GIVEu CCMS User Guide

Church content management system

http://giveu.amdtllc.com

```
This is a church content management with online giving system build on Lavar
el Framework. Other comments or concerns please email at contact@amdtllc.com
or open a ticket [https://amdtllc.com/support](https://amdtllc.com/support).
```

Requirements

- MySQL database
- PHP >5.6
- SSH access
- Composer (https://getcomposer.org/download)
- NPM (https://www.npmjs.com/get-npm) (for developers)
- NODEJS (for developers)
- Familiarity with Laravel framework (optional if you have someone else installation for you)

Features

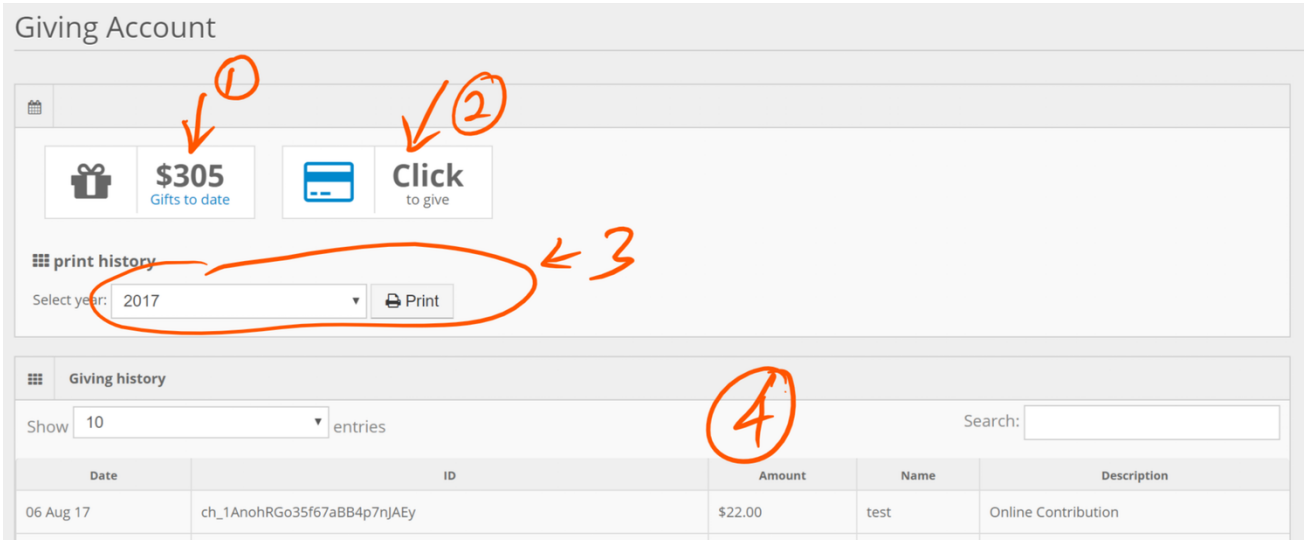
- **Online giving** - Allow your members to give online. Create custom destinations for the contributions such as outreach, missions, etc.
- **Blogs** - Publish a blog for your church. Create an editor role and assign them ability to create posts.
- **Ministries** - Publish information about ministries available to your church.
- **Messaging** - Send mass or individual emails to members from within the system.
  - **Message templates** - Create re-usable templates for sending emails such a birthday cards, weekly updates, upcoming events, etc.
- **Events** - Display upcoming events in a calendar.
- **Sermons** - Publish your sermons to include audio and video with the message.
- **Kiosk** - You can setup a touch screen terminal in the lobby such as iPad with kiosk homepage loaded to allow member to contribute conveniently.
- **Add your own theme** - Create and upload your own front-end theme. See instructions.
- **Add your own module** - Easily create and integrate a module with roles and permissions.
- Support for translation to other languages.

Installation

- Unzip contents to your installation directory

- Create your MySQL database
- run `composer install --no-dev`. Note: remove `--no-dev` if installing on development server.
- Rename `.env.example` to `.env` and file to include database connection, mailer, and default account
  - ***Important***! Enter your default admin login email and password
- run `php artisan migrate --seed`
- run `php artisan key:generate`
- run `php artisan storage:link`
- If on development server, run `php artisan serve` to load your application at http://localhost:8000. Login using default email and password.
- Navigate to `/roles`. Create new roles and assign permissions as needed.
- If on production server, set your environment variable `APP_ENV=production`

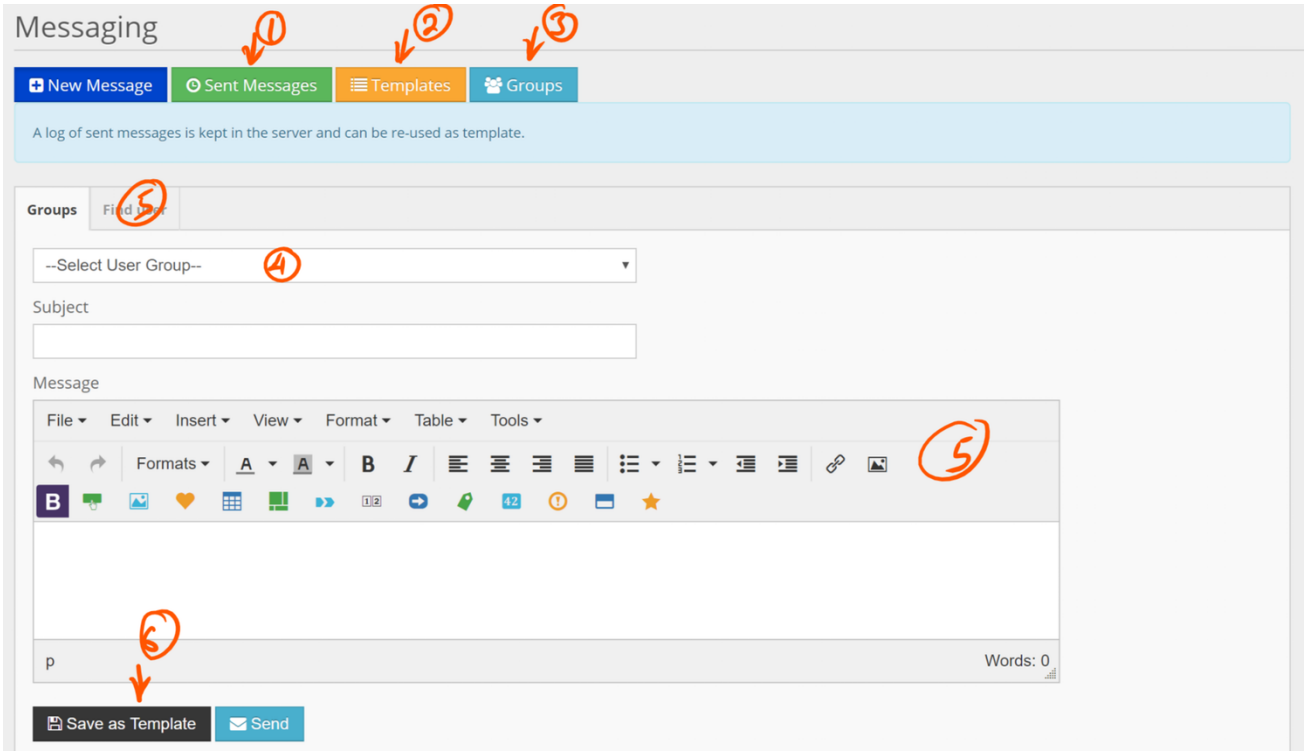
Giving account



Giving account as viewed by user

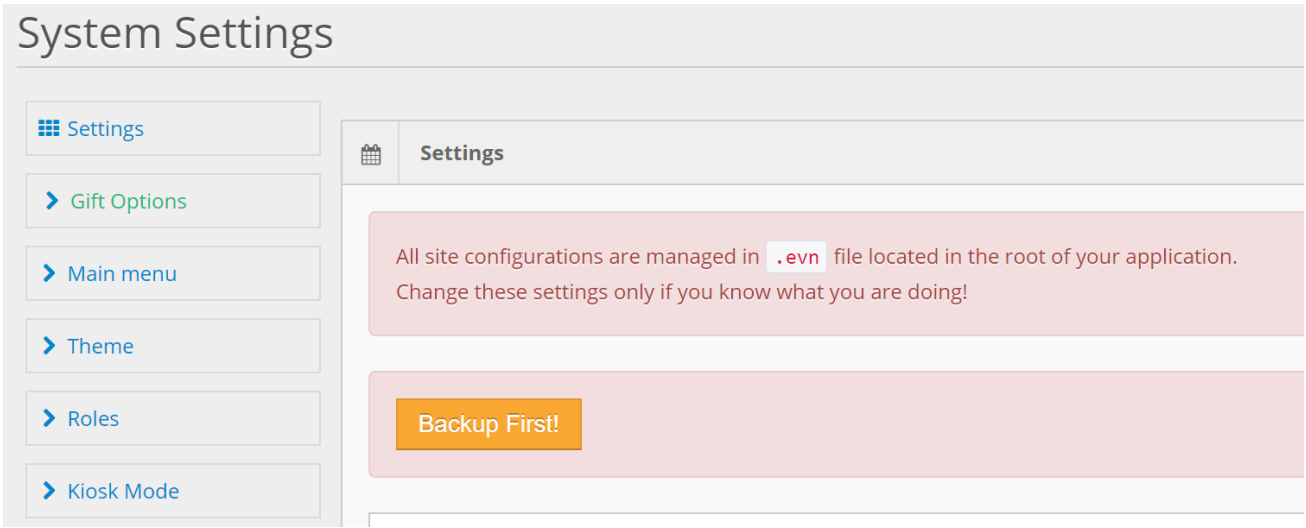
This page is accessible to all users

1. Total amount contributed to-date
2. Open modal window to enter amount, payment method and select designation
3. Select and print
4. Searchable table for all contributions to date



Messaging

1. Send messages history
2. Re-usable message templates
3. Custom groups for mass messaging
4. Select a group to mass message
5. WYSIWYG editor with ability to upload images, attach videos and create templates.
6. Save a message as a template

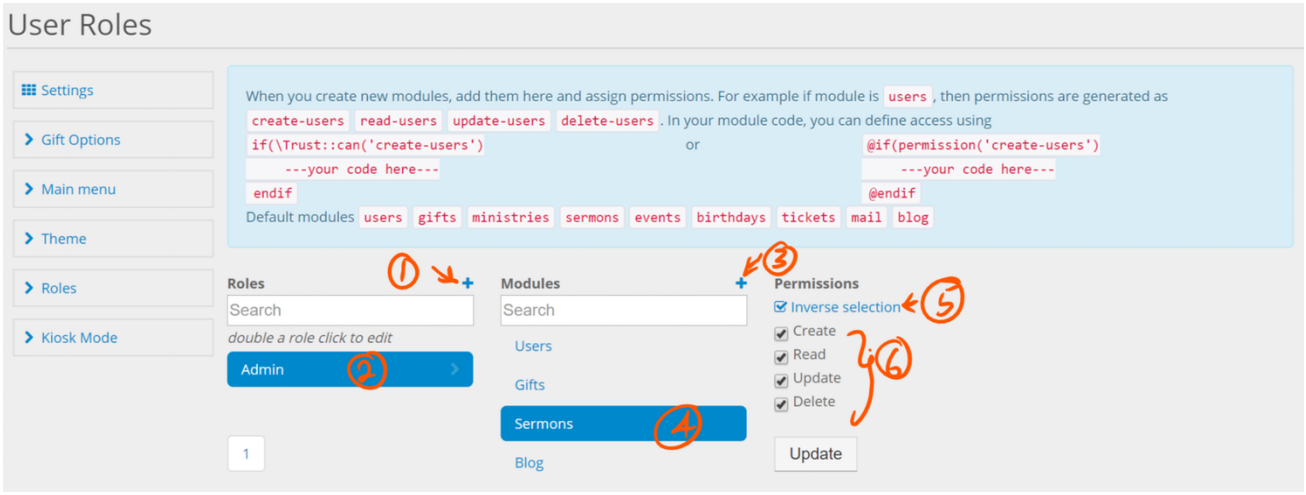


System settings

- You can edit .env directly from settings page. ***`CAUTION: You should have a very strong password for your admin account. Access to this page will compromise the security of your installation including the database.`***

- BACKUP UP settings first before making changes.
- Gift options: Options entered under this section will be available for selection in kiosk and accounts page dropdown list for designation.

User Roles



User roles

1. Click to add a new role
2. Single click to select or double click to edit
3. Register a new module
4. Single click to select, double click to edit
5. Inverse selection to quickly select/deselect items
6. CRUD permissions for the selected module of the selected role

Default modules are shipped with the application and can be generated after using the seed option during initial setup above.

Creating and defining roles for new modules

- Create a module example `job` and register is on modules. Select role, then click on that module and assign permissions.
  Permissions are stored as `create-job`, `read-job`, `update-job` , `delete-job`
- To enforce permission in routes, (`/routes/web.php`)

```
Route::post('jobs', ['middleware' => ['permission:create-job'], 'uses' => 'JobsController@storre']);
```

- To enforce permission in controllers (`/app/Http/Controllers`)

```
$this->middleware('permission:create-job',['only'=>['store']]);
$this->middleware('permission:read-job',['only'=>['index','show']]);
$this->middleware('permission:update-job',['only'=>['edit','update']]);
$this->middleware('permission:delete-job',['only'=>['destroy']]);

//within code
```

```
if(\Trust::can('create-job'){
  //your code here
}
```

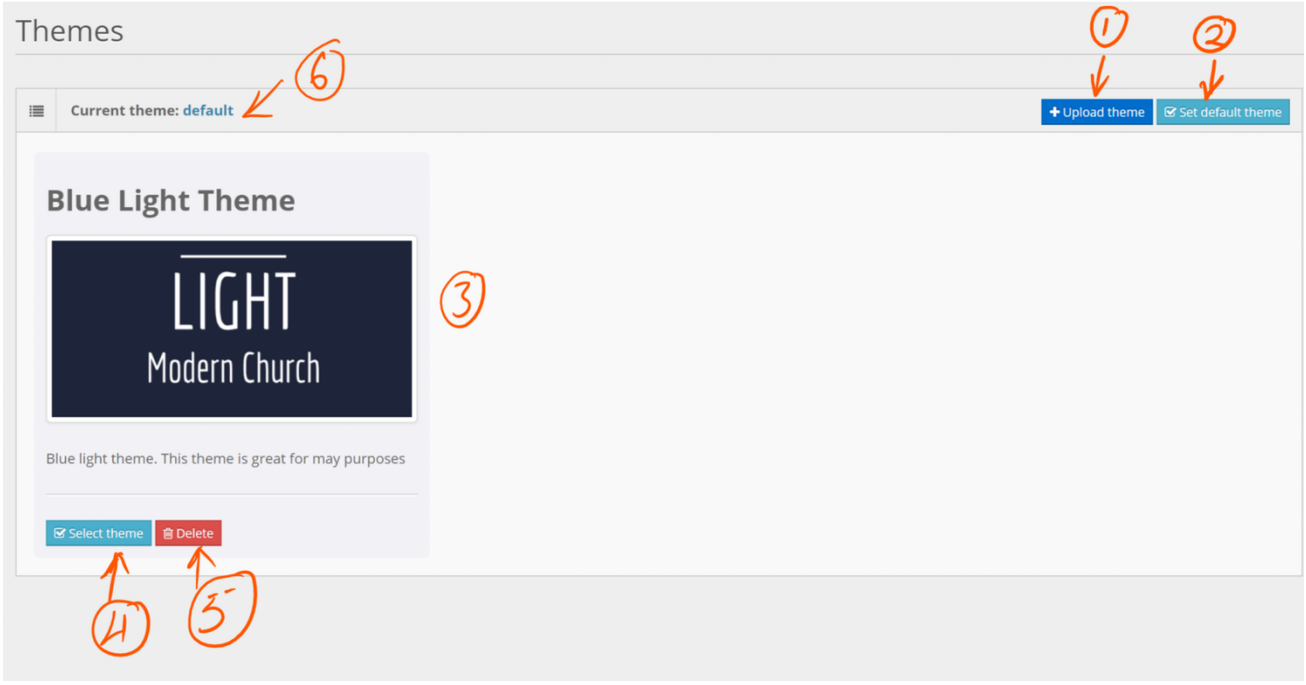
- To enforce permission in views (`/resources/views`)

```
@permission('create-job')
  //your code here
@endpermission
OR
@if(\Trust::can('create-job')
  //your code here
@endif
```

- You can also enforce within code for both control
- Roles can be setup with the same format as above e.g. `role:admin`. Refer to the Santiago's Laratrust package at http://laratrust.readthedocs.io for further details.

Front-end Themes

You can create your own front-end themes and upload them via `settings > themes`.
Themes must conform to a set of predefined standards in order to be compatible with this application.

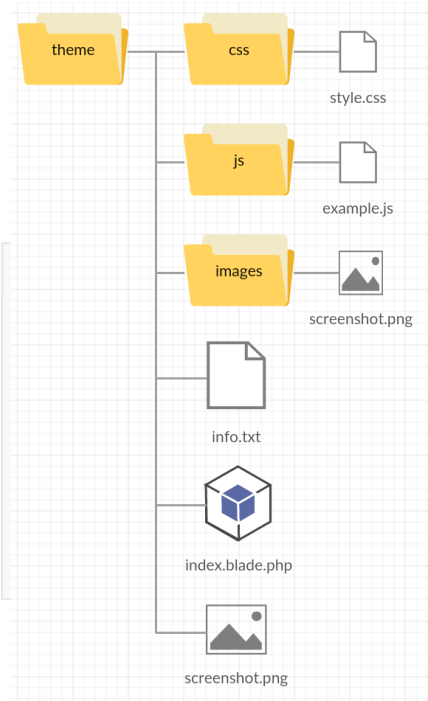


Theme dashboard

1. Upload your zipped theme and select is to set it.
2. You can always revert back to default theme.
3. Uploaded themes will appear here.
4. Select theme to set it.

5. Delete theme. If you delete an active theme, your installation front-end will revert to default theme.
6. Name of the active theme.

Theme structure



theme structure

- Theme name **MUST NOT** contain spaces or special characters. Preferably, use one work for name of your theme.
- Required files: `index.blade.php`, `info.txt`, `screenshot.png`
- Optional files and folders: `css/`, `/js`, `images/`
- **`info.txt`** - This is the theme information file and must be follow the pattern below.

```
Name=Blue Light Theme
Description=Blue light theme general purposer th
eme.
Author= JMuchiri
Url= http://example.com
Version= 1.0
Version date= 24-June-2017
```

- Theme must be unique per the name of the uploaded zip file. If another theme exists with same name, it will be overwritten.
- On uploaded, `index.blade.php` is loaded in `/resources/views/themes/<theme name>/index.blade.php`
  All other files are uploaded to `/public/themes/<theme name>/`

Theme options

`theme()`	loads an instance of theme helper through which you can call theme functions
`theme()->menu([options])`	Loads theme menu optional menu options *Options*: **`icons`** - Display font-awesome icons in front of menu item name **`no-submenu`** - Disable dropdown menu for nested menu items *Use this function as follows:* `theme()->menu(['icons','no-submenu'])` Menu order can be set by drag and dropping order in admin settings page.

`mainMenu()`	display main navigation bar
`theme()->logoNav()`	display nav-bar with or without logo. Logo image must be in /public/img/logo.png

`themeOpts()>[option]`	calls theme options stored in the database.
	Options:
	`id (integer)`
	`name (string)`
	`description (text)`
	`location (string) (path of the theme)`
	`active (bool)`

Translation

The application has 90% support for translating to other languages. To translate, open files located in `/resources/lang`. You will find a file `en.json` which has duplicate English columns. Copy it and rename it to your desired language e.g. es.json and translate the second part. See Laravel documentation for further information regarding translation.

Kiosk mode

When you select Kiosk Mode, you will be logged out of the application and directed to Kiosk page.
Users in Kiosk Mode are required to enter an email address with each transaction. An inactive account is setup for such user and will record future transactions associated with that email address. If they may chose to open a full account, they will have access to all their past contributions.

Advanced

Do not directly modify any minified assets under `/public/css` and `/public/js`. These can be modified in /resources/assets, then run `npm install` and `npm run dev` if in development mode or `npm run production` if in production mode.

Contributing

Thank you for considering contributing! If you would like to become part of this project, please contact us at **contact@amdtllc.com**.

Security Vulnerabilities

If you discover a security vulnerability within this application, please send an e-mail to safety@amdtllc.com. All security vulnerabilities will be promptly addressed.

Licenses

This software utilizes open-sourced software licences. All included open-sourced software are copyright on their respective developers.