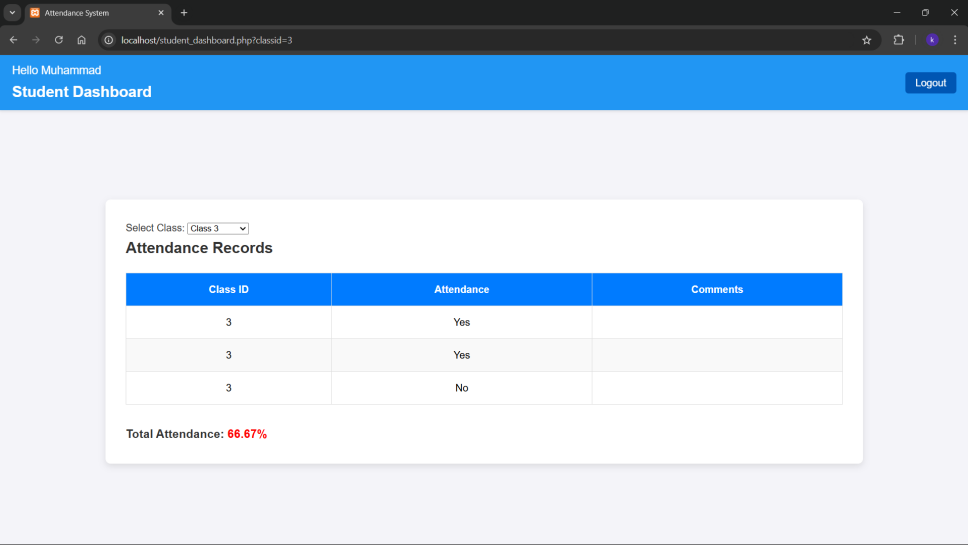
JOIN user u ON a.studentid = u.id

WHERE a.classid = ?

");

### 4. ****Student Features****

#### a. ****View Attendance****



* **Implementation**:
  + Students can view their attendance records with color-coded status:
    - **Red (Below 75%)**: Indicates critical attendance.
    - **Yellow (75-85%)**: Warns of low attendance.
    - **Green (Above 85%)**: Marks satisfactory attendance.
* **Blade php Code**:

#### A computer code on a black background Description automatically generatedb. ****Interface Design****

#### ****Teacher add class****

## ****Database Design****

The database is normalized and uses the following tables:

### 1. user ****Table****

Stores user data with roles.

CREATE TABLE `user` (

`id` int(50) NOT NULL AUTO\_INCREMENT,

`fullname` varchar(200) NOT NULL,

`email` varchar(200) NOT NULL,

`class` varchar(10) NOT NULL,

`role` enum('teacher','student','admin') NOT NULL,

PRIMARY KEY (`id`)

);

### 2. class ****Table****

Records details of all sessions.

CREATE TABLE `class` (

`id` int(50) NOT NULL,

`teacherid` int(50) NOT NULL,

`starttime` datetime NOT NULL,

`endtime` datetime NOT NULL,

`credit\_hours` int(11) NOT NULL,

PRIMARY KEY (`id`)

);

### 3. attendance ****Table****

Tracks attendance with optional comments.

CREATE TABLE `attendance` (

`classid` int(50) NOT NULL,

`studentid` int(50) NOT NULL,

`isPresent` tinyint(1) NOT NULL,

`comments` varchar(200),

PRIMARY KEY (`classid`, `studentid`)

);

## ****Visual Design****

* **Color Coding**:
  + Attendance status is displayed with consistent visual indicators:
    - **Red**: Critical.
    - **Yellow**: Warning.
    - **Green**: Satisfactory.
* **Example CSS**:

.low-attendance { color: red; font-weight: bold; }

.mid-attendance { color: yellow; }

.high-attendance { color: green; }

## Work Distribution

| **Task** | **Assigned To** | **Details** |
| --- | --- | --- |
| **Authentication Module** | Abdul Munim | Implement user login/logout functionality, verify roles, and secure pages based on roles. |
| **Database Design and Setup** | Ammar Shahzad | Create and populate the database tables (user, class, attendance) using MySQL. |
| **Master View Design** | Abdul Munim | Develop a unified interface with a header, navigation, and dynamic page content. |
| **Teacher Dashboard - View Sessions** | Abdul Munim | Implement logic for displaying past, current, and upcoming sessions for teachers. |
| **Teacher Dashboard - Mark Attendance** | Abdul Munim | Create the form and logic to allow teachers to mark or update attendance for any session. |
| **Student Dashboard - View Attendance** | Ammar Shahzad | Implement functionality to display attendance records for students with color-coded statuses. |
| **Attendance Management Logic** | Abdul Munim | Write backend logic to handle attendance recording, updating, and data retrieval. |
| **Color Coding for Attendance** | Ammar Shahzad | Add CSS and logic for color-coding attendance percentages (Red, Yellow, Green). |
| **Interface Enhancements and Validation** | Abdul Munim | Add form validations and improve user experience (e.g., error handling and dynamic updates). |
| **Testing and Debugging** | Both | Test and debug the application to ensure smooth functionality and error-free execution. |
| **Final Deployment and Documentation Preparation** | Both | Deploy the project and prepare the final documentation for submission. |

### ****Explanation of Division****:

**Backend Work**:

* 1. Abdul Munim is responsible for implementing core PHP functionality like authentication and attendance logic, Controllers, Blade views, middlewares.
  2. Ammar Shahzad focuses on setting up the database and implementing dashboard views.

**Frontend Work**:

* 1. Ammar Shahzad works on creating and improving user interfaces.
  2. Abdul Munim ensures attendance records, classes, current and upcoming sessions are displayed properly with color-coded statuses.

**Shared Tasks**:

* 1. Both work collaboratively on testing, debugging, and final deployment to ensure quality.

## ****Conclusion****

The Attendance Management Application aligns with the requirements for Lab 11, providing:

* A robust and secure authentication system.
* A unified interface with role-based access.
* Comprehensive attendance tracking for teachers and students.
* Intuitive visual cues for attendance performance.