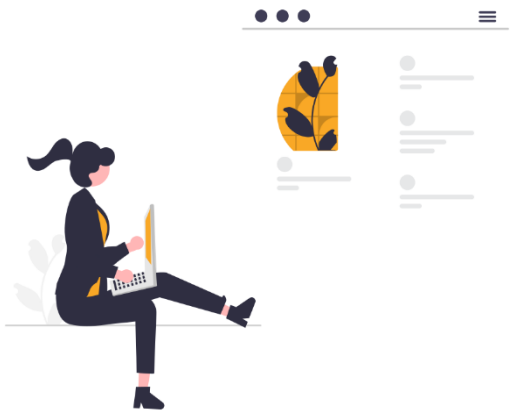


BYTEWISE FELLOWSHIP

Documentation of Progress till end of the fellowship

TRACK BYTEWISE FELLOWSHIP
Full Stack Frontend Web Development

INTERNEE'S NAME
Emaan Riaz

Task 1

Basic languages to learn in order to become a front end developer
Fundamental languages for web development

Html

To structure or webpages

CSS

To make them beautiful.

JavaScript

To program them

Analogy

A building has a skeleton, walls, etc = HTML

Different tiles are used in the building= CSS

When you add functionality to something like elevator in building = JavaScript

Another example

In developing a website like Twitter first we have to define the building blocks of layout using the HTML .e.g., an image, Twitter handle name, text to be posted , like, retweet and comment icons.

You can now give Aesthetics and visual effects to these above mentioned elements by using CSS so CSS is used for giving aesthetics and visual effects.

You can add functionality to these elements by using JavaScript.

So web development is done by three basic and fundamental languages:

- HTML
- CSS
- JavaScript

Front-end frameworks

- Front end frameworks and libraries are used when you have to use repetitive pieces of code in your webpage.
- They have a bunch of predefined codes that help to work faster.
- Now big organizations use frameworks and libraries.

Popular front end frameworks

1. React (not a framework)
2. Angular
3. Vue

Difference between framework and library

1. Framework forces our applications into structure.
2. Libraries don't force our application into structure(they provide reusable code).

React

- It was developed by Facebook.
- React is most popular because it is used to build Facebook and Instagram.

Version control systems

Track our project history.

Work collaboratively with others.

Most popular used version control system is git (used in 70% software development teams).

CSS preprocessors

When the code of CSS becomes excessive then we are unable to understand it so then we use new versions of CSS like CSS3 but these have to be supported by our browsers and it's a slow process so then comes the CSS preprocessors(it is used to generate css code that is more capable in language other than CSS)

Future CSS to preprocessor to today CSS

Names of some mostly used preprocessors:

- SASS(most popular preprocessor)
- LESS
- Stylus

Preprocessors in JavaScript

Typescript/Coffeescript scriptJavaScript(most widely used) to Transpiler(translate+compile) to JavaScript

Learning path

3 months (HTML CSS JavaScript)
1-2 month(React front end library or framework)
2 weeks (git version control system)

Optional but better to learn to become outstanding among crowd of people seeking jobs

1-2 weeks(SASS)
2-4 weeks(Typescript)
In 6 months you can apply for junior front end developer.

How the web works

Server consists of the codes.
Browsers act as an interpreter when we search something. The source code will come back from the server to our browser or the source code is generated by the server.

How does the server knows what we are entering?

IP Addresses are assigned to each address and DNS (Domain name system) is used to translate the address to the respective IP address. The address we write in the browser goes to the DNS and it translates it and then goes to the server in the form of an IP address the server then sends the source code to the browser where it is displayed.

Request

The address we write in the browser is the request(it contains some headers also which can be seen by developer tools on Google and also HTML code is present in every web page which we search that can also be seen as well in Google developer tools.

Hypertext markup language(HTML used for the structure of webpage)
CSS (cascading style sheet used for styling the webpage)
JavaScript(used for making web page dynamic or adding functionality to web page after it is opened)

HTTPS

- All the data is standardized.
- How the request looks like and it's end to end encrypted that's what the https do.
- **Nodejs, PHP and python frameworks** are used to generate the web page dynamically on the server.
  - Frameworks are packages with predefined codes and rules which can be applied to our webpage.
  - Parsing logic is also defined by the frameworks.
  - Frameworks are hidden when we are viewing the webpage.

The Web is more than a website.
Instagram and Facebook have their own way of parsing the logic.
WebSockets are the advanced technology used for real-time response.

Task 2

HTML

Today we are learning HTML5 basic tags.
HTML is not a programming language. Markup language creating web pages/documents.
Building blocks of the web.

What we need to start

- Web browser(google chrome mostly)
- Text editor(vs code and sublime mostly)

Creating a file

Files should end with .html extension and index.html file is the root/home page of a website.
Don't need a server.

Tags syntax

Tags are surrounded with angular brackets. The text is in between the tags. Each tag ends with angular brackets having the forward slash. The tag that ends itself automatically is called the remnant of XHTML.

Page structure

```
<! DOCTYPE html>
<html>
<head>
<title>
Document name
</title>
</head>
<body>
<h1>this is heading </h1>
<p>this is paragraph </p>
</body>
</html>
```

Headings

Heading are of six sizes:

```
<h1>this is heading one </h1>
<h2>this is heading two </h2>
<h3>this is heading three </h3>
<h4>this is heading four </h4>
<h5>this is heading five </h5>
<h6>this is heading six </h6>
```

Paragraph

```
<p>this is paragraph </p>
```

Google developer tools you can see the by default margin, padding, border and other things in user agent stylesheet.

Inline level and block level tags/elements

Don't take full width and don't start on the next line like (span, img, a) while the block level needs full width and start on the next line(div,h1, h6, p, form).

Tag attributes

```
<tagname attributename="value"></tagname>
```

Lists

Ordered list and unordered list
For ordered
For unordered

HTML5 semantic tags

Describe the meaning to the browser and the developer.

```
<header></header>
<footer></footer>
<aside></aside>
<main></main>
<article></article>
```

<nav></nav>
<section></section>
<details></details>
Basic tags
<table>
<thead>
<tr>
<th>
</th>
</tr>
</thead>
<tbody>
</tbody>
</table>
<form></form>

<hr>
<blockquote> </blockquote>
<cite></cite>
<button></button>

Task 3

Html5 semantic tags
In the early versions of html the code was not that meaningful for the browser. But now with HTML5 Semantic tags it has become possible for the browser to find the meaningful code. Semantic tags are a bit more descriptive.
Semantic Tags:
<main>
<section>
<article>
<aside>
<header>
<footer>

Task 4

HTML media tags
Today I learned about how to embed Audio, Video ,YouTube videos on your webpage by using media tags.
Media tags
<audio></audio>
<video></video>
<iframe></iframe>

Task 5

Fundamentals of CSS
CSS is used for design so it is much more artistic.
Cascading styles sheet used for designing.

Not a programming language.
Used alongside html.
Can be extended with SASS/LESS.

A web browser

Google Chrome

A text editor

Sublime or vs code preferably

3 methods of CSS

Inline CSS(it is written in the same line with the html code)

Internal CSS(<style> tag is used in the head of the document)

External CSS(styles.css file is created in the same folder in which the html file is present and the file is linked by using <link> tag in the head of the document).

CSS selectors

Syntax

```
a{
background-color:white;
}
```

Colors in CSS

- Rgb(red green blue)

Example:

```
color :rgb(0, 0,12);
```

- Hexadecimal

Example:

```
color:#000000;
```

- Html5 color names

Example:

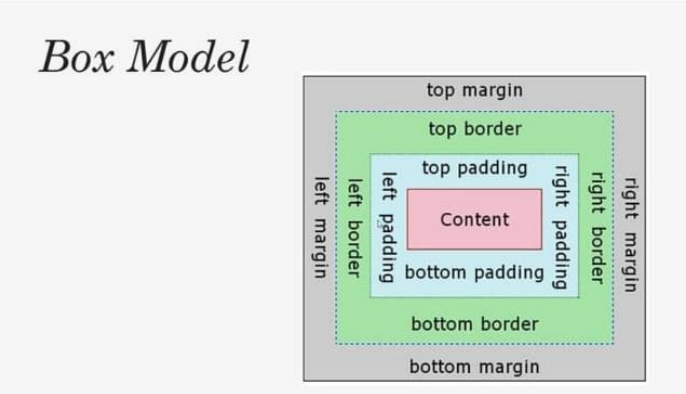
```
color: white;
```

Web safe fonts

- Serif
- Sans serif
- Arial, etc

em is used for more responsive size of the elements in CSS.

line-height:1.6em;



Positioning in CSS

1. Static it's by default.
2. Relative is a natural position but we can write top, left, right, bottom positions for it.
3. Absolute allows us to target any position inside the relative element.
4. Fixed makes the position of the element fixed.

5. Initial sets the value to its default.
6. Inherit will inherit the properties of its parents.

Task 6 and 7

Today I've learned about CSS grid

Syntax:

display:grid;

Properties:

- grid-template-rows
- grid-template-columns
- grid-row-start similarly grid-row-end
- grid-row:value/value;
- grid-column-start similarly grid-column-end
- grid-column:value/value;
- span keyword
- grid-row:

Syntax:

- grid-row: span 5;
- grid-area
- grid-auto-rows
- grid-implicit-flow
- fr is used for the fraction.
- We can apply minmax value to the elements such as (minmax (100px, 3fr)).
- grid-template-rows: repeat( ) ; it will repeat the value.
- grid-gap

Syntax:

- grid-gap: 1em 3em;
- grid-template-areas

Syntax:

- grid-template-areas: 'header'
- 'main aside'
- 'footer';
- justify-items and align items
- justify-self and align-self properties
- justify-content(align the grid around row axis) and align-content (align the grid around
- the vertical axis) and have the values like
- space-between
- space-around
- space-evenly, etc.
- auto-fit keyword

Best Practices

Best practice is to first understand these following things:

1. Box model.
2. Firefox instead of Google Chrome dev tool.
3. Use flexbox
4. Use grid
5. Clamp it down.
6. Variables for variables.

7. Fancy calculations.
8. State management counter.
9. Finding focus within.

Today I've learned about what is:

CSS float

float:left or right;
clear:left or right both;

CSS positioning

Values of CSS positioning absolute, relative, fixed, sticky, static.

Flexbox

display:flex;
flex-direction:column(will make a vertical axis)or it can be row(will make a horizontal axis) ;
And properties like justify content property which contains the following values:

- baseline
- center
- flex-end
- flex-start
- last baseline
- space-between
- space-around
- space-evenly
- left
- right
- end
- align-items which contains the following values:
- baseline
- center
- flex-end
- flex-start
- last baseline
- self-start
- self-end
- end
- flex-wrap
- align-content
- space-between
- space-around
- space-evenly
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self
- Order

Task 8

Today I read CSS animation, CSS transition, CSS transform 2D and 3D some important points are as follows:

CSS Animation

- @keyframes.
- animation.
- animation-delay.
- animation-direction.
- animation-duration.
- animation-fill-mode.
- animation-iteration-count.
- animation-name.
- animation-play-state.
- animation-timing-function.

CSS 2D Transform

- transform.
- transform-origin.

CSS 2D Transform Methods

- matrix(n,n,n,n,n,n).
- translate(x,y).
- translateX(n).
- translateY(n).
- scale(x,y).
- scaleX(n).
- scaleY(n).
- rotate(angle).
- skew(x-angle,y-angle).
- skewX(angle).
- skewY(angle).

CSS 3D Transform

- transform.
- transform-origin.
- transform-style.
- perspective.
- perspective-origin.
- backface-visibility.

CSS 3D Transform Methods

- matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)
- translate3d(x,y,z)
- translateX(x)
- translateY(y)
- translateZ(z)
- scale3d(x,y,z)
- scaleX(x)
- scaleY(y)
- scaleZ(z)
- rotate3d(x,y,z,angle)
- rotateX(angle)
- rotateY(angle)
- rotateZ(angle)
- perspective(n)

CSS transition

Attributes in transition:

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

Task 9

- I realized the task then I created the project initials.
- I created the (blog website) project header and the hero section.
- I created the topic section, featured section and recent post section.
- Then I created the popular tags section, footer and applied the media queries.
- I used JavaScript,HTML and CSS for my blog website project.

Project header and hero section

- Firstly I realized the task then I created the project initials.
- Secondly today I created the (blog website) project header and the hero section.
- I also used js today to add functionality to the website which I've created till now.
- I downloaded the zip file and extracted it, then used the elements required for initial settlement of html and css files like I added the js files, google fonts, css file.
- Moreover I also added mobile view extension to vs code.

Project Topics section, feature section and popular tags section

- Firstly I realized the task then I created the project initials.
- Secondly I created the (blog website) project header and the hero section.
- Today I created my blog website project's topic, feature and popular tags section using html, css and JavaScript.
- I also updated my github repository with the progress I've made till now.

Task 10

Task name

Adding code files to github repository.

Task progress

I'm making a project on a blog website and till now I've made the header and project initials.

Task description

- What is git and github?
- Creating a github profile.
- Making a github repository and adding code files of task 9 to it.

Task 10 next half

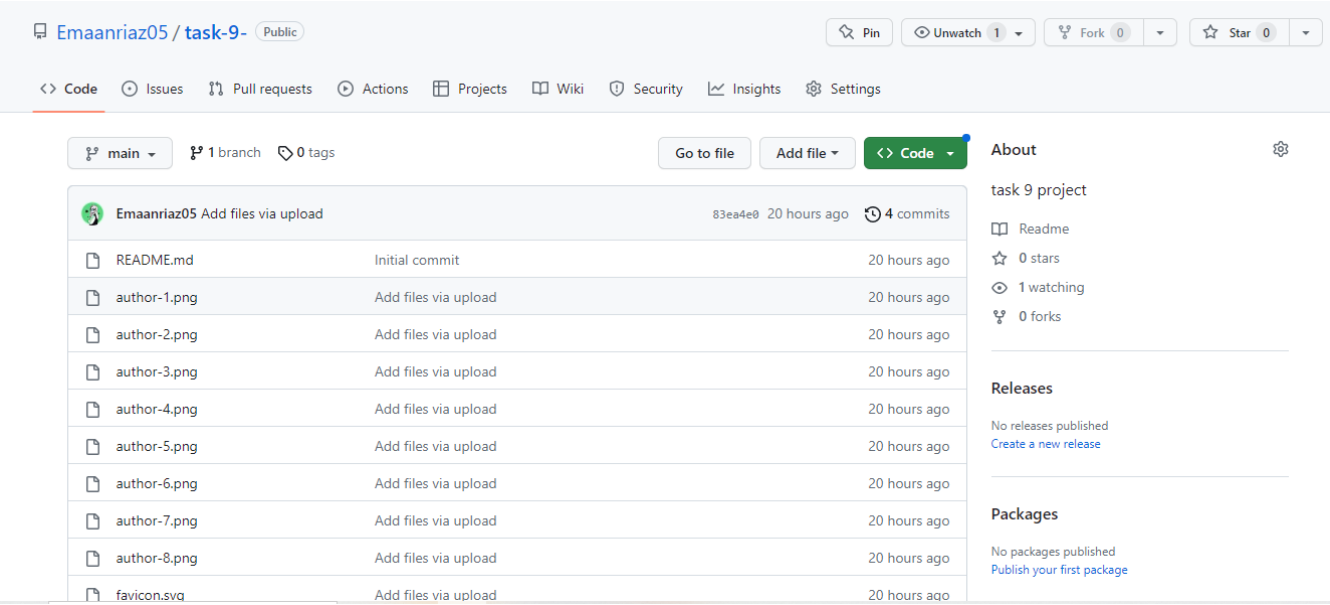
Task description

Task was to deploy my personal blog website using netlify and github live pages.

Progress

I've completed my personal blog website project.

Commit screenshot



Netlify link

https://scintillating-choux-8172f8.netlify.app/

Github live pages link

https://emaanriaz05.github.io/task-9-/

Github repository link

https://github.com/Emaanriaz05/task-9-/tree/master

Task 11

Design and UI resources

Sometimes the most experienced developers also find difficulty in finding the right resources so here are some resources for UI and design.

The following are some categories of resources:

- UI graphics
- Fonts
- Stock photos
- Stock videos
- Vectors
- Smart Mockups
- Html CSS templates
- Css frameworks like tailwind, bootstrap..,etc
- Css components
- Modern UI kit
- React desktop
- ViewViews
- Design systems
- Etrade
- Pluralsight
- Online Design Tools
- Wireframe
- Logo Maker
- Site Map builders
- Cartoon Photo

What is framework and why use framework?

● The predefined skeleton in which you perform coding is called a framework.
● It is a combination of two words frame and work.
● Like a house has a structure which is built afterwards from inside, that structure is called the frame or like a photo frame in which we put a photo.
● A framework is an arrangement in which software provides greater functionality that can be extended by additional users in written codes.
● Standard way of creating applications.
● Reusable functions.
● Simplify the process of creating web pages

Why do we use framework?

● Not start from scratch.
● Hiding startup things.
● Just a way of processing requests.
● It has an MVC(model, view, controller) pattern.
● Following a new Pattern.
● Attaching third party packages and resources.
● Extendability.

What is Bootstrap?

● Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs.
● As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a predefined grid system. The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS) and JavaScript. Web developers using Bootstrap can build websites much faster without spending time worrying about basic commands and functions.

Why Use Bootstrap?

- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

How do I use Bootstrap?

There are two ways to start using Bootstrap on a website.
You can:
● Download Bootstrap from getbootstrap.com
● Include Bootstrap from a CDN

Task 12 & 13

Task description

Task was to deploy my bootstrap bootcamp website using netlify and github live pages.

Progress

I've completed my bootstrap bootcamp project.

Github live pages link

https://emaanriaz05.github.io/Frontend-Dev-Bootcamp-Website/

Netlify link

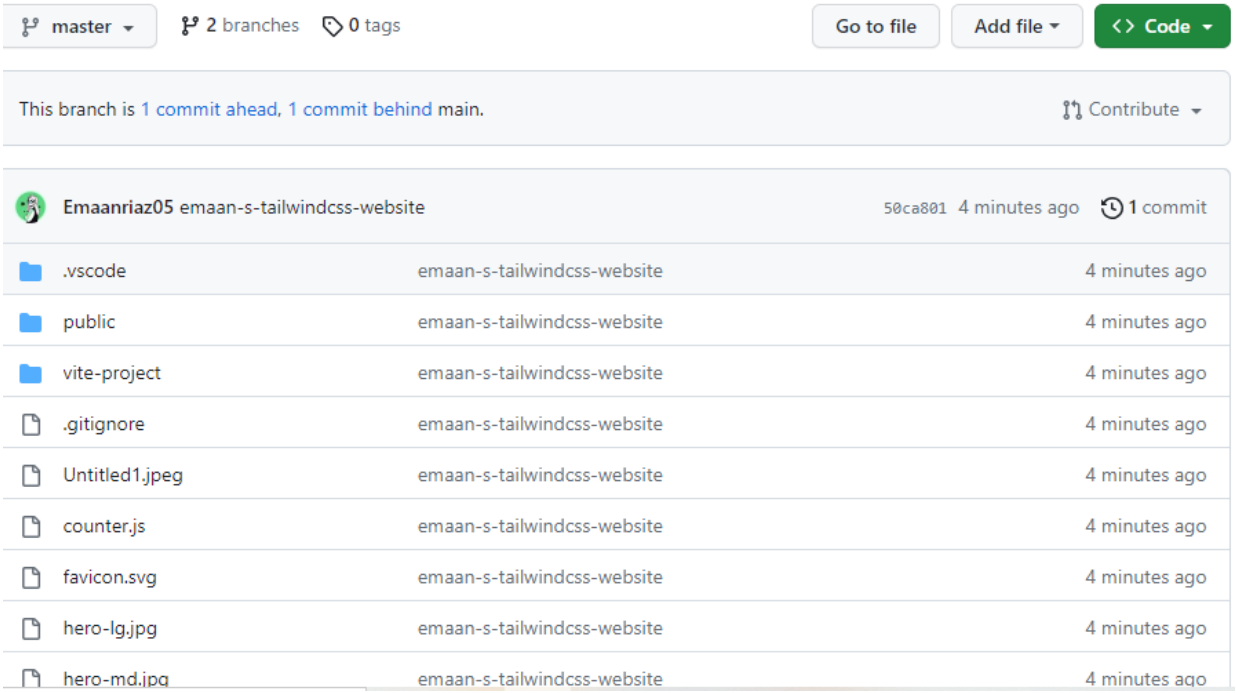
https://emma-frontend-bootcamp-website.netlify.app/

Github repository link

https://github.com/Emaanriaz05/Frontend-Dev-Bootcamp-Website/tree/master

Task 14

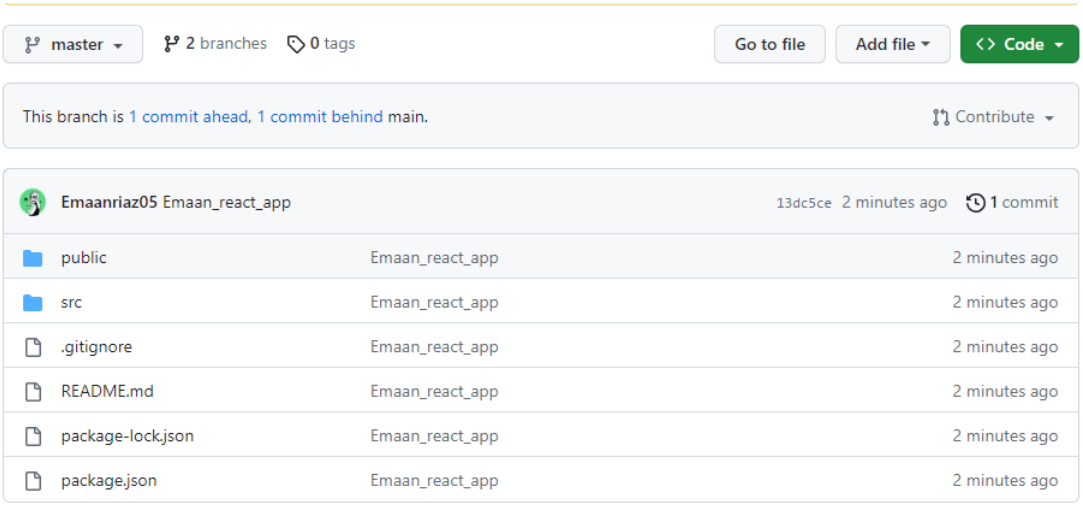
Task description
Task was to deploy tailwind css website.
Progress
I've completed my tailwind css project.
Github repository link
https://github.com/Emaanriaz05/emaan-s-tailwindcss-website
Commit screenshot



Netlify link
https://emaan-tailwindcss-website.netlify.app/
Github repository link
https://github.com/Emaanriaz05/emaan-s-tailwindcss-website/tree/master

Task 18 (Final Project)

Task description
Task was to learn about react and developing a project based on react.
Progress
I've completed my react-movie-app project.
Github repository link
https://github.com/Emaanriaz05/Emaan_react_app/tree/master
Commit screenshot



Netlify link

https://emaan-react-movie-app.netlify.app/