



By



Technical Guide

Product Version: 2.8.0

Table of Contents

<i>Server Requirements</i>	3
<i>Frameworks & Library Uses</i>	4
PHP	4
JavaScript & CSS	5
<i>File Structure</i>	7
Controllers	7
File Structure	7
public/dist Folder	8
media-storage	8
<i>Emails</i>	9
<i>Please Note</i>	10
<i>Thank You</i>	11
<i>Our New Service</i>	12

Server Requirements

Server requirements as given below:

- PHP >= 8.0.2
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- BCMath PHP Extension
- Fileinfo PHP Extension
- GD Library (>=2.0)
- Curl
- MySQL 5.7 +

Frameworks & Library Uses

To make this awesome product below are the libraries used:

PHP

- `php` - ^8.0.2,
- `fideloper/proxy` - 4.4,
- `guzzlehttp/guzzle` - ^7.2,
- `laravel/framework` - ^9.29.0,
- `laravel/sanctum` - 3.0,
- `laravel/tinker` - ^2.7,
- `barryvdh/laravel-dompdf` - 2.0.0,
- `firebase/php-jwt` - 5.0,
- `fruitcake/laravel-cors` - ^2.0,
- `intervention/image` - 2.4.1,
- `laracasts/utilities` - ^3.2.1,
- `laravel/helpers` - ^1.4.0,
- `livelyworks/laravel-yes-authority` - 2.9.11,
- `phpseclib/phpseclib` - 3.0.16,
- `ramsey/uuid` - ^4.5.1,
- `vencogroup/laravel-sparkpost-driver` - ^5.0.3,
- `league/flysystem-aws-s3-v3` - 3.3.0,
- `box/spout` - 3.3.0,
- `fzaninotto/faker` - 1.9.2,
- `gettext/gettext` - 5.5.4,
- `gettext/php-scanner` - 1.3.0,
- `laravel/socialite` - 5.5.5,
- `mk-j/php_xlsxwriter` - dev-master,

- paypal/paypal-checkout-sdk - ^1.0.2,
- pusher/pusher-php-server - ~7.0.1,
- stripe/stripe-php - ^9.6.0,
- razorpay/razorpay - ^2.8.4,
- paypal/rest-api-sdk-php - ^1.14.0,
- unn/gettext-blade - dev-main

JavaScript & CSS

- leaflet - 1.7.1
- @fortawesome/fontawesome-free - 5.12.1,
- agora-rtc-sdk - 3.6.11,
- bootstrap - 4.*,
- animate.css - 4.1.0,
- chart.js - ^3.9.1,
- datatables.net - 1.12.1,
- datatables.net-bs4 - 1.12.1,
- datatables.net-responsive - 2.3.0,
- datatables.net-responsive-bs4 - 2.3.0,
- emoji-nearea - 3.4.2,
- filepond - 4.30.4,
- filepond-plugin-file-poster - 2.5.1,
- filepond-plugin-file-validate-type - 1.2.8,
- filepond-plugin-image-preview - 4.6.11,
- filepond-polyfill - ^1.0.4,
- jquery - ^3.6.1,
- jquery-lazy - ^1.7.11,
- jquery-parallax.js - 1.5.0,

- jquery-validation - 1.19.3,
- jquery.easing - 1.4.1,
- leaflet - 1.9.1,
- lodash - 4.17.21,
- noty - 3.1.4,
- photoswipe - 5.3.2,
- pusher-js - 7.4.0,
- selectize - 0.12.6,
- smartwizard - 6.0.5,
- sweetalert2 - 11.4.34
-

File Structure

While development we follow some of our rules to organize the code. For this we have made some changes to the Laravel framework files as follows:

Controllers

By default, Laravel placed all its controllers under **Http/Controllers** folder, our component controllers are placed **App/Yantrana/Components/** in their respective component

File Structure

All the component code is placed under the folder called **Yantrana** under the **app** folder. This folder contains folders:

- **__Laraware** – a files contain a core files for the various classes which mostly extends Laravel classes & helper, which are restricted to change while development.
- **Base** – is also contains the base files for various classes which mostly extended using **__Laraware** classes.
- **Support** – contains various & common helpers & classes.
- **Services** – contains various & common required service classes.
- **Components** – This is main folder which contains all the business logic in to its own modules as follows:

- **Component Folder**

Each component has its own folder structure as follows:

- **Models** – Contains all the model files.
- **Controllers** – contains all the controller files.
- **Repositories** – contains all the repositories which treated as data-layer.
- **Requests** – contains the request files mostly for forms.
- **Blueprints** – contains the interface files for required classes.

- Engine.php – Its treated as most important file for the component

Each component can have multiple sub components.

public/dist Folder

This folder contains all the frontend related files like JS, CSS etc.

In the CSS & JS folder you will find files named like: **application.min.js**, **vendor-manage.css** etc.

For the application 2 types of files are there one is minified (.min) & other one is normal non-minified files (.src). By default, application uses **.min** files & called in to below named views which are placed under **resources/views** using **__yesset** helper functions, which dynamically finds the files on given file named & path contains *****.

- *-master.blade.php
- includes/head-content.blade.php
- includes/foot-content.blade.php

media-storage

This folder stores all the static media of the files like images etc.

For the most other things you can consider **Laravel PHP Framework** documents at below link:

<https://laravel.com/docs/6.x>

Please Note

There is certain information you should aware of.

1. Application cache (ServerSide) some queries (settings etc.) are cached for better performance if you think you have made some changes to database directly in the related tables, please consider deleting folders at **storage/ framework cache**.
 - a. If you want to disable these query cache you need to set **ENABLE_DB_CACHE=false** in the **.env** file.
2. If you set **APP_DEBUG=true** in the **.env** file. the system also logs lot of information in to the **Browser's Console** panel for debugging, so please keep it **false** for the **production** uses.
3. When the **APP_DEBUG=false** in the **.env** file. the page's markup automatically minified
4. Email are sent in **English** Only.

Thank You

This information would help you to customize understand application better

Once again thank you so much for purchasing this product. I would be glad to help you if you have any questions relating to this product. No guarantees, but I will do my best to assist.

We have small team of Designer & Developers, here are some of the Great tools and technologies we use day2day for the development: HTML5, CSS3, jQuery, PHP, Laravel, AngularJS etc.

we also build **iOS & Android Application**.

If you need any Customization Work / Support on this Product or having any project for us. Please feel free to email us at contact@livelyworks.net

Our New Service



We are happy to inform you that we just launched our new service called ServePHP for managing Servers and Applications for PHP.

ServePHP will help you in creating, provisioning and managing server tuned for PHP. It helps in performing many important tasks like Attaching Domains, Applications, Cron Jobs, Firewall Rules, Auto Deployments and many more without writing single line in Terminal, these tasks are just a click away using simple interface.

Many Advance features like Auto Deployments using Git Push requests, these deployments can be configured conditionally to avoid unnecessary commands executions.

Get started with free account without any credit card information and you can quick sign up via the Google, GitHub, Bitbucket, DigitalOcean, Slack, Gitlab to get hassle free ride to functions & features of site.

To know more about this application please visit our site at the servephp.com

Thank You & Good Luck



livelyworks.net | contact@livelyworks.net | @livelyworks |
<https://github.com/livelyworks>