1 UI TRAINING MODULE

Five Reasons to choose UI development as a career?

1. UI developers are responsible for the entire visual interface. This includes everything from putting together style guides that detail how everything from icons to menus should look.
2. You'll get a chance to design a responsive page which supports different mobile devices, tabs and desktop.
3. You'll make decisions on everything from choosing the best typeface to creating button styles - and you'll likely have to convince others about your choices.
4. HTML5, CSS3 (Flexbox) and Bootstrap are the key languages needed for UI development. It's easy to understand and its base for learning other UI technologies which is most popular now like ReactJS.
5. It's hard to find a good UI Engineer and always a demand in the market.

What are the important technologies that I must learn if I want to become a UI developer??

- **HTML5**
- **CSS3 (Flexbox and Grid)**
- **SASS**
- **Bootstrap 4 latest (Grid System and Responsive)**

2 HTML5

HTML stands for Hypertext markup language, and it is the most widely used language to write Web Pages.

2.1 Learning Objectives

- Getting started with HTML
  - Topics covered
    - Elements, Nesting Elements, Block versus inline elements, empty elements.
    - Attributes, Boolean attributes, single or double quotes.
- What's in the head? Metadata in HTM
  - Topics covered
    - Title, Metadata, character encoding, description and SEO
    - Custom icons, adding CSS and JavaScript to the page, document language
- HTML text fundamentals
  - Topics covered
    - Structure in web pages. Headings and Paragraphs
    - Lists: Unordered, Ordered, (they don't cover description lists), nested lists.
- Creating hyperlinks
  - Topics covered
    - Anatomy of a link, the title attribute, block level links.
    - URLs and paths, block fragments and ids, absolute and relative URLs.
- Advanced Text Formatting
  - Topics covered
    - Description lists, Blockquotes, Inline quotations
    - Citations, Abbreviations, Address, Super/sub-script
- Document and website structure
  - Topics covered
    - Basic sections of a web page. Structuring a page.
    - <main>, <article>, <section>, <aside>, <header>, <nav>, <footer>.
    - Non-semantic wrappers: , .
- HTML Forms
  - Topics covered
    - Your first form. Basics of the <form> element, action, and submit button
    - How to structure a form. How to use *field sets*, *legends*, *labels*, and such.
    - The native form widgets. Includes: all sorts of text input fields, select boxes, checkable items, buttons, and more. Use this more as a reference. Did you notice that things like tabbed panes, dialog boxes, drop down menus are missing?
    - Sending form data

- Multimedia and Embedding
    - Topics covered
        - Images in HTML
        - Video and audio content.
        - Other embedding techniques.
        - Adding vector graphics.
        - Responsive images

2.2 Learning Resources

(Resource link embedded in the learning objectives in 2.1)

Other Resources

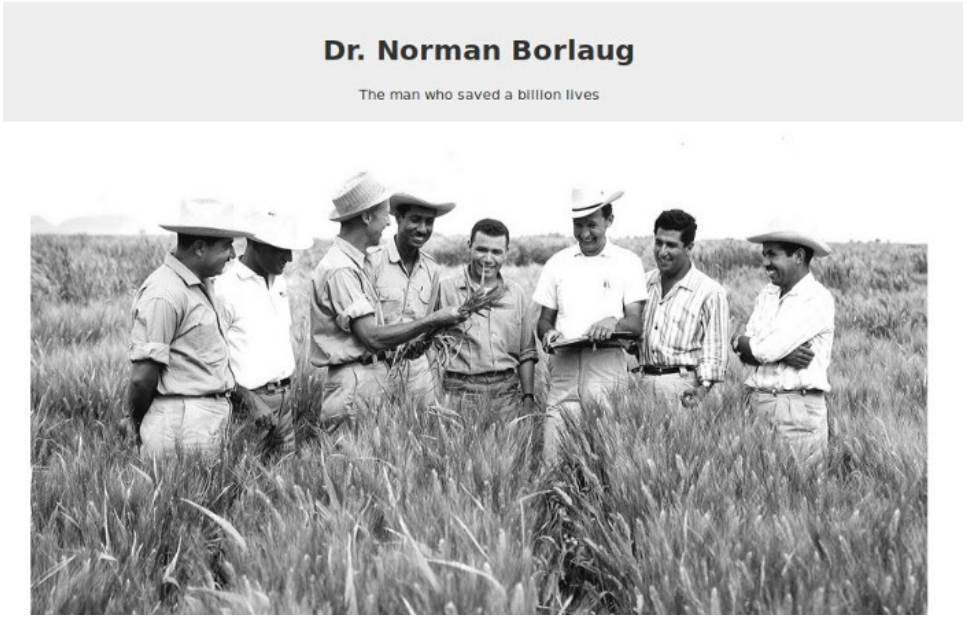
1) https://learn.shayhowe.com/
2) HTML Crash Course for Absolute Beginners 2020 [Tutorial]

2.3 Reference

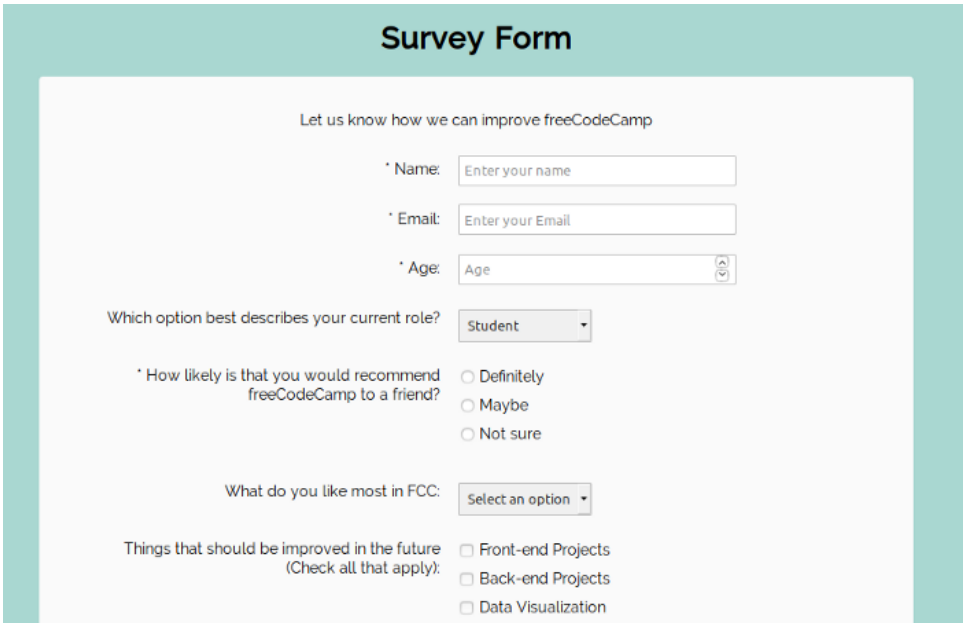
1) Web technology for developers - MDN by Mozilla
2) HTML Crash Course for Absolute Beginners 2020 [Tutorial]

2.4 Assignment

1) Create a webpage like the below screenshot and run it on the browser. Check the browser compatibility and responsive.
   (Ignore background colour)



2) Create a webpage like the below screenshot using survey form. Check the browser compatibility and responsive.
(Ignore background colour)



3 CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

3.1 Learning Objectives

- The basics about selectors, combinators and how you set up styling rules
- Flexbox How it works and how to use it
- CSS Grid and how it differs from Flexbox
- Properties, values and declarations
- Styling backgrounds (e.g. gradients) and images
- Units and dimensions, you typically use in CSS (px, rem, % and more)
- How JavaScript and CSS interact
- Styling forms and form inputs
- Working with text, fonts and text styles
- Concept of Box Model

3.2 Learning Resources

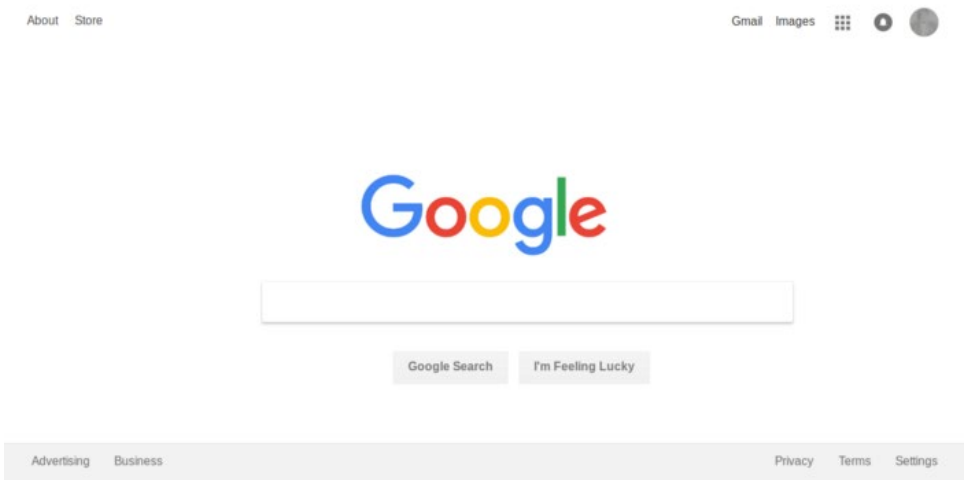
1) CSS Tutorial - Zero to Hero (Complete Course) (Mandatory)
2) Flexbox: How it works and how to use it (Mandatory)
3) CSS Crash Course for Absolute Beginners - Full Course (Optional)

3.3 References

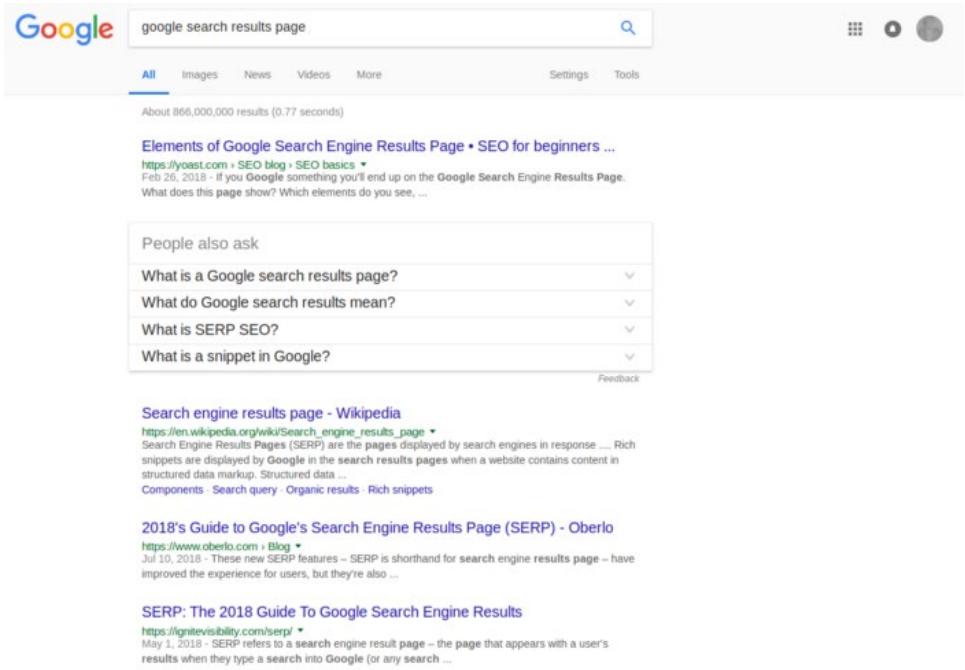
1) https://css-tricks.com/snippets/css/a-guide-to-flexbox/ (Mandatory)
2) https://www.freecodecamp.org/news/flexbox-the-ultimate-css-flex-cheatsheet/
3) https://www.freecodecamp.org/news/css-responsive-image-tutorial/ (responsive images)
4) https://lawsofux.com/

3.4 Assignment

1) Create a Google Home web page like the below screenshot using HTML, CSS(Flexbox). Please check the browser compatibility and responsive.



2) Create a web page like the below screenshot using HTML and CSS(Flexbox). Please check the browser compatibility and responsive.
Note: Add collapse using JavaScript



4 SASS

CSS on its own can be fun, but stylesheets are getting larger, more complex, and harder to maintain. This is where a preprocessor can help. Sass lets you use features that don't exist in CSS yet like variables, nesting, mixins, inheritance and other nifty goodies that make writing CSS fun again.

mixins allow you to define styles that can be re-used throughout your stylesheet. They make it easy to avoid using non-semantic classes like .float-left, and to distribute collections of styles in libraries.

4.1 LEARNING OBJECTIVES

- Variables
- Nesting
- Mixins
- Inheritance

4.2 RESOURCES

1) https://sass-lang.com/guide
2) Sass Tutorial for Beginners - CSS With Superpowers

4.3 ASSIGNMENT

1) Create a multiple web pages using CSS and SASS and check the difference.

5 BOOTSTRAP 4

Bootstrap is versatile, simple and very flexible to use. A good knowledge and combination of bootstrap and CSS3 will lead to a very smooth and responsive user interface both to the developers during development stage and the end users when it is viewed live.

5.1 LEARNING OBJECTIVES

- **Responsive Systems**: Bootstrap is very responsive if utilized properly. It utilizes two responsive patterns and alternates them on its various components depending on which serves better.
    - Grid system: Bootstrap utilizes mobile-first flexbox grid and the twelve column's width system to build layouts of all shapes and sizes. Its responsiveness comes in five tiers or classes - -sm -md etc.
    - Flex system: Bootstrap uses flex system to quickly manage the layout, alignment, and sizing of grid columns, navigation, components.
- **Responsive Breakpoints**: Bootstrap includes five set of predefined classes for building complex responsive layouts. The classes cover for extra small, small, medium, large, or extra-large screen widths.
  Here is a handy table containing all responsive breakpoints and the respective classes.

	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				

- **Components:** Bootstrap has a lot of handy components, from which you can choose and easily integrate or modify to suit your needs. Most of these components come with default breakpoints to cater for all screen widths. This can go a long way to help in fast prototyping; a navbar here, a jumbotron there, some forms and containers here and there and you are good to go.
  The best part is that some components have their JavaScript behavior incorporated into them already, so you do not need to worry about trying to write the functionalities for it.
  Some of these components are.
    - list groups
    - Modals
    - Dropdowns
    - Alerts
    - carousel etc

5.2 RESOURCES

Complete Course

Alternate Course (Optional)

Grid System

List groups

Modals

Dropdowns

Alerts

Carousel

5.3 REFERENCES

https://getbootstrap.com/docs/4.3/layout/grid/ (Grid system)

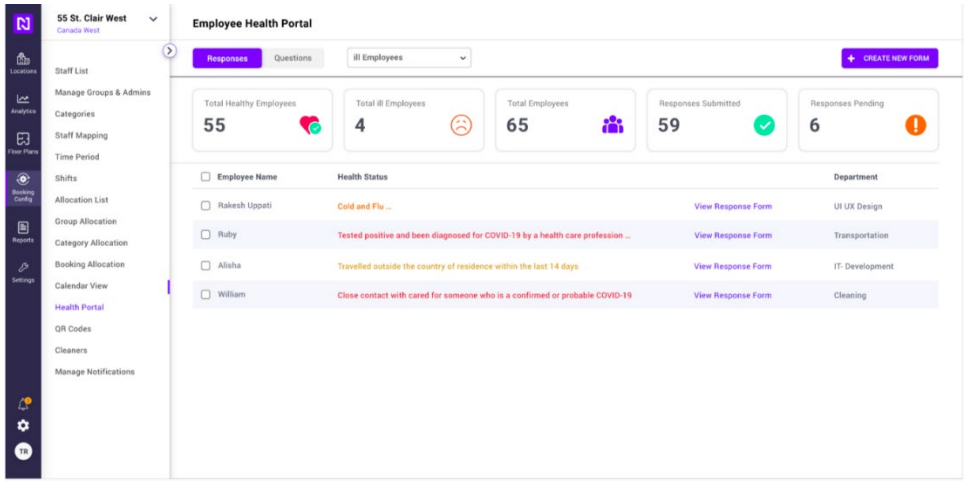
https://www.freecodecamp.org/news/learn-bootstrap-4-in-30-minute-by-building-a-landing-page-website-guide-for-beginners-f64e03833f33/

https://medium.com/wdstack/how-to-bootstrap-94abe3525442

5.4 ASSIGNMENT

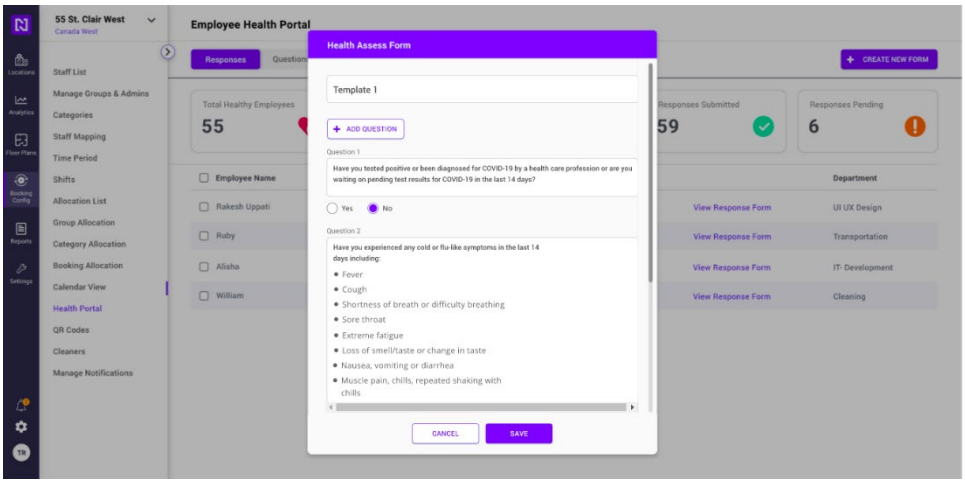
1) Develop the Employee Health Portal web page using **HTML5, CSS3 (flexbox), SASS, Bootstrap and JavaScript** for Modal (basic).

   Screen 1:



   Screen 2:

   - Below screen is visible when user click on Create New Form button and it should behave like Modal.
   - Please use Font awesome icons.



Kindly learn basic JavaScript also because it helps a lot while creating a web pages like Modal, Navbar etc.

All the best! Practice more...