



## **BYTEWISE FELLOWSHIP**

**Documentation of Progress till end of the Fellowship**

### **FRONTEND**

### **WEB-DEVELOPMENT:**

### **INTERNEE's NAME**

Aneeq Khan Sherwani

## **Task 1**

**Today I learned most important fundamentals of front-end-developers. That is HTML,CSS,and javaScript.**

**HTML:** Stands for hyper text markup language and it is used to structure web pages.

**CSS:** stands for cascading styling sheet and is used to make them beautiful.

**JavaScript:**it is used 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

If we learn all these fundamentals then web development is done by these fundamentals.

### **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.

## **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.

**It works through combination of technologies, including:**

**Hypertext Transfer Protocol (HTTP)** - a protocol that allows web browsers and servers to communicate with each other.

**Uniform Resource Locators (URLs)** - addresses that identify the location of web pages and other resources on the internet.

**Hyperlinks** - clickable links that allow users to navigate between web pages and other resources.

**Web browsers** - software applications that allow users to access and view web pages.

When a user enters a URL into a web browser, the browser sends an HTTP request to the web server hosting the requested web page. The server then responds with the requested web page, which the browser renders and displays to the user.

The web also includes other technologies such as HTML, CSS, and JavaScript, which are used to create and design web pages, and databases and server-side scripting languages, which are used to generate dynamic content.

## **Task 2**

Today I learned about HTML5.

### **What is HTML?**

HTML stands for Hyper Text Markup LaNguage.It is not a programming language.It is used for creating web pages/Document and building block of web. **We use elements in HTML. An HTML element is defined by a start tag, some content, and an end tag.**

### **How we can create an HML file?**

Files must end with .html extension. It is run in a web broswer.Index.html is a root OR home of a web page. We does not need a server.

### **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 vs Block level elements:**

**Inline Elements:** these elements do not start on a new line and take a necessary width.(<br>,<a>,<img>,<span>)

**Block Elements:** these elements start on a new line and take full width available.(<p>,<h6>,<div>,<form>)

### Tag attributes

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

### Lists

Ordered list and unordered list

For ordered <ol></ol>

For unordered <ul></ul>

### 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>  
<br>  
<hr>  
<blockquote> </blockquote>  
<cite></cite>  
<button></button>  
<div></div>

## **Task 3**

### **HTML5 semantic tags**

Today I learned about semantic element in html.

#### **Semantic Elements**

In the early versions of html the code was not that meaningful for the browser. But now HTML5 Semantic element clearly describes its meaning to both the browser and the developer. In HTML5, There are many semantic elements:

<article>,<aside>,<details>,<figcaption>,<figure>,<footer>,<header>,<main>,<mark>,<nav>,<section>,<summary>,<time>

<main>: For the main content of a webpage, unique to that page.

<section>: Defines a certain section of a webpage.

<article>: Defines a bit of content which makes up an article.

<aside>: Defines some content related to something else.

<header>: For the header of a website.

<footer>: For the footer of a website.

## **Task 4**

**Today I learned about media elements in HTML.**

**Media Elements:** on the web is sound, music, videos, movies, and animations. There are some media elements (<video>,<audio>).

**<video>:**element is used to show a video on a web page. The easiest way to play videos in HTML is to use YOUTUBE.(Playing a YT video in HTML)

- 1.GO to the YT video, which you have to use
- 2.Right click on this video and select “embed code”(This video ‘s embed code is copied)
- 3.Paste this code on your web page.

OR

- 1.Upload the video to YouTube
- 2.Take a note of the video id
- 3.Define an <iframe> element in your web page
- 4.Let the src attribute point to the video URL
- 5.Use the width and height attributes to specify the dimension of the player
- 6.Add any other parameters to the URL

**<audio>:**element is used to play a sound on a web page.

## **Task 5**

**Today I learned about CSS and fundamental of CSS**

**What is CSS?**

- CSS stands for Cascading Styling Sheet. It is not a programming language.
- It is used to style HTML Doc. CSS describes how HTML elements should be displayed .
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files.

**There are three ways to insert CSS in HTML:**



1. External CSS: With an external style sheet, you can change the look of an entire website by changing just one file. Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
2. Internal CSS: An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.
3. Inline CSS: An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS properties.

**The basic fundamentals of CSS include the following:**

#### **Selectors:**

CSS selectors are used to identify the HTML elements to which the styles should be applied. Selectors can be based on element types, class names, IDs, or other attributes.

#### **Properties:**

CSS properties are used to define the visual style of HTML elements. Examples of CSS properties include color, font-size, margin, padding, and background-color.

#### **Values:**

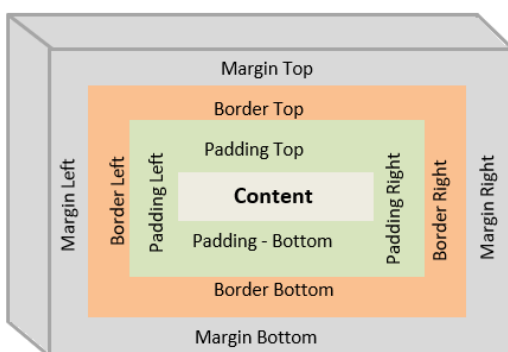
CSS values are used to set the values of CSS properties. For example, the value for the color property can be a named color, an RGB value, or a hexadecimal value.

#### **Cascading:**

CSS stands for Cascading Style Sheets, which means that styles can be inherited and overridden in a cascading manner. This allows for consistent styling throughout a website while also allowing for customization on specific pages or elements.

#### **Box model:**

The CSS box model is used to define the layout of HTML elements. It consists of the content area, padding, border, and margin.



**Units of measurement:**

CSS supports various units of measurement, such as pixels, ems, rems, and percentages, which are used to define the size and position of elements.

**Media queries:**

Media queries allow developers to create responsive websites by applying different styles based on the size of the device or screen. This enables the website to look good on various devices, such as desktops, tablets, and smartphones.

## **Task 6&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
- $\boxtimes$  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**

**Today I created the initials project.**

**I created the (blog website) project including:**

- I created the 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.

## **Task 10**

**Today I learned about Git and GitHub .**

### **What is Git?**

**Git** is a distributed version control system that is used to track changes in source code during software development. It allows developers to work collaboratively on the same codebase and keep track of changes over time. Git was created by Linus Torvalds in 2005 and has since become one of the most popular version control systems in use today.

**Here are some commonly used Git commands:**

**git init:** Initializes a new Git repository in the current directory.

**git add [file]:** Adds a file or files to the staging area for the next commit.

**git commit -m "commit message":** Commits changes to the local repository with a brief message describing the changes.

**git status:** Shows the current status of the repository, including any modified files and files that have been added or deleted.

**git push:** Pushes changes to a remote repository, such as a repository hosted on GitHub.

**git pull:** Pulls changes from a remote repository and merges them with the local repository.

**git branch:** Lists all the branches in the repository.

**git checkout [branch]:** Switches to the specified branch.

**git merge [branch]:** Merges the specified branch into the current branch.

**git clone [repository URL]:** Creates a local copy of a remote repository.

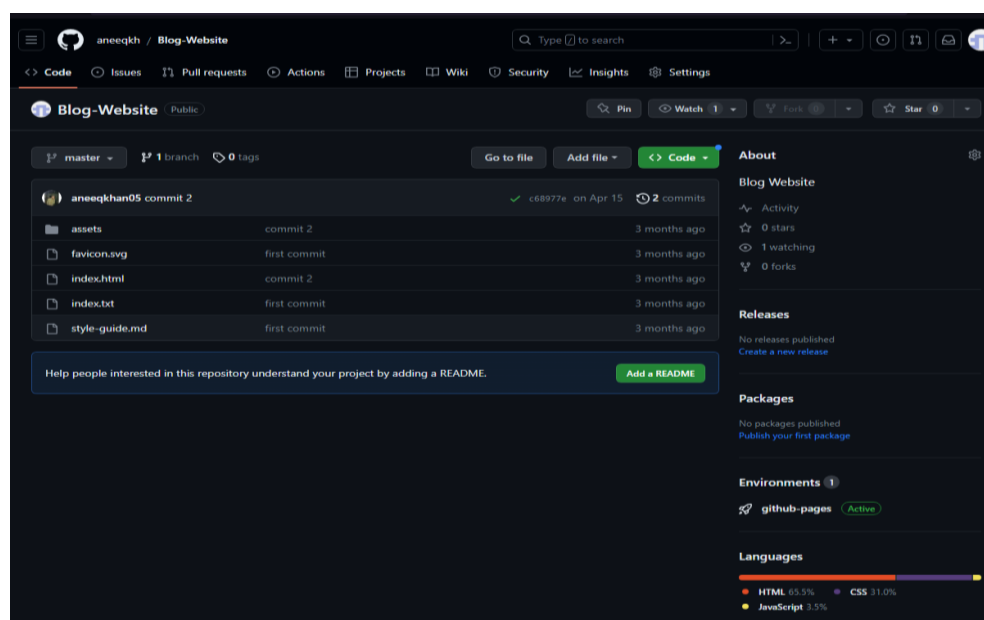
## What is GitHub?

**GitHub**, on the other hand, is a web-based hosting service for Git repositories. It provides a platform for developers to store, manage, and collaborate on Git repositories, as well as a variety of tools for managing code reviews, issue tracking, and continuous integration and deployment.

GitHub allows developers to easily share their code with others and contribute to open-source projects. It provides a platform for developers to work on code collaboratively, manage project workflows, and track issues and bugs. GitHub also provides a range of tools for project management, including wikis, project boards, and team management features.

In this task I also create a github account and make github repository of previous task 9 (blogsite) and in this task to deploy my personal blog website using netlify and github live pages.

## Commit screenshot





**GitHub Repo Link:**

<https://github.com/aneeqkh/Blog-Website.git>

**GitHub Pages Live:**

<https://aneeqkh.github.io/Blog-Website/>

**Netlify Link:**

[aneeqkh-blogsite.netlify.app](https://aneeqkh-blogsite.netlify.app)

## **Task 11&12**

### **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
- ViewView

## Design Resources Repo Link:

<https://github.com/aneeqkhan05/design-resources-for-developers.git>

## 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.

There are several reasons why developers use frameworks when building software applications:

- **Speed and efficiency:** Frameworks provide pre-built components and libraries that can help developers to build applications more quickly and efficiently than if they had to write everything from scratch.

- **Standardization:** Frameworks often come with standardized coding conventions and best practices that can help ensure that the application is consistent, maintainable, and easy to read.
- **Security:** Many frameworks provide built-in security features and modules that can help protect the application from common security vulnerabilities.
- **Scalability:** Frameworks are designed to be scalable, so as the application grows and evolves over time, the framework can adapt and support new features and functionality.
- **Community and support:** Many frameworks have active communities of developers who contribute to the development and maintenance of the framework, which can provide a valuable source of support, documentation, and resources.

Overall, using a framework can help developers save time, increase productivity, and build more robust, secure, and scalable applications.

### 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](https://getbootstrap.com)
- Include Bootstrap from a CDN

## **Task 13**

Today I create a Bootstrap BootCamp website

GitHub Repo Link:

<https://github.com/aneeqkh/FrontEnd-BootCamp/tree/master>

GitHub Pages Live:

<https://aneeqkh.github.io/FrontEnd-BootCamp/>

Netlify Link:

<https://aneeqkh-bootcamp-project.netlify.app>

## **Task 14**

I've completed my tailwind css project

GitHub Repo Link:

<https://github.com/aneeqkh/Tailwind-Project.git>

Netlify Link:

[aneeqkh-parallax-landing-page.netlify.app](https://aneeqkh-parallax-landing-page.netlify.app)

## **Task 15,16&17**

Today I learned about Javascript

JavaScript (often abbreviated as "JS"), let's cover some essential aspects of the language.

## **Purpose and Scope:**

JavaScript is a high-level, interpreted programming language primarily used for developing interactive web pages and web applications. It runs on the client-side (in the browser) and the server-side (with the help of platforms like Node.js). It enables dynamic content, interactivity, and user experience enhancements on websites.

## **Core Features:**

**JavaScript offers several fundamental features, including:**

**Variables:** Used to store data values.

**Data Types:** Primitive types (e.g., numbers, strings, booleans, null, undefined) and complex types (e.g., objects, arrays, functions).

**Operators:** Arithmetic, assignment, comparison, logical, etc.

**Arrays:** Arrays are fundamental data structures in JavaScript. Learn how to create, manipulate, and iterate over arrays. Familiarize yourself with array methods like push, pop, splice, and map.

**Control Flow:** Conditional statements (if-else, switch) and loops (for, while) for program flow control.

**Functions:** Blocks of reusable code that can be invoked with different arguments.

**Objects:** Key-value pairs used to represent entities, and object-oriented programming concepts such as inheritance and prototyping.

**Modules:** JavaScript has a built-in module system that allows you to organize your code into reusable and separate files. Learn how to import and export modules using the ES6 module syntax.

**DOM Manipulation:** JavaScript interacts with the Document Object Model (DOM) to modify web page content dynamically. The Document Object Model (DOM) allows you to interact with web page elements using JavaScript. Learn how to select and modify HTML elements, handle events, and dynamically change the content and style of a webpage.

## **Front-End Frameworks and Libraries:**

JavaScript is extensively used in conjunction with front-end frameworks and libraries. Some popular ones include:

**React.js:** A component-based library for building user interfaces.

**Angular:** A full-featured framework for building web applications.

**Vue.js:** A progressive framework for building user interfaces.

**jQuery:** A fast and concise library simplifying DOM manipulation and AJAX requests.

**D3.js:** A data visualization library for creating interactive charts and graphs.

### **Back-End Development:**

With the help of platforms like Node.js, JavaScript is also used for server-side development. It allows you to build scalable, event-driven, and efficient web servers. You can use frameworks like Express.js and NestJS to develop robust back-end applications.

### **Asynchronous Programming:**

JavaScript has extensive support for asynchronous programming, which is vital for handling tasks such as fetching data from servers or performing time-consuming operations without blocking the user interface. Promises and the newer async/await syntax are commonly used for managing asynchronous operations.

### **Additional Concepts:**

To have a solid understanding of JavaScript, it's beneficial to explore the following concepts:

**Closures:** Functions that can access variables from their outer scope even after the outer function has returned.

**Prototypes and Prototypal Inheritance:** JavaScript's object-oriented nature based on prototypal inheritance.

**Event Handling:** Capturing and responding to user interactions and events.

**Error Handling:** Techniques for handling and managing errors within JavaScript programs.

**Modules:** Encapsulation of code into separate files for better organization and code reuse.

To become proficient in JavaScript, you need to dedicate time and effort to learning and practicing these concepts. You can start with the basics and gradually progress to more advanced topics. There are numerous online tutorials, courses, and resources available that can assist you in your learning journey.

## **Task 18(Final Project)**

**Today I learned about react and also developed a react base project.**

### **React**

React is a popular JavaScript library for building user interfaces. It was developed by Facebook and released as an open-source project. React allows developers to create reusable UI components and efficiently manage the state of the application, resulting in faster and more maintainable code.

#### **Key concepts and features of React include:**

- 1. Components:** React organizes the user interface into reusable, self-contained components. Components encapsulate both the visual elements (HTML structure) and the logic associated with them. This component-based approach promotes code reusability and modularity.
- 2. JSX:** React uses JSX (JavaScript XML) as a syntax extension that allows you to write HTML-like code within JavaScript. JSX enables you to describe the structure and appearance of components in a declarative manner, making it easier to visualize and understand the UI.
- 3. Virtual DOM:** React utilizes a virtual representation of the DOM (Document Object Model) called the Virtual DOM. The Virtual DOM is a lightweight copy of the actual DOM, and React uses it to perform efficient updates and minimize the number of direct manipulations to the real DOM. This approach helps improve performance and rendering speed.
- 4. Unidirectional Data Flow:** React follows a unidirectional data flow, also known as one-way binding. The data flows in a single direction, from parent components to child components. This helps maintain a predictable state management and simplifies debugging.
- 5. State and Props:** React components can have two types of data: state and props. State represents the internal data of a component that can change over time, while props (short for properties) are the immutable data passed from parent components to child components. By managing state and props efficiently, React ensures that components can be easily reused and updated.
- 6. Lifecycle Methods:** React components have several lifecycle methods that are executed at different stages of a component's existence, such as when it is being mounted, updated, or unmounted. These methods provide hooks to perform actions or side effects at specific times during the component's lifecycle.
- 7. React Hooks:** Introduced in React 16.8, hooks allow functional components to have state and access React lifecycle methods. They provide a simpler way to manage state and side effects without using class components.

**8. React Router:** React Router is a popular library for handling routing in React applications. It allows you to define routes and navigate between different components or pages without reloading the entire page.

React has a vast ecosystem with numerous libraries and tools that complement its functionality. Some notable ones include Redux for state management, Axios for making HTTP requests, and Styled Components for styling React components with CSS.

**GitHub Repo Link:**

<https://github.com/aneeqkh/React-App.git>

**Netlify Link:**

<https://aneeq-react-app-movieland.netlify.app>



