**ACH Helping Hand Portal: Developer Documentation**

**Background**

The ACH Helping Hand Portal allows users to select from a number of different services and hours required within pre-defined categories. The system remembers the services that the user has selected and allows the user to edit and update the information. Once the user has finished selecting services from the different categories they can view information about the services selected, the number of hours and price per service category. The user then has the option to fill out a contact form at which point they will receive an email and attached PDF quote which the system automatically generates. ACH also receives an email with the user details.

ACH has control over specific services and descriptions within each category, service category pricing, service category featured images and descriptions and a few other content blocks displaying informative messages around the site. The portal is a custom application built using PHP and Javascript for server side and front-end functionality, respectively. It utilises Wordpress & Advanced Custom Fields for content management. Thus, technically the application is a custom Wordpress theme.

**Purpose of this document**

The original scope of the application was to launch six pre-defined product categories with the ability to add/edit/remove different services within each category. Shortly prior to launch the client requested to go live with the application only having five service categories and to have the ability to remove and add service categories at will. This is not possible in the Wordpress environment and a custom software solution is out of budget.

This document provides step by step instructions to remove a service category or add a service category. The combination of this document and comments in the source code of the application provides clarity to an intermediate level PHP and Javascript developer to make these changes.

\*The application source code contains logic to handle instances of the unused service category of ‘My Health’. All front-end output of these instances has been commented out. This ‘my health’ logic should be used in the event the application returns to the original scope of six service categories. In the event that the category name changes, naming conventions should be changed appropriately in the source code.

**Adding a service category**

There are numerous steps that need to be taken that concern the following:

* Adding a new page in WP, assigning the template you build & creating the required custom fields (ACF)
* Custom template creation
* Using JS sessions to remember user input on front end
* Using PHP sessions to remember user selections on server
* Sending user selection info to ACH and user via email
* Writing user selection data to PDF

All files are located in the WP theme itself. A new template file needs to be created, and the following files need to be created/edited:

**Files to Create**

**\*newCategory is name of new category. Separate words should be separated by hyphens as per existing file naming conventions.**

**\*source code line numbers are at time of document creation**

A newCategory.php

B inc/newCategory/newCategory-hours-selection.php

C inc/newCategory/validateServicesBeforeSubmit.js

D inc-app/js/savenewCategoryFormData.js

E inc-app/remove/remove-newServiceCategory-data.php

F inc-app/update/process-newServiceCategory.php

**Files to Edit**

G inc/check-session-vals-assign-vars.php

H inc/select-boxes.php

I make-pdf.php

J finish.php

K page-home.php

**FILE CREATION**

**A Creating a new template**

1. Create new template by copying existing template (about-me.php) and naming appropriately
2. Update form action, name, id on line 68 under comment <!—OPEN SERVICE SELECTION FORM -->. \*steps 2 – 9 pertain to form that handles adding and updating of services
3. Update checkbox input class, id, name (array) on line 93 (in service loop)
4. Update label for attribute on line 94 (in service loop)
5. Update textarea class, id on line 141 (first item after service loop)
6. Update classes and in radio-group starting on line 164
7. Update id and name in hidden input on line 170
8. Update id, class, namenumber input on line 178
9. Update submit id on line 193 (last element of main form)
10. Update form action, name, id on line 204 \*step 10 pertains to form that handles removal of services
11. Finally, update the JS file paths on line 238 and 244 – you will be creating these new files in later steps.

**B & C Adding inc/newCategory files**

**B = evaluate number of hours selected and maintain selected state on front end**

**C = ensure user has selected at least one service and number of hours**

1. Copy the about folder and give it an appropriate name.
2. Swap all ‘about’ related classes, id’s and attributes with your new service category name
3. Remember, these files act upon id’s, classes, and attributes you created in the new template in the previous step, so pay close attention to these. This message holds true to all further steps as well

**D Adding inc-app/newCategory js file**

**D = save user selected data on front end using JS sessions**

1. Copy the saveAboutFormData.js file and give it an appropriate name.
2. Swap all ‘about’ related classes, id’s, attributes, local storage names with your new service category name

**E Adding inc-app/newCategory/remove php file**

**E = kill all PHP session related data for a particular category**

1. Copy the remove-about-data.php file and give it an appropriate name.
2. Swap all ‘about’ related session names with your new service category name
3. Update the header location with your new service category name

**F Adding inc-app/newCategory/process php file**

**F = add or update PHP session related data for a particular category**

1. Copy the process-about-data.php file and give it an appropriate name.
2. Swap all ‘about’ related session names, variable names with your new service category name
3. Update the header location with your new service category name

**FILES TO EDIT**

**G Editing inc/check-session-vals-assign-vars.php**

**G = check values of session variables and assign variables. Application uses this file in various locations around the site to conditionally determine items to output. Take ACF option fields and convert to constants.**

1. Copy logic handling aboutServices, otherServicesAbout, aboutHours, aboutHoursMore and create logic for new service with appropriate name.
2. Convert ACF field data to constant copying existing examples

**H Editing inc/select-boxes.php**

**H = display navigation boxes to each service category. Overlay colour if service category already has locked-in services.**

1. Copy all mark-up and contained PHP within <div class=”col”>
2. Paste and update with new service category name.

**I Editing inc/make-pdf.php**

**I = loop through service categories and if they contain data push to an array. Then, loop through existing arrays and write data to a PDF.**

1. Copy logic from line 226 – 266 (professional services - array)
2. Paste below and update variable names with new service category name.
3. Copy logic from line 533 – 578 (professional services – write to PDF)
4. Paste below and update variable names with new service category name.

**J Editing finish.php**

**J = Handle contact form submission and create email containing service selection information. Attach the PDF created in previous file. Send email to user and ACH.**

1. Copy logic from line 203 – 216 (writing professional services to email body)
2. Paste below and update session name with new service category name.
3. Copy logic from line 1081 – 1225 (output professional service selection data)
4. Paste below and update session & variable names with new service category name.
5. Copy logic from line 1280 – 1287 (output professional advice price)
6. Paste below with updated constant value

**K Editing page-home.php**

**K = Display featured images, descriptions of services and provide navigation to service categories.**

1. Get featured image and assign to a variable (top of file)
2. Copy all mark-up in .col-md-4 and paste with new service category details.

**Things to do in WP Admin**

1. Head to ACF & duplicate ‘about me services’ field group and give appropriate name.
2. Set up same repeater and sub fields
3. Add new category name to ‘pricing’ field group
4. Add new description to ‘landing page’ field group
5. Create the new page, add a featured image & enter the appropriate service data on the page and in ‘service pricing category’ option field \* front page.

**Removing a service category**

Follow up the commenting out of items as per ‘My Health’ category. We need only to be concerned with output for the now missing category because we will delete the Wordpress page and therefore none of these services can be added. Commenting out is the best strategy as we can un-comment if a new service category is to be added in the future. Pages and ACF field groups can be deleted to make the environment more pleasing to ACH.

You will need to comment out anything that outputs service category details on the front end:

Finish page – (pricing of