pupsched.

Faculty loading in ease

System Technical Documentation

This is a comprehensive faculty load management system designed to streamline the scheduling process at Polytechnic University of the Philippines - San Juan Campus.

Polytechnic University of the Philippines
College of Computer and Information Sciences
Department of Information Technology

Justine Lloyd Bautista
Mark Jason Fulguerinas
Charles Ezra Ilarde
Regie San Juan
Jesse Mari Mirabel

Assoc. Prof. Noel Gagolinan
Instructor

August 29, 2025

> **Note**
>
> This document ONLY provides a brief and basic technical overview of the **PUPSCHED** project, a faculty scheduling system designed for ease of use and automation. To test and practice laTex documentation workflow.

System Overview

Key Features

Feature	Description
Secure Authentication	Implements JWT-based login, session handling, and role-based access control for system security.
Admin Management Tools	Provides full CRUD operations for users, courses, and schedules, including system configuration.
Faculty Access	Read-only access to schedules with personal dashboards for monitoring teaching loads.
Responsive Web Design	Mobile-friendly interface optimized for cross-device compatibility and performance.
Real-time Updates	Supports live schedule modifications with immediate data synchronization.
Data Integrity	Utilizes a PostgreSQL-backed database ensuring consistency, reliability, and transactional safety.

Table 1: Key technical features of the PUPSCHED system.

Architecture Diagram

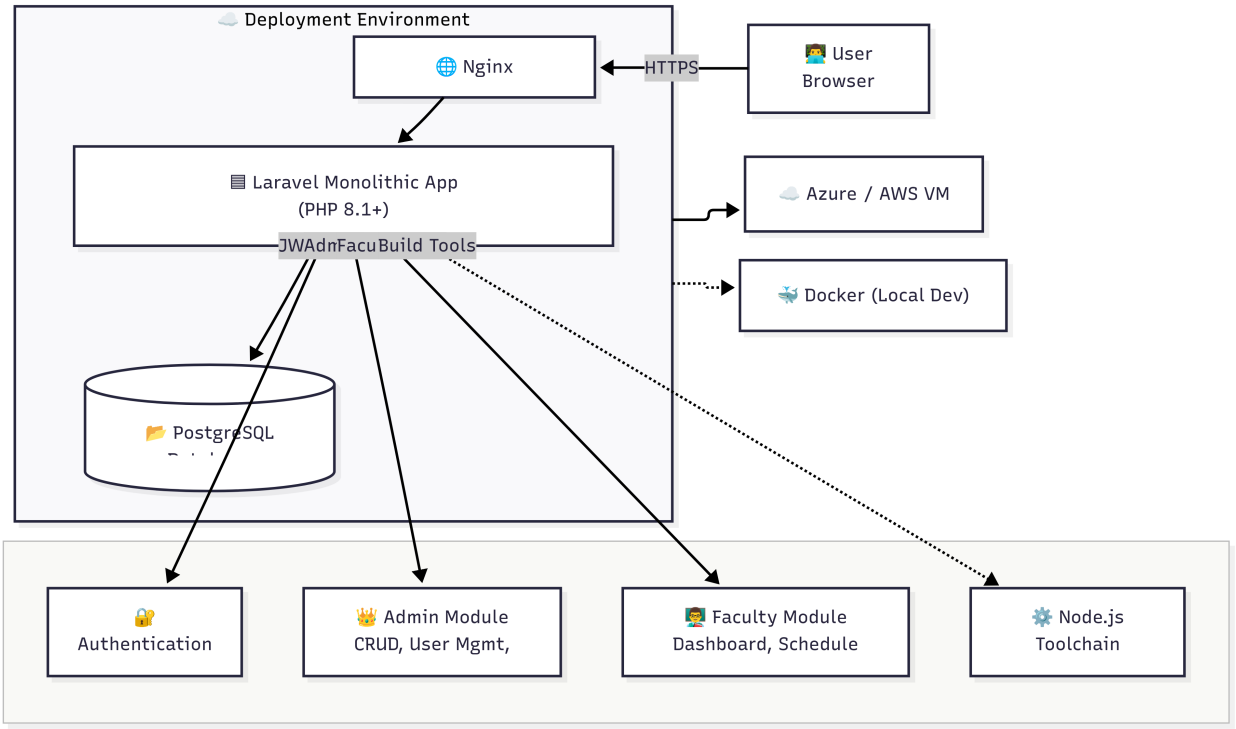

Figure 1: System Architecture of PUPSCHED.

Development Setup

Installation Requirements

Package	Version Requirement
PHP	≥ 8.1
Node.js	$\geq 16.x$
PostgreSQL	$\geq 13.x$
Composer	Latest
Docker	Latest

Table 2: Required packages for the development environment.

Step-by-Step Guide

Step 1 — Clone Repository

```
# HTTPS
git clone https://github.com/1101101011/PUPSCHED.git

# SSH
git clone git@github.com:1101101011/PUPSCHED.git
```

Step 2 — Navigate to Project

```
cd PUPSCHED
```

Step 3 — Install Dependencies

```
# Frontend dependencies
npm install

# Backend dependencies
composer install
```

Step 4 — Environment Setup

```
# Copy environment file
cp .env.example .env

# Generate application key
php artisan key:generate
```

Step 5 — Database Configuration

```
# Update .env file with your credentials
DB_USERNAME=your-username
DB_PORT=5432
```

Step 6 — Database Migration

```
# Run migrations
php artisan migrate

# Fresh migration (optional)
php artisan migrate:fresh

# Seed database
php artisan db:seed
```

Step 7 — Start Development Server

```
composer run dev
```