

ProjectWorlds Multi-School ERP SAAS - Official Installation Guide

Welcome to the **ProjectWorlds Multi-School ERP**! This guide provides comprehensive, step-by-step instructions to deploy the Laravel backend application on various server environments, including **aaPanel**, **cPanel (Shared Hosting)**, and raw **Ubuntu VPS**.

System Requirements

Before proceeding, ensure your server meets the following minimum requirements:

- **PHP Version:** 8.2 or 8.3
 - **Database:** MySQL 8.0+ or MariaDB 10.5+
 - **Web Server:** Nginx or Apache
 - **Composer:** 2.x+
 - **PHP Extensions Minimum Required:** BCMath, CType, Fileinfo, GD, JSON, Mbstring, OpenSSL, PDO, Tokenizer, XML, Zip
-

Scenario A: Deploying on aaPanel (Recommended)

aaPanel is the fastest and most reliable way to host this SAAS platform.

Step 1: Server Preparation

1. Log into your **aaPanel** and navigate to the **App Store**.
2. Install **PHP 8.2**, **MySQL 8.0**, **Nginx**, and **phpMyAdmin**.
3. Go to **Website > Add Site**, enter your domain name, and create a Database at the same time. Note down the Database Name, Username, and Password.

Step 2: Upload Files

1. Go to **Files** in aaPanel, navigate to your new website folder (e.g., `/www/wwwroot/yourdomain.com`).
2. Delete the default `index.html` and `404.html`.
3. Upload the `.zip` containing the platform code and extract it.
4. **CRITICAL:** Go to your Website Settings, select **Site Directory**, and change the **Running Directory** to `/public`. Click Save.
5. In Website Settings, click **Pseudo-static**, select **Laravel** from the drop-down template, and save.

Step 3: Environment Setup

1. Inside the file manager, rename the `.env.example` file to `.env` (or create a new `.env`).
2. Edit the `.env` file and insert your Database credentials:

```
APP_ENV=production
APP_DEBUG=false
APP_URL=https://yourdomain.com

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=your_database_name
DB_USERNAME=your_database_user
DB_PASSWORD=your_database_password
```

Step 4: Database Import

1. In aaPanel, navigate to the **Databases** tab.
2. Click **phpMyAdmin** for your newly created database.
3. Click the **Import** tab at the top.
4. Select the provided `database.sql` file (or similarly named `.sql` backup) from the downloaded package.
5. Click **Go** (or **Import**) to fully restore the database schema and dummy data.

Step 5: Configure the Cron Job (Background Tasks)

To automate fees, notifications, and biometric syncs, add this to aaPanel's **Cron** menu: - **Type of Task**: Shell Script - **Name**: ERP Master Scheduler - **Execution Cycle**: N Minutes -> 1 Minute - **Script content** (Make sure to replace your domain and match your PHP version path): `bash cd /www/wwwroot/yourdomain.com && /www/server/php/82/bin/php artisan schedule:run >> /dev/null 2>&1`

Scenario B: Deploying on cPanel (Shared Hosting)

Step 1: Upload & Extract

1. Open the **cPanel File Manager** and navigate to your domain's document root (usually `public_html` or an Addon Domain folder).
2. Upload the project `.zip` file and **Extract** it.

Step 2: Fix the Public Path (Important for cPanel)

If you cannot change your document root to `/public` via cPanel's interface, follow these steps to secure the application: 1. Move ALL files *inside* the project folder into `public_html`. 2. Move the contents of the `public/` directory into `public_html/`. 3. Edit `index.php` (now in the main folder) and adjust the pathways: `php require __DIR__.'/bootstrap/autoload.php'; $app = require_once __DIR__.'/bootstrap/app.php';`

Step 3: Database setup

1. Go to **MySQL Database Wizard**.
2. Create a database, create a user, assign the user to the database with **ALL PRIVILEGES**.
3. Open **phpMyAdmin** from your cPanel dashboard.
4. Select your newly created database and click the **Import** tab.
5. Upload the provided `.sql` file included in your package to populate the database.
6. Edit the `.env` file via File Manager and input these matching database credentials.

Step 4: cPanel Cron Configuration

1. Navigate to **Cron Jobs** in cPanel.
2. Select `**Once Per Minute (* * * * *)**`.
3. Add the following command (replace `username` with your cPanel username):

```
cd /home/username/public_html && /usr/local/bin/php artisan schedule:run >> /dev/null 2
```

Post-Installation Verification

Once you have completed the above steps on your respective hosting: 1. Navigate to `https://yourdomain.com` in your web browser. You should see the login screen. 2. Login with the default SuperAdmin credentials (provided by your administrator or seeder). 3. Immediately navigate to the **Platform Server Status** page in the sidebar. 4. Ensure all three core checks are **Green** (Production Mode, Debug Disabled, Domain Matched). 5. Ensure the **Storage Link** card says "Linked & Active".

Congratulations, your platform is now securely deployed!