# Table of Contents

Table of Figures ........................................................................ 3
Executive Summary / Abstract ...................................................... 4
Introduction .............................................................................. 5
Requirements & Deliverables ...................................................... 6
Site Structure Introduction ......................................................... 7
Wordpress Plugins .................................................................... 8
Design & Site Templates .......................................................... 10
Automation ............................................................................ 12
Testing / Evaluation / Assessment ............................................ 13
User’s Manual .......................................................................... 15
Developer’s Manual ................................................................ 17
Lessons Learned ..................................................................... 23
Acknowledgements .................................................................. 26
Bibliography / References ....................................................... 27
Appendix .................................................................................. 29
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Website Header</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Website Footer</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>People Section Landing Page</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Example Faculty Post</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Example Test Case for Adding New Person Post</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Example Test Case for Adding New Person Grid Loop</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Faculty Page</td>
<td>15</td>
</tr>
<tr>
<td>8</td>
<td>Alumni Page</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>Site Architecture</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>Plugins Search Page on Wordpress</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>CPT UI Adding a Custom Post Type</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>Form for Adding a New Post Type</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>CPT UI Taxonomy</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>A Form to Fill Out for when Creating a Taxonomy</td>
<td>20</td>
</tr>
<tr>
<td>15</td>
<td>ACF, Creating a Custom Field Type</td>
<td>21</td>
</tr>
<tr>
<td>16</td>
<td>People CPT</td>
<td>22</td>
</tr>
<tr>
<td>17</td>
<td>Form to Add People</td>
<td>22</td>
</tr>
<tr>
<td>18</td>
<td>Selecting People Types</td>
<td>22</td>
</tr>
<tr>
<td>19</td>
<td>Timeline for our website</td>
<td>23</td>
</tr>
</tbody>
</table>
Our goal for this project was to work on recreating the CTE website in order to improve the maintenance capabilities for the administrator and to fix the flaws of the site. The CTE program at Virginia Tech stands for Computational Tissue Engineering, which is an interdisciplinary graduate education program focused on the convergence of multiple scientific disciplines, with the goal of driving advances in tissue engineering. The site is split into sections that cover the Research, News, and Events of the program as well as an archive of the Faculty, Students, and Alumni of the program. The site was created with a CMS called Basecamp over 10 years ago. Because so much time had passed, it became difficult to do maintenance on the underlying code. This was mainly due to the fact that Basecamp was developed with PHP, and the specific version of PHP that the CMS was originally developed on had become outdated. This was the reason our client, Dr. Murali, who is the Associate Program Director of the CTE program, tasked us with recreating the site, as it had become too difficult to maintain the site.

The new CTE site was developed by our team over the course of three months. We began development with the research phase in February. We needed to find a new environment to develop the new site which would meet the needs of our client. To find a site that would meet Dr. Murali’s needs we met with him several times to discuss deliverables and to learn about the way in which the original site was maintained. We eventually chose WordPress as the CMS we would be building the site with. The next step we took was to develop plans for version 1 of the site and to learn how to use WordPress and its Plugins to meet our goals. We learned about Custom Post Types, Taxonomies, and Custom Fields. We developed an architecture for entering and storing data in the site, as well as building the initial backend of the site. By April we had finalized work on the basic functionality of the website, so we moved on to complete the structure and the frontend of the site. Finally, by the end of April, we had been working on the frontend, making fixes, and putting the finishing touches on the site.

The new site was built with many of the deliverables that had been discussed with Dr. Murali. In the completed site, the admin has the ability to dictate permissions to other users, including the ability to grant editing privileges. Student profiles that were entered into the site can easily be changed to alumni profiles. Although pages for the site cannot be toggled on and off, they can be functionally disabled and enabled within the WordPress Dashboard. New entries of profiles, such as students, would automatically be displayed on the site. The site could recognize different categories of data including “people” entries. The admin has the ability to preview pages when making changes. Fields can be changed to be a list of data types within certain editors. The admin also has the ability to create forms on the site.

The deliverables of the site were developed with the original site’s problems in mind. One major problem that the old site suffered from was the inability to change the structure of certain pages without editing the site’s underlying code. Thankfully the new site solves this problem. As for additional work, it would be beneficial to host the site using a Virginia Tech CS
domain name. This task would probably require a site admin to get in contact with a network liaison for the university.

Introduction

The motivation and background behind our project comes from the Computational Tissue Engineering (CTE) website that our client, Dr. T. M. Murali has been managing for the past few years. The website is currently maintained with Basecamp and PHP, but the version of PHP that Basecamp was running on quickly became outdated. This caused issues when it came to managing and making edits to the website. Some of the problems mentioned by Dr. Murali were: no live previews, a long process of staging and publishing edits to the website, and static implementations of data. This meant our client had to constantly go in and change the data when it became obsolete. The combination of these three problems meant that our client had to make changes frequently that were not available to view until after a long process to publish. Therefore, our goal going into this project was to focus on the functionality of the website and ease of maintenance. Note that our focus on this project wasn’t to change or update the look of the CTE website, but rather to make it easier and simpler for Dr. Murali to manage the website. Our solution to the overall problem was to use a different Content Management System that would simplify the process of maintaining the website for Dr. Murali. We decided to use WordPress as our CMS, as we found it fit our needs, resulting in various plugins and tools that could help with the management process.

We took a very structured approach to implementing the site. The site was developed around a simple backend architecture that used Wordpress’s data organizational features. The features used were the Custom Post Types, Taxonomies, and Custom Field Types. We used these features to categorize the data of the site into certain buckets or data types, and we then created subcategories for these data types based on the buckets. The main buckets of data for the site, which each had their own Custom Post Types made, included the People data type, the Research data type, and the News data type. Each data type’s Custom Post Type used Taxonomies, when necessary, to create subcategories. For example, for the People CPT, Taxonomies were used to create the categories of Faculty, Current Students, and Alumni. Custom Field Types were used to allow the admin to associate data with a post/entry. Custom Field Types for the People data type were developed to allow the admin to include information in posts/entries such as the name of the individual, their role (if applicable), their program (if applicable), the year they were recruited (if applicable), the year they graduated (if applicable), and much more information. The admin also had the ability to choose the Taxonomy/Category for the data type they were making an entry for.
Requirements & Deliverables

The CTE site needed to be remade in a way that the administrator was capable of maintaining the site and making changes to it in the future. One of the major problems faced by our client, Dr. Murali, was the fact that when he needed to make certain changes to the structure of the old site, he was forced to change the underlying code. This process could be quite tedious and inefficient, and we didn’t want the new site to suffer from similar major inconveniences. In order to ensure that the site is adequate for maintenance, our group discussed certain requirements with Dr. Murali to focus on during development.

- **Permissions**
  - The admin needs to be able to give permissions to other users so that they can edit the website.
  - The admin should be able to add and remove users from the site.

- **Automation**
  - The admin should be able to automate certain tasks, such as editing list properties or moving one list item to another.
  - The admin should have the ability to automatically add new entries to the list of events.
  - The admin should have the ability to recognize different list categories.
  - The admin should be able to update the site based on list entries.
  - The admin should be able to select a list of categories from a field.
  - The admin should be able to divide assets (such as media, photos, and documents) into categories.

- **Publishing changes**
  - The admin should have the ability to toggle on and off a page of the site.
  - The admin should also have the ability to preview the final appearance.

- **Site-building/maintenance**
  - The admin should be able to create forms.
  - The admin should have access to a sitemap.
  - The admin should have comprehensive customization options.
Site Structure Introduction

As mentioned before, the focus of our project is not to redesign the CTE website, but to reimplement it with features that are necessary for maintenance. Slight changes to the UI of the website may occur to make the website more modernized, but we do not plan on changing the UI too much as it is pretty practical the way it is. In order to effectively recreate the website and correctly implement the functionality that was requested, it is important to know in detail how the current website is made and functioning. The design of the website is broken down into three different dynamic sections that frequently change with data and two static sections that do not change. Each section is determined by the navigation bar located at the bottom of the header of the website.

Dynamic Sections

The three dynamic sections are People, News & Events, and Research. The reason these sections are referred to as “dynamic” is due to the fact that Dr. Murali constantly adds new posts to these sections and has to constantly update the visuals of the website depending on the change in the list of data. The People section is constantly changing due to the change in faculty and incoming students as well as the change in graduation status from current students to alumni. The News & Events section is constantly updated by Dr. Murali with the new articles and events that occur for the CTE department. Finally, the research section depends upon the different research opportunities and projects that the CTE department chooses to do.

Static Sections

The two static sections of the website are: About Us and Education & Training. These sections do not have any data that change and are not dependent on different posts. The Education & Training section includes a short description of the goal of the CTE Department and what it aims to accomplish for the students in Virginia Tech. Finally, the About Us section includes links to multiple different departments of Virginia Tech related to CTE.

The overall purpose of the site introduction is to give basic information on the site structure so that the following sections can effectively refer back to specific sections and showcase our implementation of each part of the website. In order to visualize the full site architecture, refer to the Developer’s Manual.
WordPress Plugins

Elementor

Elementor [13] is the leading website-building platform for WordPress. The Elementor website builder lets WordPress developers create and edit websites with a responsive drag-and-drop technique. This is especially helpful for developers who do not know much about code development. The goal of the whole project was to let Dr. Murali edit and manage the site without having to code anything; Elementor fits that perfectly. Due to the widespread popularity of Elementor, the plugin is constantly being updated with new features that could possibly help with the maintenance of the website. The Elementor plugin is integral to the website as the whole site’s front end is dependent on this plugin. The rest of the plugins also interact with Elementor’s front end as they are linked to the many dynamic aspects of the site.

Elementor was used to build the different templates of the website. The header, footer, and site templates were all used in the theme builder section of the website so that they could easily be accessed as a template when making a new page. Elementor was also used in the process of making grid loops (see more in the Automation section).

CPTUI & ACF - Custom Post Types/Taxonomies

CPTUI [12] and [11] ACF are both WordPress plugins that help with customizing and creating posts. The CPTUI plugin was the plugin that was used to create the Custom Post Types, without the need to personally write PHP scripts. This plugin was chosen over its alternatives as it provided a simple UI for creating and editing Custom Post Types and avoided the need to write PHP code, as one of our goals for the new site was to not require the admin to write any code. This way we could make maintenance for the admin straightforward and easy. The CPTUI plugin was also used to create the Taxonomies for the Custom Post Types, with taxonomies basically acting as categories for the CPTs. Taxonomies, for the Custom Post Types we created, could be added and edited through the CPTUI plugin on the dashboard. The plugin’s interface is simple to use, and it allows the admin to specify certain settings for the Custom Post Types and Taxonomies, such as labels, descriptions, slugs, and more.

The ACF plugin, which stands for Advanced Custom Fields, was the plugin used to create Custom Fields for the Custom Post Types. The ACF plugin allows the admin to add fields, or data inputs, to their CPT entry, like the name of an individual, their role (if applicable), their program (if applicable), the year they were recruited (if applicable), the year they graduated (if applicable), and much more information. The admin can also use the ACF plugin to include additional data in their Custom Post Types, such as images, videos, or metadata. The ACF plugin provides a wide range of field types including image, text, WYSIWYG editor, file upload, and other field types. The ACF plugin also provides an easy-to-use UI in the dashboard which avoids
the need for coding in PHP, similar to the CPTUI plugin, and makes it easy to add, delete, and edit fields.

**JetSmartFilters**

The JetSmartFilters plugin is a plugin that is used to build complex filtering systems and structures in a way that saves time and money. The plugin is made to be used with websites built with the Elementor plugin. It is very easy to use, is customizable, and doesn't rely on coding any PHP scripts. The plugin was developed by the company Crocoblock, which offers a whole suite of plugins and solutions for WordPress. The plugin allows you to create filters of different types including checkboxes, radio buttons, range sliders, and select dropdowns. Filters can be grouped together into custom filter groups that can be used to filter content on multiple criteria. AJAX filtering is supported by the plugin, meaning the site doesn't have to be reloaded to filter page content. For the site, we created several select-type filter objects that were integrated into the UI. These filters were used to allow site visitors to filter Current Students by Program and Recruiting Class, and to Filter Alumni by Program and Graduating Class.

**Elementor Custom Skin**

The Elementor Custom Skin plugin is used to create custom templates, skins, or designs to be used throughout the site. The plugin serves as an addition to the Elementor plugin as it extends the functionality of that plugin. The dynamic templates created with the plugin can be applied to multiple pages or posts. The plugin supports conditional logic, meaning that the template could be set to display only under certain circumstances. The Elementor Custom Skin plugin loops, which are instructions for how to display a group of posts, can be defined and displayed based on the admin’s specifications. We use conditionals to display these loops of objects, on certain pages, with certain templates.
Design & Site Templates

Header & Footer

The header (see Figure 1) and footer (see Figure 2) of the WordPress CTE website are present on every page of the website. They comprise the CTE Department’s logo, the department name, and the navigation bar for the different sections. They are made to be a part of Elementor’s full template so that all new pages made will have them included on the top and the bottom.

Dynamic Section Landing Pages

The landing pages for the dynamic sections -- Research, People (see Figure 3), Education & Training -- are similar in structure. They consist of three different subsections: header & footer, sidebar, and a grid loop. The sidebar is there to navigate the different subsections of each section with their own personalized menu and spotlight. The spotlight highlights a specific post tailored to each section by random, and has a link to the post at the bottom. The grid loop uses a custom-made loop item to showcase the different posts in a uniform, baseball card template.
Individual Post Templates

The individual post types have a similar structure to the dynamic section landing pages, but the grid loop subsection is replaced with a single post, and the whole page is dedicated to that particular post. For an example of a faculty post (see Figure 4).

Figure 4. Example Faculty Post
Automation

One of the big deliverables that we wanted to meet was the task of automating changes in the look of the website whenever there is a change in data. There are two ways that we were able to achieve this: linking the CPTUI database with the front end of the website, and the usage of cron jobs to update the taxonomies based on specific conditions.

Custom Forms to Populate Data

In order to simplify the process of updating the website, we wanted to avoid simply recreating the website on Elementor statically. We knew that the re-creation of the website could be done all in Elementor, but once Dr. Murali or any other manager of the website would take over, the process of updating the website would be incredibly time-consuming, thus negating the original purpose of this project. With the help of CPTUI, we could emulate the properties of a database, that we could pull the information from, and update the site dynamically. Therefore, the only aspect that the site maintainer would need to do, would be to edit or add upon the Custom Post Type Query instead of touching the front-end design of the website. The way this is possible is due to the Loop Grid element in Elementor. The loop grid element allows you to make a custom element section and link it to a database, in this case, CPTUI custom post fields, looping the element for each post in the database. This is what we implemented for the landing pages for all the dynamic sections of the website. Since every entry in the CPTUI is an individual post, we used the custom single-page template that we made in the section on automation and applied it to make a single page. Therefore, editing or creating a new post in the CPTUI plugin would dynamically change the site as well without the need of using Elementor.

Cron Jobs

In general, a cron job in computing is a command used for scheduling tasks to be executed in the future. In WordPress, you can achieve this by writing a custom function in the functions.php file to be fired at a set interval. Usually, a cron job is used to have daily maintenance on the site. We have used it to automatically update data on our Custom Post Types. For example, in the events section, there are two taxonomies: active and past events. They are used to distinguish events that are planning to be held, and events that have already been held. Once a day, the cron job function compares the date of the event to the current date and updates the taxonomy accordingly so that Dr. Murali himself does not need to change the categorization of all the events. Furthermore, the cron job function can also be utilized in other pages such as the people section. The dates of graduation for current students can be compared to the current date and can move them to the alumni section. However, there were some complications as the fields for the alumni and current students are quite different. Therefore, we decided to not include that in this project for now.
Testing/Evaluation/Assessment

The live preview function on WordPress allows someone to be able to test the different changes that they are making on the website, live. To test the functionality of adding people onto the page, we created a test case and added a user to the people page. The person was put as a student. The filter was able to filter it correctly and display it when the student's filter was clicked. Figure 5 shows an image of a test user being added.

Furthermore, the basic functionality of the site had to be tested. For instance, the navigation bar that is present in the header in all the pages of the website all correspond to their respective links on the website. There were three main tests that we wanted to see if the website would behave the way it should: updating grid loops on dynamic pages, home page functionality with sliders, and spotlight functionality.

Updating Grid Loops

The grid loop is the section with the small snippets of information that leads to individual post pages as shown in the Design & Site Templates section diagram. It should loop through all the individual posts that correspond to the pertaining Custom Post Type, and show it in a uniform manner in the main section of the page. Whenever a new person is added to the “database”, the grid loop should be updated with the new loop item in place. For example, when we added the test user “John Doe”, not only was a post page created, but the grid loop was also updated as shown in Figure 6.
We tested this updating grid loop for all the pages that utilize this method: News & Events, People, and Research pages. They all behave similarly with no problems.

Figure 6. Example Test Case for Adding New Person on Grid Loop
Some of the features of the site include being able to browse through the different people on the website that are featured as shown in Figure 7.

Figure 7. Faculty Page

There are filters on the people’s page where a user is able to filter through the different people that are featured on the website. Users can also look into news and events that occur within the CTE department. The navigation through the different pages is fairly simple and the filters make it easy to sort through the different people. Clicking the read now button will allow the user to read more about a person.
Discussion of the use environment

The use environment of this application could be on any browsing engine such as Google Chrome or Internet Explorer. This page, shown in Figure 8, can be accessed by anyone, but only the admin has permission to make changes to the site.

Use Cases / Tasks Supported

- Use Cases:
  - A user wants to look at the students that are part of the CTE program.
  - Admin trying to add a new person to the site
  - A person that wants to see the news and events that are happening
  - A person who wants to learn about the CTE program

- Tasks Supported:
  - Combing through different categories of people's entries
  - Having the ability to filter people's entries
  - Having the ability to comb through sorted & relevant news
Developer's Manual

The site architecture of the website is depicted in Figure 9.

![Site Architecture Diagram]

Figure 9. Site Architecture

Installing Software

- The WordPress account, needed to update the site, is already created and any new members can be invited to edit the site.
- The necessary plug-ins that need to be installed are:
  - CPT (Custom Post Type) UI plugin
  - ACF (Advanced Custom Fields) plugin
  - Elementor plugin
  - JetSmartFilters plugin
  - Elementor Custom Skin plugin
- To obtain these plug-ins, you would have to go to the plugins button on the left. It will take you to a page where you can search for the plugin that you want. You can search for the plugin you want and click install. This should work for all the plugins except the JetSmartFilters plugin which needs to be installed externally.
  - You can install the JetSmartFilters Plugin by buying it from crocoblock.com/plugins/jetsmartfilters/.
  - We paid for the JetSmartFilters plugin and used it to make the select filters throughout the site
Creating a Post Type

- To create a post type, you would click on the CPT UI button on the WordPress sidebar and then click on the “Add/Edit Post Types” button, which will direct you to a form page.
- In the form at the top, you can choose what you would like the name of the CPT to be by filling out the fields for the slug and the labels.
- There is a button that says “Populate additional labels based on the chosen labels”; click on that.
- There will be additional forms on the page where you can fill out different features and settings for the Custom Post Type, which are optional.
- When you’re finished, click on the “Add Post Type” button.
Creating a Taxonomy

- In order to create a taxonomy, you would click on the UI button on the WordPress sidebar.
- In the button’s dropdown, click on the “Add/Edit Taxonomies” button.
- After clicking on that, you will be directed to a form that you can fill out to make a new Taxonomy.
- Make sure to attach the Taxonomy to the Custom Post Type of your choice.
- When you’re finished click “Add Taxonomy”.

Figure 11. CPT UI Adding a Custom Post Type

Figure 12. Form for Adding a New Post Type
Figure 13. CPT UI Taxonomy

Figure 14. A Form to Fill Out for when Creating a Taxonomy
Creating a custom field type

- In order to create a field type, you would have to click on the ACF button on the WordPress sidebar.
- You will be directed to a page where you can click on the “Add New” button at the top of the page.
  - Click on that and you will be directed to a page where you can create a new Field Group and add new fields to the Field Group.
- After you finish filling out the Field Group, use the settings form, at the bottom of the page, to apply this Field Group to a specific CPT, and optionally a specific Taxonomy.

![Figure 15. ACF Creating a Custom Field Type](image)

Adding someone to the website

- To add someone to the website, you would have to fill out a form for the Custom Post Type as shown in Figure 17.
- Click on the specific Custom Post Type you’re trying to add.
- Click the “Add new” button in the dropdown shown in Figure 16.
- The form is structured in a way that it is easy to add entries.
- Click on the settings button next to the “Update” button.
  - Select the Taxonomy you want for this entry as shown in Figure 18.
  - Add an image by inputting an image into the Featured Image section.
- When you are finished filling out the form, click the “Publish” button to add the new entry to the site.
Figure 16. People CPT

Figure 17. Form to Add People
Lessons Learned

Timeline/schedule

- We were able to complete the tasks that we set forth to achieve in our timeline.
- Our Timeline follows.
- February
  - Initial Research
    - We conducted research into what type of CMS to use that would best fit the needs of the site.
- March
  - WordPress Familiarity/Plugins
    - We familiarized ourselves with how to use WordPress.
    - We also acquired different plugins to use for the site such as JetSmartFilters, Elementor, CPT UI, and so on.
- April
  - Backend Development
    - We created Templates, Custom Post Types, Taxonomies, and Custom Fields for the CPTs.
- May
  - Finalization of the Site and the Report
Milestones/Timeline

Figure 19. Timeline for our website

Problems

- JetSmartFilters
  - When adding the JetSmartFilters select-type filter to the Alumni and Current student pages, there were some issues.
  - When the filter was selected, and a user attempted to filter the post objects on a page by pressing the “FIND” button, the page would re-populate the posts with the Spotlight objects.
  - The issue with the filter was that, when you are setting it up in Elementor, the posts object or loop object has to be given a CSS ID.
  - After you assign it an ID the filter’s Query ID should be set to be equal to this CSS ID.
  - This issue was complicated to solve because the documentation wasn’t necessarily straightforward on how we should go about setting up a select filter when you have multiple post objects or loop objects on a page.

Solutions

- For the JetSmartFilters, the solution to our issue was giving our Post/Loop object a CSS ID and then setting our filters Query ID equal to the filter.
- This took a lot of time just tinkering with the settings until it worked.
Future work

- The website should be exported into a VT WordPress environment.
- The admin will need to acquire a VT domain name and finalize hosting, probably by speaking with a network liaison.
Acknowledgments

Special thanks to:

Dr. T. M. Murali
Email: murali@cs.vt.edu

Dr. Edward Fox
Email: fox@vt.edu
Bibliography/References


Appendix

Goals

- The goals of the website are to allow for easier updates.
- Rebuild the website using a superior CMS and tech stack
- Maintain the website using a more intuitive CMS
- Allow the owner to make substantial changes to the underlying code
- Create an optimal user experience for website visitors
- Update the User Interface to follow modern trends

Break-down of goals

- Rebuild and maintain the website using a superior CMS and tech stack
  - Research CMS options
    - Look into WordPress
      - Find articles and tutorials on website maintenance
    - Choose a tech stack that is suitable for the website
      - Choose a language that works best for the frontend
        - Look into frontend frameworks to make development easier
          - Find tutorials and articles on these frameworks
        - Choose a language that works best for the backend
          - Look into backend languages that are best suited for WordPress
            - Find tutorials and articles on these frameworks
    - Begin building the frontend with these frameworks
    - Begin building the backend with these frameworks
- Allow the owner to make substantial changes to the underlying code
  - Research ways to implement the CMS in ways that are suitable for the owner
    - Build the site so that the owner has the ability to update the code through the CMS
- Create an optimal user experience for website visitors
  - Make CTE Program information easily accessible
    - Plan the frontpage to have panels and articles with information that is relevant
      - Build out frontpage based on these plans
    - Allow users to access updates from the website owner
      - Build the website to update with changes made by the owner
    - Allow users to find past information from the site when updates are made
      - Build the website to have a backlog to store past information
        - Implement this backlog when building the backend
    - Make the website quick and easy to navigate
- Use a powerful frontend framework to reduce loading times
- Update the User Interface to follow modern trends
  - Create modern wireframes for the website
- Build a site based on these wireframes

Workflows for the four main goals follow.

Goal 1: Workflow 1 = Look into WordPress + Choose a language that works best for the frontend + Begin building the frontend with these frameworks + Look into backend languages that are best suited for WordPress
Goal 2: Workflow 2 = Build the site so that the owner has the ability to update the code through the CMS

Goal 3: Workflow 3 = Plan the frontpage to have panels and articles with information that is relevant

Goal 4: Workflow 4 = Create modern wireframes for the website