Instructions on How to Set Up and Run the Student Management System Locally

This guide will walk you through the steps required to set up and run the **Student Management System with Task Assignment** locally on your machine.

Prerequisites

Before setting up the project, ensure you have the following installed:

- **PHP** (>= 8.0) – You can download it from here.
- **Composer** – The PHP package manager for managing dependencies, install it from here.
- **Laravel** – You can install Laravel by following the instructions here.
- **MySQL or SQLite** – Install MySQL or use SQLite for the database.
- **Node.js** and **NPM** – For front-end assets, install from here.

Step 1: Clone the Repository

1. First, clone the project repository to your local machine using git.

bash

Copy code

git clone https://github.com/AkkiD7/student_managemnet_laravel

Step 2: Install Project Dependencies

2. **Install PHP dependencies** using Composer:

composer install

3. **Install front-end dependencies** (for compiling assets like CSS/JS) using NPM:

npm install

Step 3: Set Up the Database

4. **Create a new database** in MySQL (or use an existing one) for the project. You can do this via MySQL command line or a GUI tool like PhpMyAdmin.

```
CREATE DATABASE student_management;
```

5. **Configure database settings** in the .env file in the root directory of the project. Open .env and set the following values:

```
DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=student_management

DB_USERNAME=root

DB_PASSWORD=your_password
```

6. **Run database migrations** to create the tables for students, courses, enrollments, tasks, and users.

```
php artisan migrate
```

7. If required, **seed the database** with sample data (users, courses, etc.):

```
php artisan db:seed
```

Step 4: Set Up Authentication

8. **Run the Laravel authentication scaffolding** (this will generate the necessary views and routes for login, registration, and user management):

```
php artisan make:auth
```

This will create the authentication views (login, registration, etc.) in the resources/views/auth directory.

Step 5: Configure the Roles (Admin, Teacher, Student)

9. **Seed users with roles** (teacher, student). You can do this in the UsersTableSeeder:

Copy code

```
// database/seeders/UsersTableSeeder.php

use Illuminate\Support\Facades\Hash;
use App\Models\User;

class UsersTableSeeder extends Seeder
{
   public function run()
   {
      // Teacher User
      User::create([
         'name' => 'Teacher User',
         'email' => 'teacher@example.com',
         'password' => Hash::make('teacher123'),
         'role' => 'teacher',
      ]);

      // Student User
      User::create([
```

```
            'name' => 'Student User',
            'email' => 'student@example.com',
            'password' => Hash::make('student123'),
            'role' => 'student',
        ]);
    }
}
```

Then, run the seed command:

```
php artisan db:seed --class=UsersTableSeeder
```

Step 6: Set Up the Web Server

10. **Serve the application** using Laravel's built-in development server:

```
php artisan serve
```

By default, this will run the application on http://127.0.0.1:8000.

Step 7: Access the Application

11. Open a web browser and go to:

```
http://127.0.0.1:8000
```

- **Login as Teacher**: Use teacher@example.com with password teacher123.
- **Login as Student**: Use student@example.com with password student123.

Step 8: Testing and Functionality

After logging in, the system will allow you to:

- **Teacher**: Create, update, and delete tasks for students, as well as track task completion.
- **Student**: View their assigned tasks and mark them as completed.

----------END----------