Final Project Proposal – Harry Potter Characters by House

Purpose:

The purpose of the project is to have a fun page that displays characters from the world of Harry Potter, sorted by the four Hogwarts houses. The Home screen will display four door images/links, colored according to the house colors. When the user clicks a door, they will be taken to a page displaying character cards from that house. They will be able to sort these characters A-Z or Z-A, and filter with a name search. Each character card, when clicked, will link to a character detail page. In the details, there will be a link to another page giving a few details on the actor who played that character.

The main purpose of this page is to provide entertainment and information to people who love and are curious about the characters in the world of Harry Potter.

Audience:

My audience is people who love Harry Potter books and movies.

Files, Modules and Functions:

HTML

- index.html – home page
- house/index.html – character list by house
- character/index.html – character details for one character
- actor/index.html – actor details for one actor
- partials/header.html
- partials/footer.html - partials

CSS

- style.css

JS

- **main**
- **house**
- **character**
- **actor**
- **utils**
  - Functions
    - loadHeaderFooter
    - getParam
    - getLocalStorage(key)
    - setLocalStorage(key, data)

- save the following types of data to localStorage:
  - sort: “A-Z”
  - house: “gryffindor”
  - username: “Heather”
  - userHouse: “ravenclaw”
- loadPersonalization
  - based on user selections that were saved in localStorage, adjust some display elements on the homepage
- savePersonalization
  - save as username and preferred house to localStorage for use in loadPersonalization function

- **HouseDetails**
  - Attributes
    - houseName
    - dataSource
    - element
  - Functions
    - init
      - call getCharactersByHouse(houseName) to get list of characters
      - calls getStorage(key = sort) to find out whether to sort the list of characters ascending or descending
      - call renderCharacterList(list) to display the character cards
    - renderCharacterList(list)
      - given an array of characters, display to the webpage
    - filterCharacters(search)
      - given search criteria, filter the character list to only include names containing search string
      - call renderCharacterList(list) with the filtered list as the parameter

- **CharacterDetails**
  - Attributes
    - characterId
    - dataSource
    - element
  - Functions
    - init
      - call getCharacterById to get character details
    - renderCharacterProfile(data)
      - given an array of details, display to the webpage

- **ActorDetails**
  - Attributes
    - actorId

- dataSource
- element

- Functions
  - Init
    - call getActorByName(name) to get actor details
  - renderActorProfile(data)
    - given an array of details, display to the webpage

- **ExternalServices**
  - Attributes
    - type (house, character, or actor) – determines which API to use
  - Functions
    - getData(url)
      - given the url to the API, fetch the data, convert to JSON and return result
    - convertToJson:
      - return a result in JSON format
    - getCharactersByHouse(house)
      - house = “gryffindor “, “ravenclaw”, “hufflepuff”, or “slytherin”
      - determine the correct API url, then call getData
    - getCharacterById(id)
      - determine the correct API url, then call getData
    - getActorByName(name)
      - given a name from the Harry Potter API array, look for a match in themoviedb API, if no match, give the user a message

Functionality:

Home:

The page will load with 4 Hogwarts house doors images. Each is a link to a new page. At the bottom of the page, there is a button for personalization, where the user can enter their name and select their preferred house. This will cause the background color and other styling to change to match their selection.

House Pages:

From the home page, the user will select a Hogwarts house by clicking on an image link. The house page will load with a grid of all the characters in that house. The grid is sortable A-Z or Z-A and can be filtered by name. Each character card will be a clickable link to a character page.

Character Pages:

From the house pages, the user will select a character. The character page will load with an image and details pertaining to that character. Included will be a link to the actor’s information (if applicable).

Actor Pages:

The actor page will load with an image and details about other movies they have been in.

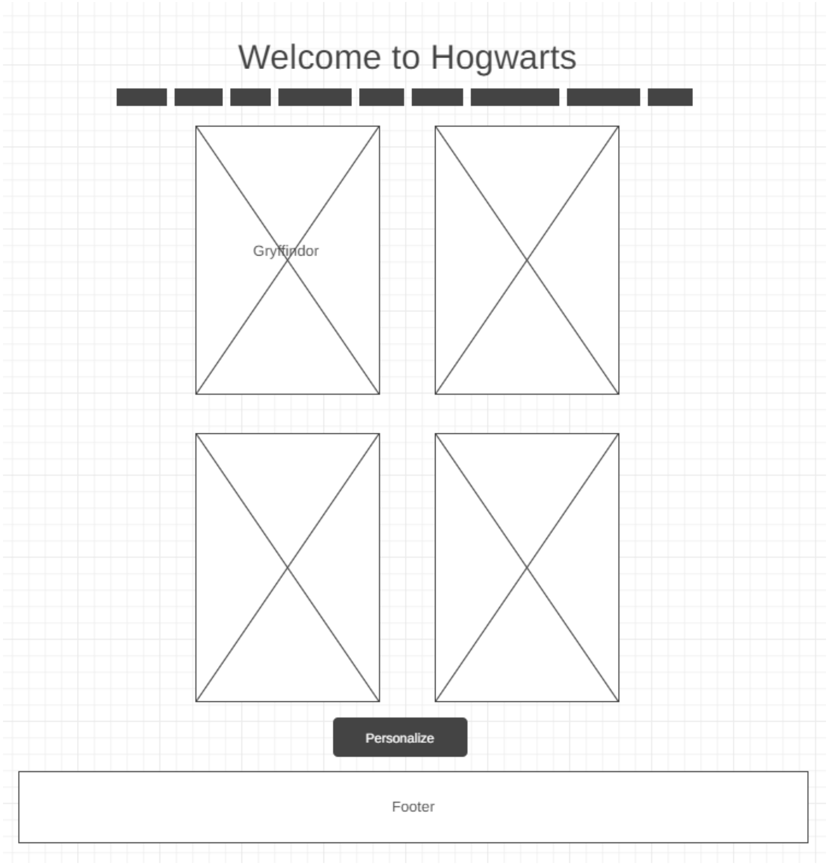
External API Data Sources:

- https://hp-api.onrender.com/api/characters
  - API Key: No key required

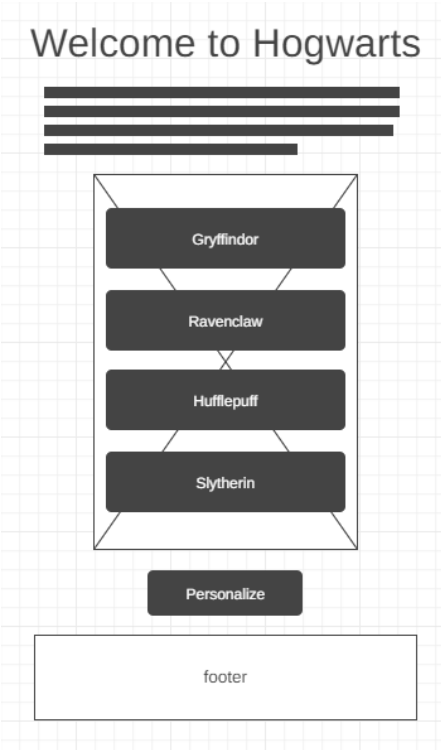
- https://www.themoviedb.org/
  - API Key: 59ff68b3de9b08b1553c8eeb39adec4e
  - API Read Access Token: eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiI1OWZmNjhiM2RlOWIwOGIxNTUzYzhlZWIzOWFkZWM0ZSIsIm5iZiI6MTcyODk0NjMzNS41MjkxOTksInN1YiI6IjY3MGQ2YjhjZjU4YTkyMDZhYTQxOGVhNCIsInNjb3BlcyI6WyJhcGlfcmVhZCJdLCJ2ZXJzaW9uIjoxfQ.AvTmWB60u1jPfQe_XQkq0kpDDt_g24JFZBzflhJ6Gtg

Wireframes:

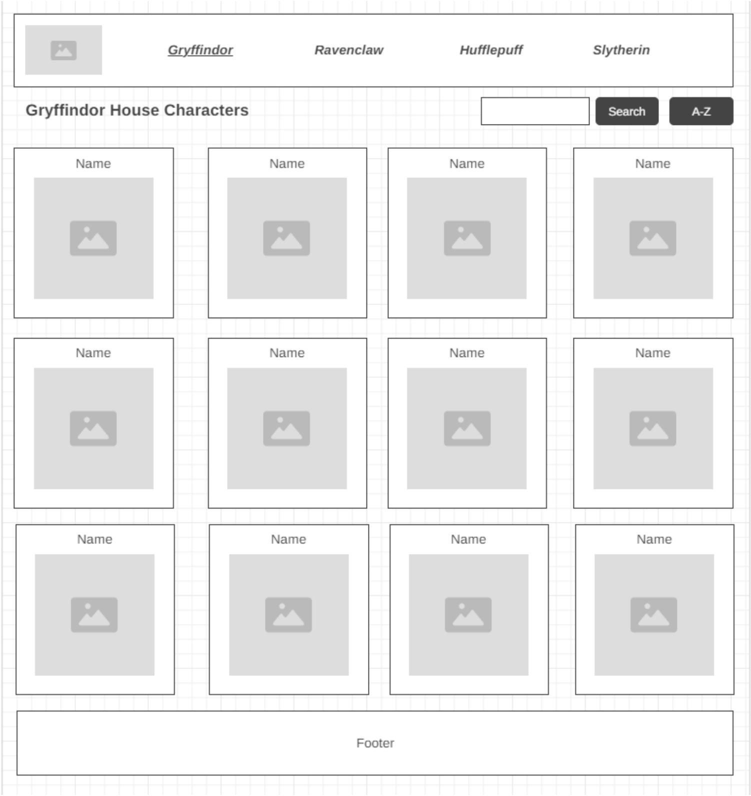
<u>Home Desktop</u>



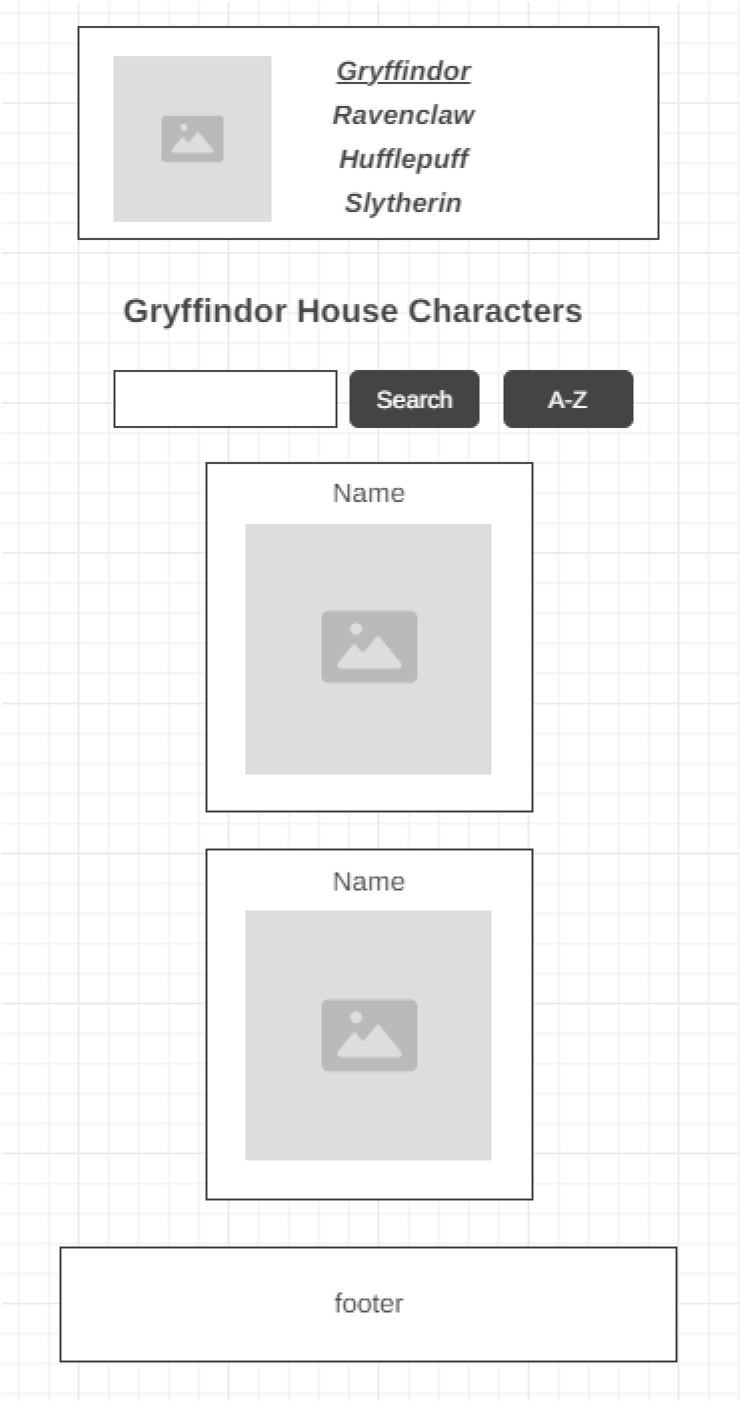
Home Mobile



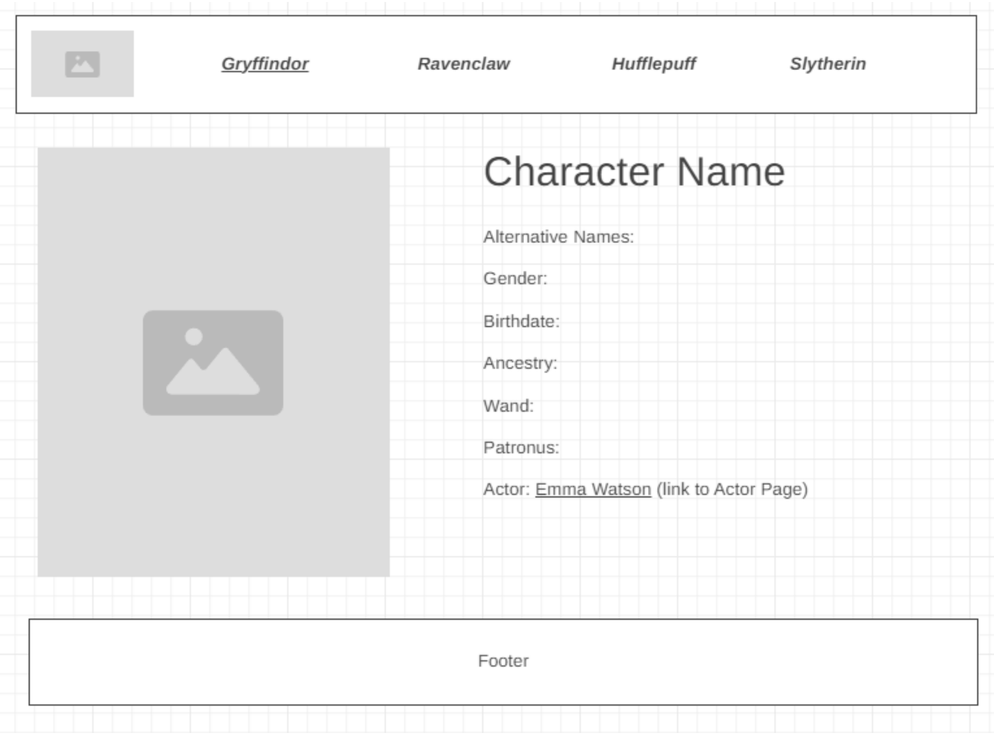
House Detail – Desktop



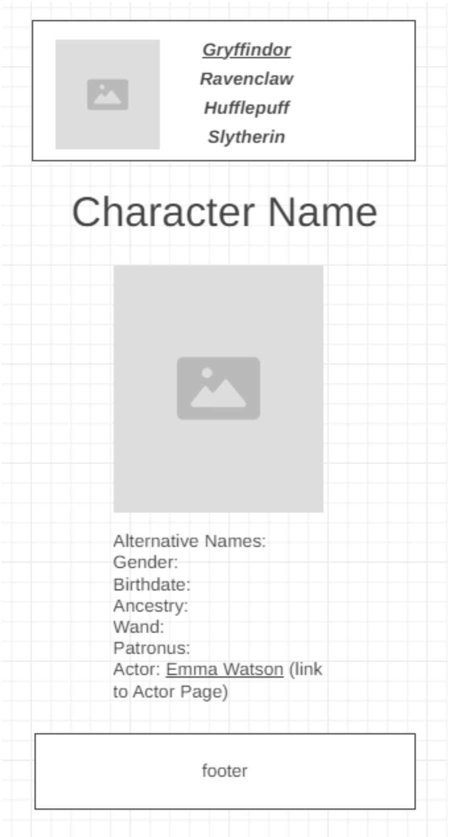
House Detail – Mobile



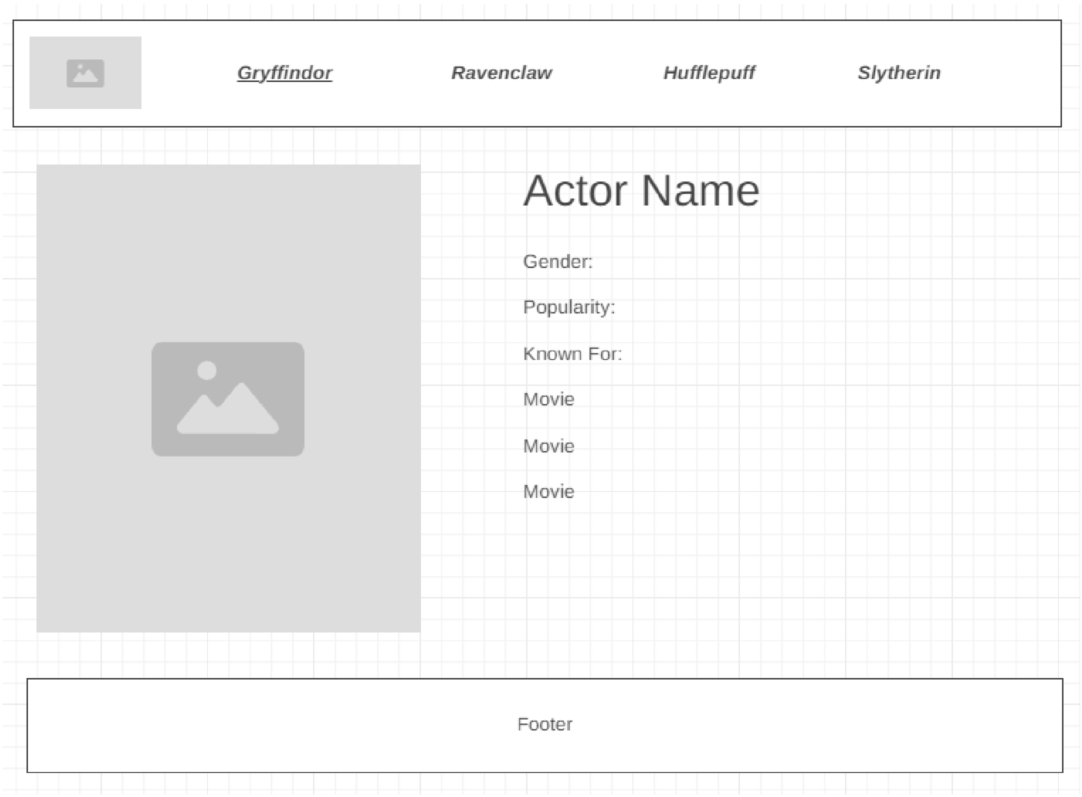
Character Detail – Desktop



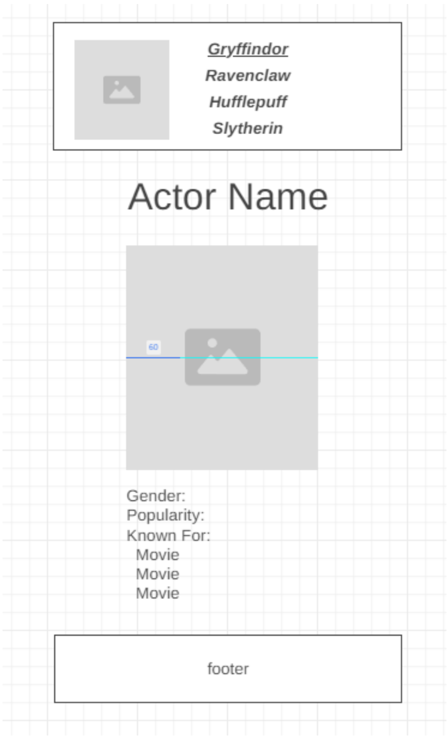
Character Detail – Mobile



Actor Detail – Desktop



Actor Detail – Mobile



Styling:

Header/Nav Bar

- Background: Black
- Font: Bigelow Rules, Georgia, serif
- Color: #EBEBEB

Titles

- Font: DM Serif Display, Georgia, serif
- Color: It depends on the background (see below)

Details/Text:

- Font: Lato, Arial, sans-serif
- Color: It depends on the background (see below)

Background Color and Font Colors: change dynamically depending on the selected house.

- Gryffindor
  - Background: #740001
  - Accent: #D3A625
  - Text: White
- Ravenclaw
  - Background: #0E1A40
  - Accent: #BEBEBE
  - Text: White
- Hufflepuff
  - Background: #ECB939
  - Accent: #372E29
  - Text: Black
- Slytherin
  - Background: #1A472A
  - Accent: #5D5D5D
  - Text: White

Schedule:

Oct 14 – 19

- Get the basic file structure set up.
- Complete the basic header, footer, and home page functionality.
- Find and optimize images for the home page, favicon, and backgrounds.
- Create and test functions to fetch data from both APIs
- Begin creation of Character List by House page, get all html, CSS, and dynamic loading complete

Oct 21 – 26:

- Add Sort and Search functionality to Character List page
- Complete Character Details page
- Complete Actor Details page
- Add transition effects, animations and other CSS
- If time, make nav menu a hamburger menu on Mobile View
- If time, add extra Personalization features to home page

Oct 28 – 29:

- Test, test, test
- Complete Skills Assessment Document
- Make a demonstration video
- Submit!!!

Trello Board:

https://trello.com/invite/b/670dcca8f3806cbc02690cff/ATTIb039dd722bcd90807c35af414502366a72B7D7CC/wdd330-final-project