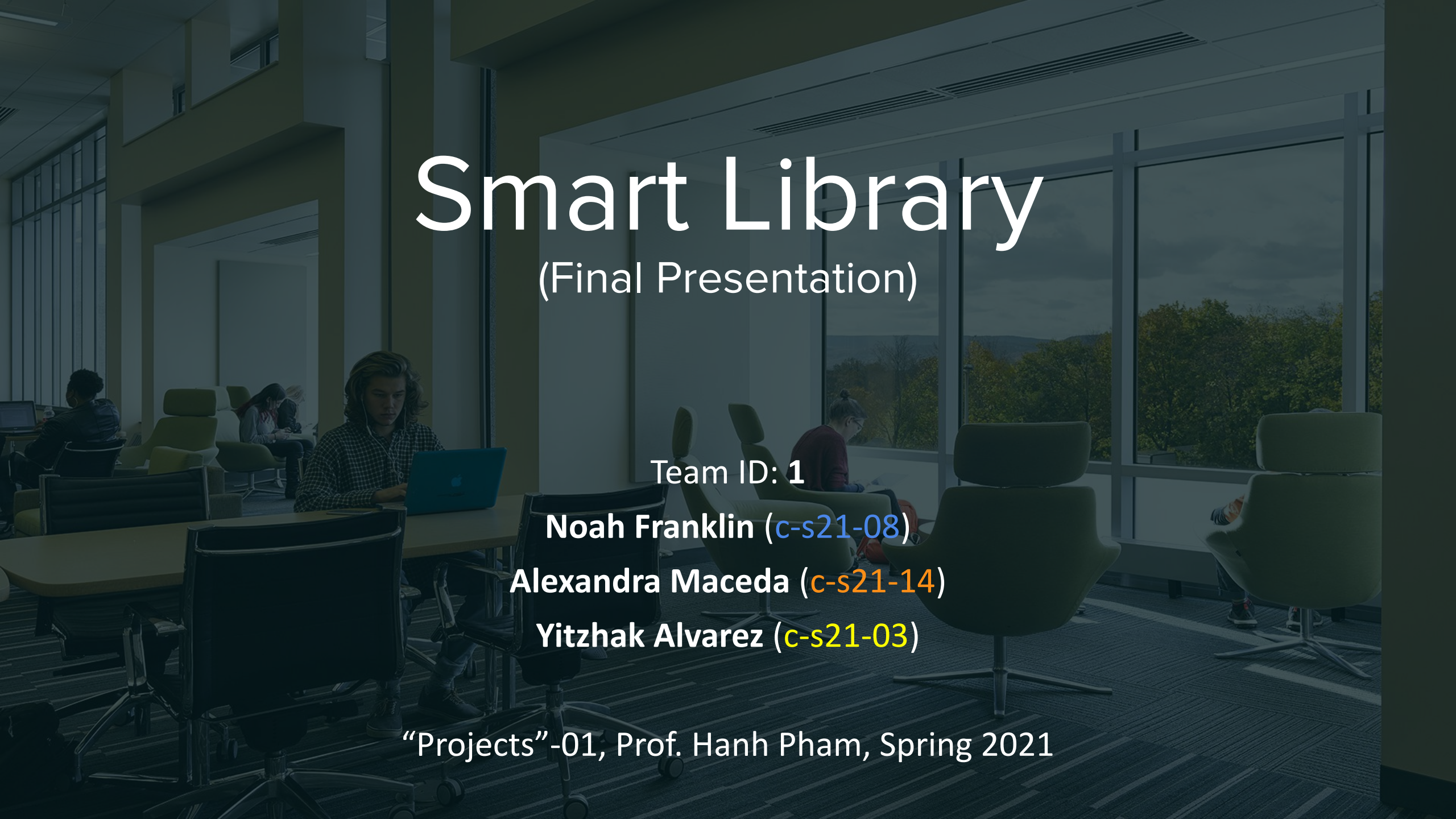


Smart Library

(Final Presentation)

Team ID: **1**

Noah Franklin (c-s21-08)

Alexandra Maceda (c-s21-14)

Yitzhak Alvarez (c-s21-03)

“Projects”-01, Prof. Hanh Pham, Spring 2021

Outline

1. Problem Description
   1.1. Goals and Functionality

2. Technologies
   2.1. Related Technologies

3. Design
   3.1. System Architecture
   3.2. Components

4. Software/System Description
   4.1. Map of Files
   4.2. Developer’s Guide

5. System Demo

1. Problem Description

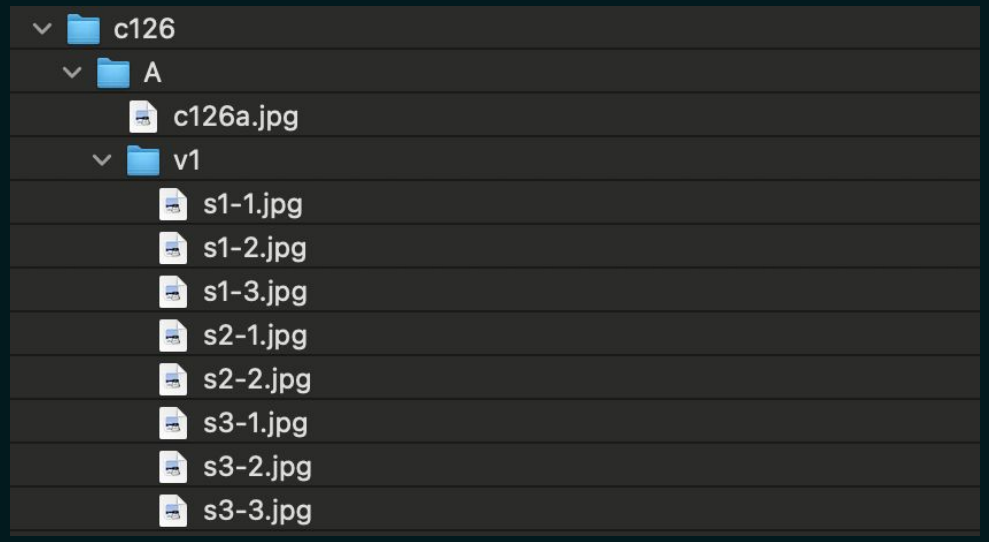
GENERAL DESCRIPTION

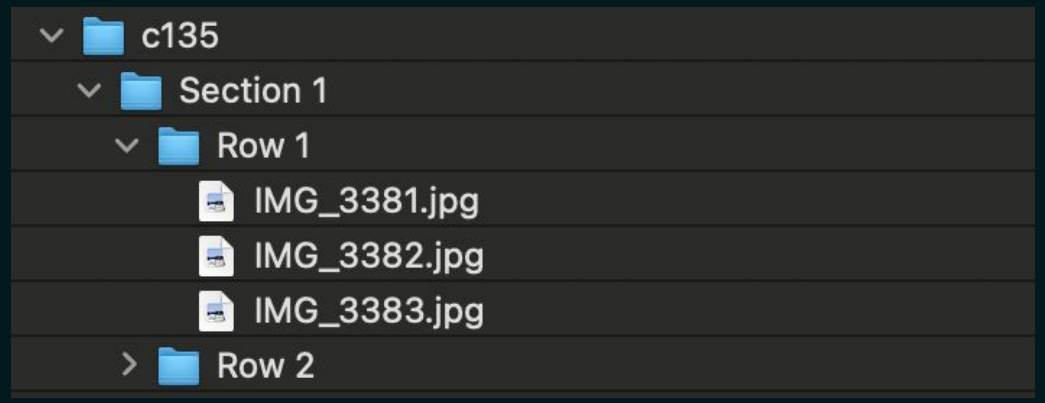
Creating a virtual library for SUNY New Paltz which accurately reflect book locations without adding new tags to books. Users can also browse library floors (Main and Concourse), and bookshelves. When clicking on books students will be able to go to library link of that book. Goal is to closely mimic a virtual experience while keep the application simple and easy to use.

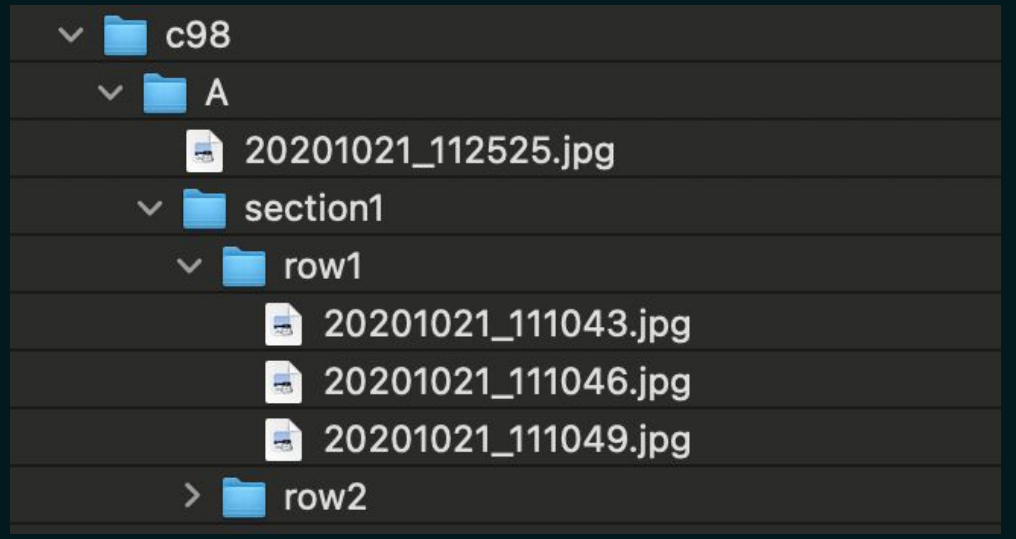
Made for the Sojourner Truth Library

PROBLEM #1

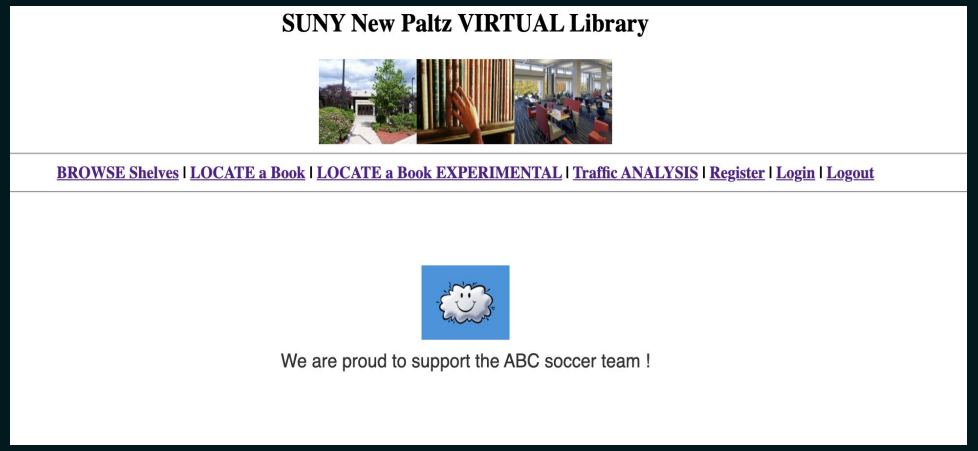
The book images in the A server were not following a consistent structure. This would make it harder to access each image when dynamically mapping in the site.

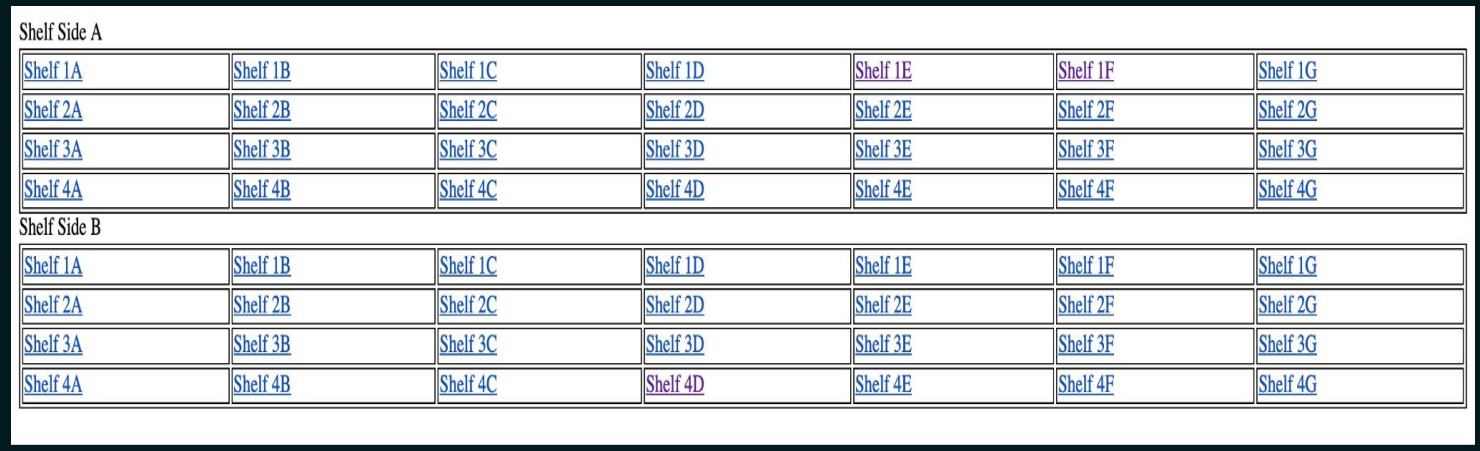




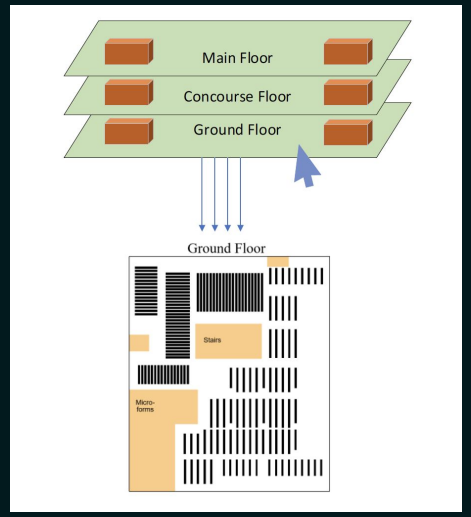


PROBLEM #2





Previous web design lacked a user friendly interface, maps did not link to shelves properly or book images.



SUBGROUP GOALS

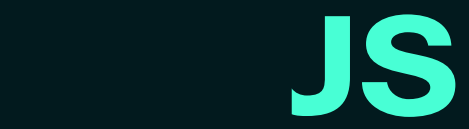
File Reorganization

Our goal was to reorganize all shelves in the A server, so it followed the following structure:

Shelf>>Side>>Section>>Row>>Book

Website Redesign

Our goal with the new version was to finish linking the maps and shelves, all while updating the user interface.



2. Technologies

2.1 Related Technologies

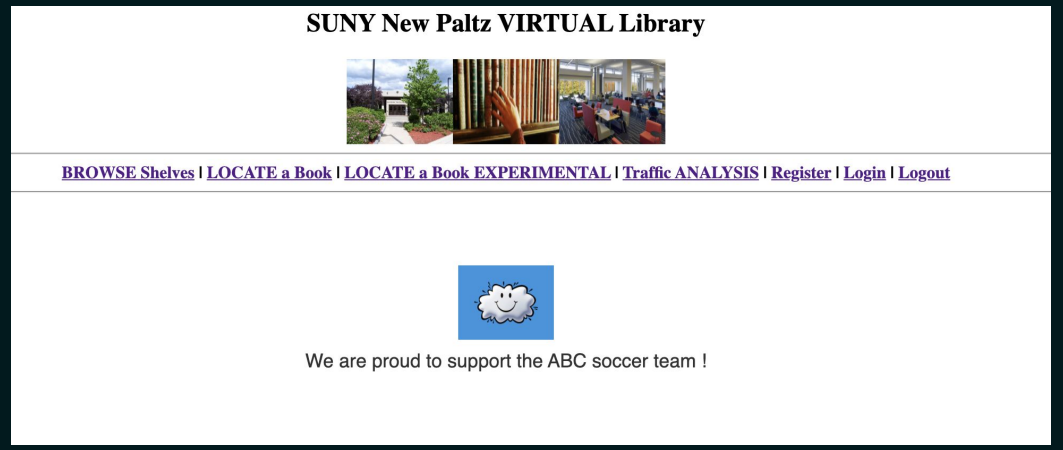
Frontend/UI

- HTML - Used to structure the web pages
- CSS - Used to style
- Bootstrap - CSS framework for responsiveness and styling
- JavaScript - Used to make the webpages interactive
- jQuery - Much easier use of JavaScript on our website

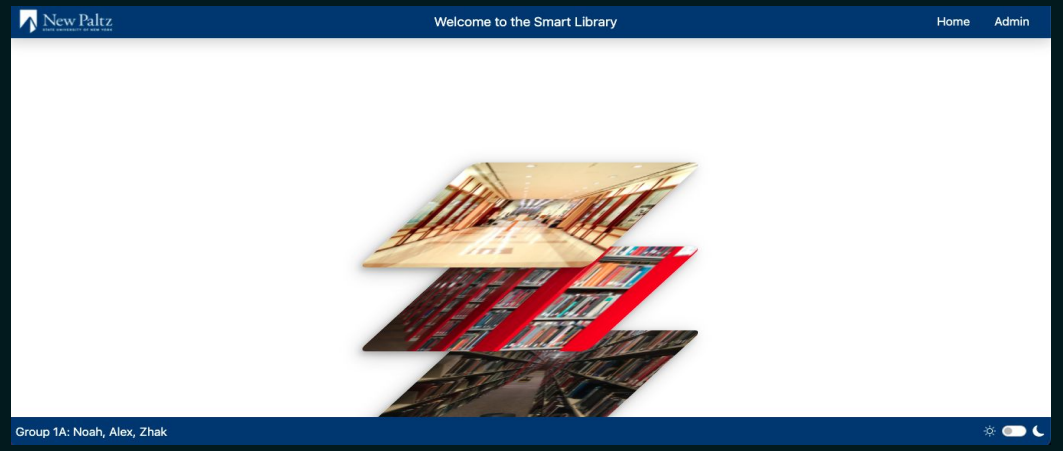
Backend

- PHP - Interact with the database to display on the front end
- MySQL - Create tables and queries for the database
- Python - Used to make shelfgenerator.py which goes thru every organized image and creates HTML files for bookcases & bookshelves

Existing Homepage of Smart Library



Newly Homepage of Smart Library



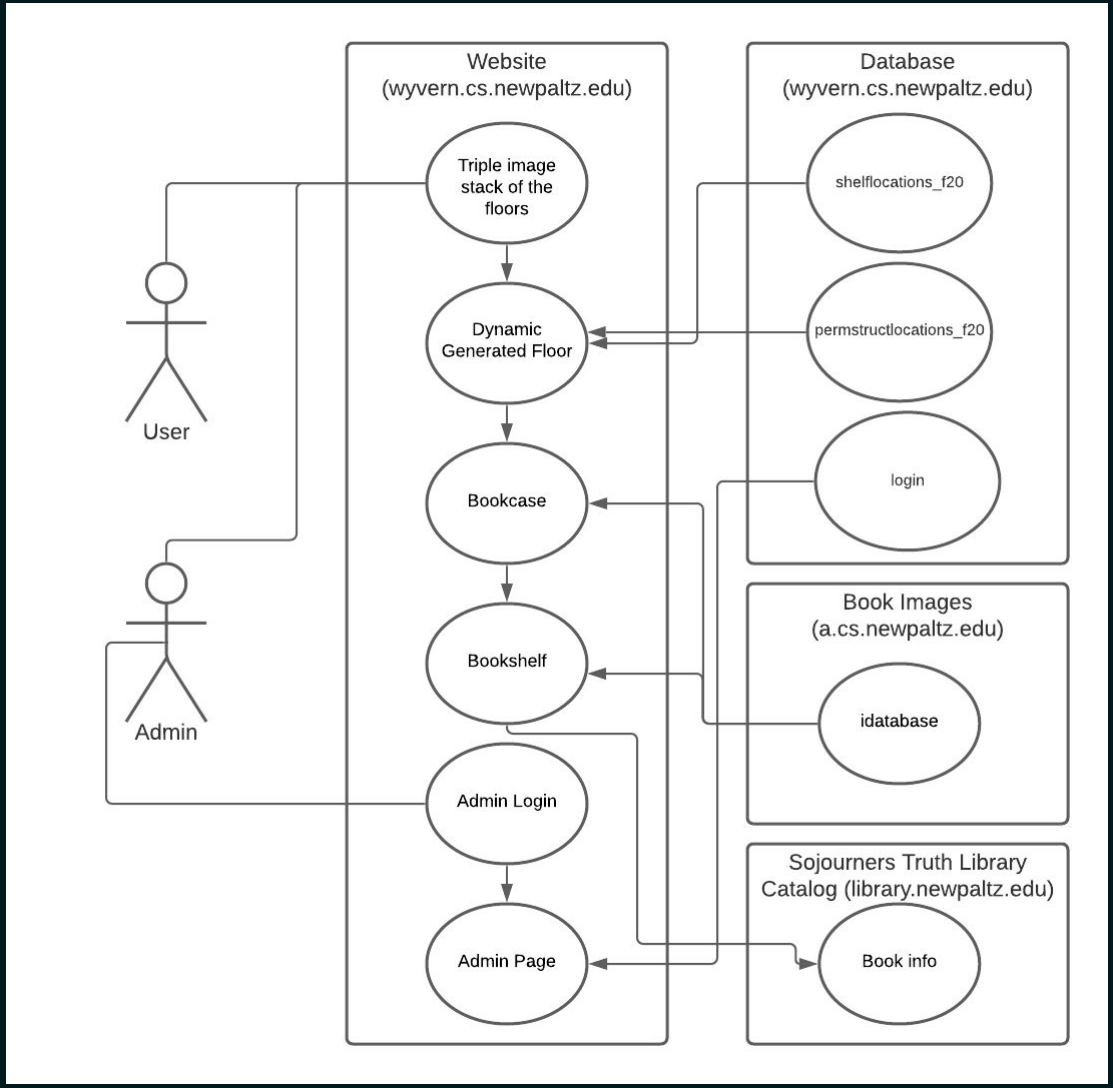
2.2 Newly Learned Skills/Technologies

- Figma - Great design software tool for us to prototype our website
- FileZilla - Great for file transferring and renaming folders
- PHP - Good for backend development
- jQuery - Simple but advanced JavaScript
- Python - Great for making scripts

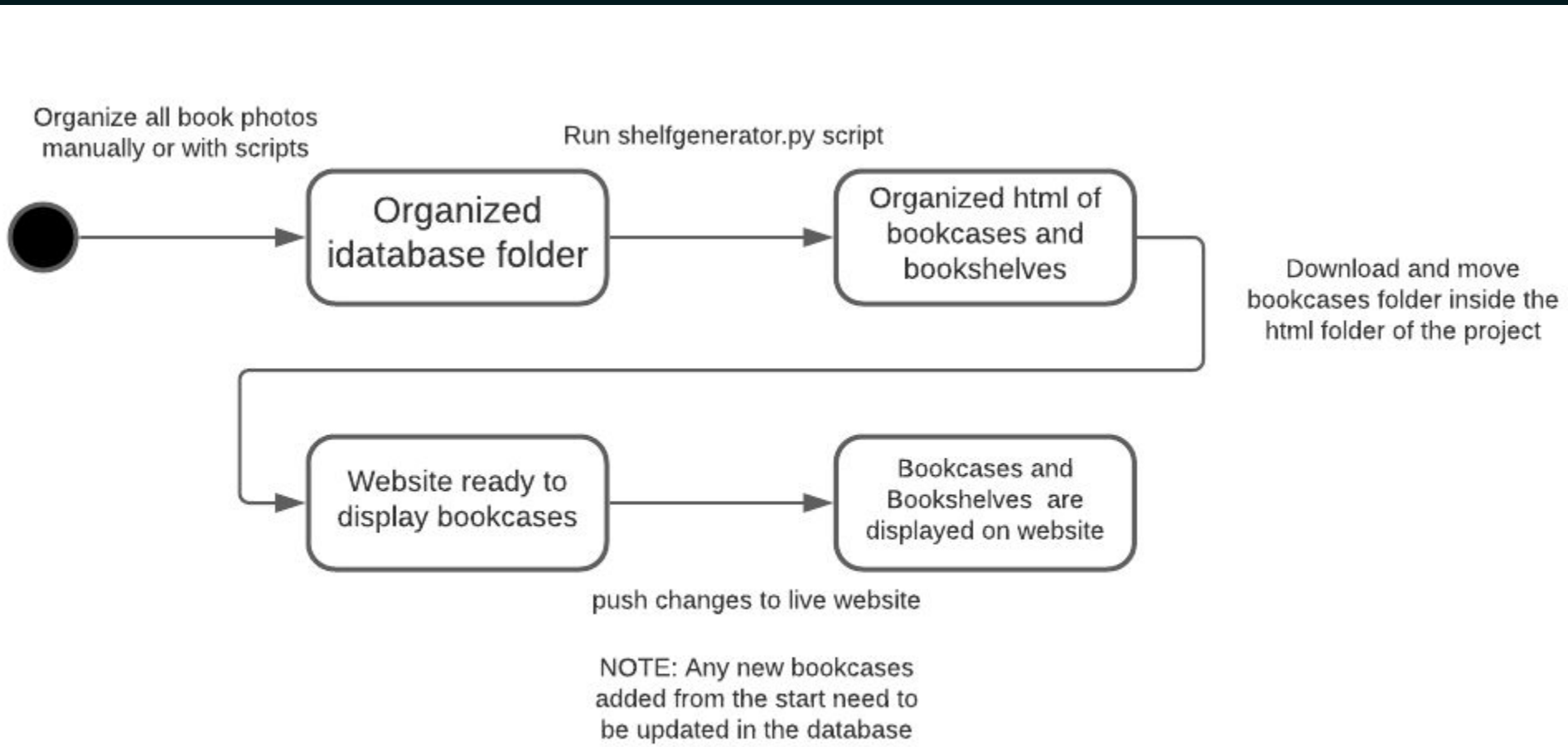


3. Design

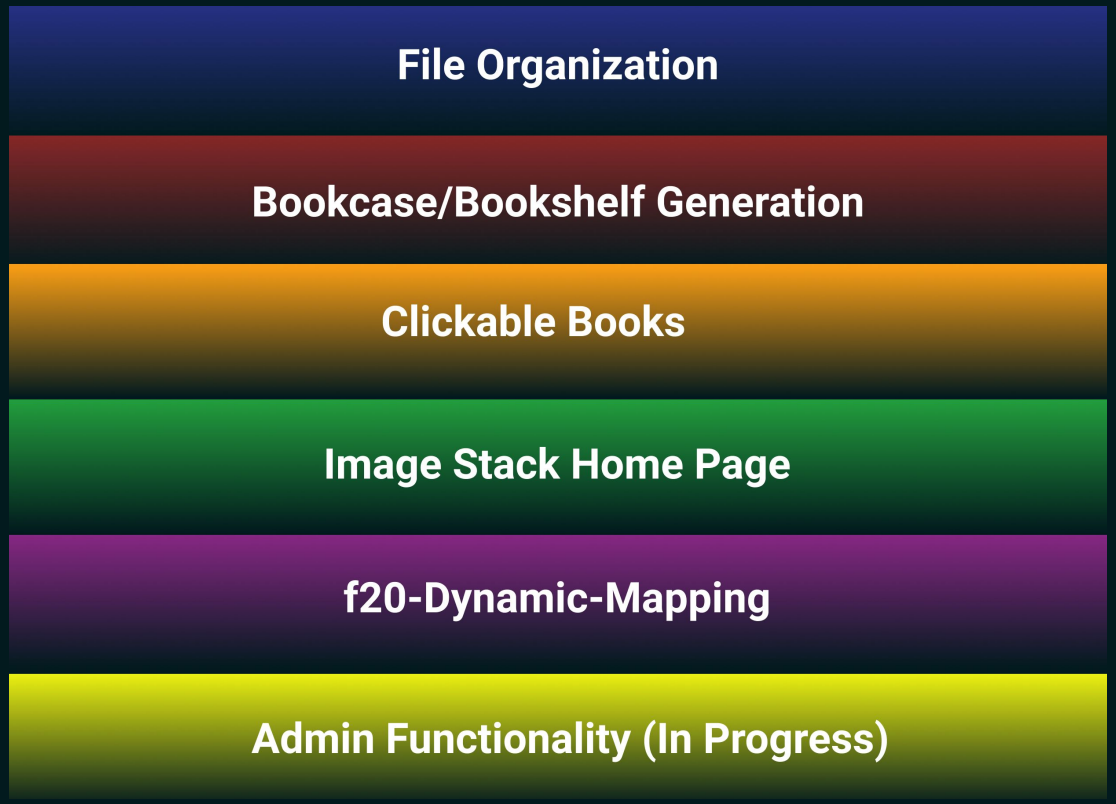
3.1. System Architecture



3.1. System Architecture



3.2. Components



3.2. Components

File Organization

Organize all the images of the books to follow a consistent structure to be used later to display the bookcases and bookshelves on the website

Format: **vlib/idatabase/number-name/shelfnumber/side/sectionnumber/rownumber/imagefile**

This was done with the combination of manually moving and renaming files and running scripts

Scripts: noah.sh, alex.sh, zhak.sh

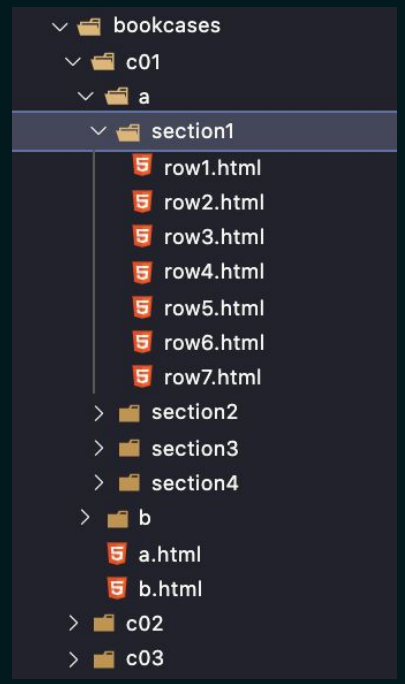
Bookcase/Bookshelf Generation

Create all the html files for every bookcase page and every bookshelf from the organized images

This was also done with the use of scripts.

Script: shelfgenerator.py

```
36  if file.endswith(".jpg"):
37      continue
38  if (file.startswith("c")):
39      # global parent
40      # global case
41      case = file
42      parent = "./bookcases/"+file
43      os.mkdir(parent)
44      get_stuff(directory+file+"/", "A or B")
45
46  if (file.startswith("a")):
47      # global side
48
49      side = 'a'
50      os.mkdir(parent+"/a")
51      f = open(parent +"/a.html" , "w")
52      f.write('<head>\n')
53      f.write('<link href="./css/bookCase.css" rel="stylesheet" />\n')
54      f.write('</head>\n')
55      # f.write("<h1>case: "+case+"\nSide: a</h1>\n")
56      f.write('<div class="bookcaseInfo">\n')
57      f.write('<span>Bookcase: <strong>'+case+'</strong></span>\n')
```



3.2. Components

Clickable Books

Add clickable sections on a book image to open up a new tab to show more info about a book on the sojourners truth website. This also has to be done based on a book call number.

Since CLARA is not fully implemented yet, we developed a solution using html maps with onclick functions but the book call numbers are hard coded in. This can be easily changed later.

Search URL :
https://suny-new.primo.exlibrisgroup.com/discovery/fulldisplay?docid=alma990002702100204844&context=L&vid=01SUNY_NEW:01SUNY_NEW&lang=en&search_scope=MyInstitution&adaptor=Local%20Search%20Engine&tab=LibraryCatalog&query=any,contains,<bookcallnumber>&offset=0

```
function getBookInfo() {
  callNumbers = ["F294.S2 J64 1996", "F294.S2 S58 2014", "F311 .D66"];
  callNumber = callNumbers[Math.floor(Math.random() * callNumbers.length)];
  callNumberFormated = callNumber.replaceAll(" ", "%20");
  console.log(callNumberFormated);
  query =
    "https://suny-new.primo.exlibrisgroup.com/discovery/fulldisplay?
docid=alma990002702100204844&context=L&vid=01SUNY_NEW:01SUNY_NEW&lang=en&search_scope=MyInstit
ution&adaptor=Local%20Search%20Engine&tab=LibraryCatalog&query=any,contains," +
    callNumberFormated +
    "&offset=0";
  console.log(query);
  window.open(query, "_blank");
}
```

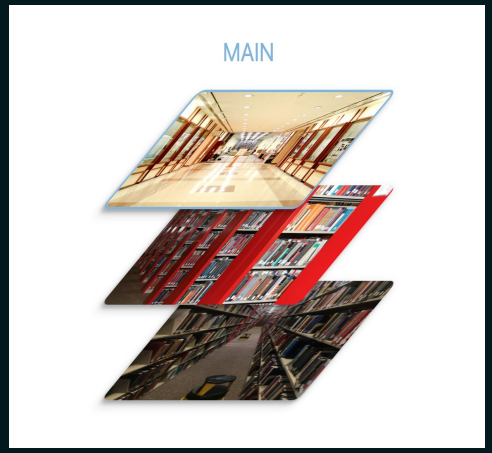
Image Stack Home Page

Create a home page displaying the three floors similar to how it appeared in the project proposal

Using some advanced CSS was able to get this done

Proposal

Live



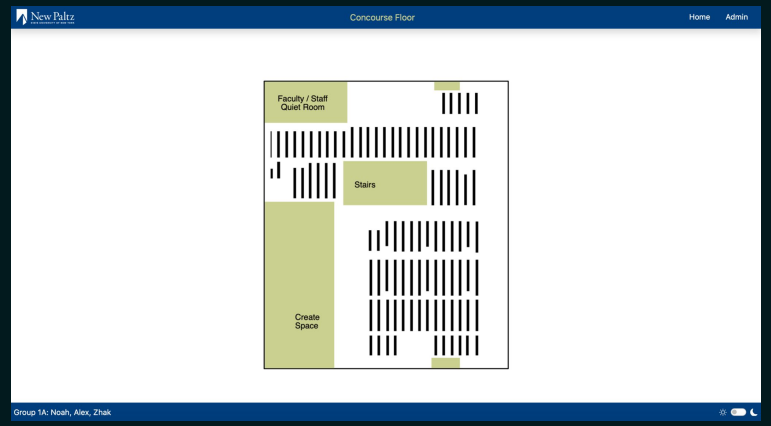
3.2. Components

f20-Dynamic-Mapping

This part of the project was already completed from previous semesters, but we needed to take and edit only certain parts of it.

We copied the code responsible for laying out the base map of each floor and populated it with the black rectangular bookshelves using the php floor displayer files which we also edited to add onclick functions to link to our bookcases.

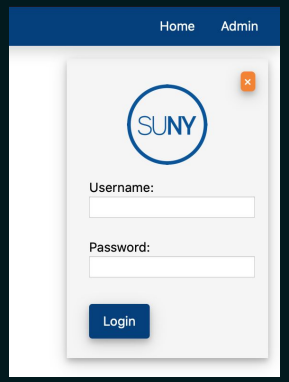
```
function getBookShelf(dir, row) {
  $(".main").load("./html" + "/" + dir + "/" + row +
".html");
}
```



Admin Functionality (In Progress)

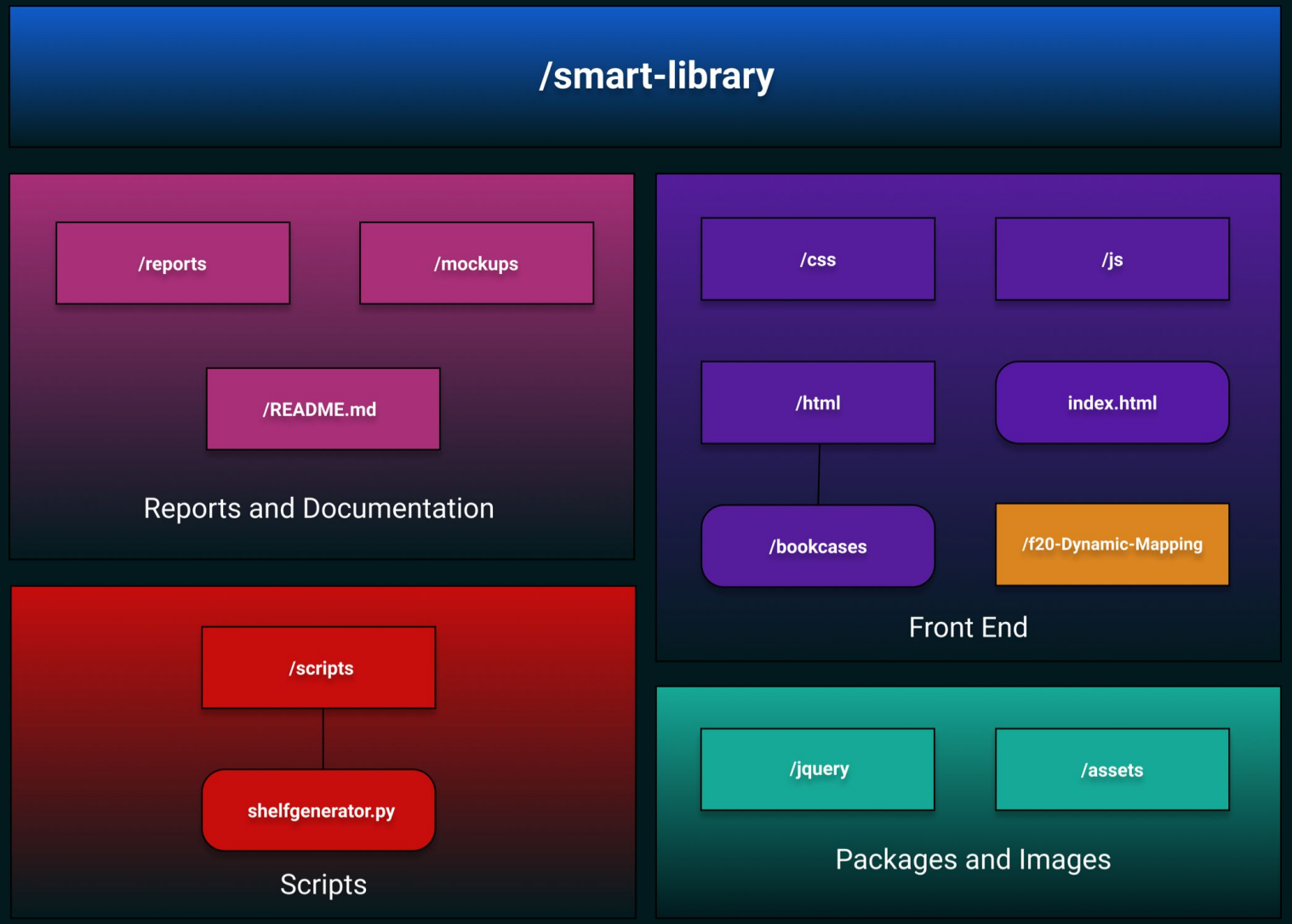
This part of the project was also already included in the f20-Dynamic-Mapping software and we got a basic version of it running.

Currently the admin page is not route protected so anyone with a direct link can access it and it also can not be accessed if the browser blocks popups which in most cases it will.

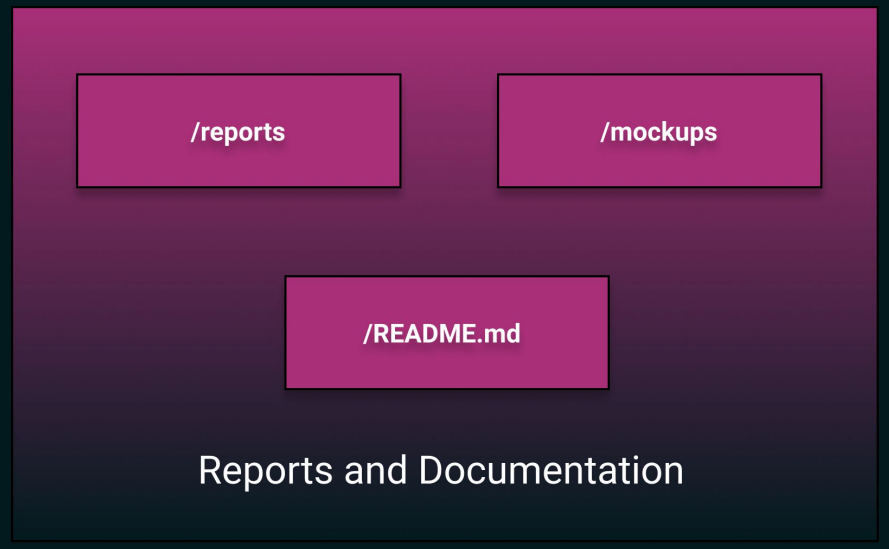


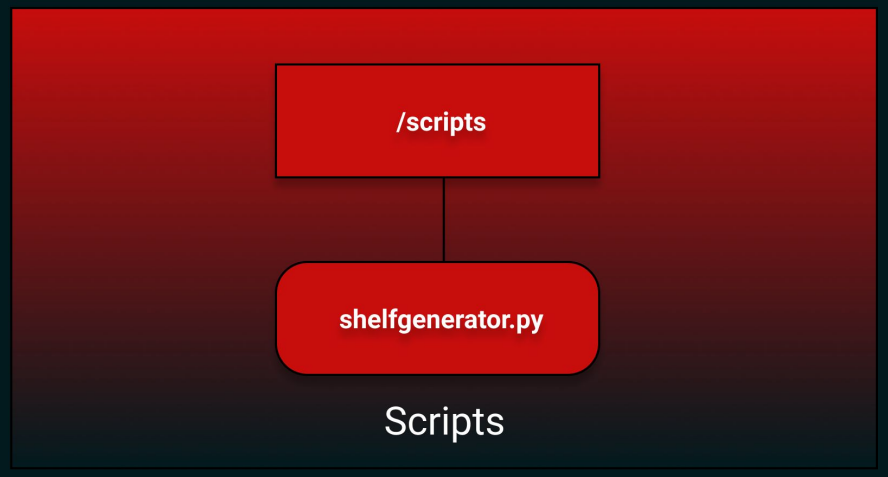
4. Software/System Description

4.1. Map of Files



4.1. Map of Files





Reports and Documentation

- **/reports –** midterm and final slides/report for the project
- **/mockups** – early website design mockups and images for the README
- **/README.md** – Project summary, running the website, and any additional description of the project to help anyone who works with it

Scripts

- **/scripts** – shell scripts used to organize folders and book images in the idatabase folder on the “a” server
  - shelfgenerator.py - python script used to generate all the html files for every bookcase and bookshelf of the images in the idatabase folder

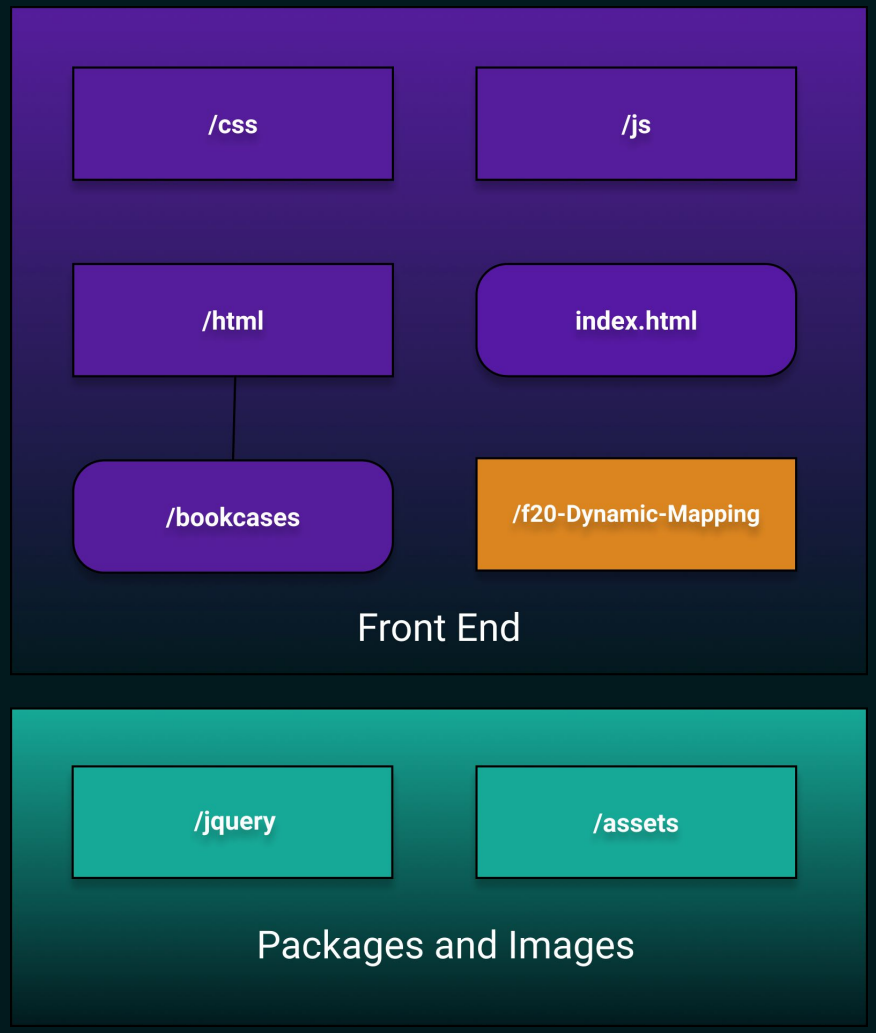
4.1. Map of Files

Front End

- **/css** – css files for styling the website
- **/js** – javascript files to make the website dynamic
- **/html** – html files to structure the website and is inserted into the website with jquery when we need it
  - **/bookcases** – organized html files that display the bookcases and bookshelves for the library, this is produced by the shelfgenerator.py script
- **index.html** – the main html file where the website is brought together, this is loaded by default when viewing the website
- **/f20-Dynamic-Mapping** – project from previous semesters to dynamically display shelves

Packages and Images

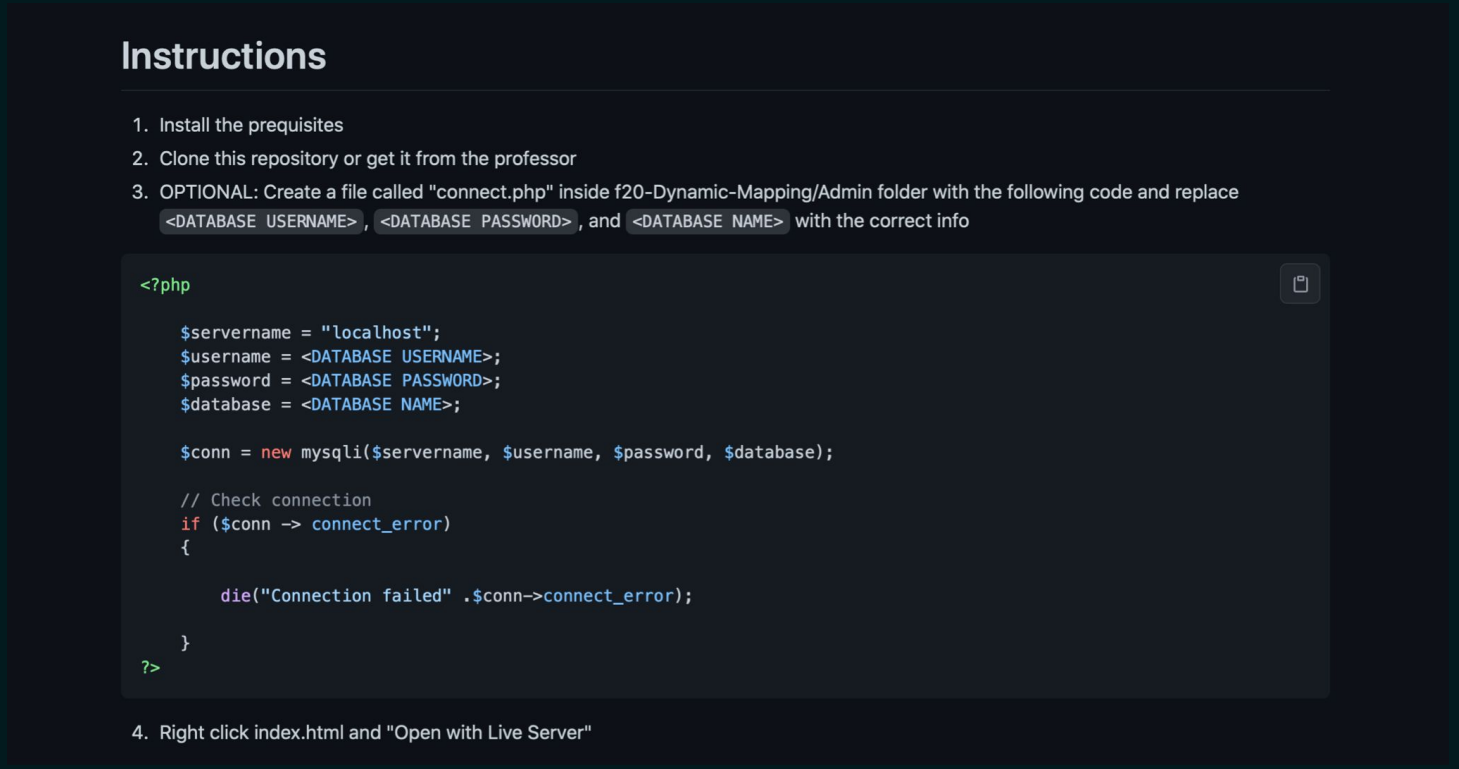
- **/jquery** – jquery package to let use easy and powerful javascript
- **/assets** - images files used throughout the website



4.2. Developer's Guide

Setting up a dev environment with the project is really simple since we didn't use a framework. As long as you have the project by cloning with git or getting it elsewhere, you can simply start the website with a local web server using the VSCode extension Live Server or XAMPP.

The only thing to note is that the dynamic mapping page will not display the black rectangular shelves locally, only on the live wyvern server.



Instructions

1. Install the prequisites
2. Clone this repository or get it from the professor
3. OPTIONAL: Create a file called "connect.php" inside f20-Dynamic-Mapping/Admin folder with the following code and replace `<DATABASE USERNAME>`, `<DATABASE PASSWORD>`, and `<DATABASE NAME>` with the correct info

```php
<?php

    $servername = "localhost";
    $username = <DATABASE USERNAME>;
    $password = <DATABASE PASSWORD>;
    $database = <DATABASE NAME>;

    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection
    if ($conn -> connect_error)
    {

        die("Connection failed" .$conn->connect_error);

    }
?>
```

4. Right click index.html and "Open with Live Server"

4.2. Developer's Guide

shelfgenerator.py is the python script responsible to creating all the html for our bookcases and bookshelves.

Adding any new bookcases, images, or moving them requires running the script to create the html for the website.

Instructions for `shelfgenerator.py`

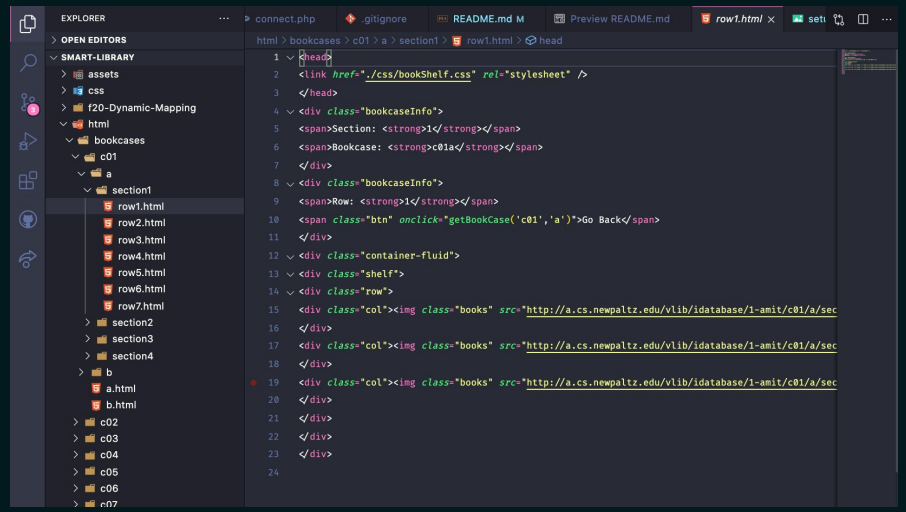
1. Use FileZilla to create a bookcases directory inside the idatabase folder (if this is already done make sure to delete everything inside the bookcases directory before running the script)
2. Move the `shelfgenerator.py` script into the idatabase folder
3. SSH into the "a" server to run the script file (./shelfgenerator.py)
4. Wait for the script to finish and all the generated html should be located in the bookcases folder

Output

The script output format is as follows:

```
/bookcases/shelfnumber/side/sectionnumber/rownumber.html
```

4.2. Developer's Guide



Attaching the bookcases html generated from the script should require nothing more then moving the generated bookcases folder to the html folder of our project.

However, if any new bookcases are added to the idatabase folder, the BookCase attribute in the shelflocations_f20 table of that bookcase will need to be added.

5. System Demo

http://cs.newpaltz.edu/p/s21-01/smart-library/