**Church Management System (CMS) – Documentation**

**Bootstrap Template:** [**https://www.bootstrapdash.com/demo/azia/v1.0.0/template/dashboard-six.html**](https://www.bootstrapdash.com/demo/azia/v1.0.0/template/dashboard-six.html)

**Membership Section:**

**Ministry Section:**

**TO DO:**

**1.)** When clicking Add Ministry, it should show a modal where you can add to a ministry (or multiple ministries): `Graphical user interface, application

Description automatically generated



Similar to this:

A screenshot of a computer

Description automatically generated with medium confidence

\*This modal should list the ministries and the user can choose where to add the member to the ministry. It should be a toggle button to add/remove.

**2.)** Then once you click the Ministries view, it should show a table where the members are listed for that selected ministry, this table should be at the bottom of the page.

**3.)** Then create a description section on top. So Description on top, then table of members at the bottom. Description section can look like this:

Text

Description automatically generated

**4.)** Then on the Ministry members table on the view, it should grab data from ci\_member table (TO DO: add a column of Ministry to each member, if NULL then they will not be listed on the Ministry view)

**5.)** TO DO: Edit ministry…

**Technical Side of CMS: Using PHP CodeIgniter as a framework:**

Creating a brand-new project:

1.) on config, update baseurl to localhost/<foldername>/

2.) on routes, update default\_controller to landing controller <nameOfController>

3.) Create a controller for the default, or copy paste welcome page.

4.) Create the view of that default controller.

5.) Create the database in phpMyAdmin, update database.php, input database information

6.) In model (or created model), update database name

7.) rinse and repeat for different processes…

How to search for templates:

- google ERP System Template HTML

- google Bootstrap admin templates

**Bugs (Backlog):**

- img user logo is not showing when in viewMember or viewMinistry.

**Church Management System – Project Deadline and Milestones:**

**Ministry Section:**

What needs to be done?

- Add a member to ministry modal on members’ page (members and viewMember page for the select box is not working properly) and also save button.

- Add member button to link to members’ page

- Print the members of the ministry in a table to the ministry view.

- Edit ministry information

- Ministry description, a select statement when Edit Ministry is clicked.

Review Points (Aljon):

Ministry Page:

Delete Ministry:

- After deleting a ministry, it is not displaying the members after redirecting.

Add Ministry:

- After adding too many ministries, the scrollbar is not working.

Add Members Tab on the Left Pane

When editing a Ministry, redirect to the specific Ministry edited

Change function names in the model to get/set/insert/delete for uniformity

Change function name in the controller to view/edit etc. instead of viewMinistry etc.

Change the position of the Ministry Description imitating how the Network is displayed in the members page.s

Remove colon on the ministry members

Retrieve Attendance Functionality on the Windows laptop

**Deadline:** **June 8, 2022 (Wednesday)**

**Events and Attendance Section:**

- Do delete event button to delete an event.

- Initialize Status to Open.

- Adhoc task when an attendee is not a member yet.

- Create a “Non-member” button on the 2nd tab to input name (required). Then optional fields are network, address, and contact number. When clicked, it should pop out a modal with these fields, then add it to the ci\_event\_attendee, and print on the next tab.

- So you have to create a new table called “ci\_evt\_non\_member” with fields (name, address, contact number, first\_timer\_flg (yes or no values only)).

- Extract to pdf or excel.

- Add another tab(third tab) on the wizard to print attendees on the event on a table.

**Deadline: ??**

**Finance Section:**

**Learnings along the way of developing this project:**

- Think in a business perspective. What do the admin users want and need to see for their purposes?

- When debugging, find the cause of the problem first and start there.

- Always comment so other programmers can understand the code and be able to help you fix the problem faster.

- Know the design of the system first before setting the deadline.

- The skills I need to acquire is learning how to find the problem, put them into tiny pieces, and fix it.

- The starting point of debugging is to comment codes or print codes line by line, see if the value being passed is correct, where does the code explodes, etc. Once you spot where the problem is, work towards fixing that.

- In models, it should only contain getters and setters. Naming convention is important.