Introduction

The project is about the design and development of the SparcLab team web page. This initially started with the aim of developing and integrating circuit stimulation initially work on the existing website. The lack of flexibility on the previous website and different other content management systems (CMS) in addition to the familiarity with Django made us choose Django as a framework.

The first version is developed on Django. The `download.html` file is only present in the Django version. The mirror copy of the website is added to the Purdue Server. There have been several changes done on the mirror version website on the Purdue server. Additionally, the Django-CMS version of the current website is developed.

Infrastructure

In order to develop the Spar Lab team website. The website is completely developed using Django (a Python-based web framework) on a lab computer. The code is also found on a private GitHub repository. As the website was being developed, the different version was made available to Heroku server (a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud).

Each development stage was tracked using Verison Control System (VCS) like Git. Git is a free and open-source distributed version control system (VCS). VSC is a system that records changes to a file or set of files over time so that you can recall specific versions later.

Since Purdue University server (where the actual publicly available website is hosted) doesn't allow the software to be installed, we made a mirror copy of the website from the Heroku server using HTTrack and made the website available as local files.

Frameworks

We chose the Django framework because of familiarity, compatibility, flexibility, and robustness. The website is developed using a `conda` environment with Python 3.7.4 and Django 2.0.5 along with a few additional packages. Django is a Python-based free and open-source web framework that follows the model-template-view (MTV) or model-view-control (MVC) architectural pattern.

Django-CMS is a Django based Content Management System. Which is well tested and maintained. Django-CMS is different from the Django project because it includes a middleware that handles the create, read, update, and delete (CRUD) basic functions. And It is a faster way to make ordinary websites quickly without much backend development. Once the template is properly configured with Django-CMS or built using the different bootstrap modules in Django-CMS, we can easily make a change by login into the backend.

If you would like to login to the admin page or the backend on Django-CMS write the code below after the URL on your address bar.

```
http://localhost:8000/?edit
```

This should ask you for the appropriate credentials (user and password). Once you provide the right credentials, you should be able to use all the functionalities at the backend and frontend to make the necessary changes on the website.

On the front side, we can use the structure bar to make the edit. The structure toggle is the quickest way to make structural changes. From the structural bar, we can choose pages to get to the backend or we can simply log in as an admin using the code below on our address bar and provide the proper user credentials.

```
http://localhost:8000/admin/
```

List of Pages

1. Home
2. Research Areas
    a. Ongoing Research Area
    b. Research Area
        i. High-Speed IO
        ii. Low Power
        iii. RF Test
        iv. Species Identification
3. Highlights
    a. Media
    b. News
4. Awards
5. Team
6. Publications
7. Teaching
8. Service
9. Chip Gallery
10. Openings

The three web pages that were specially made are the publication, team, and chip gallery pages. In the publication page, there are three custom js functions (namely `filterSelection`, `publicationAdd`, `publicationRemove`) that do the selection of content when a button is clicked.

On the team page, the original `Card` module has been modified to remove the marge and padding around the items. The size has been slightly enlarged. The hovering functionality is custom added to the added image in each card. The chip gallery page includes the same functionality as the team's page in terms of showing description when hovered over the image.

Purdue Web Server

The Purdue server doesn't allow us to install any kind of packages other than the ones that they already installed on there. Since both our sites are developed using some sort of framework (Django or Django-CMS), the website cannot be functional with a simple file copy-paste process. Therefore, we mirrored the website using HTTrack and stored the file locally on the Purdue server to make it live.

Contribution

There is a total of 18 pages that are custom designed and developed with a unique look and template. Unique functionality was also custom coded using Javascript. A large amount of data on the site was reviewed to make sure each link and video embedding are still available. Accent mark on the site was maintained to preserve highlighting of useful information. The website went through a series of design and development iterations. Both Django and Django-CMS versions are locally available on the lab computer. To make the website live on Purdue server, HTTrack was used.