Project Development Tracking

Technologies to apply

Once I've read carefully the whole assignment, I'm considering resolving it using React JS because I've realized that I won't need to develop any databases or overly complex backend implementations, which would take more time. This is because all the data is provided by the APIs.

Another alternative would be to develop the challenge using HTML, CSS and JavaScript. However, I want to take a step forward and apply React JS, which offers many benefits, such as:

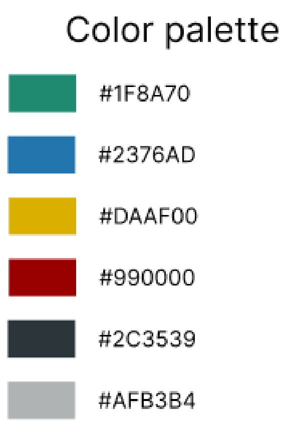
- **Component-Based Architecture:** This means it's made up of reusable and modifiable components, facilitating construction, scalability and maintenance.
- **Virtual DOM**: it achieves high perfomance by rendering only the changed parts rather than the entire page.
- **JSX**: it's a JavaScript extension that allows writing HTML code within JavaScript, improving readability.

On the other hand, I will use Chakra UI to facilitate the building of components. Chakra UI is a library for UI design that provides building blocks for React applications.

UI/UX Design

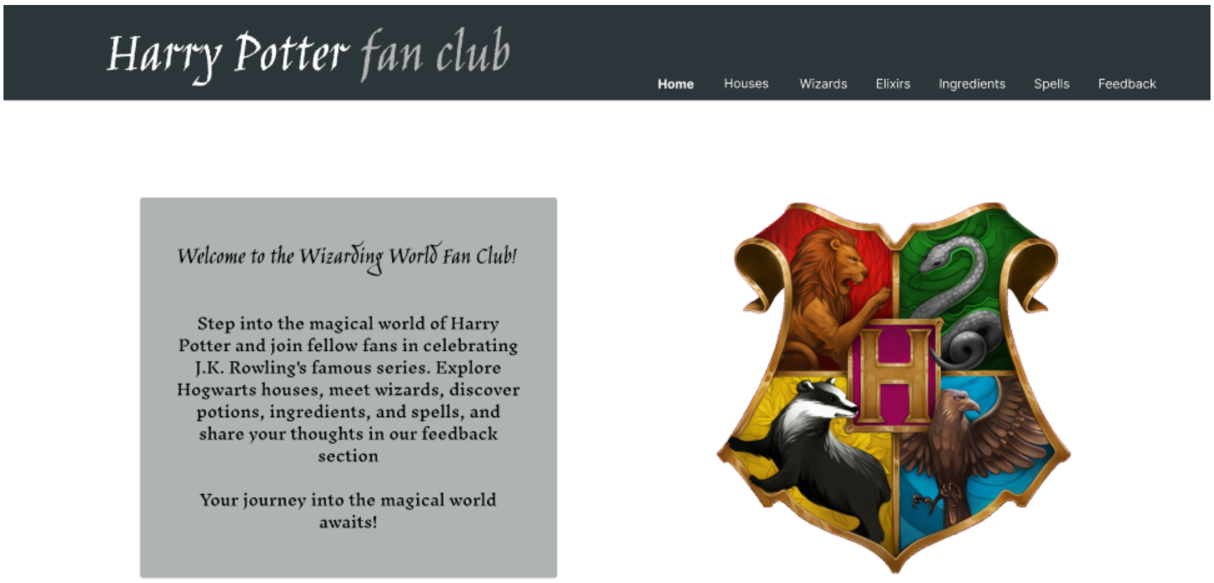
For this stage, I used Figma to simulate the initial idea of what I wanted my page to look like.

First of all, I defined my color palette:



It's based on the four colors of Hogwarts houses.

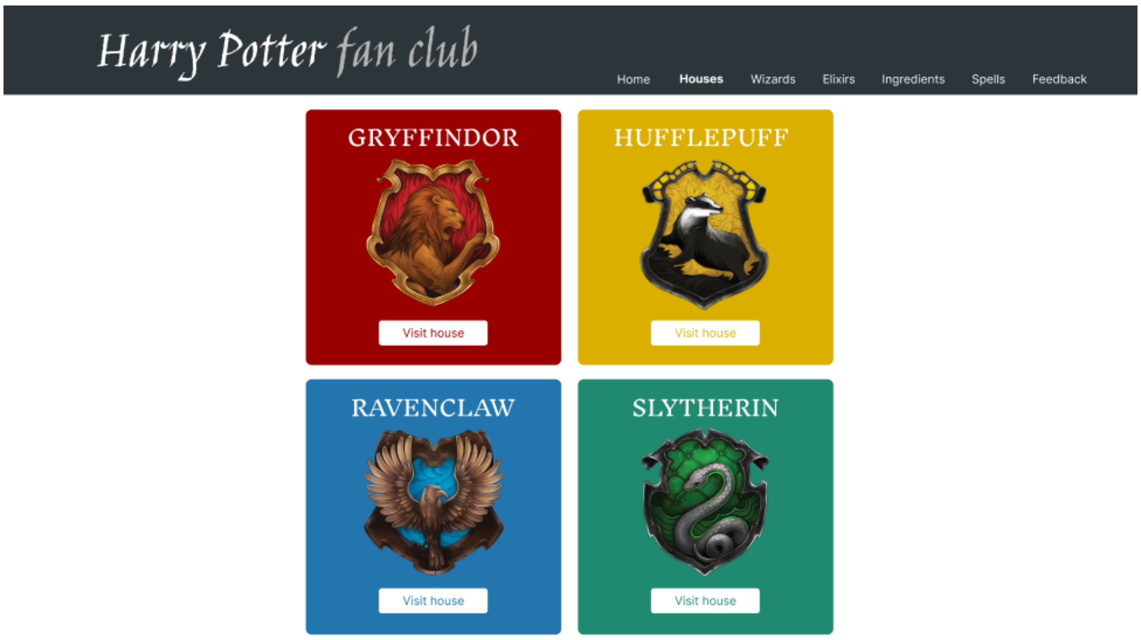
Then, I designed the frames that will contain my app:

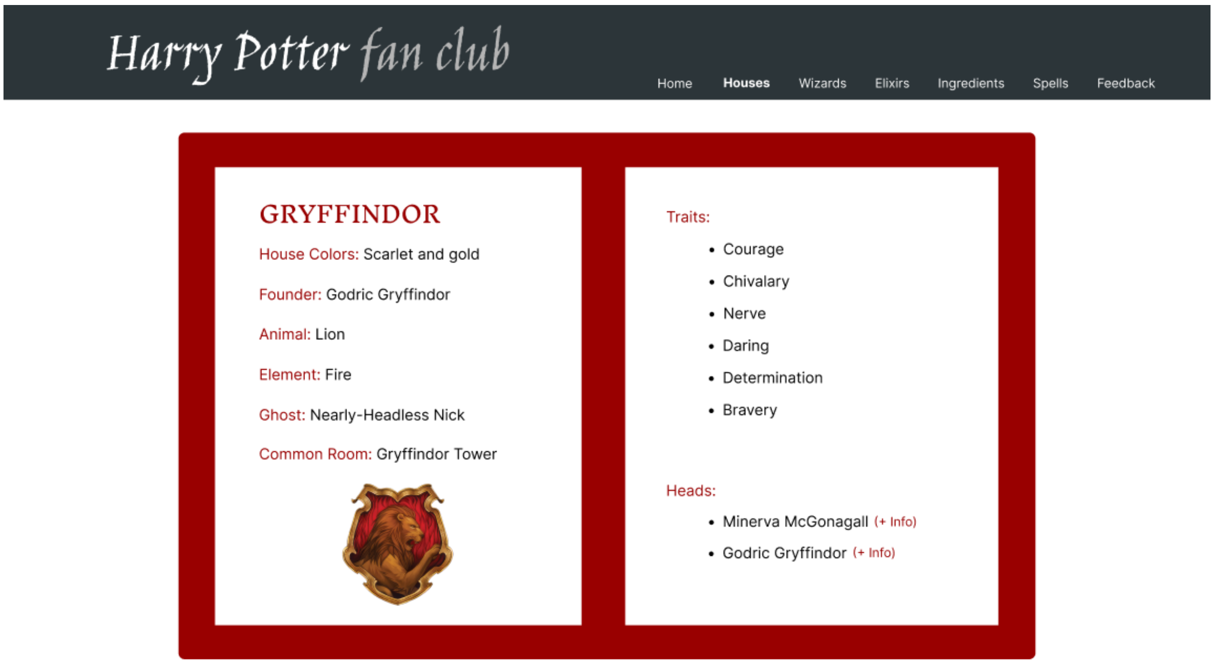


Welcome to the Wizarding World Fan Club!

Step into the magical world of Harry Potter and join fellow fans in celebrating J.K. Rowling's famous series. Explore Hogwarts houses, meet wizards, discover potions, ingredients, and spells, and share your thoughts in our feedback section

Your journey into the magical world awaits!





Harry Potter fan club

Home Houses Wizards Elixirs Ingredients Spells Feedback

GRYFFINDOR

House Colors: Scarlet and gold

Founder: Godric Gryffindor

Animal: Lion

Element: Fire

Ghost: Nearly-Headless Nick

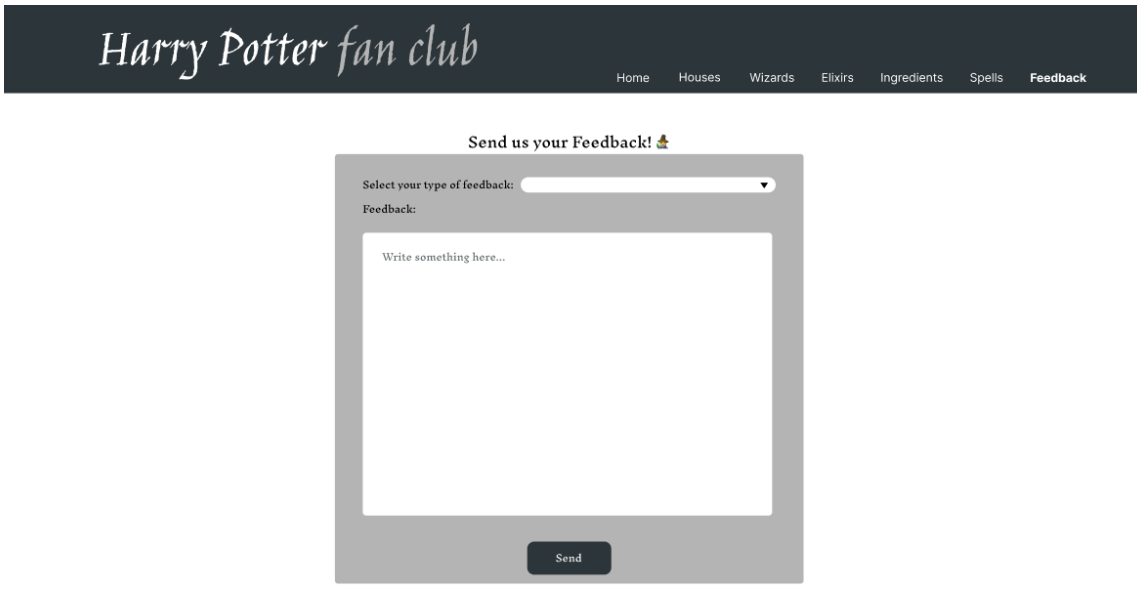
Common Room: Gryffindor Tower

Traits:

- Courage
- Chivalary
- Nerve
- Daring
- Determination
- Bravery

Heads:

- Minerva McGonagall (+ Info)
- Godric Gryffindor (+ Info)



Harry Potter fan club

Home Houses Wizards Elixirs Ingredients Spells Feedback

Send us your Feedback!

Select your type of feedback:

Feedback:

Write something here...

Send

Also, I choose this two fonts to work along with:

- Inknut Antiqua (houses names)
- Jim Nightshade (main title of the page)

The link to the project UI/UX design is:

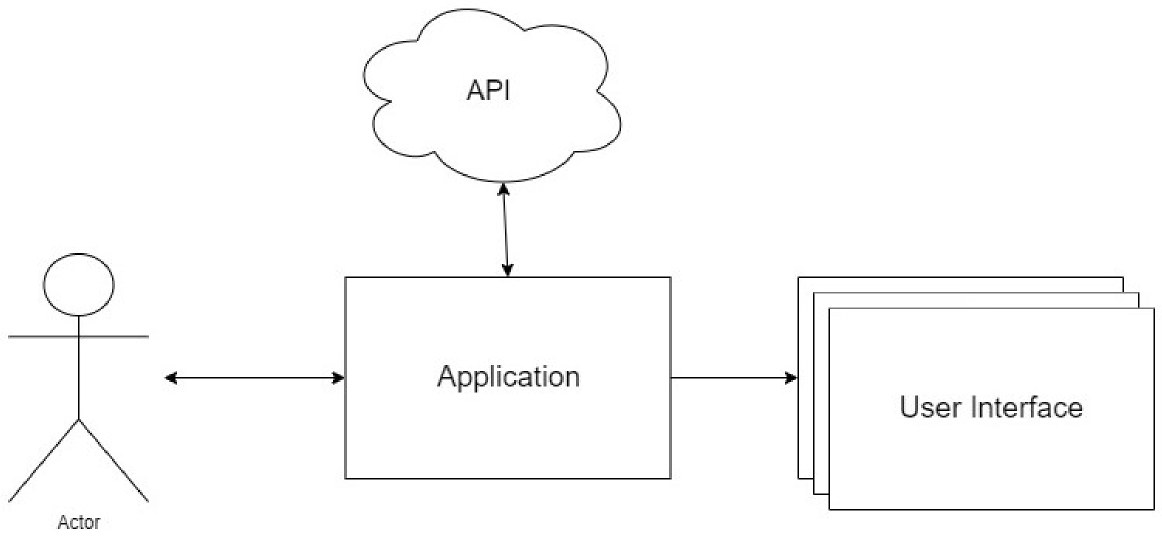
```
https://www.figma.com/design/XqfrIXyRF8qD1i1NKUn8rH/Harry-Potter-fan-club?node-id=0%3A1&t=SsUzEMhCIDEjYM9p-1
```

Tool Development

In the initial phase of designing the system architecture, I developed a series of diagrams to visually represent both the high-level and low-level aspects of the architecture.

High Level Diagram

The high-level diagrams provide an overview of the system's structure, focusing on the major components and their interactions:



Actor: Represents the user interacting with the application via a browser

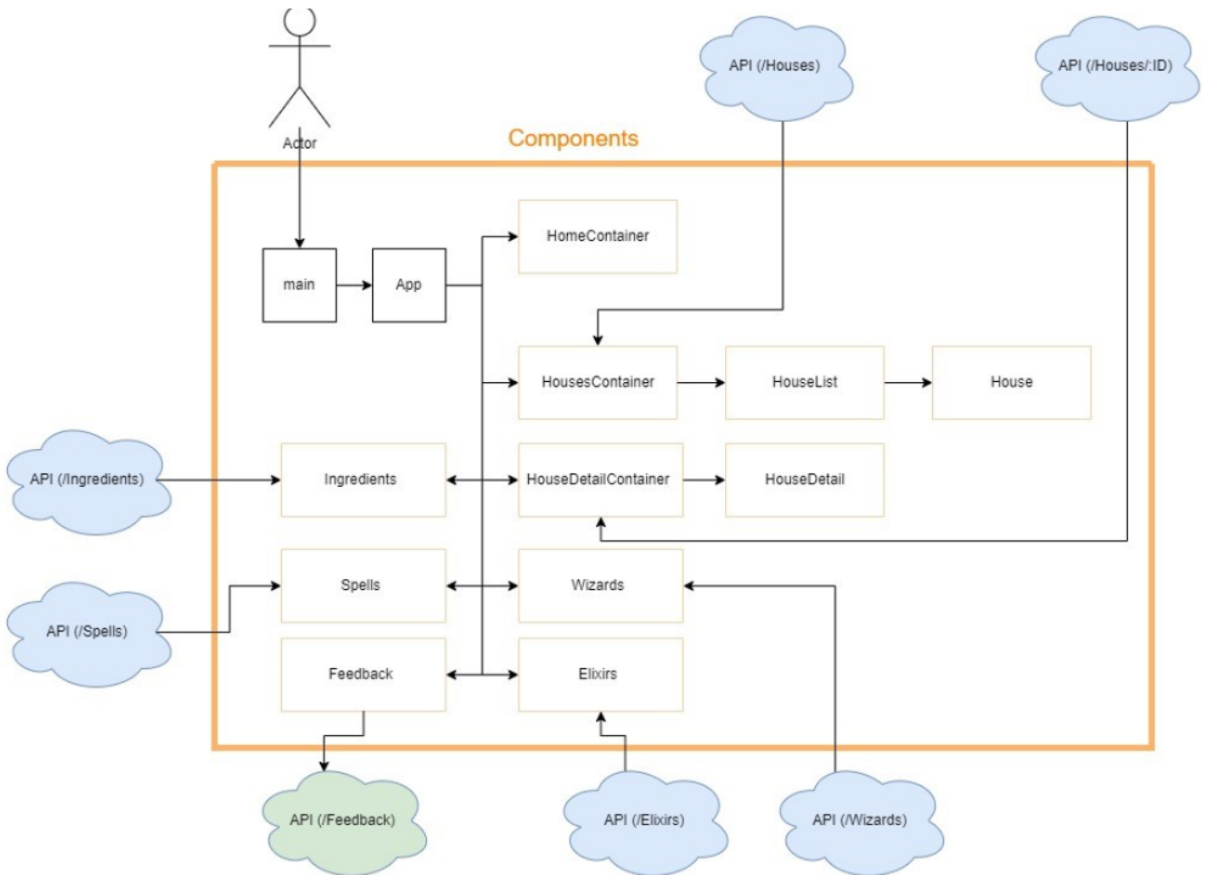
Application: Represents the main React application. It handles routing between different parts of the UI and manages state

API: Represents the external APIs that provide the data to the application

User Interface: Different parts of the application represented by React components. Each component might make specific API calls to fetch the data it needs.

Low Level Diagram

The low-level diagrams delve into the finer details of the system, showing specific parts, how they connect, and the data flow between them.



The blue clouds represent the API GET connections, while the green ones represent the POST interactions. Inside the orange box are all the system's React components necessary to display the UI.

Tool Implementation

Navbar

This component will always be shown. I started implementing this part first. I downloaded the font “Jim Nightshade” from https://fonts.google.com/specimen/Jim+Nightshade?query=jim for the page title.

Houses

This is the next step, in order to priorize the mandatory tasks. In this case I decided to bring only the names and IDs of each house from the API https://wizard-world-api.herokuapp.com/Houses

With this information I created a page were cards will be exposed with the name, shield and visit buttons of each house. Since colors and images weren’t specified, I decided to add them to a dictionary:

```
const default_color_house = "#B06E3F"
const default_img_house = "/img/default.png"
const dicc_color_house = {
    "Gryffindor" : "#990000",
    "Ravenclaw" : "#2376AD",
    "Slytherin" : "#1F8A70",
    "Hufflepuff" : "#DAAF00"
}
const dicc_img_house = {
    "Gryffindor" : "/img/gryffindor.png",
    "Ravenclaw" : "/img/ravenclaw.png",
    "Slytherin" : "/img/slytherin.png",
    "Hufflepuff" : "/img/hufflepuff.png"
}
```

In case the house isn't in the json or a new house is added in the future, the program won't break, default values were considered.

House Detail

When the "Visit house" button is clicked, the website will redirect you to the house details page. The information is retrieved from the API https://wizard-world-api.herokuapp.com/Houses/:id where the ID is obtained from the URL parameters (the button sets the url with the house ID).

From this page, you can access the Head Wizard information, which will be retrieved by the API https://wizard-world-api.herokuapp.com/Wizards/:id

Feedback

This page is a dedicated feedback space for users. Users select a feedback type and provide their comments in a textarea field. Upon form submission, the system utilizes the API at https://wizard-world-api.herokuapp.com/Feedback. The API returns a response indicating the success or failure of the POST action.

Amplitude Implementation

After reading Amplitude's documentation, I set up my own organization on their app and created a project called 'Harry Potter Fan Club'. Then, I began developing a tracking plan. Along with using some pre-made tracking options, I also created my own custom events:

- Click Visit Feedback

Description: Event tracking when a user clicks to visit the feedback section either from the homepage or the navbar

Properties: *from* (It takes de values "home" and "navbar")

- Click Visit Head Wizard

Description: Event tracking when a user clicks to visit the Head Wizard's profile from the details of a specific house

Properties: *houseWizard* ("Gryffindor", "Hufflepuff", "Slytherin", "Ravenclaw") and *wizard* (for example "Rowena Ravenclaw")

- Submit Feedback

Description: Event to track when users submit feedback through a form

Properties: *feedbackType* ("General", "Bug", "DataError", "Suggestion") ans *status* ("error","success")

- Visit House

Description: Event tracking when a user clicks the "Visit house" button of a specific house

Properties: *nameHouse* ("Gryffindor", "Hufflepuff", "Slytherin", "Ravenclaw")

- Visit NavLink

Description: This event is triggered when the user clicks any of the sections in the navbar

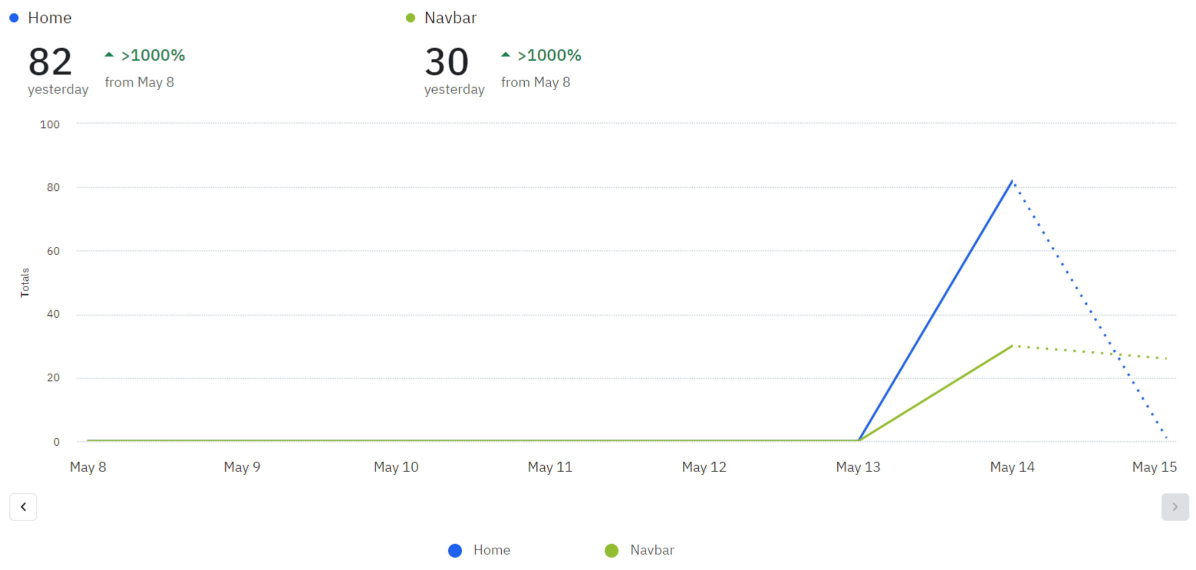
Properties: *nameLink* ("Home", "Houses", "Wizards", "Elixirs", "Ingredients", "Spells", "Feedback")

Amplitude charts

Once i've implemented the event triggers on my application, I started to gather data and made this charts on the Amplitude's app:

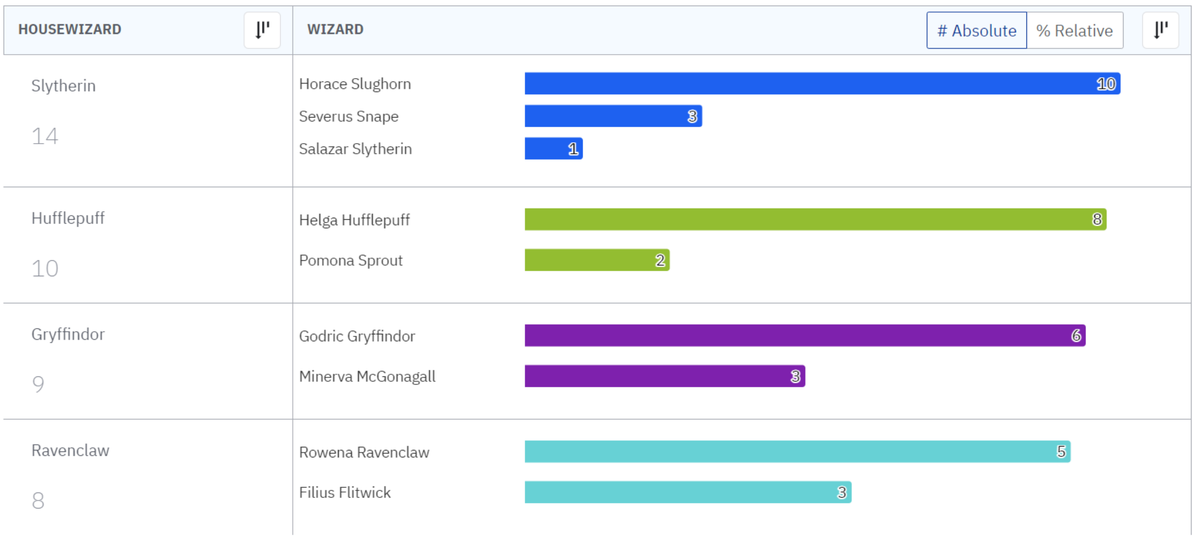
Click Visit Feedback by From - Total Event Count

It takes the data gathered by the "Click Visit Feedback" event and groups it by the event property *from*. It counts the number of events triggered per day.



Click Visit Head Wizard by houseWizard - Total Event Count

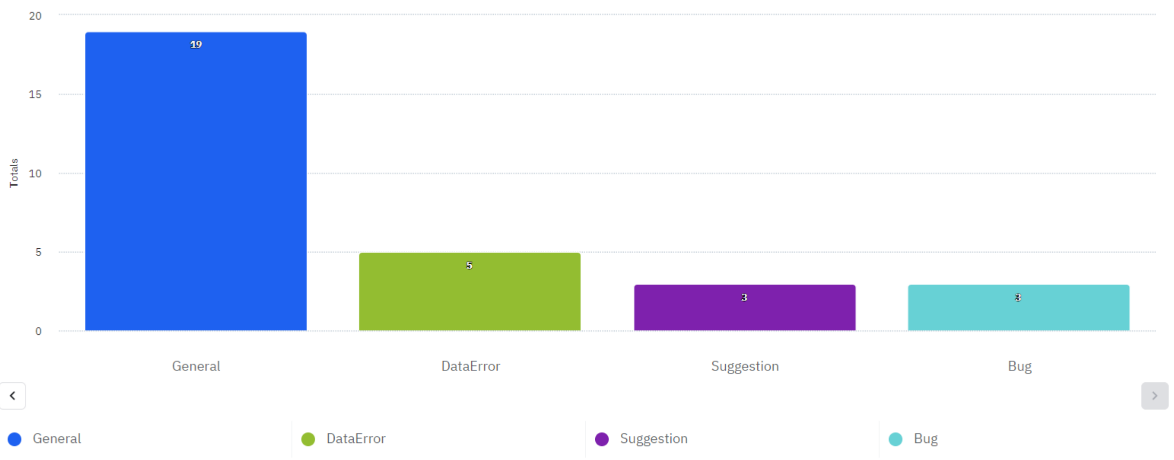
Using the "Click Visit Head Wizard" event, I created this chart grouped by *houseWizard* and *wizard* event properties. It measure the event totals and displays them on horizontal bars, showing information from the last 7 days.



HOUSEWIZARD		WIZARD	
Slytherin	14	Horace Slughorn	10
		Severus Snape	3
		Salazar Slytherin	1
Hufflepuff	10	Helga Hufflepuff	8
		Pomona Sprout	2
Gryffindor	9	Godric Gryffindor	6
		Minerva McGonagall	3
Ravenclaw	8	Rowena Ravenclaw	5
		Filius Flitwick	3

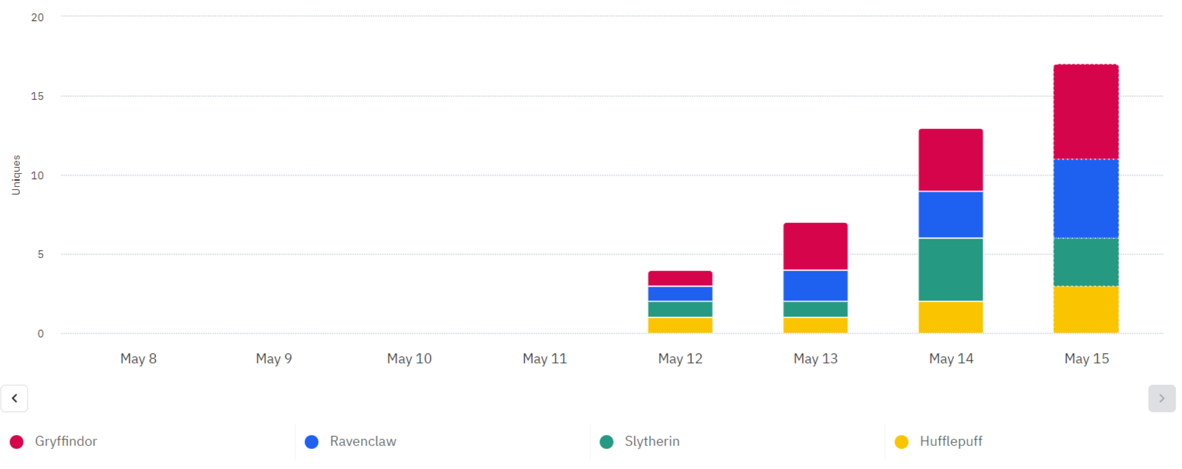
Submit Feedback by Feedback Type - Total Event Count

It shows the “Submit Feedback” event totals grouped by feedbackType for the last 7 days.



Most viewed House (Unique Users)

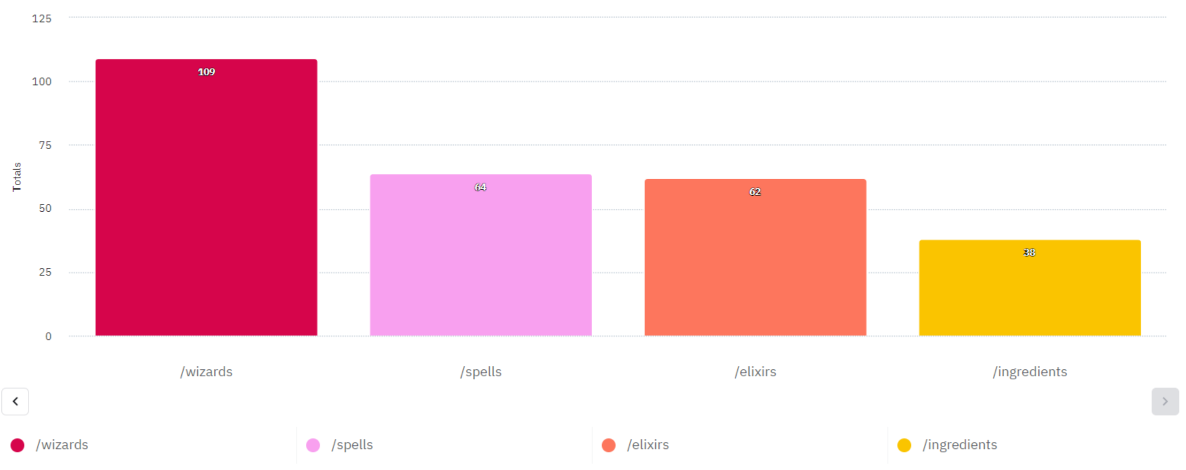
This chart was made using the “Visit House” event, grouped by *nameHouse* event property. It measures how many unique users triggered these events in the last 7 days.



It is also possible to create this exact chart using Amplitude's pre-made "Page-Viewed" event, filtered by the *Page Path* property. In this case, the paths selected to filter the chart will be "/houses/:(id_gryffindor)", "/houses/:(id_slytherin)", "/houses/:(id_ravenclaw)", "/houses/:(id_hufflepuff)" .

Coming soon pages

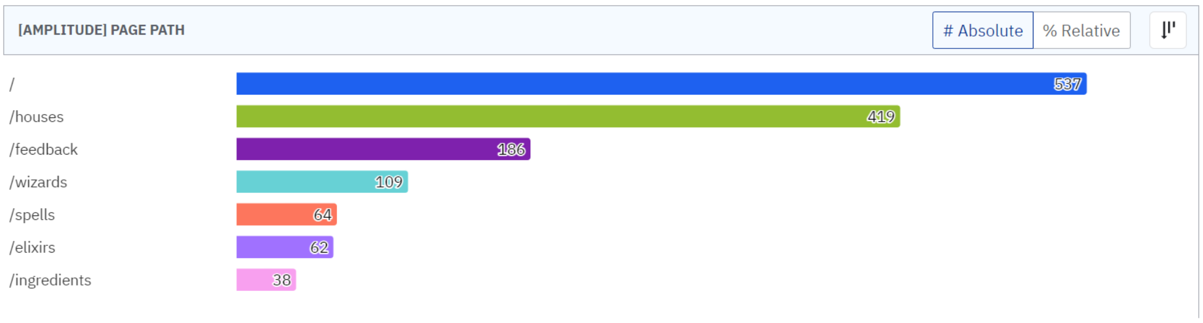
This chart was built using Amplitude's pre-made "Page-Viewed" event, filtered by the *Page Path* property for "/elixirs", "/wizards", "/ingredients", "spells". These paths represent those pages not yet developed, and the aim is to register the demand for the coming soon pages to potentially prioritize which one to implement next. The information represents the event totals for the last 7 days.



This information is also reachable applying the "Visit NavLink" event and filtering by *nameName* property on "Wizards", "Spells", "Elixirs" and "Ingredients".

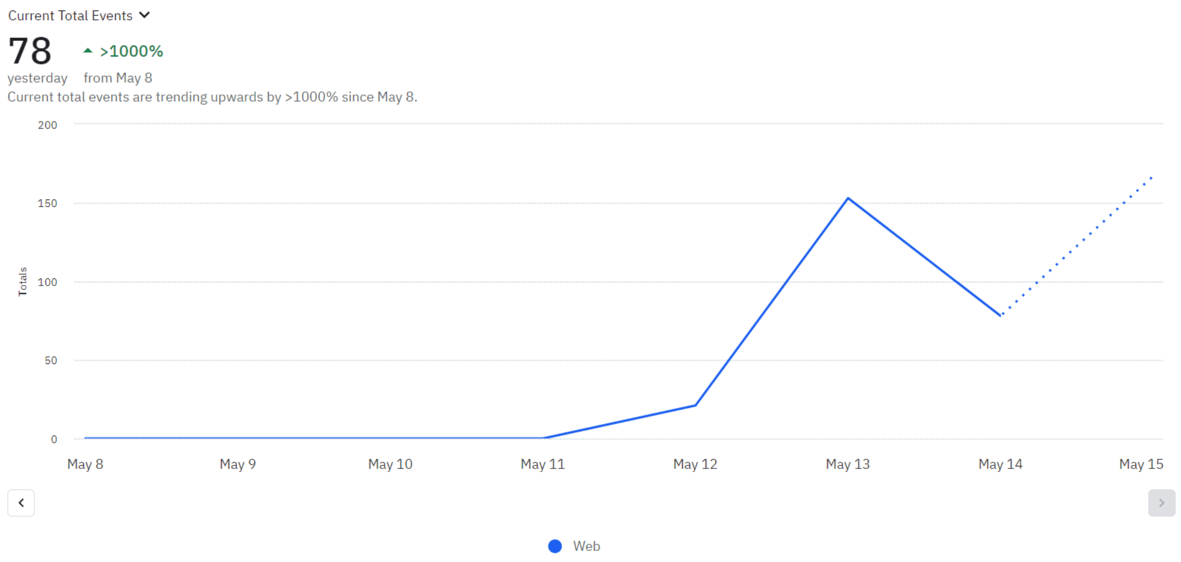
All Pages Views

In a similar manner, I could utilize the "Page-Viewed" event from Amplitude and obtain the total count of events tracked when a user visits a page, grouped by *Page Path.*



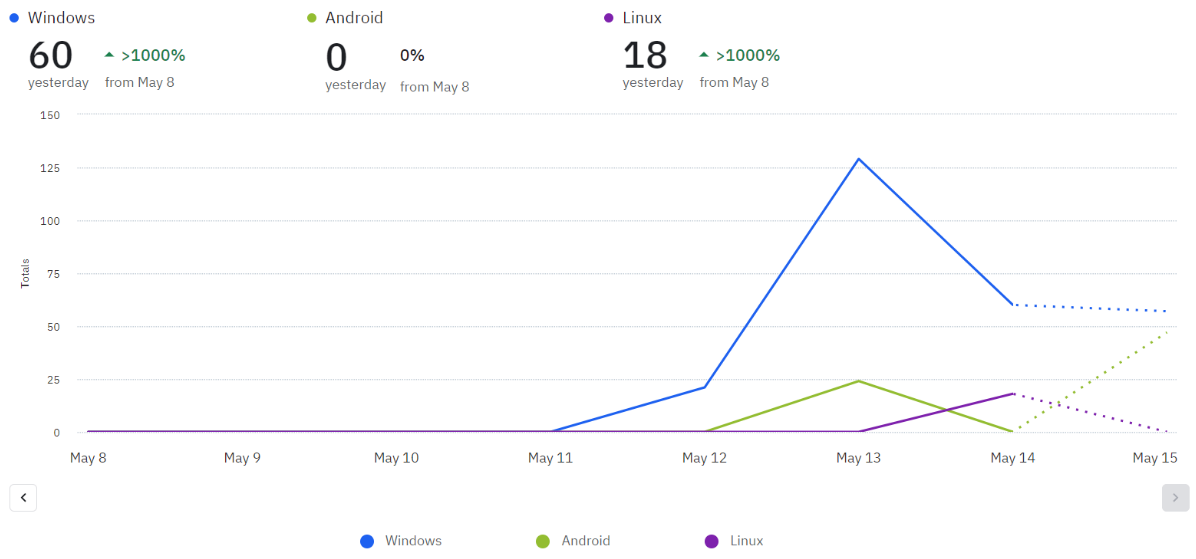
Page "All Houses" viewed by Platform

This chart represents the total events of visits to the "All Houses" page, grouped by *Platform* (Amplitude user property). In order to achieve this, I utilized the Amplitude "Page-Viewed" event and filtered it by *Page Path* when its value is "/houses"



Page "All Houses" viewed by Device type

In a similar manner to the previous chart, I designed this chart to track the total events of visits to the "All Houses" page, but in this case, grouped by *Device Type* (Amplitude user property). I utilized the Amplitude "Page-Viewed" event and filtered it by *Page Path* when its value is "/houses"



Application Repository

https://github.com/valenpontaut/Harry-Potter-fan-club

Hosting server

I chose to host my app on Netlify. I connected my GitHub repository to enable automatic deployment every time I make changes to my code.

This is the link:

HP fan club
https://hpfanclub.netlify.app/