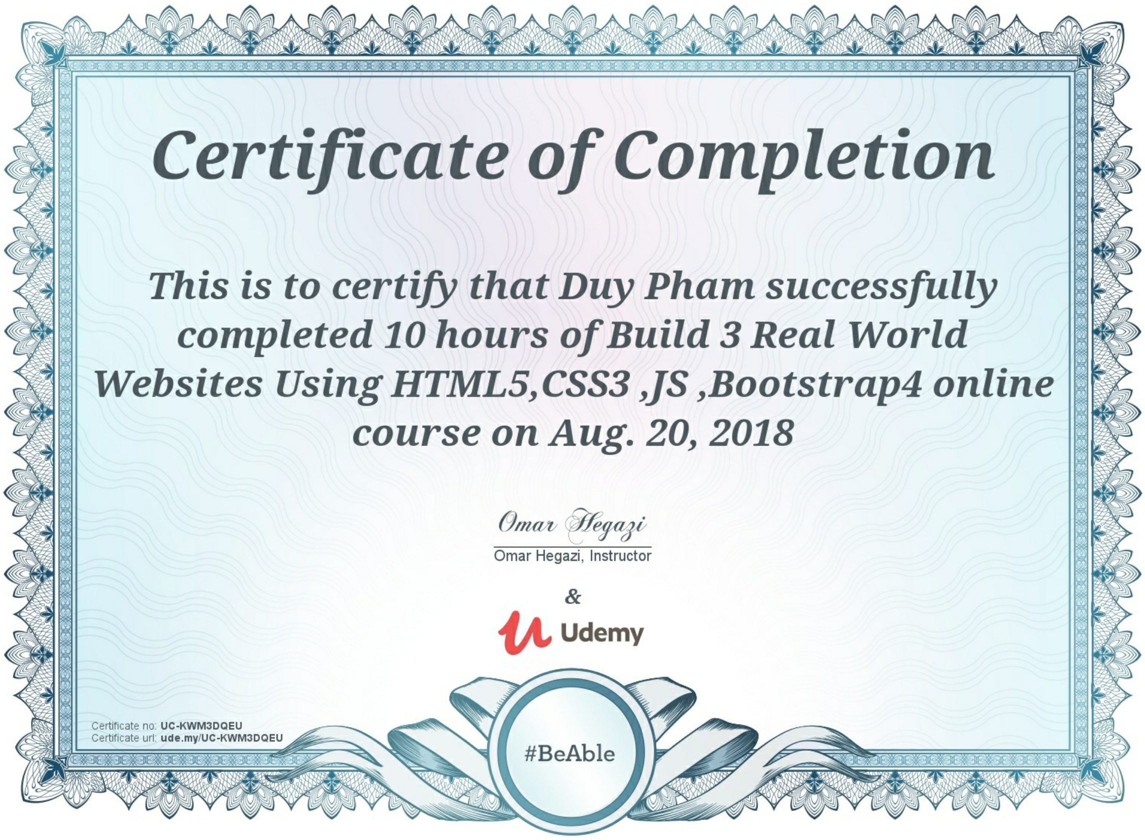
Table of Contents

Building 3 Responsive Real World Website Using HTML5, CSS3, JS, And Bootstrap 4

Planing

- Expected time: 59 hours
- Finish day: 20/08/2018
- Study Schedule

Certificate



Certificate of Completion

This is to certify that Duy Pham successfully completed 10 hours of Build 3 Real World Websites Using HTML5,CSS3 ,JS ,Bootstrap4 online course on Aug. 20, 2018

Omar Hegazi

Omar Hegazi, Instructor

&

Udemy

Certificate no: **UC-KWM3DQEU**
Certificate url: **ude.my/UC-KWM3DQEU**

#BeAble

Day 1: 12/08/2018

Today's progress

Finish lesson 1: Why this Course and Course plan

- Get the purpose of the course
- List out overall content for the course

Finish lesson 2: Editor

- Choose for yourself an IDE(VS Code as the course suggested)
- Discover some built-in features of VS Code
- Install some package for VS Code

Finish lesson 3: Adobe Photoshop Essentials

- How to convert a PSD to a full working website
- Explain photoshop tools

Thoughts

- Struggle with Git a little bit.
- The documentation is more time-consuming than expected.
- Spend some time on markdown hyperlink
- The markdown preview of VSCode show wrong indentation.
- Markdown show up in gitlab and atom differently which blocks
- The lesson is like a walk in the park not as hard as i expected

Links to work

Details of lesson 1

Details of lesson 2

Details of lesson 3

Day 2: 13/08/2018

Today's progress

Finish lesson 4: Bootstrap 4

- Discuss some of the neccessary and most used bootstrap components and ultilities

Thoughts

- Phew, Pretty long documentation
- The video is a bit hard to listen

Links to work

Details of lesson 4

Day 3: 14/08/2018

Today's progress

Finish lesson 5: Preprocessors SASS vs LESS

- Find out what is SASS and LESS, why use them and give some examples

Thoughts

- A little confusion comes from the naming of Sass and Scss.
- The `yarn global add node-sass` does not work, use `npm install node-sass -g` instead

Links to work

Details of lesson 5

Day 4: 15/08/2018

Today's progress

Finish lesson 6: Trello The Project Manager

- What is Trello and why it's helpful
- How to use

Finish lesson 7: OpenCharity Overview

- An overview on the website that we are going to clone

Finish lesson 8: Prepare files

- Build project folder structure
- Convert PSD file to some neccessary assets
- Integrate Bootstrap 4

Finish lesson 9: Linking Files + Implement Style Sheet

- Link CSS and JS files to index.html
- Implement CSS of our own with less

Thoughts

- The Trello lesson is easy.
- Finding fonts and download them for free is time-consuming
- The photoshop move tool does not work properly, so inpecting elements to export struggle a bit, attempting to fix it.
- Lesson 9 is plain coding with no explaination or visual comparision

Links to work

Link to GitHub Project: OpenCharity

Details of lesson 6

Details of lesson 7

Details of lesson 8

Details of lesson 9

Day 5: 16/08/2018

Today's progress

Finish lesson 10: Navbar Part 1

- Implement styles for Navigation Bar

Finish lesson 11: Navbar Part 2

- Implement event-handlers for Navigation bar

Finish lesson 12: Header Part 1

- Implement page header sections

Finish lesson 13: Header Part 2

- Finish page header sections

Finish lesson 14: Get Involved Section

- Implement UI for GET INVOLVED Section

Thoughts

- Just mirror coding without any further explaination

Links to work

Link to GitHub Project: OpenCharity

Details of lesson 10

Details of lesson 11

Details of lesson 12

Details of lesson 13

Details of lesson 14

Day 6: 17/08/2018

Today's progress

Finish lesson 15: Mission Section

- Implement UI for Mission Section

Finish lesson 16: Members Section

- Install Owl Carousel 2
- Implement UI for Member Section

Finish lesson 17: Event Section

- Implement UI for Event Section

Finish lesson 18: Blog Section

- Implement UI for Blog Section

Finish lesson 19: Contact Section and Footer

- Implement UI for Contactc Section
- Implement UI for Footer

Finish lesson 20: Enhancements

- Add animation, fix responsive bugs, make changes to UI

Finish lesson 21: Testing

- Cover some remaining issues about UI, animation,..
- Increase website performance
- Do some testing

Thoughts

- Just mirror coding

Links to work

Details of lesson 15

Details of lesson 16

Details of lesson 17

Details of lesson 18

Details of lesson 19

Details of lesson 20

Details of lesson 21

Day 7: 18/08/2018

Today's progress

Finish lesson 22: Recap

- Review what we have built

Finish lesson 23: 404-Overview

- Prepare for the 404-penguins-page

Finish lesson 24: Prepare Files

- Build project folder structure
- Extract neccessary images from psd file
- Import in neccessary css.

Finish lesson 25: Start Code

- Implement UI
- Styling Page
- Add animation

Finish lesson 26: Recap

- Advices on project

Finish lesson 27: Merkury Overview

- Overview on what are we going to build

Finish lesson 28: Prepare Files

- Apply Trello steps

Finish lesson 29: Linking Files

- Link image and library with index.html
- Apply some custom CSS

Thoughts

- Spend some time on Trello stuffs

Links to work

Details of lesson 22

Details of lesson 23

Details of lesson 24

Details of lesson 25

Details of lesson 26

Details of lesson 27

Details of lesson 28

Details of lesson 29

Day 8: 19/08/2018

Today's progress

Finish lesson 30: Navbar Part 1

- Implement styles for Navigation Bar

Finish lesson 31: Navbar Part 2

- Add some enhancements to make our navbar more interactive

Finish lesson 32: Header

- Implement UI for page header section

Finish lesson 33: Service Section

- Implement UI for service section

Finish lesson 34: Features Section

- Implement UI for Features section

Finish lesson 35: Drag and Drop Section Part 1

- Implement Skeleton for Drag and Drop section

Finish lesson 36: Drag and Drop Section Part 2

- Implement Style for Drag and Drop section

Finish lesson 37: Template Section

- Implement UI for Template Section

Finish lesson 38: Pricing Section

- Implement UI for Pricing Section
- Remind of Box shadow CSS

Finish lesson 39: Join and Footer Section

- Implement UI for Join Section
- Implement UI for Footer Section
- Remind of `:before` , `:after` and `display: flex`

Thoughts

- Problem with git pushing, SSH key added but still require manually Authentication, pushing takes too much time, fail sometimes

- The course does not go into details of CSS keyframe, jQuery and jQueryUI, just mention syntaxes and some functionalities. Research on them later on.

Links to work

Details of lesson 30

Details of lesson 31

Details of lesson 32

Details of lesson 33

Details of lesson 34

Details of lesson 35

Details of lesson 36

Details of lesson 37

Day 9: 20/08/2018

Today's progress

Finish lesson 38: Pricing Section

- Implement UI for Pricing Section
- Remind of Box shadow CSS

Finish lesson 39: Join and Footer Section

- Implement UI for Join Section
- Implement UI for Footer Section
- Remind of `:before` , `:after` and `display: flex`

Finish lesson 40: Enhancements

- Make website more responsive
- Apply some effect to increase smoothness

Finish lesson 41: Testing

- See what can we improve

Finish lesson 42: Recap

- Revise what we have learnt so far

Finish lesson 43: Important Tips

- Share some important tips and advices

Thoughts

- Problem with git pushing, SSH key added but still require manually Authentication, pushing takes too much time, fail sometimes
- The course does not go into details of CSS keyframe, jQuery and jQueryUI, just mention syntaxes and some functionalities. Research on them later on.
- More stuffs to research: `flexbox` , `box-shadow` , psuedo classes CSS
- The guideline on Scroll Reveal section is outdated
- The deployment and image compressing and speed test take freaking long time

Links to work

Details of lesson 38

Details of lesson 39

Details of lesson 40

Details of lesson 41

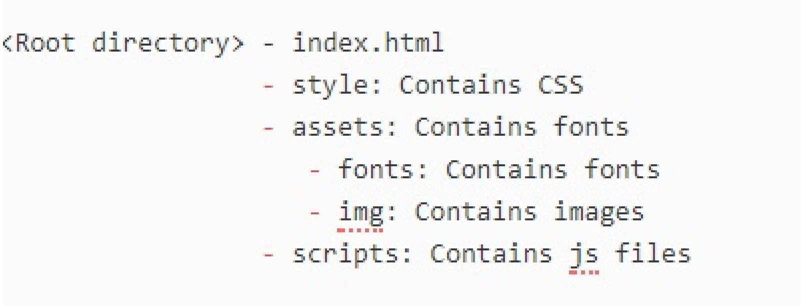
Details of lesson 42

Details of lesson 43

Day 10: 26/08/2018

Some tips on front-end developments:

Folder structure



```
<Root directory> - index.html
                 - style: Contains CSS
                 - assets: Contains fonts
                    - fonts: Contains fonts
                    - img: Contains images
                 - scripts: Contains js files
```

Step to convert PSD to HTML and CSS

- Build html files first.
- Inspect color, font size, spacing
- Convert font-size, color, spacing to CSS
- Add animation, effects

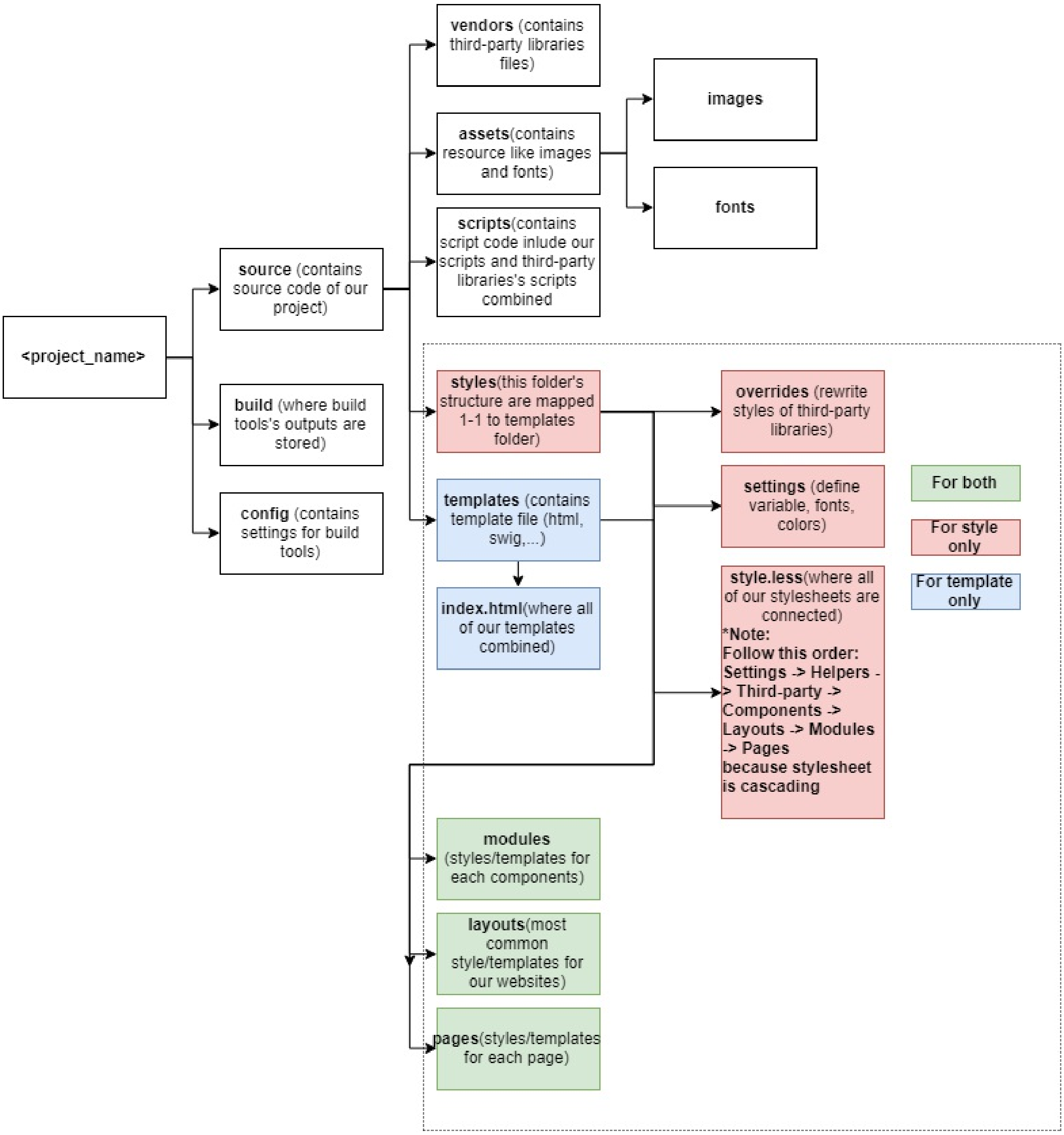
Thoughts

- Add CSS to ensure color and spacing need 1 to 1 precision.

Day 11: 29/08/2018

Config build tools and Steps to create skeleton HTML

New folder structure



vendors (contains third-party libraries files)
images
assets(contains resource like images and fonts)
fonts
scripts(contains script code inlude our scripts and third-party libraries's scripts combined
source (contains source code of our project)
<project_name>
styles(this folder's structure are mapped 1-1 to templates folder)
overrides (rewrite styles of third-party libraries)
build (where build tools's outputs are stored)
For both
settings (define variable, fonts, colors)
For style only
templates (contains template file (html, swig,...)
config (contains settings for build tools)
For template only
index.html(where all of our templates combined)
style.less(where all of our stylesheets are connected)
*Note:
Follow this order:
Settings -> Helpers -> Third-party -> Components -> Layouts -> Modules -> Pages
because stylesheet is cascading
modules (styles/templates for each components)
layouts(most common style/templates for our websites)
pages(styles/templates for each page)

Note:

The `index.js` in scripts folder should import style.less

The `style.less` should include banner, like:

/ Open Charity style /

Tips on setting up Build tools

Webpack

Should hash output filename after build to support auto clearing cache at client

Gulp

Steps to install

1. Create `gulp` folder and `gulpfile.js`
2. Install gulp `yarn add gulp --ignore-engines`
3. Install `require-dir`
4. Config `gulpfile.js`
5. gulp/util.js: for logging
6. gulp/concat.js: concat files
7. gulp/gulp-size.js: Alert file size after build

Note

Gulp's tasks naming:

For development: `scripts:vendor:compile`

For production: Add prefix `production`

Create Skeleton HTML

Prepare resource:

1. Open design file with Photoshop
2. Resize PSD file if it's too large using Image/Image Size...
3. Extract color, fonts and images and note down/save them

Connect resource to our project

With fonts

Always include all font type : oet, woff, ttf for cross-browser support

Suggest fontsquirrel to generate font kit from one file type

The font kits from internet usually include demo document, use their css to quickly define fonts

With color

Use chir.ag to name color and turn colors into `less` variables

Define color class for text and background

- Text: .text-<color_name>
- Background: .bg-<color_name>

With image

Rename them to describe where they would be used in website

Naming convention: Using _ to separate words, __ to represent child component

Ex: Image used for item 2 of section 2: `section_1__item_2.png`

Writting HTML

Name `class` and `id` right:

With class

Follow Semantics UI: `primary button` , `facebook item` , `facebook item-1`

With ID

Use _ to separate words and **to represent child components: `footer**social_links`

Note

- Class and ID should be used mixed.
- Classes are used for common styling, describing component
- IDs are used for quicker queries.
- Writing HTML should avoid over thinking

Day 12: 07/09/2018

Today's progress

Perfect pixel website without resizing design file

Front-end tips and tricks

Introduce some linting tools and optimize tools

Details

Perfect pixel website without resizing design file

Scale down screensize for fitting PSD file better

Front-end tips and tricks

How to use padding and margin properly:

- Padding should be used for nesting containers
- Margin should be used for adjusting spacing between elements
- Margin must not be used together with position
- Tag containing background should be put on top of its container

---

How to use `button` and `a` tag with class `button` properly:

- `a` with class `button` tags should be used for navigating
- `button` should be used for form and action only

Naming convention update:

- `.list` should contain `.item`
- `.row` should contain `column`

- Avoid abbreviations to keep template semantic.

Introduce some linting tools and optimize tools

Some linting tools to be applied in near future:

- ESlint
- CSSlint
- Husky + lint-staged

Linting configuration file: `.editorconfig`

Some linting tools for reference:

- prettier

Optimizing tools to be applied:

- Minify CSS.
- Minify JS
- Minify HTML (experimenting)

Some Optimizing tools for reference:

- prepack.io (developing)

Day 13: 12/09/2018

Today's progress

Naming convention update

Front-end tips and tricks

Some strays on the modern web design world

Introduce some web optimization techniques

Introduce regression test

Details

Naming convention update

Vocabularies:

Classes:

`.title` (For section, for card),

`.content` (For section),

`.description` (Esp for card),

`.meta` (Additional information, use with descriptive classes like `time` , `date` , `price` , `discount` , wrapped with `<span>`),

`.action` (For button),

`.thumbnail`

`.row > .column` ,

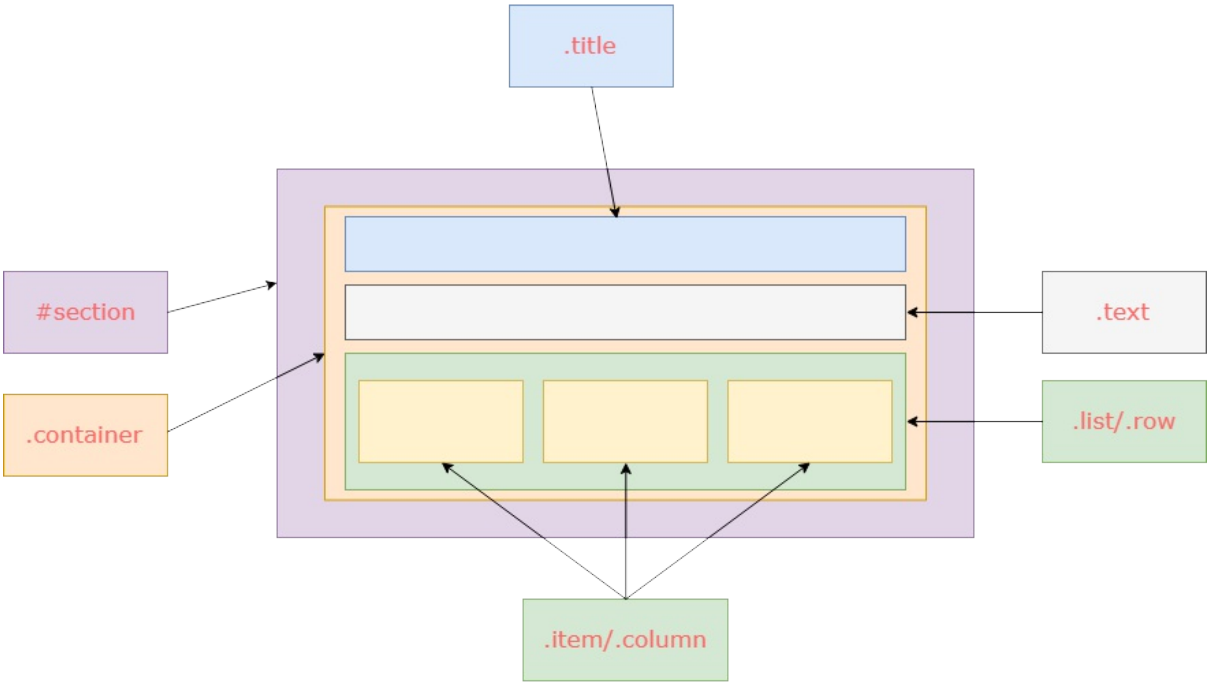
`.list > li.item`

Front-end tips and tricks

New rules:

1. Restrict on using `.container-fluid` , using `.container` and override `@container-md` , `@container-sm` , `@container-sm`
2. Using css `class` for defining fonts
3. Color and background color class pattern: `color-[color-name]` , `background-[color-name]`
4. Bootstap `.row` s should be only applied to `.row > .column` s
5. Font's filename: `"Font-Name.[ext]"`
6. font-family define: `'Font Name'`
7. Using CSS over JS creating animations

Template structure:



Some strays on the modern web design world

Avoid FOUT, FOIT, FOFT

Readmore

Some popular Design Systems

1. Bootstrap
2. Zurb Foundation

3. Semantic UI

States of digital websites

1. Full (Interacting)
2. Loading (Interacting)
3. Error
4. Error
5. Empty
6. Edge cases

Introduce some web optimization techniques

With Javascript

1. Minify JS
2. Obfuscate JS (encode to avoiding reversing)

With CSS

1. Minify CSS
2. Critical CSS (determine some important CSSes to be applied before all CSSes are loaded to create `First meaningfull paint`)
3. Remove unused CSS using

With HTML

1. Minify

With Images and fonts0

1. Compress images and fonts

Introduce regression test

Apply regression test and its importance in frontend development

Chapter 1: Introduction + Course Essentials

Why this Course and Course plan

Overview:

Understand the purpose of taking this course and give out the course plan

Details:

- Top 5 reasons:

    1. Technology are introduced along with practical projects
    2. Step by step lead you from exporting elements of Photoshop Design till you Deliver a full working responsive production website to client (the combination of design and coding is very usual in big companies)
    3. Concern testing and optimization.
    4. Provide three different stunning website in one course with different ideas and carefully chosen design.
    5. Simplify the knowledges of technologies used and combine them.
- The testing includes:

- Cross browser test
- Markup validation service
- Speed and performance test

- Enhancements includes:

- Responsive test
- Compressing images

- Course planning:

  1.Adobe Photoshop:

    - How to use to export elements to psd files
    - How to get the colors and fonts of your website
    - Tools that front-end developers/designer should know
    - How to control layers and useful shorcuts

  2.Bootstrap 4:

    - Introduce Bootstrap 4 grid system by examples
    - How to use components like navbar, cards, forms, dropdown, scroll spy and more
    - Further explanation of the most common bootstrap 4 classes

  3.CSS Preprocessors SASS and LESS

    - Full explanation about preprocessors by examples
    - Definition and why use them ?
    - Different between SASS and LESS
    - How to install
    - Features of SASS and LESS by examples

  4.Trello Project Manager

    - What is Trello

- Why we use it
- How we can use it to organize our project steps and keep the clients aware of the project progress

5.The Open Charity website

- How to prepare website files
- Write clean and reusable code with comments for every section
- Use LESS in our style sheet
- Use some of the bootstrap components
- How to make scroll spy as well as smooth scroll
- Write some jQuery animations
- User a plugin called owl slider to make two different sliders and how to customize your slider easily
- Use animated.css to animate some elements
- Check the responsive plus optimize the website speed and performance and run three different tests on website

6.The 404-penguins page

- Learn how to make advanced css keyframes and animations plus positioning

7.Merkury Website

- Go over some of subjects that we have learned in the past projects
- Learn how to make advanced animations with css keyframes and jQuery
- Use bootstrap 4 ultility classes
- Use CSS before and after selectors
- How to combine between bootstrap 4 cards dropdown and customize them
- Learn how to use jQuery UI library to apply advanced events on elements
- Learn how to use scroll reveal plugin to animate elements while scrolling
- Finding solutions and workarounds and more
- Check responsive and optimize website performance and speed along with running test on website

Bookmarks: None

Editor

Overview:

Introduce VS Code Editor

Details:

- Customize setting: File/Preference/Settings a tab appear to let you put on your own style
- VS Code comes with built-in terminal, (expand from the bottom)
- Install the following package for compling preprocessors SASS and LESS
  1. Ctrl + Shift + X to open Extensions
  2. Search and install Easy LESS(compile LESS to CSS on save), Live SASS compiler(compile SASS or SCSS to CSS, come with live server to support hot reloading), Sass (Sass syntax highlighting, autocomplete and snipets)
- Install Dracula Theme through Extensions.

Bookmarks: Download and Documentation VSCode

Adobe Photoshop Essentials

Overview:

Adobe Photoshop Essentials

Details:

- Image Exporting program that help to convert PSD to HTML and CSS: Figma, Gravit, Photoshop, ...
- Explain some common used tools of Photoshop with examples
- The left sidebar
    - **Select tool/Move tool(V)**: Select elements and export it
    - **Marquee tool(M)**: Makes rectangular, elliptical, single row and single column selections
    - **Crop tool(C)**: Trim, straighten, and change the perspective of images
    - **Eyedropper tool(I)**: Inspect color
    - **Horizontal Type tool(T)**: Create text, edit or delete it, get font-size, font-family
    - **Hand tool(H)**: Move image
    - **Zoom tool(H)**: Zoom in, Zoom out
- Some useful shorcuts:
    - **Zoom(Alt + Scroll)**: Zoom in and out
    - **Step Back(Ctrl + Alt + Z)**: Traverse history
    - More details in **Bookmarks**
- Change unit to pixels: Edit/Preference/General or Ctrl + K then choose Units and Rulers. Change the Rulers.
- Extract some components from template:
    - Single component
        1. Create new file: Ctrl N
        2. Use Move Tool to drag and drop the component onto that files
        3. Make background transparent
        4. User crop tool to extract the component then export to PNG
    - Multiple component
        1. Hold Ctrl select multiple component
        2. Click `Create new group` under Layers panel
        3. Export as PNG
- The usage of Zoom tool, Text tool, and Hand tool are pretty straight-forward
- History panel: Open history panel by click on Window/History.

Bookmarks: 60 useful shorcuts in Photoshop

Explain photoshop tools

Bootstrap 4

Overview:

Discuss some of the neccessary and most used bootstrap components and ultilities

Details:

About library itself

Bootstrap is an open source toolkit for developing with HTML, CSS and Javascript

Bootstrap includes HTML and CSS based design templates for typographic, form buttons, table, navigation, model, image carousels and many others as well as optional javascript plugins

Bootstrap is used to build responsive mobile first projects on the web with world's most popular front-end component library

Why use Bootstrap ?

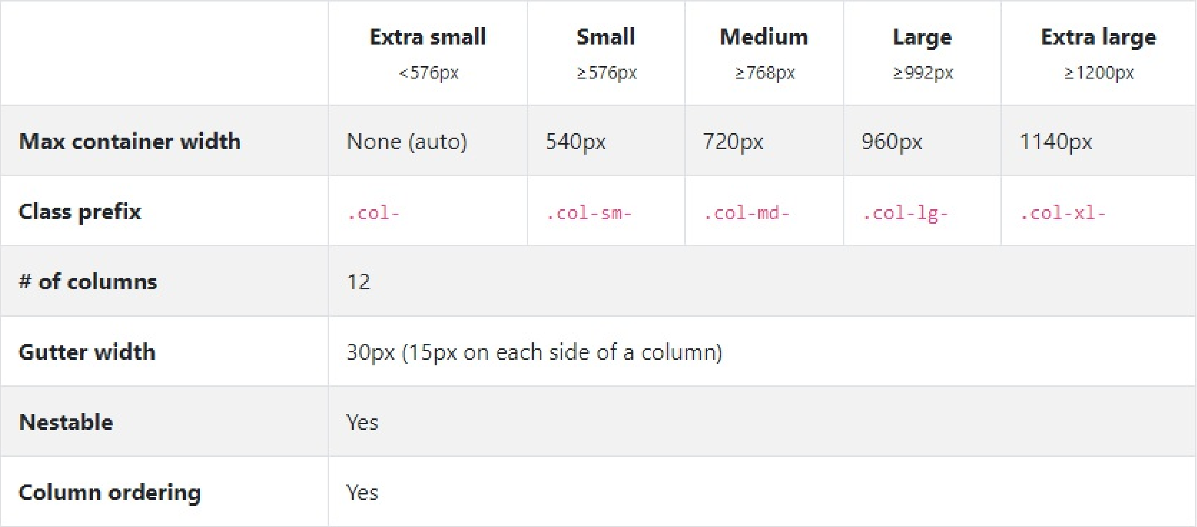
1. Easy to Use
2. Responsiveness
3. The Speed of the Development
4. Customizable (open-source)
5. Consistency
6. Big community, great support
7. Simple Integration
8. Grid system

Explain more in the **Bookmarks**

Most important to learn about bootstrap

Grid System

- Grid System use a series of container **row** and **column** to layout and align content, it's built with **flexbox** and fully responsive
- Flexbox was designed as one dimensional layout model and as a method that could offer space distribution between items in an interface
- Bootstrap grid system consists of `container` , `container fluid` (span the entire width of viewport), `rows` and `column`



	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Example here

Components

- Button
- Classes: `btn btn-primary` `btn btn-secondary` `btn btn-success` `btn btn-danger` `btn btn-warning` `btn btn-info` `btn btn-light` `btn btn-dark`
- Visual Example
- Card
- Classes: `card` > `card-img-top` `card-body` `card-title` `card-text`
- Visual Example
- Dropdown
- Classes: `dropdown` `dropdown-toggle` `dropdown-menu` `dropdown-item`
- Visual Example

Important components: cards, drop down, forms, nav bar, scrollbar

- Form

  1.Custom Form

  - Classes: `form-group` `form-control`
  - Visual Example

  2.Inline Form

  - Classes: `form-inline` `form-group` `form-control`
  - Visual Example **Bookmarks:**

  3.**Note**: `<input>` and `<label>` should be put inside form-group

  4.**Note**: You will notice some mysterious classes like `mb-2` `mx-sm-3` , they are spacing classes(ultilities class). Check it out in Bookmarks

  5.**Note**: Form in Bootstrap has many variants, see Bookmarks for more

- Navbar

- Some features:
  - Navbars and their contents are fluid by default. Use optional containers to limit their horizontal width.
  - Use our spacing and flex utility classes for controlling spacing and alignment within navbars.
  - Navbars require a wrapping .navbar with .navbar-expand{-sm|-md|-lg|-xl} for responsive collapsing and color scheme classes.
- Supported content: `.navbar-brand` `.navbar-nav` `.navbar-toggler` `.form-inline` `.navbar-text` `.collapse.navbar-collapse` (Detail in Bookmarks)
- Visual Example
- Scrollspy
- Features: Automatically update Bootstrap navigation or list group components based on scroll position to indicate which link is currently active in the viewport.
- Visual Example

- Ultilities (Link to details of Bootstrap Ultilities in Bookmarks)

  1.Colors

  - Classes: `*-primary` `*-secondary` `*-success` `*-danger` `*-warning` `*-info` `*-light` `*-dark` `*-white`
  - **Note** Replace * with `text` or `bg`
  - Visual Example

  2.Float

  - Features: Works like its name suggest
  - Classes: `float-left` `float-right` `float-none`

  3.Position

  - Features: An alternative to `position` CSS property
  - Classes: `position-static/relative/absolute/fixed/sticky`

  4.Shadow

  - Features: An alternative to `box-shadow` CSS property
  - Classes: `shadow` `shadow-none/sm/lg`

  5.Spacing

  - If you got here, you may have checked out the Spacing reference link in the Bookmarks section, so i won't recall.

6.Text

- Features: Documentation and examples for common text utilities to control alignment, wrapping, weight, and more.
- Classes:
  - Alignment with `text-left` `text-right` `text-center` `text-justify`
  - Text wrapping and overflow with `text-nowrap` `text-truncate`
  - Text transform with `text-lowercase` `text-uppercase` `text-capitalize`
  - Font weight and italics with `font-weight-bold` `font-weight-normal` `font-weight-light` `font-italic`

7.Vertical Alignment

- Features: Easily change the vertical alignment of inline, inline-block, inline-table, and table cell elements.
- Classes: `align-baseline` `align-top` `align-middle` `align-bottom` `align-text-bottom` `align-text-top`
- **Note** These classes are only applied for `inline` and `inline-block` elements

Bookmarks:

Bootstrap Quiz

Why use Bootstrap for amazing web design

CSS selector with regex

Spacing classes Bootstrap

Bootstrap Forms

Navbar Supported content

Bootstrap Ultilities

Preprocessors SASS vs LESS

Overview:

Find out what is SASS and LESS, why use them and gave some examples

Details:

What is Preprocessors ?

- It's basically a scripting language that extends CSS and compiled into regular CSS.

Why Go for CSS Preprocessors

- CSS is primitive and incomplete. Building a function, reusing a definition or inheritance are hard to achieve. For bigger projects, or complex systems, maintenance is a very big problem.
- CSS preprocessor is DRY(Don't Repeat Yourself), Not WET(Write It Twice)
- Aids in Modularization and Portability
- Enjoy the Power of Code Reuse
- More Organized
- Easy to Write
- Easy to Setup and Maintain

Difference between SASS and LESS

SASS	LESS
Built on Ruby	Built on Javascript
	Better Error messenger
Documentation is much more of a knowledge-based wiki sheet	More User-friendly documentation

How to install

With SASS: yarn global add sass

With LESS: yarn global add less

- Install compiler
- VSCode: Install Easy Less, Live Sass Compiler
- Atom: Install `yarn global add node-sass` , `apm install less-autocompile sass-autocompile`

You may be confused

Sass vs **SCSS**

Sass has 2 syntaxes available:

- Sass itself (Syntactically Awesome StyleSheets) in .sass files
- SCSS (Sassy CSS) in .scss files, which is something halfway between regular CSS and Sass

Remember that:

- Sass is the name of the preprocessor
- SCSS is easier to learn
- All resources on the internet mention Sass, not SCSS
- All features are available for both syntaxes everything in SCSS is available in Sass
- We're actually going to write SCSS but still call it Sass.

Find out more in the Bookmarks

Features

Link to SASS examples: SASS Example

Link to LESS examples: LESS Example

Variables

- Variables were all time wanted feature for CSS. Every developer, wanted to define a base color and use it all over the CSS file, in stead of writing the hex or named color in a property each time
- Variables in LESS start with @ and those in SASS start with $, value of variable is assigned with semicolon :-

Mixins

- The idea is you put together some set of css rules to create effect of your own, much like a reuseable css class, it resembles function in other languages

Nesting

- CSS lacks visual hierarchy while working with child selectors.
- Nesting provides a visual hierarchy as in the HTML and increases the readability.
- **Note**: In some cases, nesting causes oversizing the selectors, so use it wisely.

Extends

- Extends are useful for sharing a generic definition with selectors rather than copying it in
- SASS extends every instance of extended selector that includes its child selectors and inherited properties
- With LESS you can choose to extend only the main instance or all by `extend(.<className> all)`

Imports

- Allow you to split one large file into small pieces

There are some other features like Color Operation, If/else Statements, Loops, Math but not covered in the course.

Checkout the Bookmarks for all of them

Bookmarks:

Sass and Less Quiz

Document from course

Sass, SCSS and Less

More Feature of Sass and Less

Trello The Project Manager

Overview:

- What is Trello and why it's helpful
- How to use

Details:

What is Trello

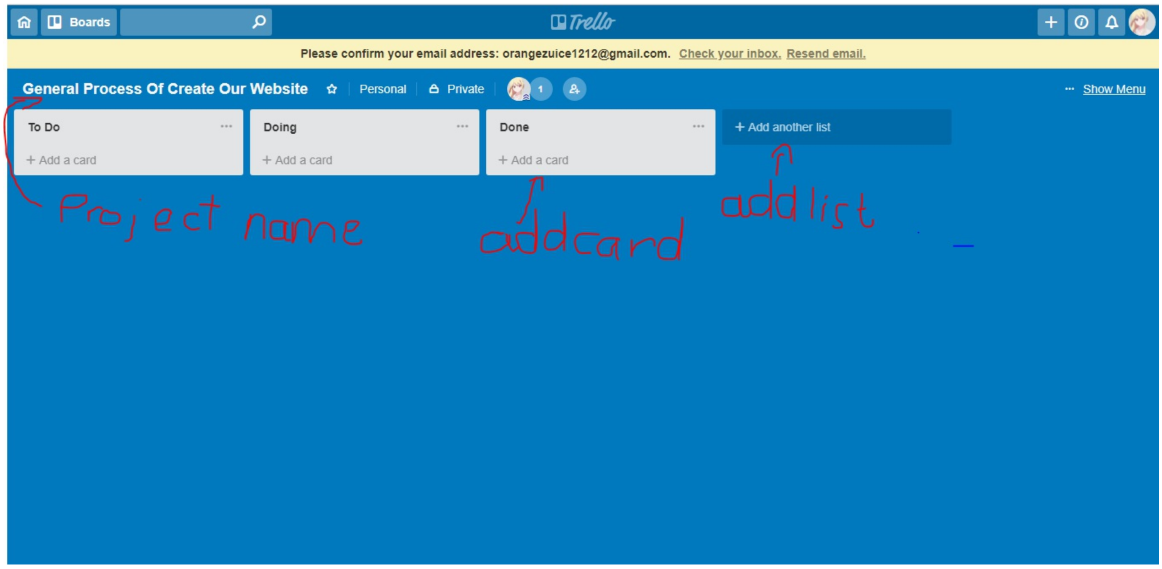
Trello is a Project Management Software that use cards to organize tasks and visualize purpose.

Why use Trello ?

- Trello can be very helpful if you have more than one project at the same time and you want to organize the time between projects
- More than one person can work at the board and you can assign task to members
- Keep clients aware of your project progress and the due date
- The details of Trello will be explained on real project

How to use

1. Go to Trello
2. Sign up a free account and login
3. Rename the board (to the name of your project may be)
4. Add some list of task

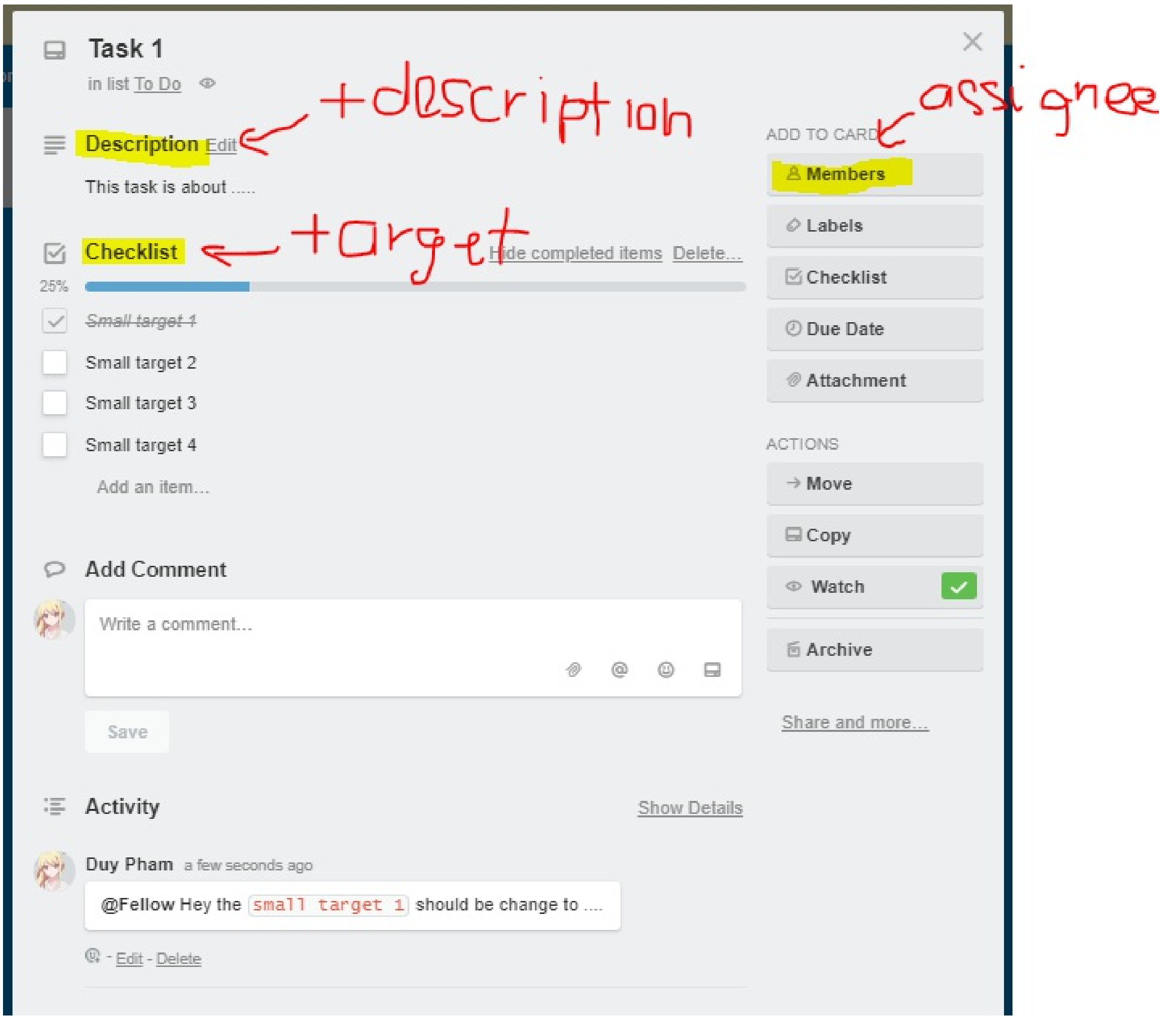


Please confirm your email address: orangezuice1212@gmail.com. Check your inbox. Resend email.

General Process Of Create Our Website ☆ Personal Private ··· Show Menu

To Do	Doing	Done	+ Add another list
+ Add a card	+ Add a card	+ Add a card	

5. Create a card to describe and delegate tasks to members



Task 1

in list To Do

Description Edit

This task is about .....

Checklist Hide completed items Delete...

25%

☑ ~~Small target 1~~

□ Small target 2

□ Small target 3

□ Small target 4

Add an item...

Add Comment

Write a comment...

Save

Activity Show Details

Duy Pham a few seconds ago

@Fellow Hey the `small target 1` should be change to ....

- Edit - Delete

ADD TO CARD

Members

Labels

Checklist

Due Date

Attachment

ACTIONS

Move

Copy

Watch ✓

Archive

Share and more...

Bookmarks:

Trello Document

Chapter 2: OpenCharity Website

OpenCharity Overview

Overview:

An overview on the website that we are going to clone

- What components is it made up of
- Point out some techniques and features used or built.

Details:

Features

- Scrollspy
- Blink button effect
- Slider with mouse drag
- Responsiveness

Bookmarks:

A look on the prototype website

Prepare Files

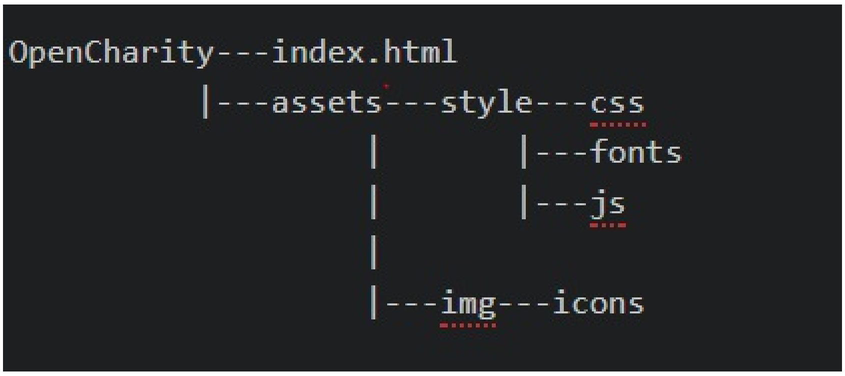
Overview:

- Build project folder structure
- Convert PSD file to some neccessary assets
- Integrate Bootstrap 4, jQuery and Popper.js

Details:

Step by Step

1. Create new project folder with structure below



```
OpenCharity---index.html
          |---assets---style---css
                   |       |---fonts
                   |       |---js
                   |
                   |---img---icons
```

2. Find and download the following fonts then save them to assets/style/fonts : `Museosans 500` `OpenSans` `OpenSans semibold` `Museosans rounded 300`
3. Download the OC.PSD file from this link
4. Open file with Photoshop and export neccessary components to PNG, details in here Trello task
5. Inspect colors from PSD file and note them down for later use.
6. Download Bootstrap 4 and save bootstrap.min.js and bootstrap.min.css in "assets/style/js" and

"assets/style/js"

7. Download and save jquery-3.3.1.slim.min.js in "assets/style/js"
8. Download and save popper.min.js in "assets/style/js"
9. Go to Bootstrap Starter Template and copy paste its content to index.html

Now we are ready

Bookmarks:

Link to GitHub Project: OpenCharity

Linking Files Implementing Style sheet

Overview:

Link CSS and JS files to index.html

Implement CSS of our own with less

Details:

Features:

Using LESS

Use variables for code reuse and function darken().

CSS Reset

Is a short, often compressed (minified) set of CSS rules that resets the styling of all HTML elements to a consistent baseline.

Why use CSS Reset ?

Every browser has its own default 'user agent' stylesheet, that it uses to make unstyled websites appear more legible.Using a CSS Reset, CSS authors can force every browser to have all its styles reset to null, thus avoiding cross-browser differences as much as possible.

Style Sheet Implementation

- Declare some variables that hold values of mainly-used colors and font-families
- CSS reset
- Customize components that will be used like headers, buttons, list, anchors.

Bookmarks:

Link to GitHub Project: OpenCharity

Navbar Part 1

Overview:

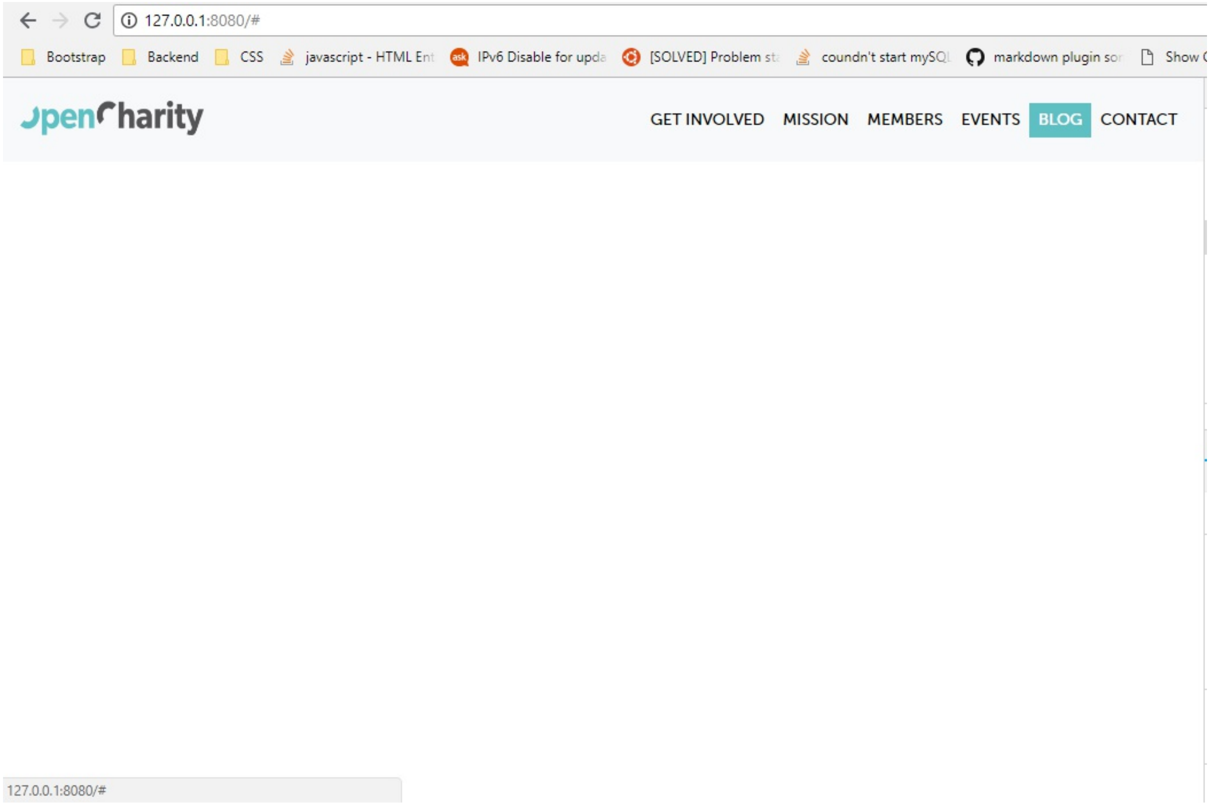
Implement styles for Navigation Bar

Details:

- Display favicon
- Display Open Charity logo
- Implement navbar, navbar links and navbar items styles

    - Add padding
    - Add hover effect
    - Add transition
    - Change font
    - Change background color

The Navbar now look like this



Bookmarks:

Link to GitHub Project: OpenCharity

Navbar Part 2

Overview:

Implement event-handlers for Navigation bar

Details:

Effect implemented

- Close Responsive menu when a link is clicked
- Activate scrollspy to add active class to navbar items on scroll
- On scroll down if the offset is higher than 50px the navbar shrink

Note

`@media` rule CSS: The `@media` rule is used in media queries to apply different styles for different media types/devices.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Details about syntax and usage are in Bookmarks

Bookmarks:

Link to GitHub Project: OpenCharity

@media rule in CSS

Header Part 1

Overview:

Implement page header sections

Details:

Components implemented

- Header with centered inspiration texts
- Event introduction and schedule section
- Register button

Note

- Can use the `Text tool` of Photoshop to extract text from PSD file

Bookmarks:

Link to GitHub Project: OpenCharity

Header part 2

Overview:

Finish page header sections

Details:

Components implemented

- Add background image for header
- Change background-color footer
- Add border-bottom footer
- Reposition event detail and `Register` button
- Change color and font of texts

Note

- You may confuse about background css properties. Read more in the Bookmarks.

Bookmarks:

Link to GitHub Project: OpenCharity

CSS background properties

Get involved section

Overview:

Implement UI for get-involved section

Details:

Components implemented

- Add Group Logos, Sologans, and Intro

Note

Differences between inline elements and block-inline elements

Elements with `display:inline-block` are like `display:inline` elements, but they can have a width and a height. That means that you can use an inline-block element as a block while flowing it within text or other elements.

Difference of supported styles as summary:

- inline: only margin-left, margin-right, padding-left, padding-right
- inline-block: margin, padding, height, width

Bookmarks:

Link to GitHub Project: OpenCharity

Inline block and inline elements

Mission section

Overview:

Implement UI for Mission Section

Details:

Components implemented

Section with `id = mission` containing:

- Three col represent three mission
- Cols contain a card with icon, title and content

Note

`border-radius` CSS property

The border-radius property defines the radius of the element's corners.

This property allows you to add rounded corners to elements!

This property can have from one to four values. However there are other rules, see in the Bookmarks

Bookmarks:

Link to GitHub Project: OpenCharity

border-radius CSS

Members section

Overview:

Install Owl Carousel 2

Implement UI for Member Section

Details:

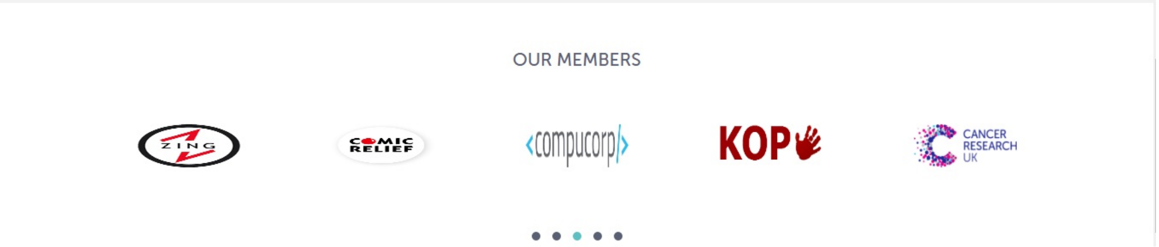
Implementation

Owl Carousel 2

- What is it ?
- Owl Carousel is a touch enabled jQuery plugin that lets you create a beautiful responsive carousel slider.
- Why use Owl Carousel 2 over Bootstrap Carousel ?
  - Fully Customisable: Over 60 options. Easy for novice users and even more powerful for advanced developers.
  - Touch and Drag Support: Designed specially to boost mobile browsing experience. Mouse drag works great on desktop too!
  - Fully Responsive: Almost all options are responsive and include very intuitive breakpoints settings.
  - Modern Browsers: Owl uses hardware acceleration with CSS3 Translate3d transitions. Its fast and works like a charm!
  - Zombie Browsers: CSS2 fallback supported for older browser.
  - Modules and Plugins: Owl Carousel supports plugin modular structure. Therefore, you can detach plugins that you won't use on your project or create new ones that fit your needs
- Steps to Install
  - Download Owl Carousel 2
  - Move to dist folder. Copy `owl.carousel.min.js` to /assets/style/js
  - Move to dist/assets folder. Copy `owl.theme.default.min.css` `owl.carousel.min.css` to /assets/style/css
- How to used
  - Put class `owl-carousel` and `owl-theme` on the `div` where you want the carousel to appear
  - Then put on images inside as carousel items
- JavaScript
  - Call Carousel constructor with syntax: $('<div_id>').owlCarousel({

```
  &lt;Object containing setting properties>
})
```

- Result



Note

Checkout the list of options for Additional Carousel settings in the Bookmarks.

Bookmarks:

Link to GitHub Project: OpenCharity

Carousel Settings

Event section

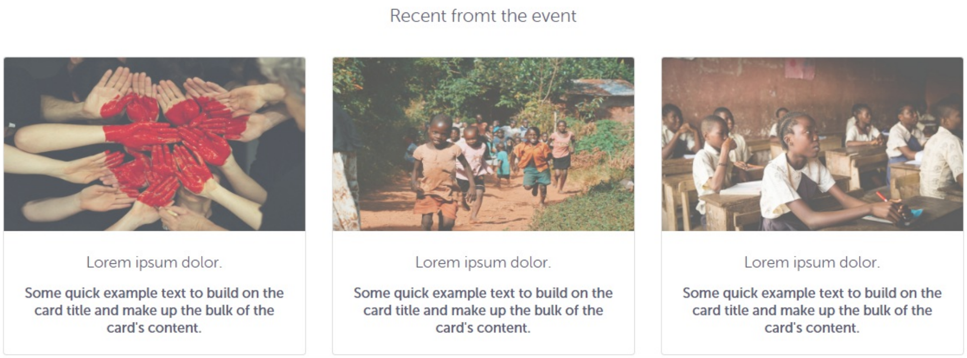
Overview:

Implement UI for Event Section

Details:

Implementation

- Using Bootstrap `card` to make three columns containing three cards describing three recents events
- Result



Recent fromt the event

Lorem ipsum dolor.

Some quick example text to build on the card title and make up the bulk of the card's content.

Lorem ipsum dolor.

Some quick example text to build on the card title and make up the bulk of the card's content.

Lorem ipsum dolor.

Some quick example text to build on the card title and make up the bulk of the card's content.

Note

Bookmarks:

Link to GitHub Project: OpenCharity

Blog section

Overview:

Implement UI for Blog Section

Details:

Implementation

- Using Owl Carousel 2 to create a slider with 4 blogs
- Result

```
![CarouselResult](./blogs.PNG)
```

Note

Bookmarks:

Link to GitHub Project: OpenCharity

Contact Section and Footer

Overview:

Implement UI for Contactc Section Implement UI for Footer

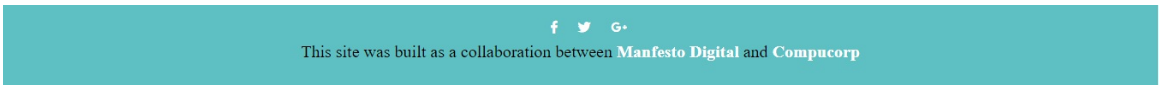
Details:

Implementation

- Implement Contact form with input fields `Name` , `Email` , and `Message` textarea
- Result

  ```
  ![CarouselResult](./contact-form.PNG)
  ```

- Implement Footer with social sharing buttons `Facebook` , `Twitter` and `Google Plus`
- Result



This site was built as a collaboration between **Manfesto Digital** and **Compucorp**

Note

- Always structure css similar to HTML to minimize code repetition and narrow down css effects range.

Bookmarks:

Link to GitHub Project: OpenCharity

Enhancements

Overview:

Add animation, fix responsive bugs, make changes to UI

Details:

Implementation

- Integrate Animate.css library
  - Go to https://daneden.github.io/animate.css/
  - Click on `Download Animate.css`
  - Save content as `animate.css` to following path: `./assets/style/css/animate.css`
- Add flash animation to `Slack Group` button
- Active Scrollspy to `navbar`
- Smooth Scrolling with jQuery
- Add fade in effect to `Mission Section`
- Change letter-spacing of Event Footer for better mobile look
  - Before

    ```
    ![Before](./before-header.PNG)
    ```

  - After

    ```
    ![Ater](./header-after.PNG)
    ```

Note

Bookmarks:

Link to GitHub Project: OpenCharity

Testing

Overview:

Cover some remaining issues about UI, animation Increase website performance Do some testing

Details:

Issues Spotted

Compressing Image

Steps

- Go to https://imagecompressor.com/
- Upload all project images and wait for image compressing to complete
- Download them all and replace old images with new ones

Cross-browser Testing

Try out website on different browser and make sure everything works fine

Animation

Add `fade in up` animation to header

With Mobile devices

Remove `fade in` animation from Mission Section while on mobile cuz mobile does not have hover effect

Put website online

Steps

- Go to https://app.netlify.com/drop
- Drag and Drop your project folder into white box
- Wait for uploading and enjoy.

My website

W3C markup validation service

What is ?

- The Markup Validator is a free service by W3C that helps check the validity of Web documents.
- Most Web documents are written using markup languages, such as HTML or XHTML. These languages are defined by technical specifications, which usually include a machine-readable formal grammar (and vocabulary). The act of checking a document against these constraints is called validation, and this is what the Markup Validator does

How to use

- Go to https://validator.w3.org/about.html
- Paste your website url on the input
- Click `Check` and wait for result

Website Speed Test

- Go to https://tools.pingdom.com
- Paste website url into the input
- Click `Start Test` and wait for result

Note

Bookmarks:

Link to GitHub Project: OpenCharity

Recap

Overview:

Review what we have built so far

Details:

- How we can export images from PSD file as well as getting colors and fonts for our website
- Then moving to setting our variables and style sheet
- Learned how to nest our code in `less` so it's more readable, understandable.
- Learned how to use Bootstrap 4 and its components plus adding animation to our project with `jQuery` or `Animate.css`
- Learned how to organize our project by applying our `Trello` steps that we had written before starting
- Learned how to write clean code with comment
- Dealt with responsive part so our website is available on all devices
- Increase website performance and run the markup and speed test

Bookmarks:

Link to GitHub Project: OpenCharity

Chapter 3: 404 Penguins Page

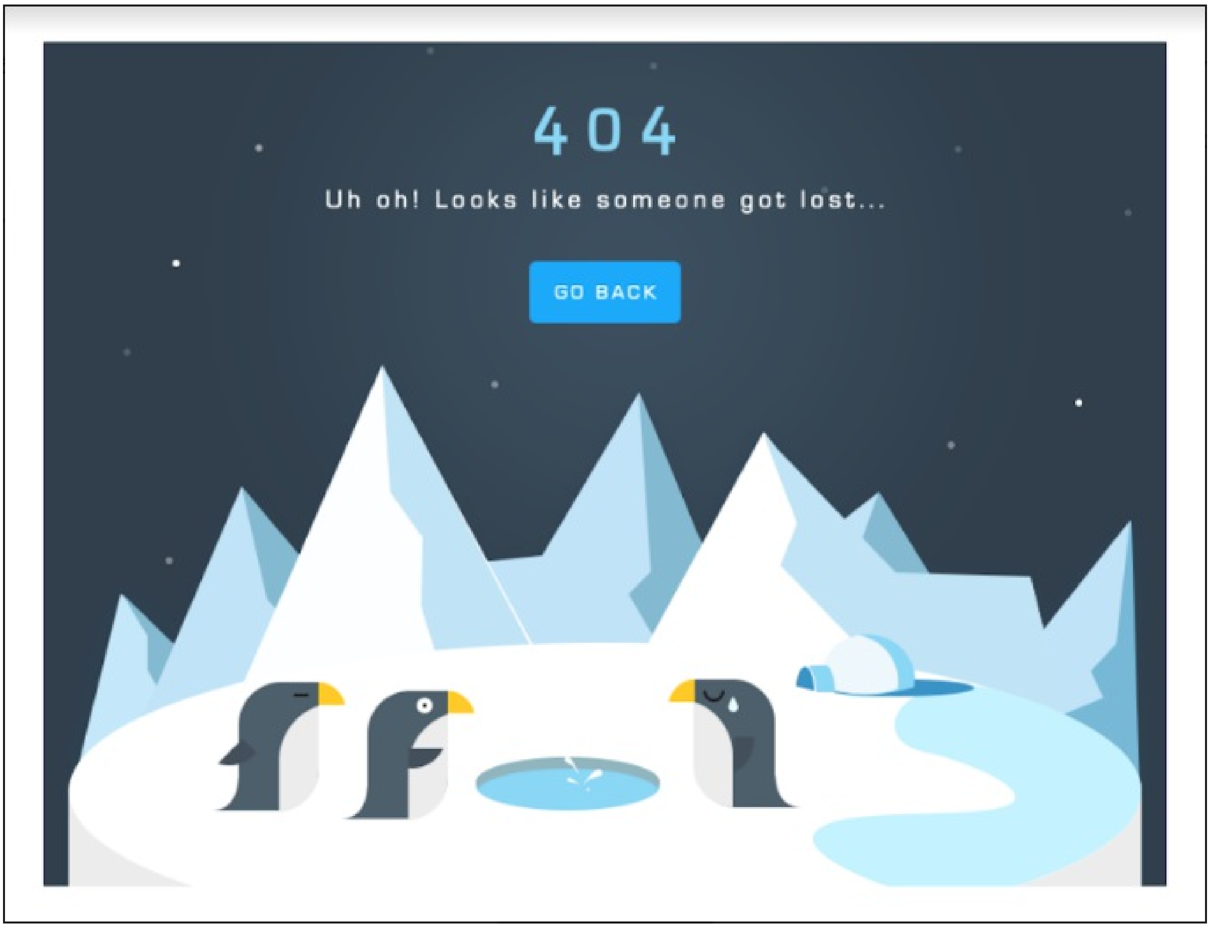
404 Overview

Overview

An overlook on what we gonna build

Details

Prototype



404

Uh oh! Looks like someone got lost...

Target

- Create snowfall animation
- Make image stick to the page bottom no matter screen size

Components

- 404 logo
- Error message
- `Go back` button
- Sticky background image
- Snowfall animation

Bookmarks

Prepare Files

Overview

Bootstrap the project

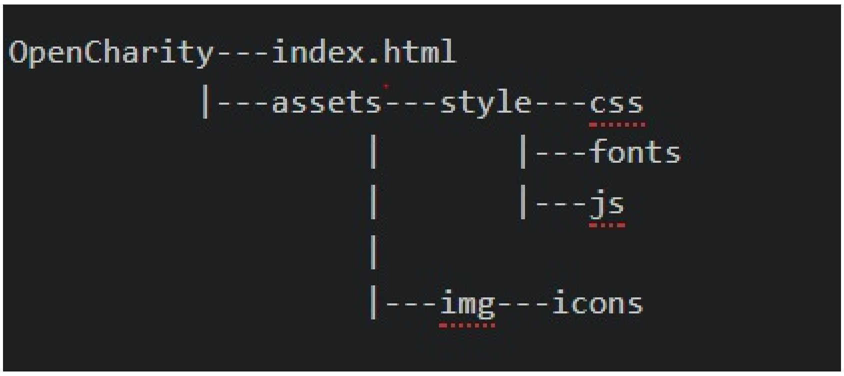
Details

What we gonna use

- `bootstrap.min.css`
- CSS keyframe

Building folder structure

The folder structure is pretty similar to `OpenCharity` project



```
OpenCharity---index.html
          |---assets---style---css
                   |       |---fonts
                   |       |---js
                   |
                   |---img---icons
```

Bootstraping steps

- Go to https://drive.google.com/drive/folders/1WAn16FJ0s-6j1DiR2-LcJ9HAQoPIIBGY to download PSD file and save it to project
- Go to https://www.dafontfree.net/freefonts-eurostile-f484.htm to download eurostile font
  - Use photoshop to extract neccessary images.
  - One penguin for favicon
  - Three penguins with iceberg as background
- Note down colors to be used.
- Put `bootstrap.min.css` into .assets/style/css
- Run following commands `git init` , `echo > README.md` , `git add .` , `git commit -m "Bootstrap project"`

Bookmarks

Eurostile font

PSD File

Start code

Overview

Implement UI Styling Page Add animation

Details

Components implemented

- One wrapper container containing

  - Header: 404 with response text and `GO BACK` button
  - One image as background
  - One `div` to handle the animation

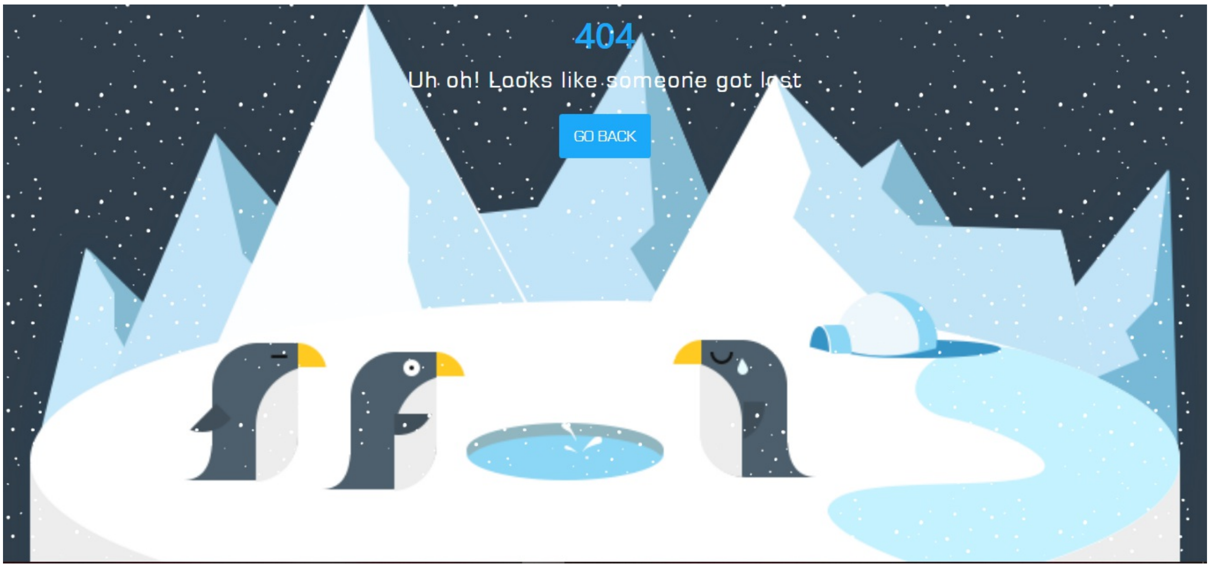
Styling

Mainly used colors and fonts

@gray_color: #313f4c; @blue-color: #1ca9f9; @white-color: #ffffff; @Eurostile: 'Eurostile';

Animation was created using CSS `@keyframe` . It's a long explaination. Check out the Bookmarks

Result



404

Uh oh! Looks like someone got lost

Bookmarks

CSS keyframe

Link to GitHub project 404 Penguins Page

404 Recap

Overview

Advices on project

Details

- If you fail try again and again
- Should not add animation to every section cuz it could be annoying
- Mention the final project

Bookmarks

Link to GitHub project 404 Penguins Page

Chapter 4: Merkury Website

Merkury Overview

Overview

Discuss on what are we going to build

What will we learn along the project

And some small reminders

Details

What are we building

An introductory website of the merkury dashboard which displays:

- The important features
- The workflow
- The powerful design of the Merkury dashboard

What are we learning

Making powerful animation by CSS and positioning elements

Deal with two different plugins: `scroll reveal` and `jQueryUI`

Components to implement

The Navbar

- Apply transition effect
- Apply Scrollspy
- Collapse Navbar when item clicked by jQuery

The Header

- Chart animation in the background
- A title and a subtitle
- A `Get Started` button with a nice hover effect
- An image display between header and features section

Features Section

First Part

- Images represent features of the dashboard with cool over effect
- A short description text with `See more` button below

Second Part

- On the left, we have title and some text
- On the right side, we have image with the fade-in animation

Drag and Drop Section

- On the right, a title with some text and two navigation button

- On the left, we have customize drop down button, allow drag and drop item

Special Section

- Title and some text
- Running line effect when hover
- Infinite floating effect on the image

Pricing Section

- Three Bootstrap card with fade-up-in effect with different delay

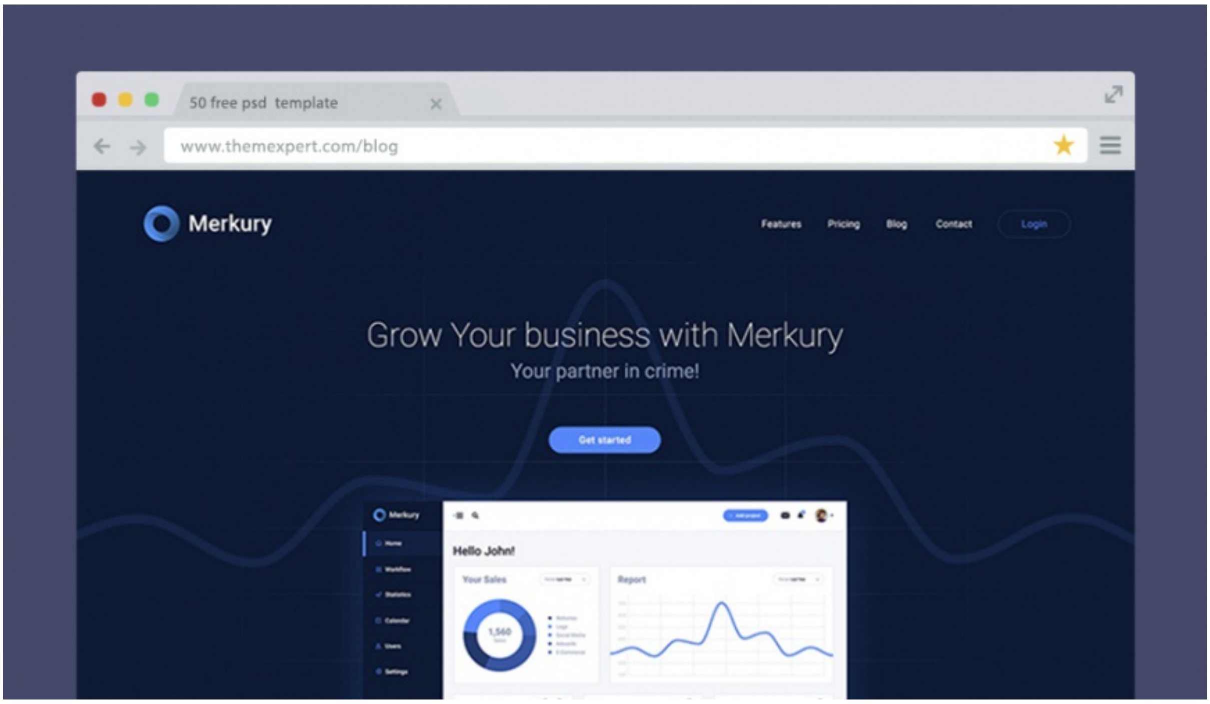
Join Section

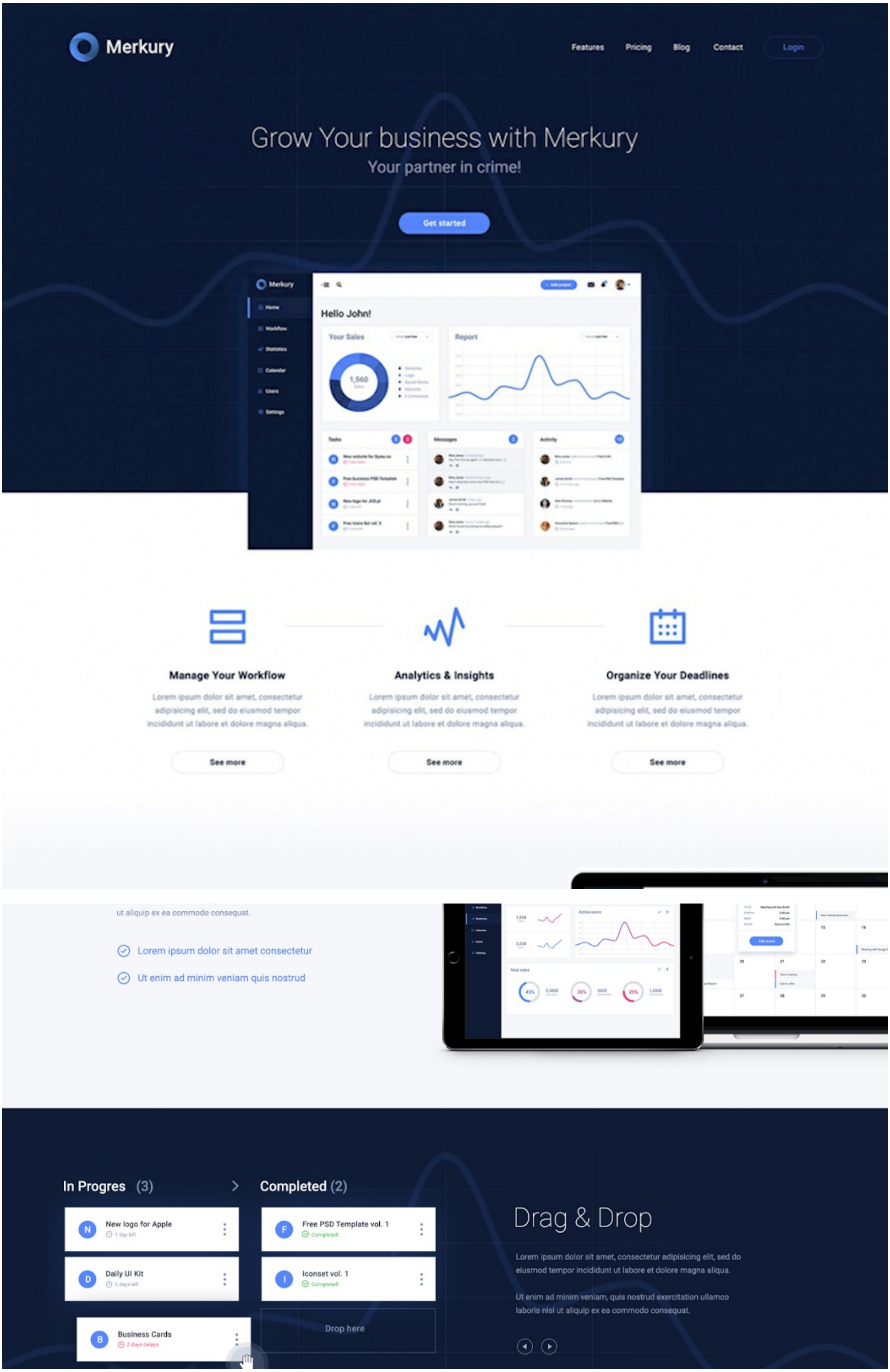
- A text with `Join now!` and separate line

Footer Section

- All right reserved and social buttons

Some visual images





Grow Your business with Merkury

Your partner in crime!

Manage Your Workflow

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Analytics & Insights

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Organize Your Deadlines

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

ut aliquip ex ea commodo consequat.

- Lorem ipsum dolor sit amet consectetur
- Ut enim ad minim veniam quis nostrud

In Progres (3)

- N New logo for Apple — 1 day left
- D Daily UI Kit — 5 days left
- B Business Cards — 2 days delays

Completed (2)

- F Free PSD Template vol. 1 — Completed!
- I Iconset vol. 1 — Completed!

Drag & Drop

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Note

Our website will be mobile first, which mean it have to fit any devices

And never forget the Testing part

Bookmarks

None

Prepare Files

Overview

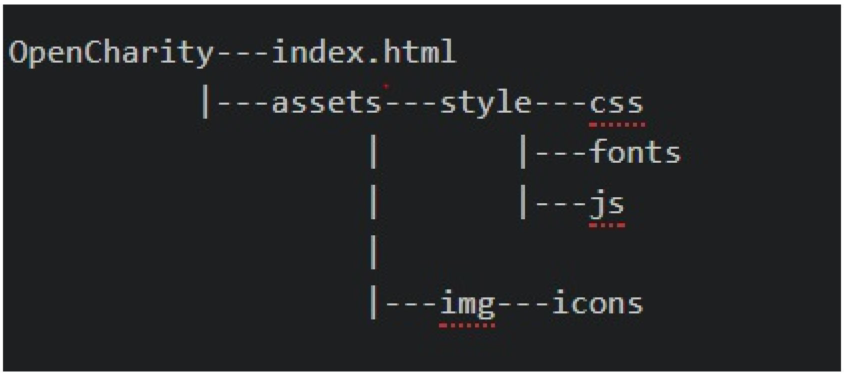
Build folder structure

Extract images and colors from PSD

Details

Folder structure

Similar to OpenCharity folder structure



```
OpenCharity---index.html
          |---assets---style---css
                    |       |---fonts
                    |       |---js
                    |
                    |---img---icons
```

Colors used

dark-blue-color: #oe1a35;

light-blue-color: #5584ff;

light-grey-color: #8492af;

black-color: #000000;

white-color: #ffffff;

Fonts used

Roboto-regular-font Roboto-bold-font Roboto-thin Roboto-medium Montserrat-regular Montserrat-bold Font-awesome

Include libraries

Bootstrap

- Download `bootstrap.min.css` and `bootstrap.min.js` from Bootstrap Download Link
- Save them into `./assets/style/css/` and `./assets/style/js/`

jQuery

- Go to jQuery Download and save it to `./assets/style/js/`

Popper.js

- Go to Popper Download and save it to `./assets/style/js/`

Bookmarks

None

Linking Files

Overview

Link image and library with index.html

Apply some custom CSS

Details

Linking Files

The index html should appear like below

Custom CSS

- Define `fontface` s
- Assign colors and fonts to variables
- Copy General Reset CSS code from Open Charity project into `style.less`
- Customize some tag with background-color, margin and padding

Bookmarks

Link to GitHub Project

Navbar Part 1

Overview

Implement styles for Navigation Bar

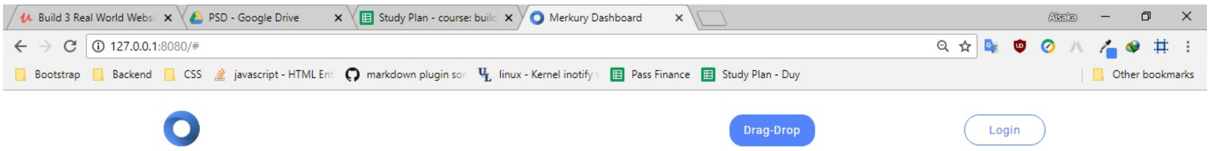
Details

Implementation

- Display favicon
- Display Merkury logo
- Implement navbar, navbar links and navbar items styles
    - Add padding
    - Add hover effect
    - Add transition
    - Change font-family
    - Change background color

Result

The background of navbar is transparent at present because it will inherit the website background later on



Bookmarks

Link to GitHub Project

Navbar Part 2

Overview:

Add some enhancements to make our navbar more interactive

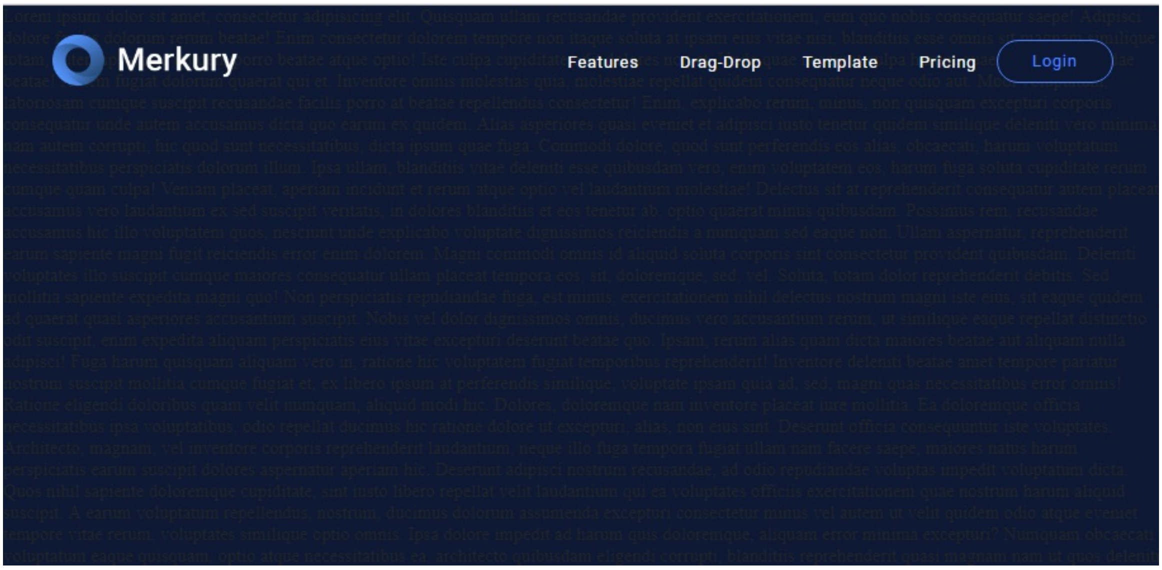
Details:

Effect implemented

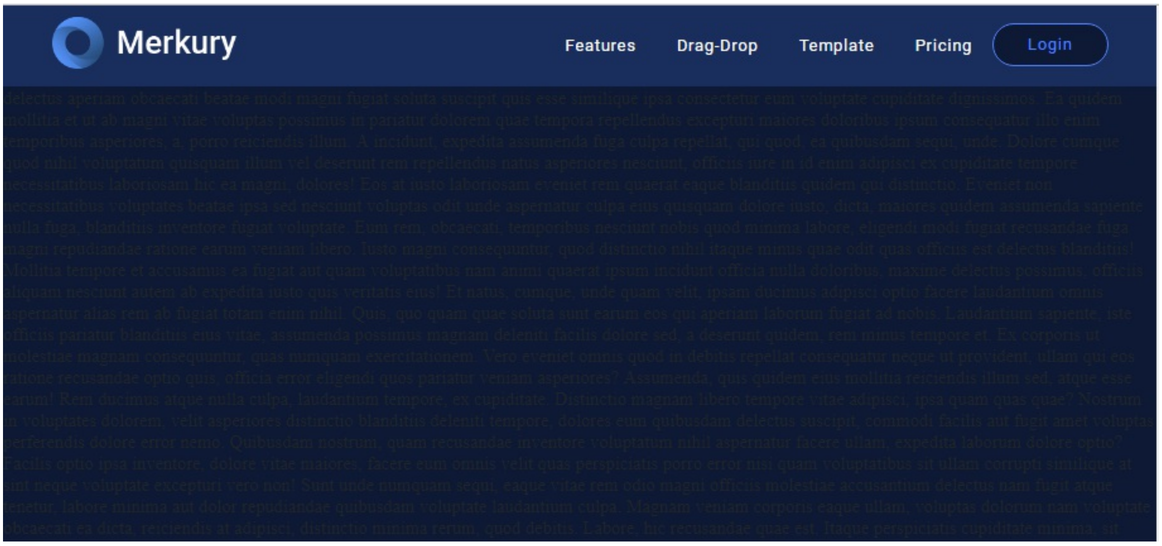
- Close Responsive menu when a link is clicked
- Activate scrollspy to add active class to navbar items on scroll
- On scroll down if the offset is higher than 50px the navbar shrink and change color

Result

- Navbar Before shrinking



- Navbar After shrinking



delectus aperiam obcaecati beatae modi magni fugiat soluta suscipit quis esse similique ipsa consectetur eum voluptate cupiditate dignissimos. Ea quidem mollitia et ut ab magni vitae voluptas possimus in pariatur dolorem quae tempora repellendus excepturi maiores doloribus ipsum consequatur illo enim temporibus asperiores, a, porro reiciendis illum. A incidunt, expedita assumenda fuga culpa repellat, qui quod, ea quibusdam sequi, unde. Dolore cumque quod nihil voluptatum quisquam illum vel deserunt rem repellendus natus asperiores nesciunt, officiis iure in id enim adipisci ex cupiditate tempore necessitatibus laboriosam hic ea magni, dolores! Eos at iusto laboriosam eveniet rem quaerat eaque blanditiis quidem qui distinctio. Eveniet non necessitatibus voluptates beatae ipsa sed nesciunt voluptas odit unde aspernatur culpa eius quisquam dolore iusto, dicta, maiores quidem assumenda sapiente nulla fuga, blanditiis inventore fugiat voluptate. Eum rem, obcaecati, temporibus nesciunt nobis quod minima labore, eligendi modi fugiat recusandae fuga magni repudiandae ratione earum veniam libero. Iusto magni consequuntur, quod distinctio nihil itaque minus quae odit quas officiis est delectus blanditiis! Mollitia tempore et accusamus ea fugiat aut quam voluptatibus nam animi quaerat ipsum incidunt officia nulla doloribus, maxime delectus possimus, officiis aliquam nesciunt autem ab expedita iusto quis veritatis eius! Et natus, cumque, unde quam velit, ipsam ducimus adipisci optio facere laudantium omnis aspernatur alias rem ab fugiat totam enim nihil. Quis, quo quam quae soluta sunt earum eos qui aperiam laborum fugiat ad nobis. Laudantium sapiente, iste officiis pariatur blanditiis eius vitae, assumenda possimus magnam deleniti facilis dolore sed, a deserunt quidem, rem minus tempore et. Ex corporis ut molestiae magnam consequuntur, quas numquam exercitationem. Vero eveniet omnis quod in debitis repellat consequatur neque ut provident, ullam qui eos ratione recusandae optio quis, officia error eligendi quos pariatur veniam asperiores? Assumenda, quis quidem eius mollitia reiciendis illum sed, atque esse earum! Rem ducimus atque nulla culpa, laudantium tempore, ex cupiditate. Distinctio magnam libero tempore vitae adipisci, ipsa quam quas quae? Nostrum in voluptates dolorem, velit asperiores distinctio blanditiis deleniti tempore, dolores eum quibusdam delectus suscipit, commodi facilis aut fugit amet voluptas perferendis dolore error nemo. Quibusdam nostrum, quam recusandae inventore voluptatum nihil aspernatur facere ullam, expedita laborum dolore optio? Facilis optio ipsa inventore, dolore vitae maiores, facere eum omnis velit quas perspiciatis porro error nisi quam voluptatibus sit ullam corrupti similique at sint neque voluptate excepturi vero non! Sunt unde numquam sequi, eaque vitae rem odio magni officiis molestiae accusantium delectus nam fugit atque tenetur, labore minima aut dolor repudiandae quibusdam voluptate laudantium culpa. Magnam veniam corporis eaque ullam, voluptas dolorum nam voluptate obcaecati ea dicta, reiciendis at adipisci, distinctio minima rerum, quod debitis. Labore, hic recusandae quae est. Itaque perspiciatis cupiditate minima, sit

Note

Remind of media rule

`@media` rule CSS: The `@media` rule is used in media queries to apply different styles for different media types/devices.

Media queries can be used to check many things, such as:

- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution

Details about syntax and usage are in Bookmarks

Bookmarks

Link to GitHub Project

@media rule in CSS

Header

Overview:

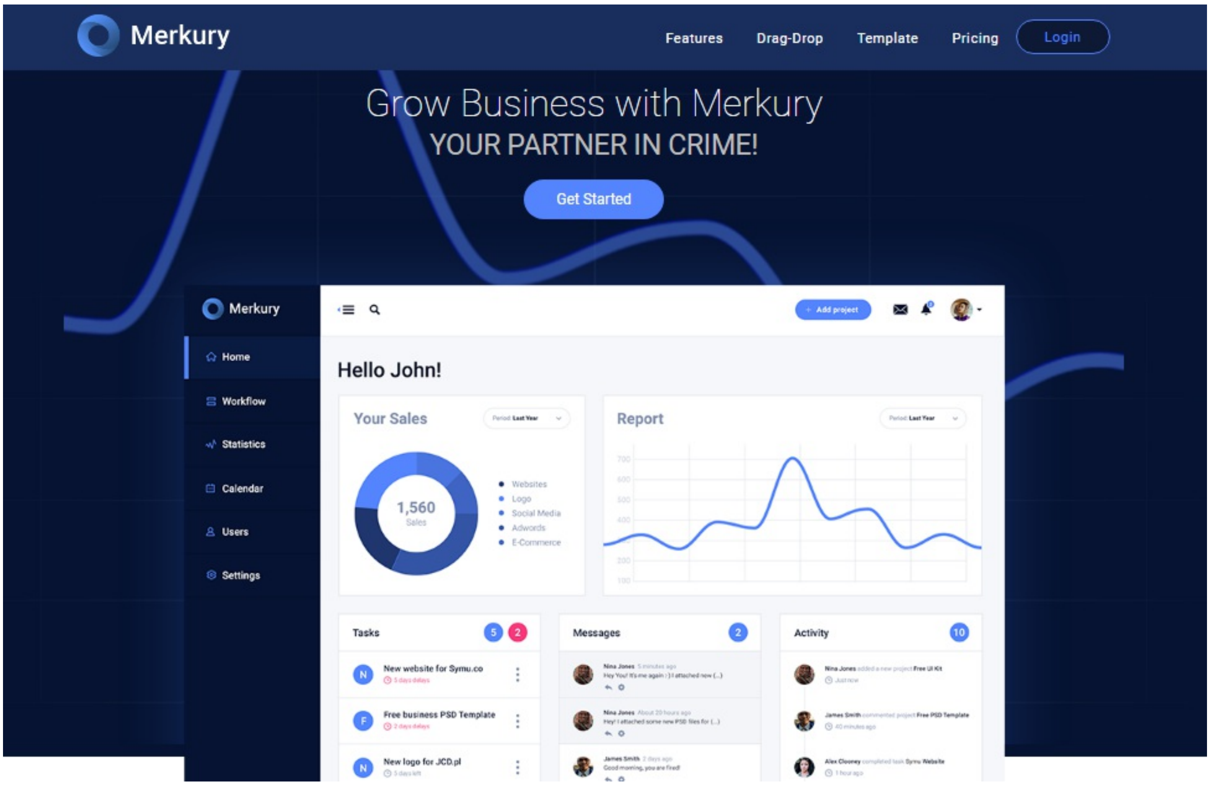
Implement UI for page header sections

Details:

Components implemented

- A title, a sologan and `Get started` button below
- Dashboard image with glowing sides extracted from PSD
- Grid background with chart on top
- Chart background with left-to-right linear moving effect

Result



Note

- Remind of CSS `@keyframe` , checkout link in the Bookmarks

Bookmarks:

Link to GitHub Project

CSS keyframe

Service Section

Overview:

Implement UI service section

Details:

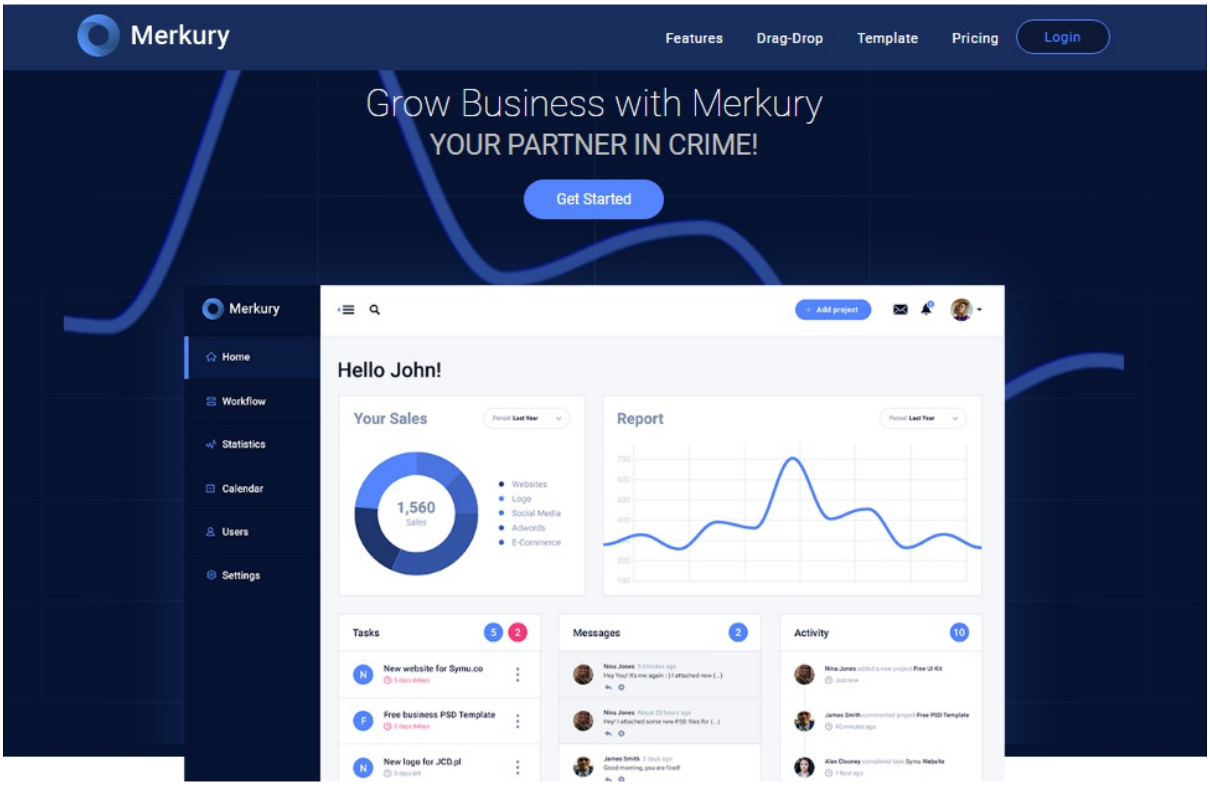
Components implemented

- Three columns represent three service with logo, title and detail text
- A `See More` button with hover effect
- Light gray divider between each image

Styling

- Customize font weight, font size and font family on titles and texts
- Add hover effect on `See More` button
- Add hover effect on logo images

Result

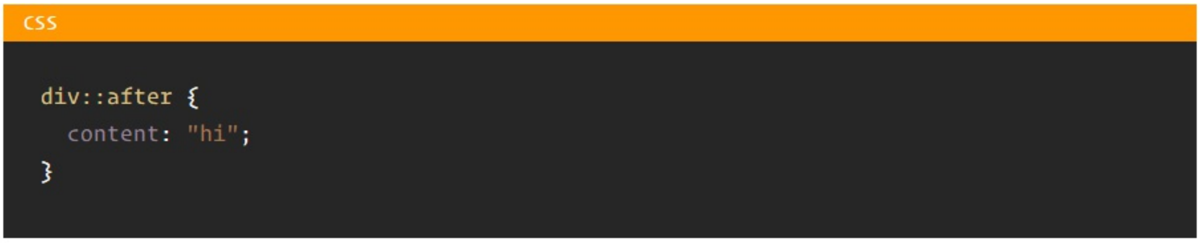


Note

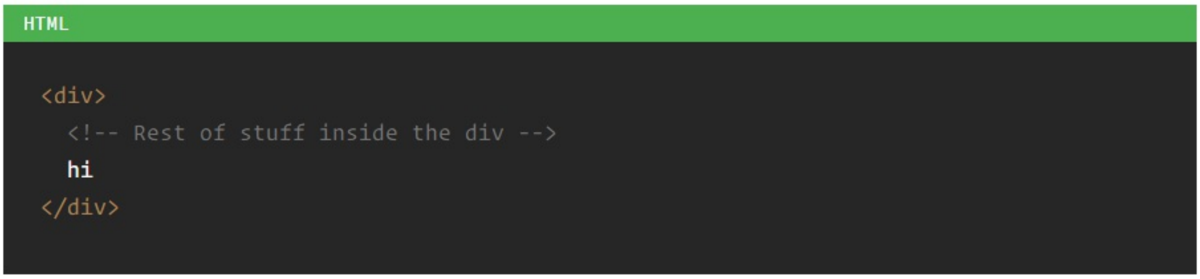
- Psuedo element `::after`

What is it?

`::after` is a pseudo element which allows you to insert content onto a page from CSS (without it needing to be in the HTML). While the end result is not actually in the DOM, it appears on the page as if it is, and would essentially be like this:


CSS

```css
div::after {
  content: "hi";
}
```


HTML

```html
<div>
  <!-- Rest of stuff inside the div -->
  hi
</div>
```

The capability of psuedo elements is tremendous. Check out the Bookmarks

Bookmarks:

Link to GitHub Project

Psuedo Elements CSS

Features Section

Overview:

Implement UI for Features section

Details:

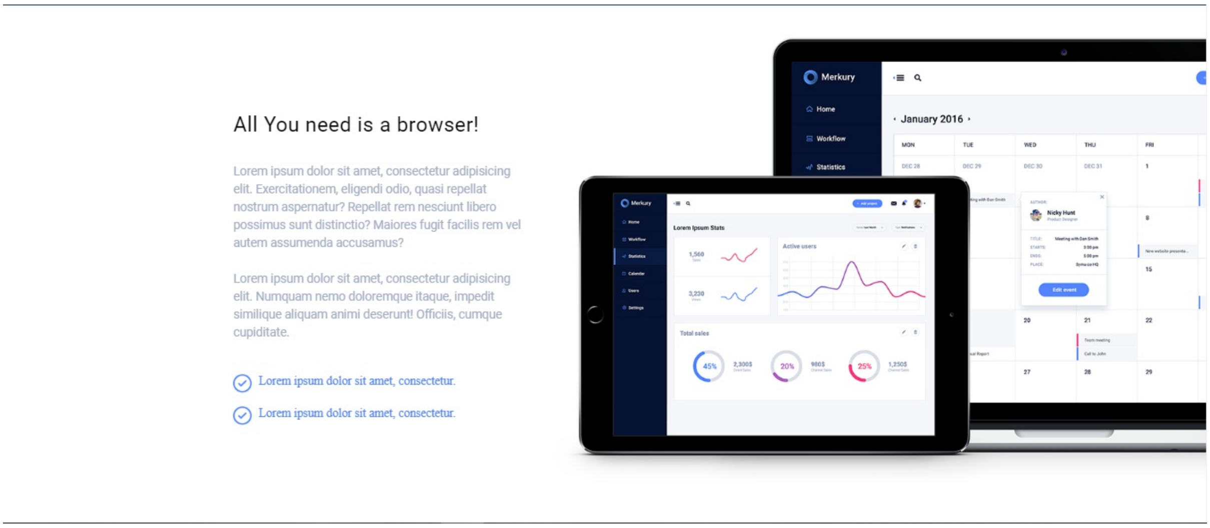
Components implemented

- Two Column, one take care of the text and one contains the image
- The text includes: A title, a paragraph and two checkpoints.

Styling

- Customize font weight, font size and font family on titles and texts
- Add padding and margin so everything look balanced

Result



All You need is a browser!

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Exercitationem, eligendi odio, quasi repellat nostrum aspernatur? Repellat rem nesciunt libero possimus sunt distinctio? Maiores fugit facilis rem vel autem assumenda accusamus?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Numquam nemo doloremque itaque, impedit similique aliquam animi deserunt! Officiis, cumque cupiditate.

✓ Lorem ipsum dolor sit amet, consectetur.

✓ Lorem ipsum dolor sit amet, consectetur.

Note

Nothing

Bookmarks:

Link to GitHub Project

Psuedo Elements CSS

Drag and Drop Section Part 1

Overview:

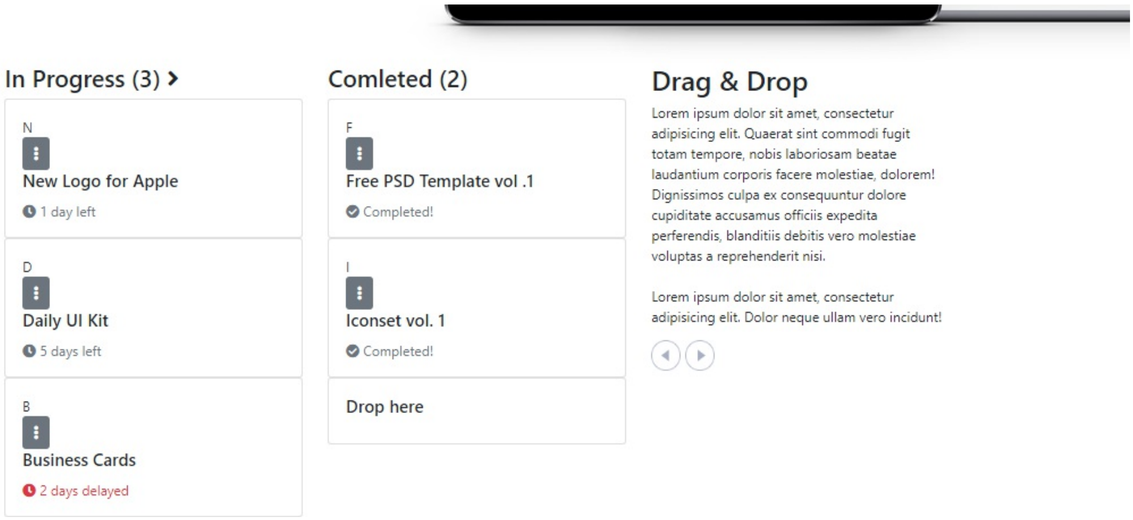
Implement skeleton for Drag and Drop section

Details:

Components implemented

- Three Column, one take care of the text and one contains card represent In-progress tasks, one contains card represent Completed tasks
- The text includes: A title, a paragraph and two checkpoints.

Result



In Progress (3) ›

N

New Logo for Apple

1 day left

D

Daily UI Kit

5 days left

B

Business Cards

2 days delayed

Comleted (2)

F

Free PSD Template vol .1

Completed!

I

Iconset vol. 1

Completed!

Drop here

Drag & Drop

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quaerat sint commodi fugit totam tempore, nobis laboriosam beatae laudantium corporis facere molestiae, dolorem! Dignissimos culpa ex consequuntur dolore cupiditate accusamus officiis expedita perferendis, blanditiis debitis vero molestiae voluptas a reprehenderit nisi.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolor neque ullam vero incidunt!

Note

Nothing

Bookmarks:

Link to GitHub Project

Drag and Drop Section Part 2

Overview:

Implement Style for Drag and Drop section

Details:

Requirement

Download and install jQueryUI from here

Copy and paste `jquery-ui.min.js` into `./assets/style/js`

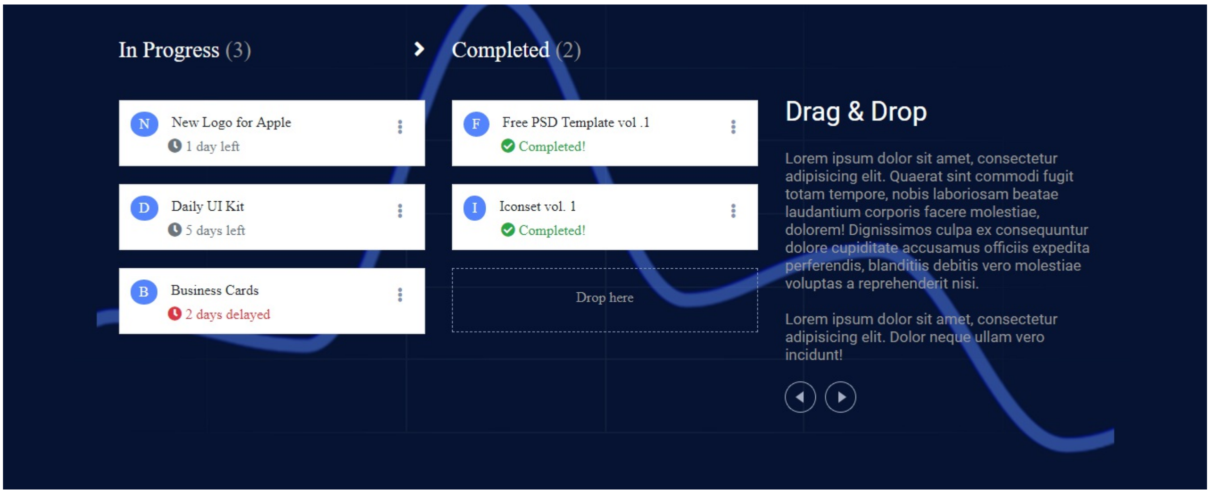
Styling

- Add Grid background to `#drag-drop` section
- Add Chart background to `.container` and apply animation on using `@keyframe`
- Add padding and margin to cards to to make them look nicer
- Turn `dropdown` button to ellipsis
- Turn mouse to crossarrow when hover on draggable item

Effect

- Efface the border and `Drop here` text from the `#droppable` item when a card is dropped onto using jQueryUI
- Restrict dragging to horizontally.

Result



Drag & Drop

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quaerat sint commodi fugit totam tempore, nobis laboriosam beatae laudantium corporis facere molestiae, dolorem! Dignissimos culpa ex consequuntur dolore cupiditate accusamus officiis expedita perferendis, blanditiis debitis vero molestiae voluptas a reprehenderit nisi.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolor neque ullam vero incidunt!

Note

Draggable and Droppable are just two of many interactions supported by jQueryUI. Find out more in the Bookmarks

Bookmarks:

Link to GitHub Project

More on jQueryUI

Template Section

Overview:

Implement UI for Template section

Details:

Idea

- A header at the top, horizontally centered
- Some description below
- Two animated line running until reach the dashboard at the middle
- Two paragraph float left and right from where the animation start
- The dashboard image will fade up at the beginning then float infinitely
- All animation start when the section got hovered on

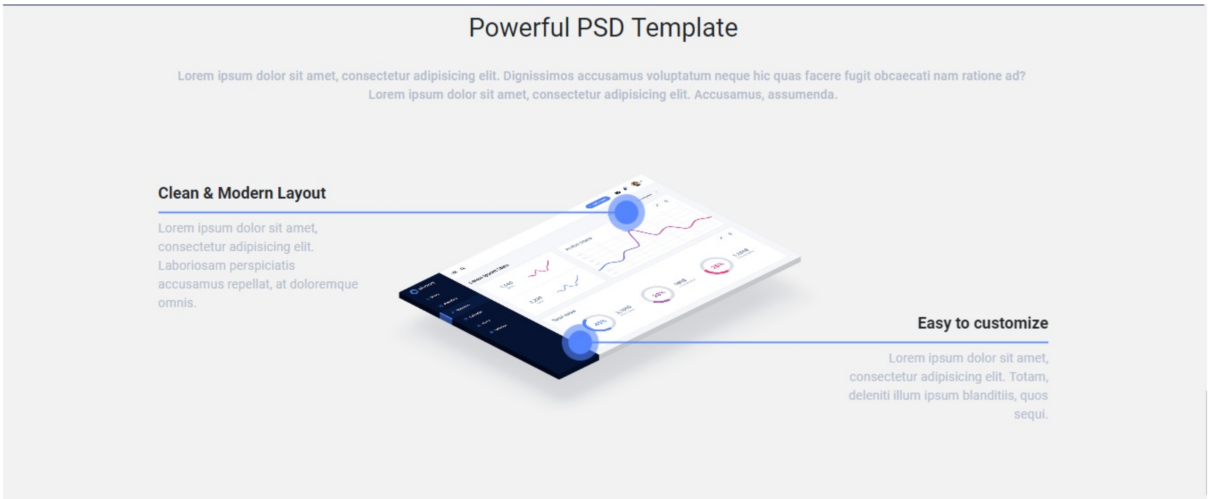
Components implemented

- First row contain the one Header, one sub-text
- Second row contains three Columns, one left-side, one right-side, one middle
- Left-side and right-side col contain a title, a paragraph, the middle one contain an image as background

Styling

- Customize font and color of header and text
- Add animation on side columns using `::before` and `@keyframes` : a blue circle running toward the center image
- Apply floating effect on the center image

Result



Powerful PSD Template

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dignissimos accusamus voluptatum neque hic quas facere fugit obcaecati nam ratione ad? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusamus, assumenda.

Clean & Modern Layout

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laboriosam perspiciatis accusamus repellat, at doloremque omnis.

Easy to customize

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Totam, deleniti illum ipsum blanditiis, quos sequi.

Note

- Remind of CSS `@keyframe` , checkout link in the Bookmarks

Bookmarks:

Link to GitHub Project

CSS keyframe

Pricing Section

Overview:

Implement UI for Pricing section

Remind of `box-shadow` CSS property

Details:

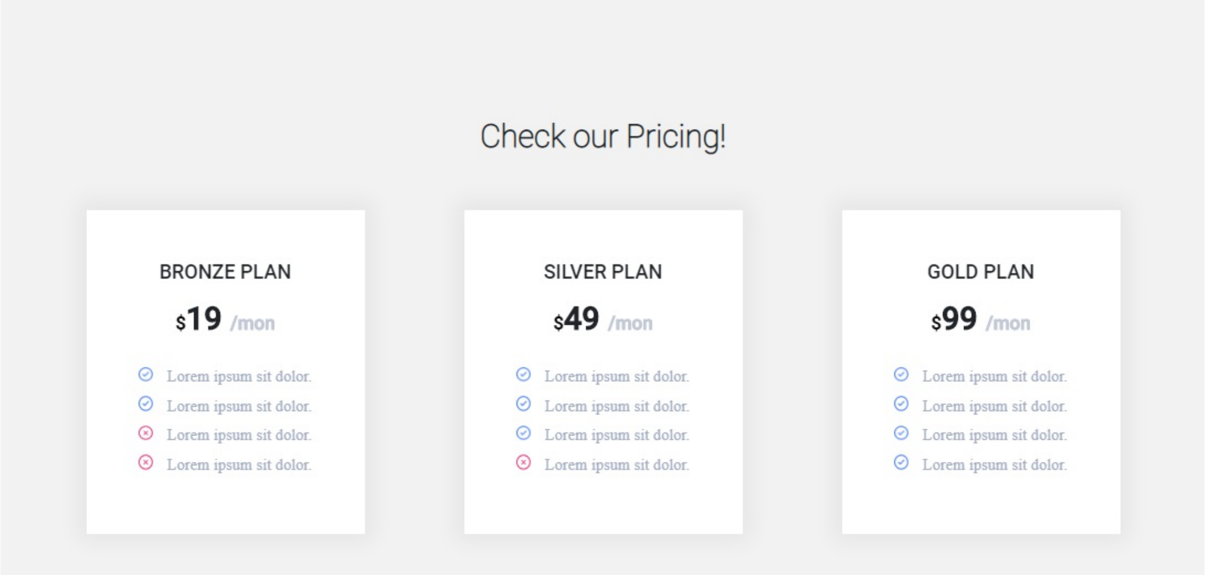
Components implemented

- Three Column, representing three pricing model: BRONZE, SILVER and GOLD
- Column contain title, price, benefits (blue-check circle) and fallbacks (x-circle) of each model

Styling

- Customize font and color of header and text

Result



Check our Pricing!

BRONZE PLAN

$19 /mon

✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.
⊗ Lorem ipsum sit dolor.
⊗ Lorem ipsum sit dolor.

SILVER PLAN

$49 /mon

✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.
⊗ Lorem ipsum sit dolor.

GOLD PLAN

$99 /mon

✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.
✓ Lorem ipsum sit dolor.

Note

- Box shadow CSS property params explain:
  - **The horizontal offset** (required) of the shadow, positive means the shadow will be on the right of the box, a negative offset will put the shadow on the left of the box.
  - **The vertical offset** (required) of the shadow, a negative one means the box-shadow will be above the box, a positive one means the shadow will be below the box.
  - **The blur radius** (required), if set to 0 the shadow will be sharp, the higher the number, the more blurred it will be, and the further out the shadow will extend. For instance a shadow with 5px of horizontal offset that also has a 5px blur radius will be 10px of total shadow.
  - The spread radius (optional), positive values increase the size of the shadow, negative values decrease the size. Default is 0 (the shadow is same size as blur).
  - Color (required) - takes any color value, like hex, named, rgba or hsla. If the color value is omitted, box shadows are drawn in the foreground color (text color). But be aware, older WebKit browsers (pre Chrome 20 and Safari 6) ignore the rule when color is omitted.
- Using a semi-transparent color like rgba(0, 0, 0, 0.4) is most common, and a nice effect

Find out more about `box-shadow` in the Bookmarks

Bookmarks:

Link to GitHub Project

Box Shadow CSS

Join and Footer Section

Overview:

Implement UI for Join Section Implement UI for Footer Section Remind of `:before` , `:after` and `display: flex`

Details:

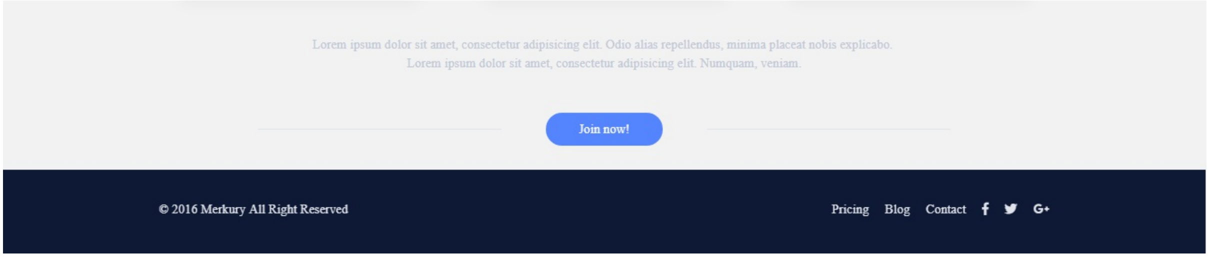
Components implemented

- Join Section: A presentation text with `Join now!` button below
- Footer Section: Copyright text float left, `Pricing` , `Blog` , `Contact` and social links float right

Styling

- Customize font and color of text, icon, style `Join now!` button
- Apply `::before` and `::after` to `Join now!` button

Result



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio alias repellendus, minima placeat nobis explicabo.
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Numquam, veniam.

© 2016 Merkury All Right Reserved

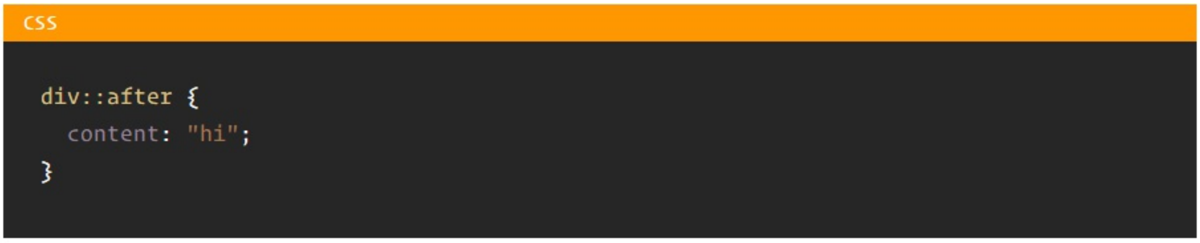
Note

Remind of psuedo classes

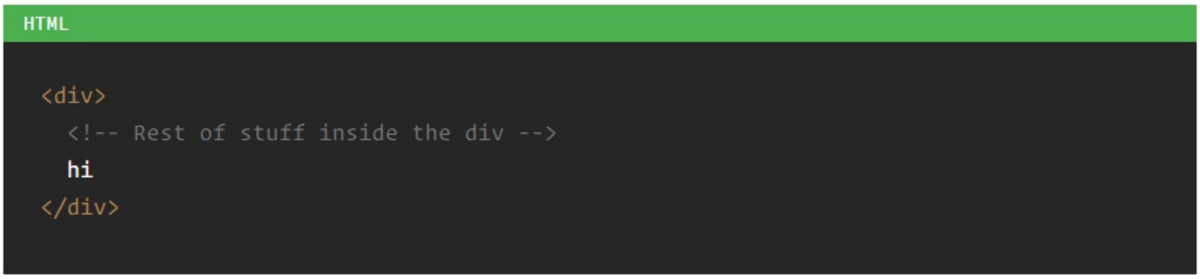
- Psuedo element `::after`
- Psuedo element `::before`

What is it?

Pseudo element allows you to insert content onto a page from CSS (without it needing to be in the HTML). While the end result is not actually in the DOM, it appears on the page as if it is, and would essentially be like this:



```css
div::after {
  content: "hi";
}
```



```html
<div>
  <!-- Rest of stuff inside the div -->
  hi
</div>
```

The capability of psuedo elements is tremendous. Check out the Bookmarks

Remind of `flex-box` CSS

Flexbox is the secret behind Bootstrap's grid system, help you build neat and responsive layout Checkout the tutorial in the Bookmarks

Bookmarks:

Link to GitHub Project

Psuedo Elements CSS

Flexbox CSS

Enhancements

Overview:

Make website more responsive

Apply some effect to increase/// smoothness

Details:

Enhancements

Test the website with every screen size and make sure everything looks nice

---

Apply Scrollspy

Apply Smooth Scrolling: Copy the code from this link and paste it into `main.js`

/ Apply Scroll Reveal

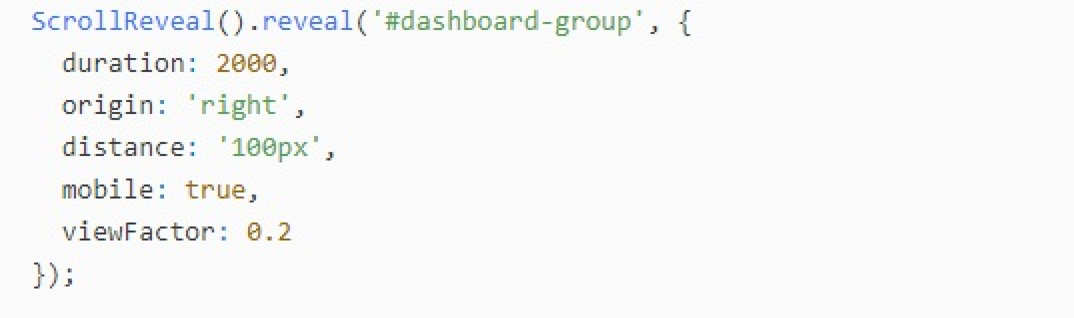
What is Scroll Reveal ?

ScrollReveal is a JavaScript library for easily animating elements as they enter/leave the viewport.

It was designed to be robust and flexible, but hopefully you’ll be surprised below at how easy it is to pick up.

Steps:

1. Go to https://github.com/scrollreveal/scrollreveal to download the library
2. Paste the srollreveal.min.js into our js folder
3. Initialize effect on an element by using this syntax:



```
ScrollReveal().reveal('#dashboard-group', {
  duration: 2000,
  origin: 'right',
  distance: '100px',
  mobile: true,
  viewFactor: 0.2
});
```

4. Checkout all options of the library through the link in Bookmarks section

Note

If using jQuery, check every jQuery effect with browser reload.

Bookmarks:

Link to GitHub Project

Scroll Reveal Doc

Testing

Overview:

- See what can we improve

Details:

Compressing images

Upload images to Optimizilla to minify images

Cross-browser testing

Open website on others browser to make sure it works everywhere

---

Put website online

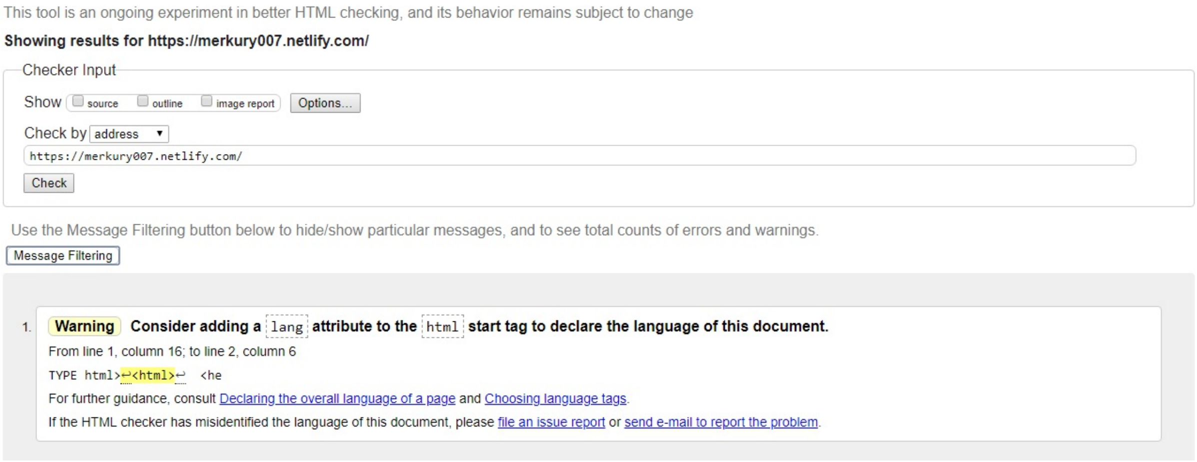
The same as we did with OpenCharity website: Go to https://app.netlify.com/sites/tender-banach-172fcd/overview to deploy your website to Netlify

Link to deployed website

Markup Validation Service

Go to W3C Markup Validation Service to check for error in your markup files

Only one warning with lang property of `html` tag



This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for https://merkury007.netlify.com/

Checker Input

Show □ source □ outline □ image report

Check by address

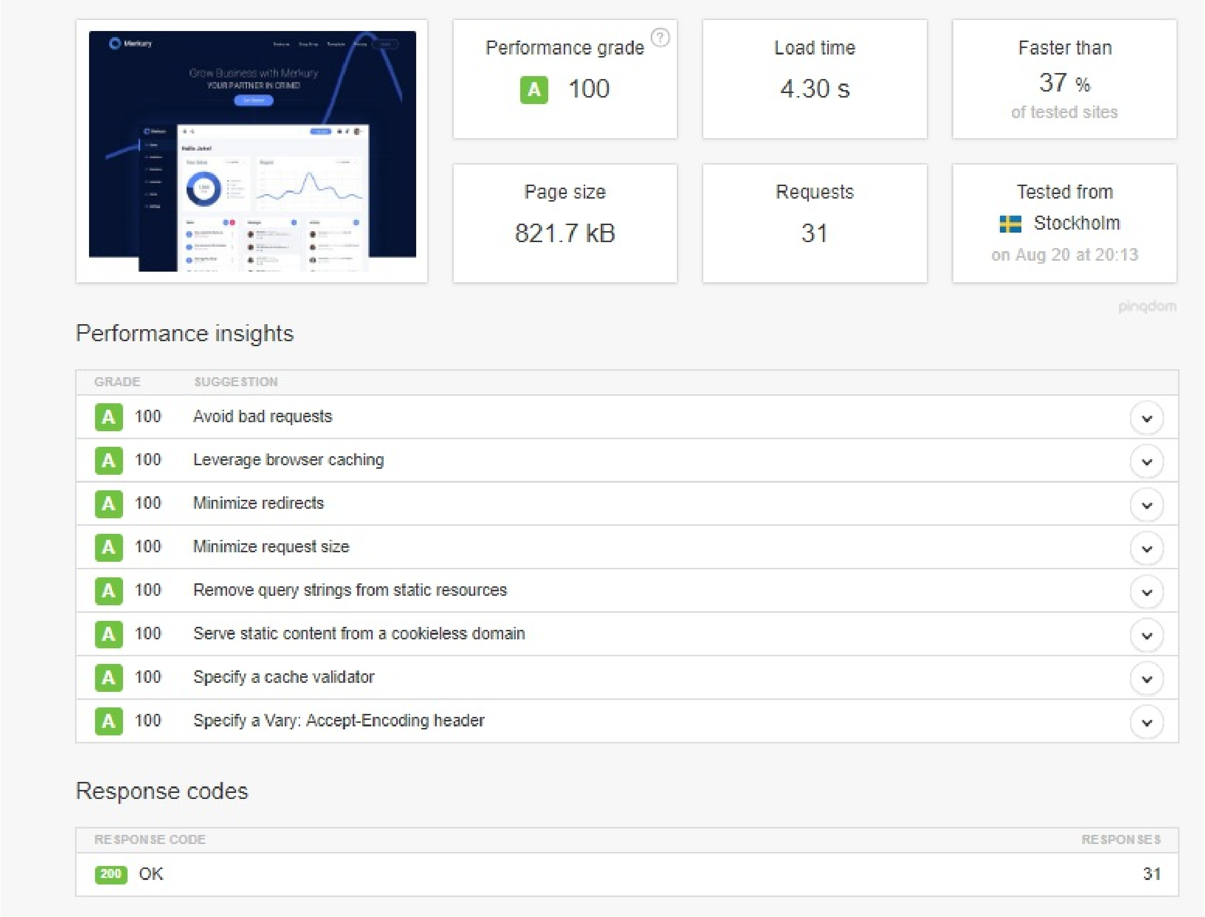
https://merkury007.netlify.com/

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

1. **Warning Consider adding a `lang` attribute to the `html` start tag to declare the language of this document.**
From line 1, column 16; to line 2, column 6
`TYPE html>↩<html>↩  <he`
For further guidance, consult Declaring the overall language of a page and Choosing language tags.
If the HTML checker has misidentified the language of this document, please file an issue report or send e-mail to report the problem.

Website speed test

Test result from Pingdom



Performance grade	Load time	Faster than
A 100	4.30 s	37 % of tested sites

Page size	Requests	Tested from
821.7 kB	31	Stockholm on Aug 20 at 20:13

pingdom

Performance insights

GRADE		SUGGESTION
A	100	Avoid bad requests
A	100	Leverage browser caching
A	100	Minimize redirects
A	100	Minimize request size
A	100	Remove query strings from static resources
A	100	Serve static content from a cookieless domain
A	100	Specify a cache validator
A	100	Specify a Vary: Accept-Encoding header

Response codes

RESPONSE CODE	RESPONSES
200 OK	31

Note

Bookmarks:

Link to GitHub Project

Recap

Overview:

- Revise what we have learnt so far

Details:

Learned

- How to export images, getting colors from PSD files
- Prepare project files and folders using Trello to-do list
- Setting variables and custom stylesheet for our website
- How to nest CSS code in less for better understanding and readability
- How to use Bootstrap 4 and its components
- Create animations using CSS, jQuery or normal JS
- Dealt with two different plugins jQueryUI and ScrollReveal and know how to customize them
- Write clean code with comments
- Dealt with website responsiveness
- Increase website performance by compressing images, running markup test and speed test to know what to improve

Fallbacks

- Still need a lot more practices on CSS, CSS animations, jQuery, jQueryUI cuz what the course brought are just the tip of the iceberg.

Bookmarks:

Link to GitHub Project

Important Tips

Overview:

- Share some important tips and advices

Details:

The best three directions for you as a front-end developer/designer\

1. Start learning about CMSs like WordPress or Drupal...
2. Start learning a backend language like PHP with Laravel framework or Python with Django framework or so on.
3. Continue improving your JavaScript skills by learning a single page app framework like AngularJS, or ReactJS or VueJS

You don't need to know everything to be productive, just be confident

COURSE FINISHED: 20:44 PM 20/08/2018