# Templates

The code takes advantage of the PHP include function to template portions of the page, such as the left nav. Thus to put the left nav in the page you use:

<?php include("../templates/left-nav.php"); ?>

To edit the left nav then modify the included file and it will update everywhere.

On a given page edit the CSS locally, using the ID of the link, such as

<style>

#general {

background-color: grey;

color: white;

}

</style>

# Database Schema

We decided to store information about created events in a database. The information stored in the database can then be pulled when the events need to be edited or when the JSON files need to be created using **getevent.php** or **getnotifications.php**.

The database used to store information about the events that have been created currently contains 13 tables. The database is created with **eventApp-data.sql** and populated using the php files in the **edit** folder. The database schema is as follows:

*event*(*ID, internal\_ID, name, year, refresh, refresh\_expire, time\_zone, welcome\_message, logo, contact\_nav, contact\_icon, sched\_nav, sched\_icon, housing\_nav, housing\_icon, prayer\_nav, prayer\_icon, notif\_nav, notif\_icon, theme\_dark, theme\_medium, theme\_color, visible*)

*themes*(*ID, event\_ID, sequential\_ID, theme\_name, theme\_color*)

*contact\_page\_sections*(*ID, event\_ID, sequential\_ID, header, content*)

*contacts*(*ID, event\_ID, sequential\_ID, name, address, phone*)

*schedule\_items*(*ID, event\_ID, sequential\_ID, date, start\_time, length, description, location, category*)

*housing*(*ID, event\_ID, sequential\_ID, host\_name, driver*)

*prayer\_partners*(*group\_ID, event\_ID, sequential\_ID*)

*attendees*(*ID, event\_ID, sequential\_ID, name, house\_ID, prayer\_group\_ID*)

*notifications*(*ID, event\_ID, sequential ID, title, body, date, refresh*)

*info\_page*(*ID, event\_ID, sequential\_ID, nav, icon*)

*info\_page\_sections*(*ID, info\_page\_ID, sequential\_ID, header, content*)

*users*(*ID, username, password*)

*event\_users*(*ID, user\_ID, event\_ID*)

For a given *event\_ID*, the *sequential\_ID* of any table acts like an auto-incrementing ID within that event.

The database schema pictured below reflects the schema above and shows the relationships between the tables.

**Suggested changes:** Change the primary keys of the tables to the *event\_ID* paired with the *sequential\_ID*, instead of using an auto-incrementing integer. This change must be reflected in most of the existing files.

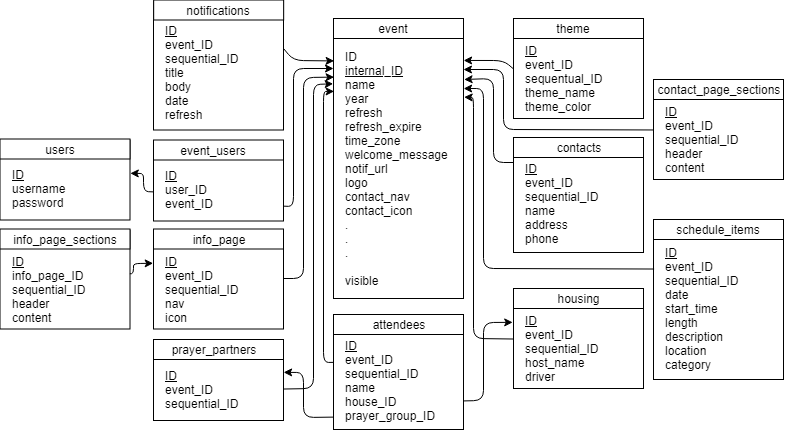


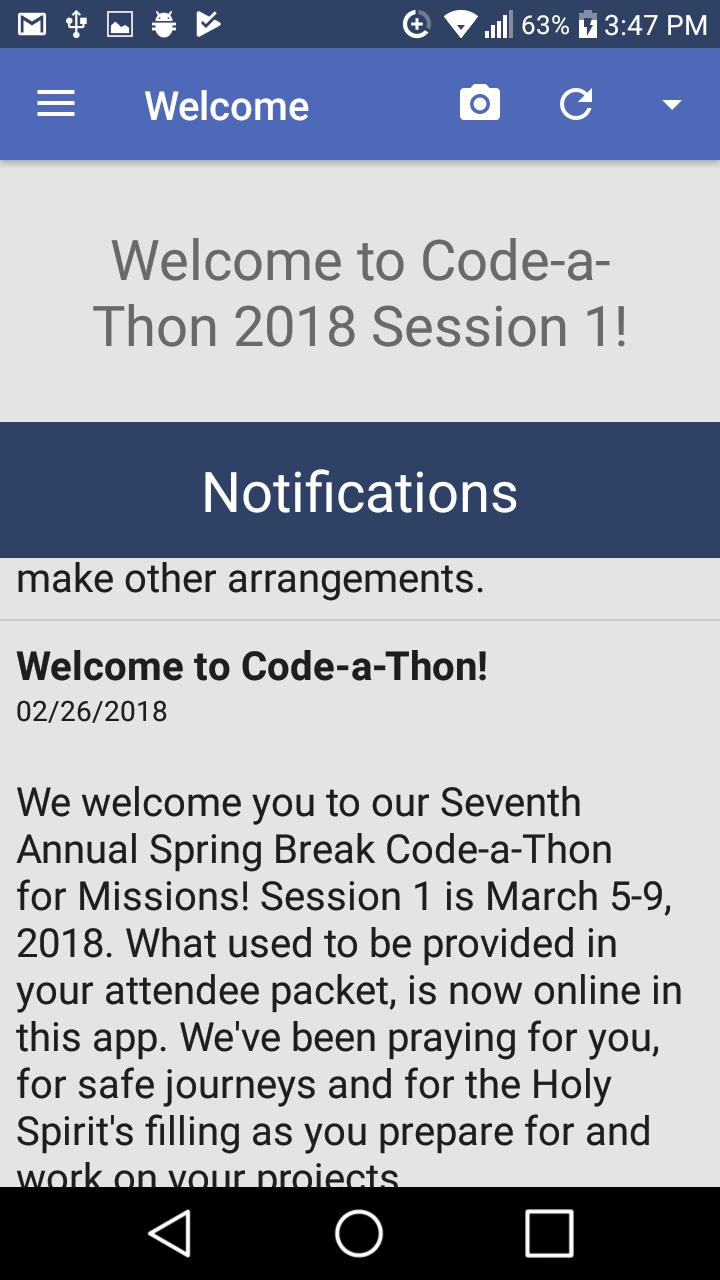
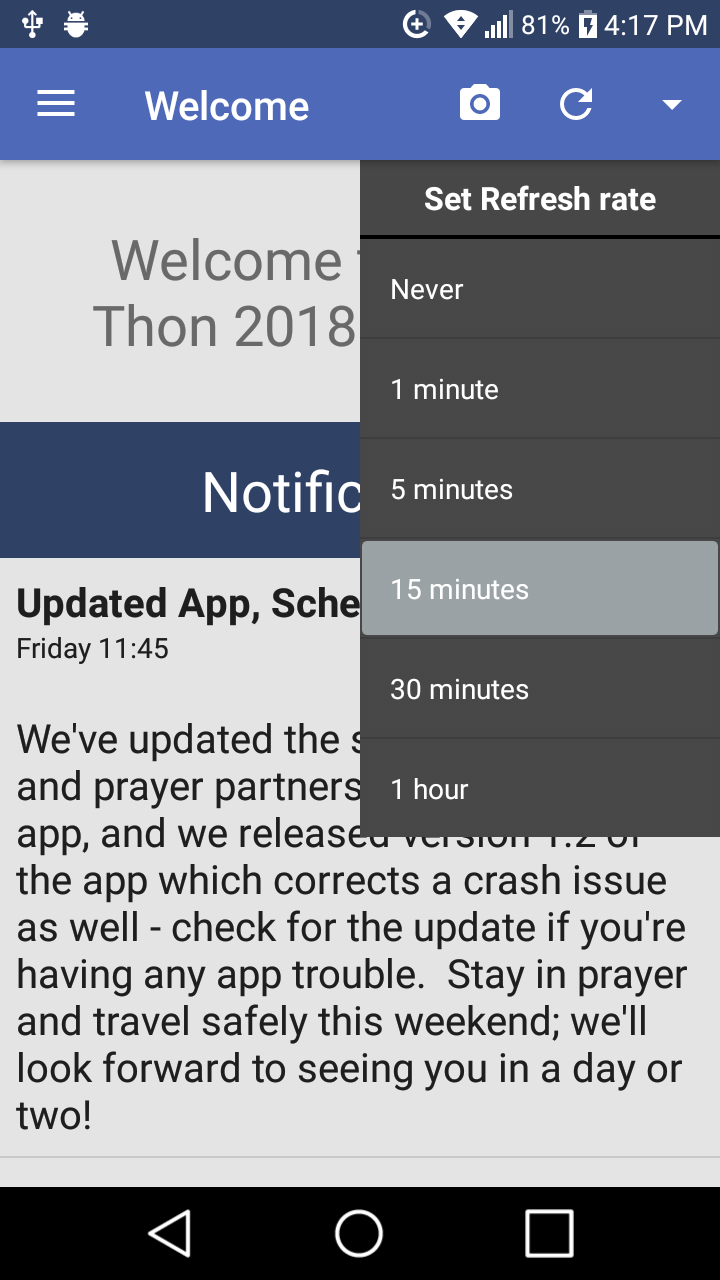
Figure 1. Event App Database Schema.

# 

# Database Relation to Event App

This section demonstrates how the information contained in each table relates directly to the information shown in the app and available to users. Although the database is used to generate the JSON files necessary for the app to display an event and only can be used by the app with this intermediate step, this section shows what information from the database is eventually made visible in the app, using the LightSys Code-a-Thon 2018 Session 1 event as an example. (This section does not discuss every field in every table.)

The first table, *events*, contains two ID attributes, and both individually distinguish each event. The first ID attribute, called *ID*, differentiates them for the users, while the other, called *internal\_ID*, differentiates them to the rest of the database and is only used internally. The *refresh* attribute contains an integer specifying the refresh rate in minutes, which can be set by the app user using the menu shown in Figure 1:



Figures 2 and 3: Relation of the *refresh* and *welcome\_message* attributes.

The text contained in the *welcome\_message* attribute is displayed at the top of the Welcome screen, as shown in Figure 2. In this case, the welcome message is “Welcome to Code-a-Thon 2018 Session 1!”

The contents of the *logo* attribute as well as each *nav* and and *icon* attribute in the *events* table are represented as the images and headers in the navigation menu. Note that these may not include every icon and associated header in the navigation menu; additional headers that can appear will be discussed below. The *logo* is pulled from the database and shown at the top of the navigation menu, as shown in the figure below. The data contained in the *notif\_nav*, “Notifications” in this case, as well as the bell icon associated with it in the *notif\_icon* attribute, make up the first link in the navigation menu, and the other associated *navs* and *icons* are similarly displayed:

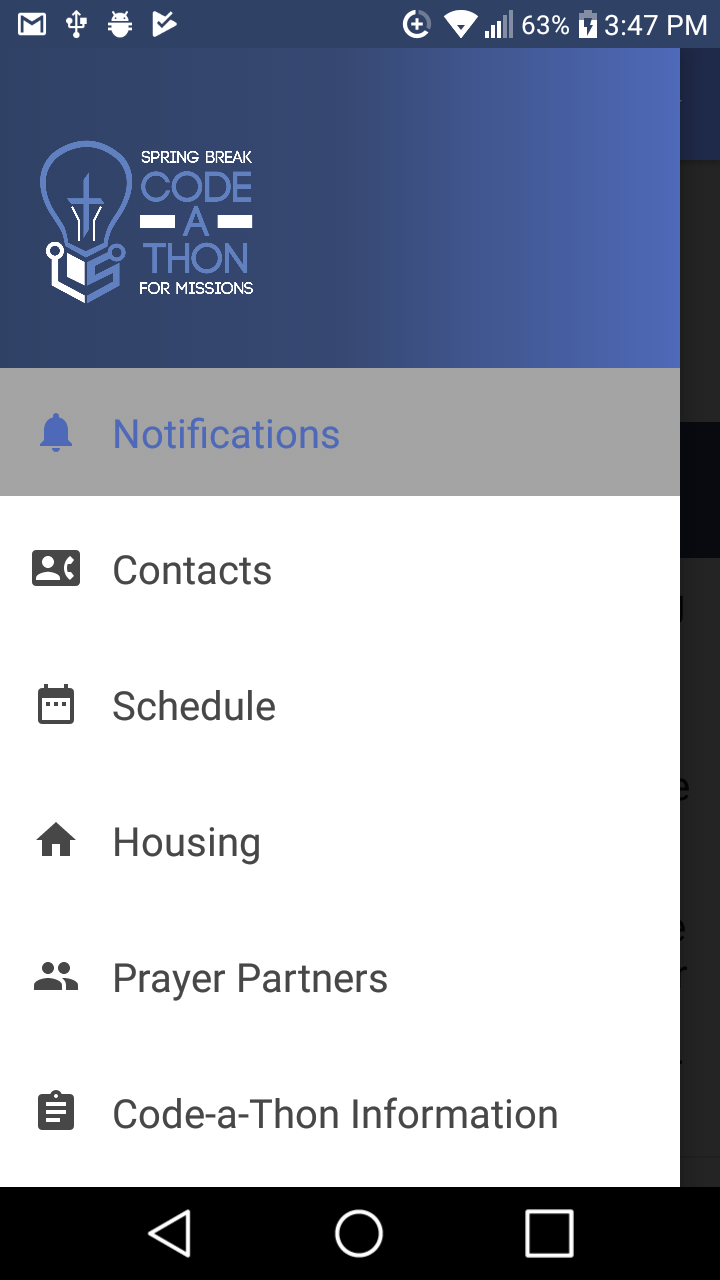


Figure 4: Contents of the *logo*, *nav*, and *icon* attributes displayed in the app.

Pages with additional information about the event can be specified by the event creator. The database stores information about these pages in the *info\_page* and *info\_page\_sections* tables. Each of these pages has a *nav* and *icon* associated with it (contained in the *info\_page* table), and these are displayed in the app’s navigation menu below those that are pulled from the *event* table. In the figure above, “Code-a-Thon Information” is the *nav* of one of these user-defined pages.

Each of these user-defined pages can have one or more sections, and the content of each of these sections is contained in the *info\_page\_sections* table. For the “Code-a-Thon Information” page, the *header* attribute of the first section is “WiFi,” and the contents of its associated *contents* attribute are displayed directly beneath it, as shown below:

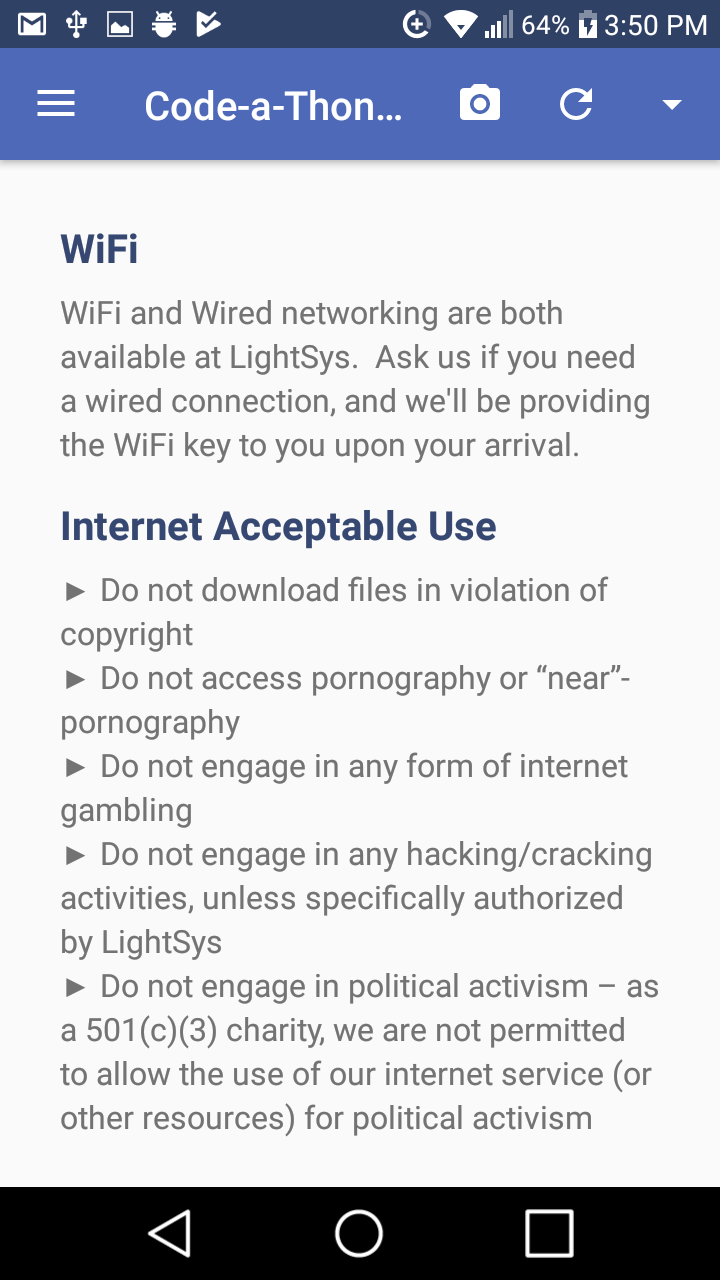


Figure 5: “Code-a-Thon Information” as an additional user-defined page.

“Internet Acceptable Use,” as expected, is the *header* of another of the sections on this page.

Information for the notifications that populate the section at the bottom of the Welcome page is contained in the *notifications* table. Each notification has a title pulled from this table’s *title* attribute a message pulled from the table’s *body* attribute, and the date it was created and posted pulled from the *date* attribute. In the figure below, the *title* of the notification shown is “Welcome to Code-a-Thon!”, and its associated date and message are shown below it:

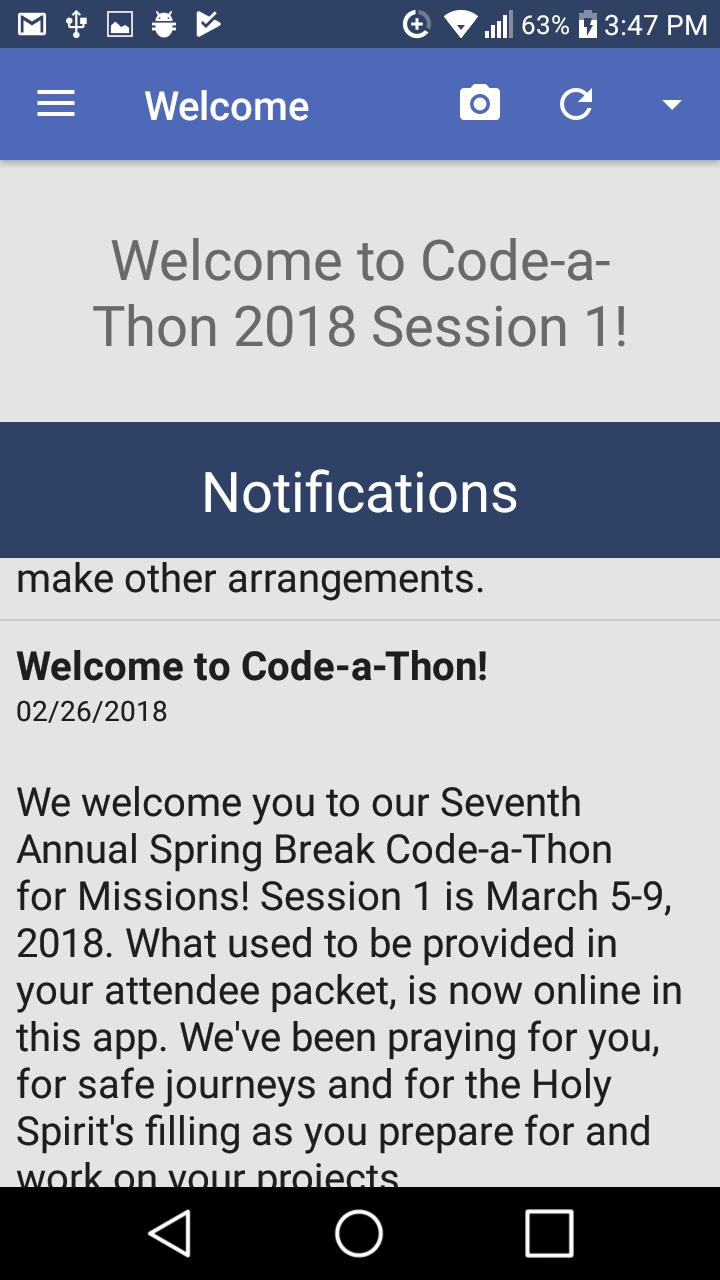


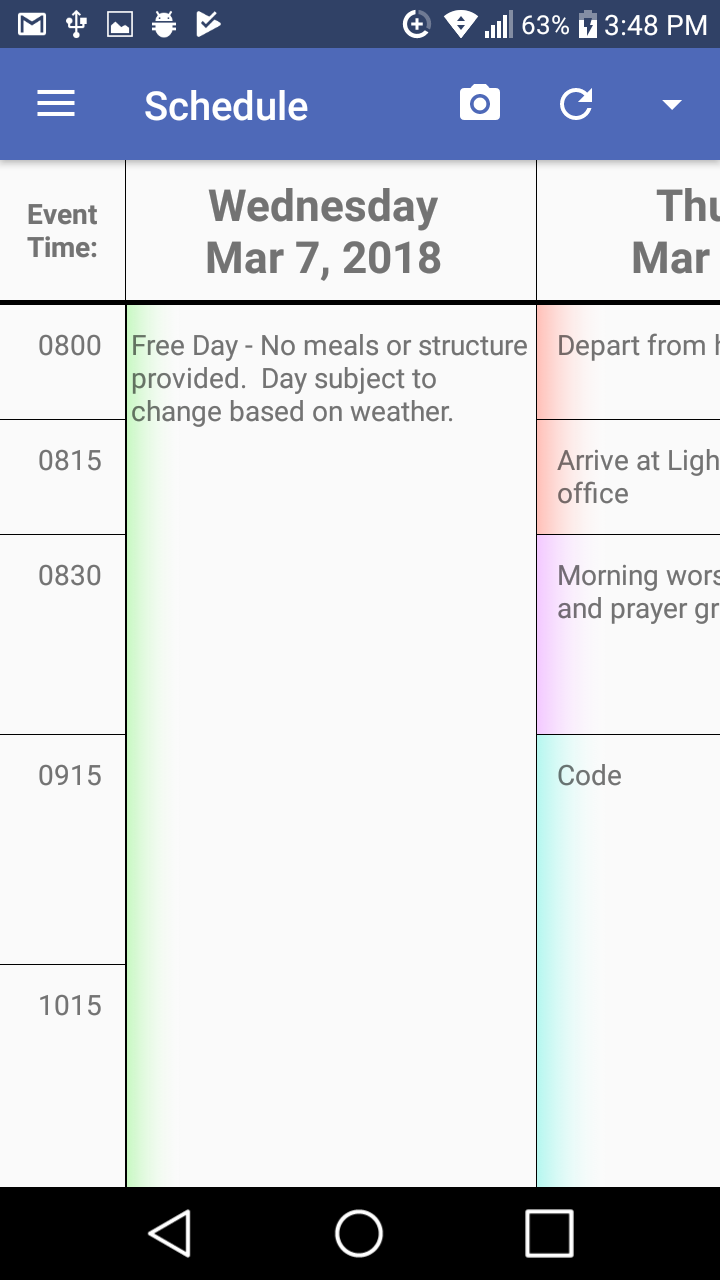
Figure 6: “Welcome to Code-a-Thon!” notification.

The Contacts page is set up and populated using the information stored in the *contact\_page\_sections* and *contacts* tables. The information contained in the *contact\_page\_sections* table structures a contact page in essentially the same way that information in the *info\_page\_sections* table structures a user-defined information page. Contacts are pulled in by looking at the names in the *content* attribute of the final contact page section and finding and displaying the information (from the *address* and *phone* attributes of *contacts*) associated with those names.

The Housing page also draws from the *contacts* table, as every host (defined in the *host\_name* attribute) is also a contact with a name, address, and possible phone number. The *housing* table also contains a *driver* attribute that tells who will drive the attendees housed at that event to and from the event. This name is displayed above the associated addresses as shown below. The names of the guests listed on the right next to the addresses where they will be housed are pulled from the *name* attribute of the *attendees* table.

Both the Housing page and the Prayer Partners page pull from the *attendees* table. For the Prayer Partners page, names from the *attendees* table are displayed next to the number of the prayer groups the attendees have been placed in.

The Schedule page lays out a schedule for each day of the event. The *date* attribute of the *schedule\_items* table is used for the date at the top of each column, and the *start\_time* and *length* determine where in a column something is displayed, using the times shown on the left side. The *category* attribute references the *theme* table, which determines the color of something on the schedule.



# Temp Folder

The temp folder will need to have 777 permissions so that users can upload logo images to it from the general section. The images will be stored in the folder briefly while they are stored in the database, and then they are deleted.

Note that currently this is the same folder for all users, so multiple users happening to save it the same time could cause an error.

# Still To Do and Known Issues

* Fix temp folder (see note in Temp Folder above)
* Authentication, roles, and user management
  + Only allow users to view events that they have access to
  + User management: currently anyone can create an account
  + Add logout button
  + Session expiration on login
* General and Advanced pages need to retrieve items from the DB
* Contacts page clears any data entered if the Submit button is not clicked before the Add Contact button. Other pages work similarly
* General will delete the logo when you save it again without re-uploading the logo, since it does not check for a change.
* The sidebar and buttons change color with the theme, but the text remains white, so it might be hard to read with a light theme. Consider changing text color to black if a high color value is used.
* Add screen tips! We want specific information from the user, but there is currently no way for them to know what kind of information to provide for some blanks.
* Make dropdowns check the database to see what was previously selected and select that option. Particularly on schedule this does not currently exist. As is it will overwrite them all with the wrong theme unless you select the correct themes again when you edit the schedule.
* Add form validation to make sure people don’t miss important things or enter random garbage.