Ahoy potential candidate! Congratulations on passing the first screening. Hopefully you enjoyed the interview as much as we did. Now that you moved on, we have created an exercise to gauge your Javascript, PHP, HTML, and CSS skills along with your ability to learn on the fly.

This assignment is divided into two parts: Section (A) and Section (B). You only need to complete the task(s) we have specified for you. In addition to these task there are bonus point tasks and requirements that you can feel free to attempt with no repercussions (they should however, not impede the specified functionality for your required task(s)). If you a complete a task we have not specified for you, those will also count as bonus points.

The Task:

Section (A):

The primary task of this section is to ***create*** the UI/UX functionality of filtering with the designs provided in **Template_A1, Template_A2, Template_A3**. There are 4 types of filters:

- Color
- Occasion
- Pattern
- Season

You can find all the information about our suits at the following API endpoint:

https://blacklapel.com/api/product/category/suits

The endpoint is domain restricted so you'll need to visit the url in your browser and figure out a way to pull and manipulate the resulting JSON object.

Filtering Rules

We want to be able to filter such that

`WITHIN Filter is *OR*`
`ACROSS Filters is *AND*`

So that I can have a filter combinations such as:

Filter By:
Color: ☑ Blue ☑ Brown ☐ Black
Occasion: ☐ Wedding ☑ Play ☑ Business Casual
Pattern: ☐ Solid ☐ Subtle Pattern ☐ Stripe
Season: ☑ Spring/Summer ☐ Fall/Winter

The filtering options you will need to be able to sort by are located in the JSON object at the api endpoint mentioned above

The Requirements

When visiting the index page, I should have the **option** to specify filters in the url params if I so choose, and the products displayed should reflect those choices.
i.e. `index.php?color=blue&pattern=solid`

The application should operate in this manner:

1) Initial parsing and filtering(if needed) should occur in a php class
2) Initially parsed and filtered data should be fetched with an AJAX call to a php resolver
3) Any filtering that occurs post page load should happen asynchronously on the entire category data set
4) Filtered data must be dynamically injected list-elements (li) into ul.product-list with the name of the product.

Any filters applied or parsing done post page load, **must** be done with JS. Products filtered data must be persistent (such that if I click another filter, it filters the previously filtered dataset. No redundant filtering after page load).

Section (B):

The primary task of this section is to ***create*** the UI functionality of customizing a product. For this section I should be able to click through a product displayed from the previous section(A) to a customization page (unless otherwise specified by us). This page should display the product name and begin the customization process.

You can find a product's customization options at the following endpoint:
https://blacklapel.com/api/product/option/{ product slug }

i.e. https://blacklapel.com/api/product/option/black-custom-suit

The endpoint is domain restricted so you'll need to visit the url in your browser and figure out a way to pull and manipulate the resulting JSON object. You will also need to pass in the product slug to pull the correct customizations for a product.

When going through the customization of a product you only need to display the customization option name and the group name for each option grouping. You can style this section however you best see fit.
You can ignore the Monogramming/Text groupings as well as the Vest Groupings.

The Requirements

The following requirements should all occur on a single page asynchronously.

Only **one** option grouping (i.e. Fit Type: Slim Fit, Tailored Fit, etc.) should be displayed at a time.

I should be able to step **forwards** and **backwards** through options as I am customizing.

Upon completing customizations, I should shown a **"Review Section**" where I should be able to view all the options selected and be able to go back into the process and change any individual customizations as I please.

On the "**Review Section**", I should also be able to see the price of the product, and if there is an additional price associated with an option, the price should reflect this.

The customization options should be stored in a "final payload" JSON with the following format for each "group":

```
{
        Fit Type: {     selected: "Tailored Fit",
                        Slug: "tailored-fit",
                        additionalPrice: "0",
                        group: "Fit Type"
        },
        Jacket Type: { selected: "Two-Button",
                        slug: "two-button",
                        additionalPrice: "0",
                         group: "Jacket Type"
        }
```

}
Console log out the final payload whenever the “**Review Section**” is reached.

Bonus Points

You earn more wizardry points should you complete any of the following (or all of them)! For each one you complete we will provide hints accordingly in our third round, in-depth technical interview.

Both Sections:

All Javascript is done Vanilla (no frameworks, libraries etc.).

Responsive design.

Section (A):

Asynchronously be able to switch in between and filter these additional categories:

- Shirts
- Blazers
- Vests
- Outerwear
- Pants

Section (B):

The ability to customize a product from any of the follow categories:

- Shirts
- Blazers
- Vests
- Outerwear
- Pants

Stylize the customization process to follow the following templates; **Template_B1, Template_B2,** and **Template_B3.** Don’t worry about the images, instead use a image placeholder from the following link :

https://placeholder.com/