PERSONAL WRITE UP

FINAL PROJECT: GiveMe5Recipe

Zexu Jin from Team 5
2nd December, 2019

OVERVIEW:

For the final project of SEAINFO6150, our team planned to develop a recipe website. The goal was to develop a website that's fully usable and accessible. In order to achieve that, we invested a lot of time in finding examples and talking with classmates for feedbacks. I was in charge of developing the subscribe page, about us page, and error page.

TIMELINE:

WEEK 1:

1. In discussion of the final project:
   a) Discuss the wireframes of different pages
   b) Layout of the pages and difference between different breakpoints.
2. Wireframes of the "Category pages (in three different breakpoints)"
   a) Wireframe of the category page of GiveMe5Recipe. Breakpoint: 500
   http://framebox.org/AJFqN
   b) Wireframe of the category page of GiveMe5Recipe. Breakpoint: 800
   http://framebox.org/AJFqU
   c) Wireframe of the category page of GiveMe5Recipe. Breakpoint: 1000
   http://framebox.org/AJFQU

3. Wireframes of the "Detail pages (in three different breakpoints)"
   a) Wireframe of the detail page of GiveMe5Recipe. Breakpoint: 500
   http://framebox.org/AJFrg
   b) Wireframe of the detail page of GiveMe5Recipe. Breakpoint: 800
   http://framebox.org/AJFrt
   c) Wireframe of the detail page of GiveMe5Recipe. Breakpoint: 1000
   http://framebox.org/AJFRm

WEEK 2 & 3:

1. Discussed project milestones and wrote the json file for desserts.
2. Worked on the html of the subscribe page and the about us page.
3. Worked on the css of subscribe page and about page.
4. Discussed user test with classmates from other teams and asked for feedbacks on their website experience.

WEEK 4:

1. Continued working on the css and html of the subscribe/about us/error page.
2. Worked on personal write up of the project.

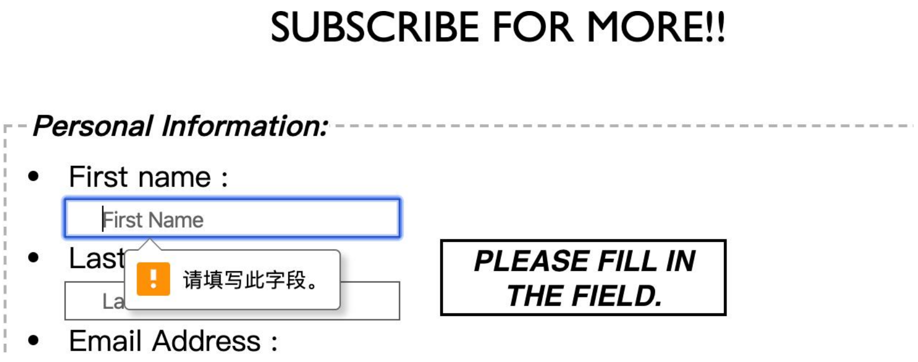
KEY DELIVERABLES

SUBSCRIBE PAGE

I used two different fieldsets to hold the subscribe information. The first one is personal information, where people should type in there name / email address / phone number (optional), as well as choose their gender.

The second one is other information including their favorite section on our website, as well as their preferred newsletter frequency.

I also add two types of form validations. The first type applied when press submit. All the inputs are supposed to be complete before submitting except marked as optional. If there's something left out, there will be a short tip near the left out field. The second type specifically applied to phone numbers and the email address. If the customer decides to fill in those two fields, the input must be in specific format, otherwise, there will be a red hint for instruction.



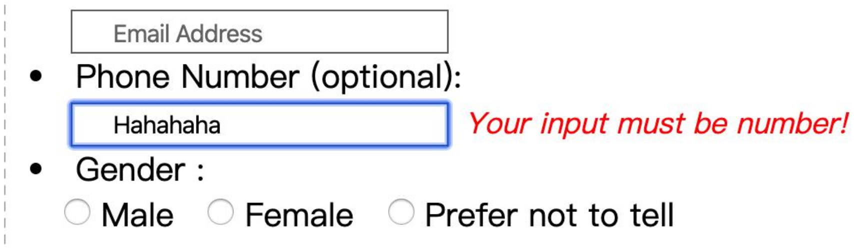
SUBSCRIBE FOR MORE!!

Personal Information:

- First name :
  First Name
- Last
  La
  ! 请填写此字段。
- Email Address :

PLEASE FILL IN THE FIELD.

F1 If there's something left out, there will be a short tip near the left out field.



Email Address

- Phone Number (optional):
  Hahahaha *Your input must be number!*
- Gender :
  ○ Male ○ Female ○ Prefer not to tell

F2 Instruction when input is not number.

ABOUT US

The about us page show some basic information about the website and our contact information. Our customers can find our customer service information at this page.

ERROR PAGE

The Not Found Error page is supposed to shown when our customers trying to reach pages could not be found on the server. I also add the back to home page link in the error page.

CHALLENGES

The challenges I face are basically all about the form and how to make it valid and elegant at the same time. I'm not familiar with form and has never dealt with forms before. I'll list some challenges I faced during the process.

CHALLENGES	HOW TO FIX
The overall layout of the form seems really messy, all the information packed without any organization, even when applied basic CSS styles.	I used the fieldsets to hold fields together. In this way, the entire page appears organized. And customers would lose focus when using it.
There are different ways for React and normal html to hold input information. At first, I didn't realize that and used normal html statements (yet I left out php file) to develop my page, which apparently wouldn't be valid for information collect when applied to back end.	I rewrote the entire statements. At first, I wasn't familiar with the React state, I didn't know how to write my handleChanges function properly. I managed to deliver it at last after reading many examples.
At first, I didn't know how to apply CSS styles to my form. I got stuck when trying to apply a better style to all the input components. The default style got them all stuck to one another. I also got confused when trying to add different style to radio, which was overwritten by the input style.	After reading many examples, I figured out the proper way to write the CSS.
I used bootstrap to write my entire page and I found it really convenient since you don't have to write any CSS. All the styles are prepared within the bootstrap plugin.	My teammates helped me realize the bootstrap might not be permitted in our project. So I rewrote the statements.
I didn't know how to apply proper form validations, especially for the email input.	I wrote the error statement within the handleChanges function and applied them in the render(). For the email, I used regex to restrict its format. For other input, I simply used Required to restrict the field to be completed before submitting.

CONCLUSION

- Working with classmates whom I've have no previous cooperation can be somehow challenging, but we managed to deliver it at the end. I'm very pleased what we've accomplished, and grateful for my teammates' help.
- Working on the form is nothing like I had in mind before. I made a lot of mistakes and some of them require an entire rewrite. The failures inspired me to do more beyond class.