Configuration of Drupal Webpage for IT

Pre-steps:

Spent three or four days brainstorming options for the platform of the systems. Original idea was to use GitLab as the environment for the ticketing and inventory systems. However, after some research, it was determined that most of the platform was preset and not customizable. This led to the need for a content management system to house the IT homepage. After completing a trade study, Drupal was found to be the most capable and secure webpage environment for what needed to be accomplished.

Step 1 MAMP (localhost):

- The first step in this process was to download a program that would allow myself to run a local server environment on my computer to develop the website.
- MAMP was decided to be the best local host program to install.

Step 2 Configuring Drupal:

- Once MAMP was downloaded, I moved the Drupal folder into MAMP's directory and renamed the folder 'htdocs' so that MAMP would recognize the information and allow the information to appear online.
- Next, I created an SQL database on the server's admin page to house the data of the website.
- Once the local host was up and running, the website was available to configure on the server.
- After mapping to the SQL database and adding the administrator account, the base webpage was visible.

Step 3 Website Theme:

- Clicked on the appearance tab in the main navigation and was directed to theme configurations.
- Modified the colors of the website to green and blue to match PDS.
- Changed the Drupal icon to the PDS logo.

Step 4 Implementing Modules – Lightweight Directory Access Protocol (LDAP):

- One of the requirements for the website was the ability to connect an active directory that enabled the user to authenticate by means of LDAP during login.
- Found a module through the Drupal directory that possessed these attributes.
- Tried installing this module but was getting an error that there was no PHP extension enabled through MAMP.
- Found a php.ini file in MAMP's directory that needed to be modified to fix the issue. Went into the file by means of a text editor and removed a semicolon (;) in front of the php extension that was previously disabling it. Problem solved.
- Proceeded to install the contributing modules to LDAP: *Authorization and External Authentication.*
- Went into the configuration portion of Drupal and started to modify the LDAP settings.
- Created LDAP Query to locate Drupal Users.
- Configured LDAP Server that enabled me to bind to the active directory using the login credentials of a non-anonymous search user.
- Modified user mappings for the PDS active directory.
- Logged into Drupal using my active directory credentials and a new account was created within the system.

Implementing Modules – Super Login:

- Downloaded the Super Login module from the online Drupal directory.
- Configured the login page so that it is more visually pleasing for the user.

Implementing Modules – Webform:

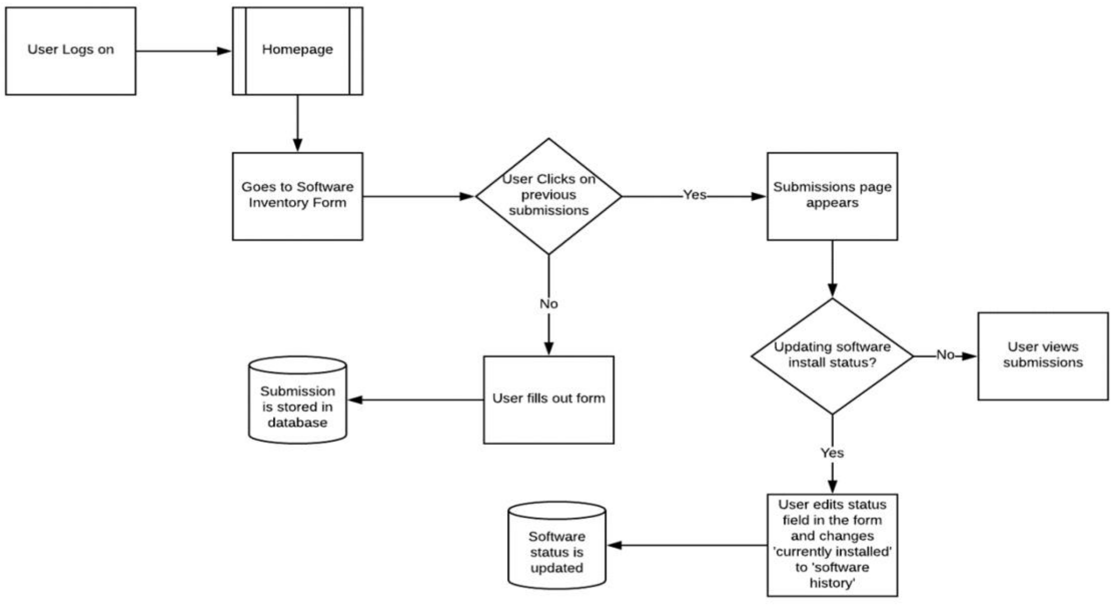
- After downloading an unfinished development version of a support desk module for Drupal, it was determined that the configurations were not up to speed with what needed to be implemented into Drupal.
- The webform was then determined to be the most effective module for the inventory and issue ticketing systems.
- Through the structure tab in the navigation, we find webform newly installed at the bottom of the list.
- Once here we build the webforms for the inventory system and the issue ticketing system.
- Throughout the creation process fields and elements were created for each individual snippet of information we needed to extract.
- When the forms were completed, I went to the menus where I added links to both the webforms on the main navigation.
- Users now have access to both webforms easily at the home page of the website.

Implementing Modules – Require Login:

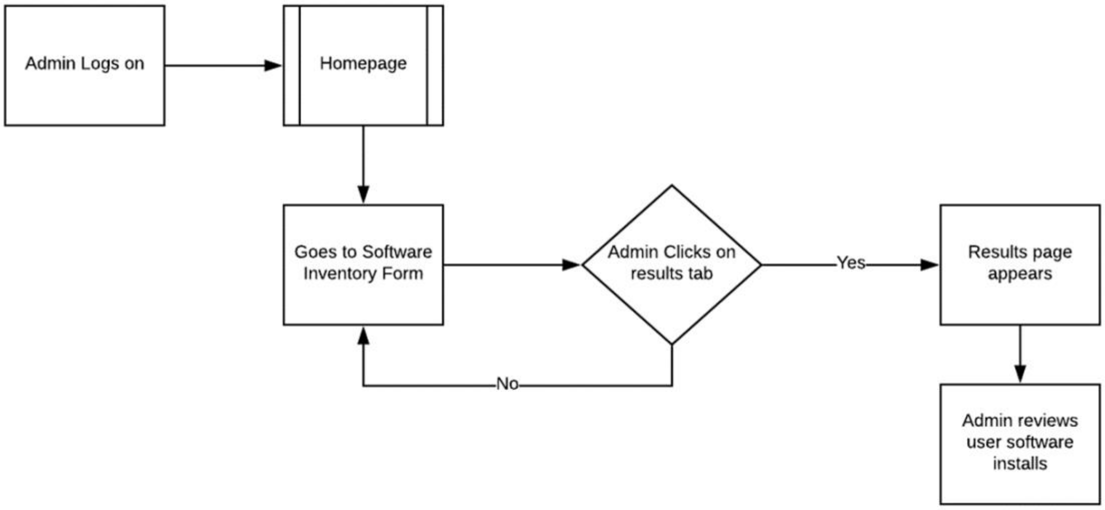
- When everything seemed to be completed, I logged out of the website and attempted to see the view of an unauthenticated user at the login page. It was then discovered that an anonymous user had access to both webforms without login. A major problem.
- To fix this, I found the module require login, that I then installed and configured so that only authenticated users would have access to view and modify the webforms.

Work Flow Charts

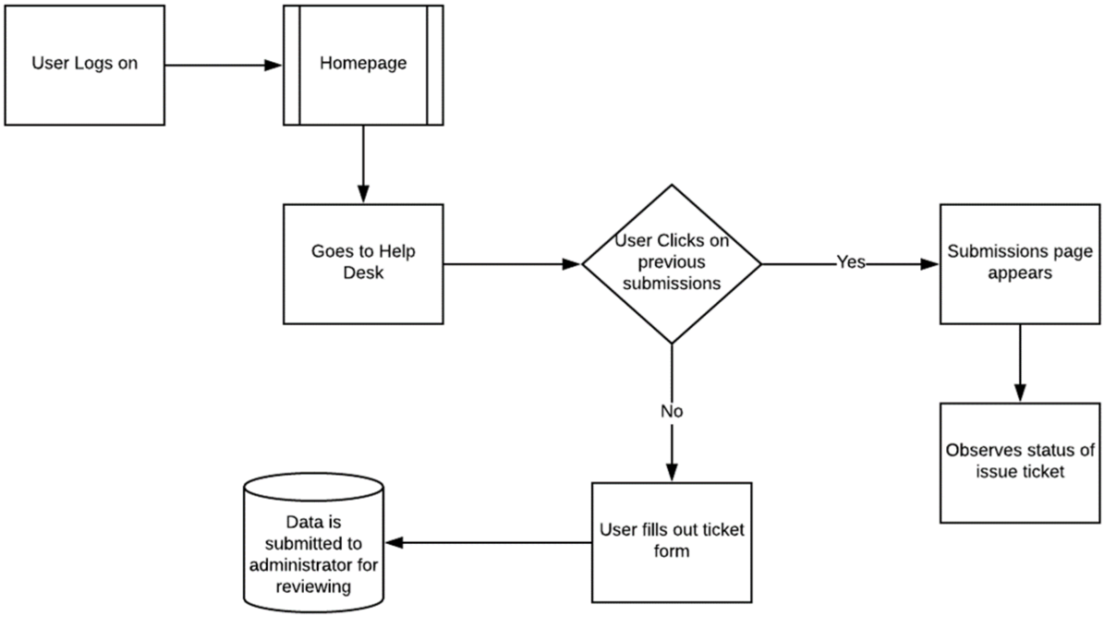
User Process Flow (Software Inventory):


User Logs on
Homepage
Goes to Software Inventory Form
User Clicks on previous submissions
Yes
Submissions page appears
No
Updating software install status?
No
User views submissions
Submission is stored in database
User fills out form
Yes
Software status is updated
User edits status field in the form and changes 'currently installed' to 'software history'

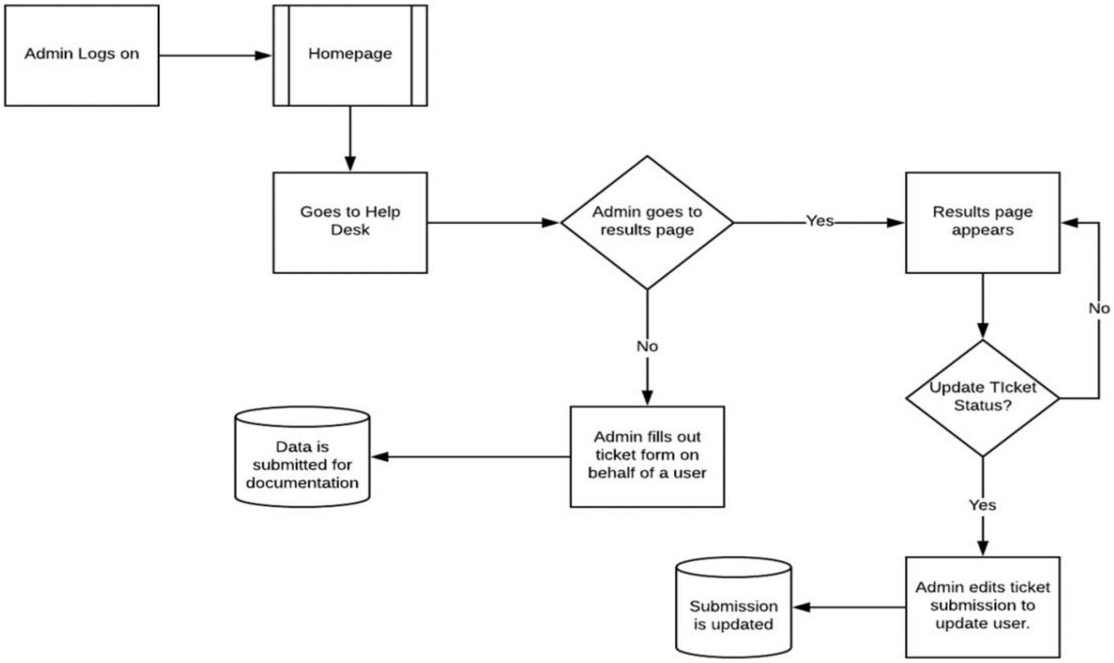
Administrator Process Flow (Software Inventory):


Admin Logs on
Homepage
Goes to Software Inventory Form
Admin Clicks on results tab
Yes
No
Results page appears
Admin reviews user software installs

User Process Flow (Issue Ticketing):


User Logs on
Homepage
Goes to Help Desk
User Clicks on previous submissions
Yes
No
Submissions page appears
Observes status of issue ticket
User fills out ticket form
Data is submitted to administrator for reviewing

Administrator Process Flow (Issue Ticketing):


Admin Logs on
Homepage
Goes to Help Desk
Admin goes to results page
Yes
Results page appears
No
No
Update TIcket Status?
Admin fills out ticket form on behalf of a user
Data is submitted for documentation
Yes
Admin edits ticket submission to update user.
Submission is updated