Marsa Maroc Port Management

Introduction

Marsa Maroc Port Management is a web application developed to streamline port operations and manage ship activities efficiently. Built using Laravel and TailwindCSS, this application provides a robust solution for managing daily reports, ship information, personnel, and equipment operations within a port environment.

Features

- **User Authentication:** Secure login and registration system with role-based access control.
- **Daily Reporting:** Create and manage daily reports with search functionality by date.
- **Ship Management:** Detailed information about ships, including operational dates, types, and more.
- **Personnel Management:** Manage personnel information and their assignments to ships and equipment.
- **Equipment Management:** Track equipment operations and their assignments to ships.
- **Stoppage Management:** Record and manage stoppages related to ships and equipment.

Installation

Prerequisites

- PHP 8.0 or higher
- Composer
- Node.js (for compiling assets)
- MySQL

Setup Steps

1. **Clone the Repository:**

```
git clone https://github.com/your-username/your-repo.git
cd your-repo
```

2. **Install Dependencies:**

```
composer install
npm install
```

3. **Environment Configuration:**

```
cp .env.example .env
php artisan key:generate
```

Update the `.env` file with your database credentials.

4. **Database Migration:**

```
php artisan migrate
```

Optionally, you can seed the database with dummy data using the `--seed` flag.

5. **Compile Assets:**

```
npm run dev
```

6. **Start the Development Server:**

```
php artisan serve
```

Visit `http://localhost:8000` in your browser to access the application.

7. Login:
    - **Admin:**
        - Email: **admin@marsa.com**
        - Password: **admin**
    - **User:**
        - Email: **user@marsa.com**
        - Password: **user**

Project Structure

1. **Controllers:** Handle HTTP requests and return responses.
2. **Models:** Interact with the database.
3. **Views:** Blade templates for rendering HTML.
4. **Routes:** Define the application's endpoints.
5. **Seeders:** Populate the database with initial data.

Technologies Used

1. **Laravel:** PHP framework for web application development.
2. **TailwindCSS:** Utility-first CSS framework for styling.
3. **MySQL:** Relational database for data storage.

4. **Laravel Breeze:** Authentication scaffolding for Laravel.
5. **UML:** Utilized for database schema design.

Challenges

- **Solo Development and Technical Hurdles:** The primary challenge was building the project alone. This solo effort resulted in an incomplete implementation and uncovered several technical issues. Despite these hurdles, the experience has provided invaluable insights into the importance of collaboration, and it has highlighted areas for technical enhancement in future projects.