Project Setup and Running Guide

Prerequisites

- PHP (>= 8.0)
- Composer
- MySQL
- XAMPP (for local server) or Wamp server

Steps:-

1. Clone the Repository:- git clone <repository-url>
cd <project-directory>

2. Install Dependencies

Composer install

npm install

npm run dev

3. Set Up Environment Variables

Copy .env.example to .env:-

cp .env.example .env

Edit the .env file to set your database configuration and other necessary configurations.

APP_NAME=Laravel

APP_ENV=local

APP_KEY=base64:uymPiG4gaCgm+gpgl0UO1l9UMkmd6tO5mLQeW5x8K6o=

APP_DEBUG=true

APP_TIMEZONE=UTC

APP_URL=http://localhost

DB_CONNECTION=mysql

```
DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=task_management_system

DB_USERNAME=root

DB_PASSWORD=yourpassword
```

Generate Application Key

```
php artisan key:generate
```

Run Migrations

```
php artisan migrate
```

Start the Local Development Server

```
php artisan serve
```

The application will be available at http://localhost:8000.

Database Configuration

- Make sure MySQL is running (if using XAMPP, start MySQL from the XAMPP control panel).
- Create a new database named task_management_system.
- Ensure the database credentials in the .env file match your MySQL setup.

Folder Structure Overview

- **app/Http/Controllers:** Contains the controllers, including

  TaskController.php.

- **app/Models:** Contains the Eloquent models, including **Task.php.**

- **resources/views:** Contains the Blade templates for the front-end.
- **routes/web.php:** Contains the route definitions.

TaskController Overview

- **index():** Fetches and displays all tasks with search functionality.

- **create():** Shows the form for creating a new task.
- **store(Request $request):** Validates and stores a new task.
- **show(Task $task):** Displays a specific task.
- **edit(Task $task):** Shows the form for editing a task.
- **update(Request $request, Task $task):** Validates and updates a task.
- **destroy(Task $task):** Deletes a task.

Views Overview

- **resources/views/tasks/index.blade.php:** Displays the list of tasks with search.
- **resources/views/tasks/create.blade.php:** Form for creating a new task.
- **resources/views/tasks/edit.blade.php:** Form for editing a task.
- **resources/views/tasks/show.blade.php:** Displays a specific task.

Commands Overview

- **Starting the server: php artisan serve**
- **Running migrations: php artisan migrate**
- **Generating application key: php artisan key:generate**

Additional Tips

- For development, use php artisan serve to run the server.
- Use php artisan migrate:refresh to reset and re-run all migrations if needed.
- Regularly commit your changes and push to your repository to keep track of your progress.

Troubleshooting

- **MySQL Connection Issues**: Ensure MySQL is running and the credentials in the .env file are correct.

- Permission Issues: Make sure you have the necessary permissions for the project files and directories.

- Error Logs: Check the storage/logs/laravel.log file for detailed error messages.

Folder Structure Overview

- **app/Http/Controllers:** Contains the controllers, including TaskController.php.
- **app/Models:** Contains the Eloquent models, including Task.php.
- **resources/views:** Contains the Blade templates for the front-end.
- **routes/web.php:** Contains the route definitions.

TaskController Overview

- **index():** Fetches and displays all tasks with search functionality.
- **create():** Shows the form for creating a new task.
- **store(Request $request):** Validates and stores a new task.
- **show(Task $task):** Displays a specific task.
- **ediľ(Task $task):** Shows the form for editing a task.
- update(Request $request, Task $task): Validates and updates a task.
- destroy(Task $task): Deletes a task.

Views Overview

- resources/views/tasks/index.blade.php: Displays the list of tasks with search.
- resources/views/tasks/create.blade.php: Form for creating a new task.
- resources/views/tasks/edit.blade.php: Form for editing a task.
- resources/views/tasks/show.blade.php: Displays a specific task.

Commands Overview

- Starting the server: php artisan serve
- Running migrations: php artisan migrate
- Generating application key: php artisan key:generate

Troubleshooting

1. MySQL Connection Issues: Ensure MySQL is running and the credentials in the .env file are correct.
2. Permission Issues: Make sure you have the necessary permissions for the project files and directories.
3. Error Logs: Check the storage/logs/laravel.log file for detailed error messages.