CSS frameworks

1. Bootstrap
Offers a consistent and tested codebase for consistent results. The trade-off is that it leaves little room for flexibility. Bootstrap classes help front-end developers quickly add styling to their HTML code.

Advantages:
- Large community and support
- Built with accessibility in mind
- Supports both SASS and LESS

Disadvantages:
- Lack of design flexibility
- Large file sizes with increased download times
- Overreliance on classes
- Compatibility issues with older browser versions

Code:
```
<code>&lt;section&gt;</code>
```

2. Bulma
Provides a modern, minimalist alternative to other CSS frameworks. Instead of using classes for styling and layout, Bulma uses a modular approach. And this allows developers to create custom designs without writing much CSS code. Bulma also has a strict CSS-only approach and includes no JavaScript components.

Advantages:
- Highly flexible and customizable
- Lightweight and fast
- Classless and modular design

Disadvantages:
- Smaller community of developers
- Steep learning curve for those new to Flexbox

Code:
```
<button class="button is-primary is-small">Small</button>

<button class="button is-link">Default</button>
<button class="button is-warning is-normal">Normal</button>
<button class="button is-success is-medium">Medium</button>
<button class="button is-danger  is-large">Large</button>
```

JS Frameworks

1. Angular
An open-source framework that is used for developing a Single Page Application (SPA).

It extends the HTML into the application and interprets the attributes to perform data binding.

Features:
Progressive Web Apps: Has modern web platform capabilities to deliver app-like experiences that are high performance, offline and zero-step installation.

Desktop: Desktop-installed apps across Mac, Windows, and Linux can be created using the same Angular methods as the web plus the ability to access native OS APIs.

Advantages:
- Open source
- Single page applications
- Easier DOM manipulation
- Two-way binding
- Very testable

Disadvantages:
- Possible performance issues
- Can get complex for larger scenarios
- Some features may be difficult to understand for newcomers

Code:
~~~
``` <div ng-app="">

<p>Name: <input type="text" ng-model="name"></p>
<p>You wrote: {{ name }}</p>

</div>```
~~~

2. React
Used to develop and operate the dynamic User Interface of web pages with high incoming traffic. It makes use of a virtual DOM, and hence, integration with any application is more straightforward.

Features:
Declarative: Creates interactive and dynamic UI for websites and mobile applications. Declarative views make the code readable and easy to debug.

Virtual DOM: For every DOM object, there is a corresponding "virtual DOM object." It creates a virtual copy of the original DOM and is a representation of a DOM object.

Advantages:
- Easy to learn
- Reusable components
- Virtual DOM
- SEO friendly

Disadvantages:
- Documentation may be an issue
- JSX is a double-edged sword

Code:

~~~
``` function Welcome(props) {
   return <h1>Hello, {props.name}</h1>;
}

const element = <Welcome name="Faisal Arkan" />;
ReactDOM.render(
   element,
   document.getElementById('root')
);
```
~~~