

# **FRONT END WEB DEVELOPER**

## **ASSIGNMENT – 3**

### **BACHELOR IN TECHNOLOGY**

**(COMPUTER SCIENCE ENGINEERING)**



**Name:** Vikash Chaubey

**Registration No:** 11906626

**Roll No:** 68

**Section** KM082

**Submitting to** Ruchita Duggal

**LOVELY PROFESSIONAL UNIVERSITY**

**(Jalandhar - Delhi G.T. Road, Phagwara, Punjab (India) – 144411)**

## **ACKNOWLEDGEMENTS**

I would like to express my gratitude towards for providing this course which has helped me in improving my Java language. I am also grateful to LPU for providing learning opportunity.

*I am thankful to and fortunate enough to get constant encouragement support and guidance from all Teaching staffs of LPU which helped me in successfully completing my course.*

***Vikash Chaubey***

## **TABLE OF CONTENTS**

- Introduction
- What Is a Front-End Developer?
- Feature
- What is the basic Structure of a Front End?
- Technology Learnt
- Reason for choosing this technology
- Learning outcome of the technology Learnt
- Conclusion
- References

# **INTRODUCTION**

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data.

## **Hyper Text Markup Language**

Hyper Text Markup Language (HTML) is the backbone of any website development process, without which a web page does not exist. Hypertext means that text has links, termed hyperlinks, embedded in it. When a user clicks on a word or a phrase that has a hyperlink, it will bring another web-page. A markup language indicates text can be turned into images, tables, links, and other representations. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation. This version contains new and efficient ways of handling elements such as video and audio files.

## **Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) controls the presentation aspect of the site and allows your site to have its own unique look. It does this by maintaining style sheets which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

## **JavaScript**

JavaScript is an event-based imperative programming language (as opposed to HTML's declarative language model) that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the Document Object Model (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

Using a technique called AJAX, JavaScript code can also actively retrieve content from the web (independent of the original HTML page retrieval), and also react to server-side events as well, adding a truly dynamic nature to the web page experience.

## **WebAssembly**

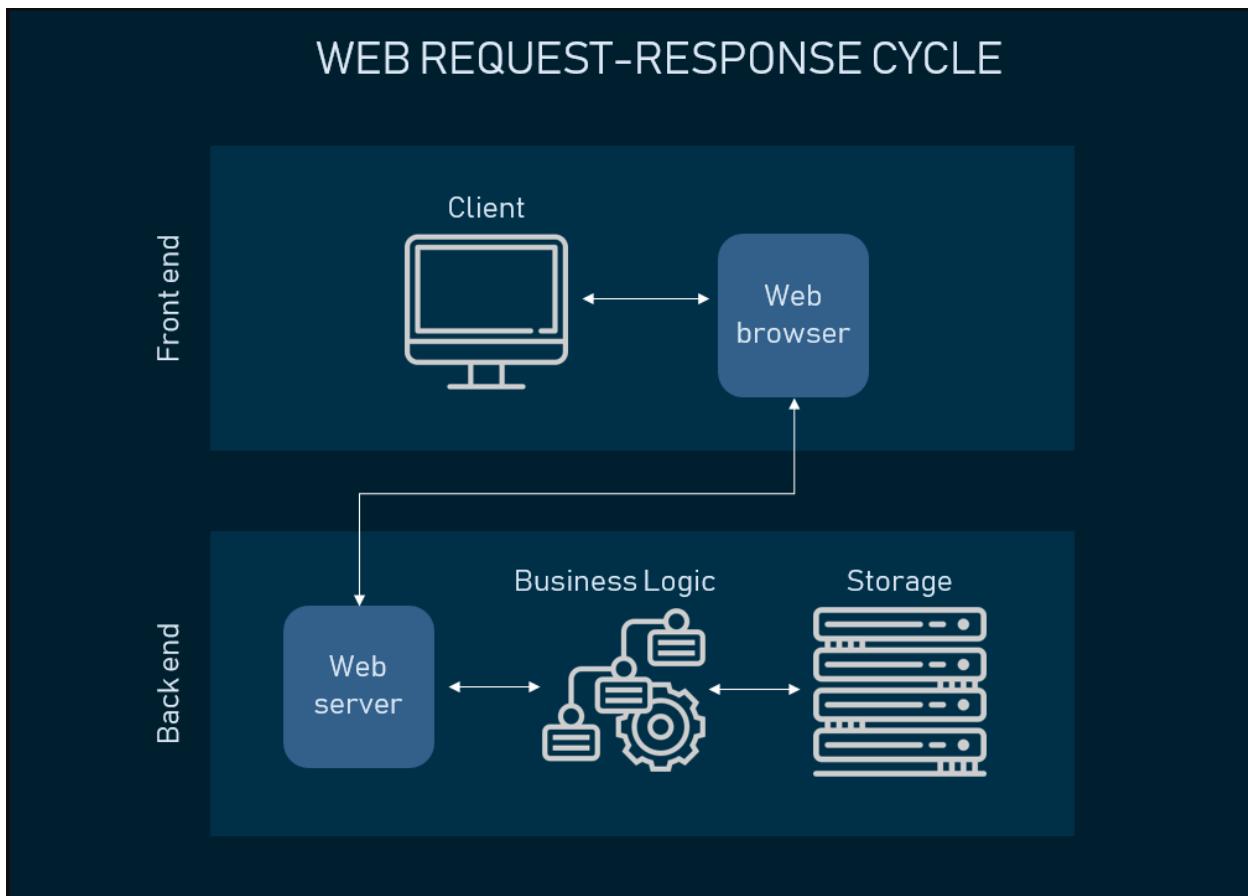
WebAssembly, supported by all the major browsers (i.e. from the major vendors Google, Apple, Mozilla and Microsoft), is the only alternative to JavaScript for running code in web browsers (without the help of plug-ins, such as Flash, Java or Silverlight; all being discontinued, as browsers are dropping plug-in support). Prior to its adoption, there was asm.js (a subset of JavaScript; and thus strictly works in all browsers), that's also used as a compiler target with efficient support in browsers such as Internet Explorer 11; and for such browsers that do not support WebAssembly directly, it can be compiled to asm.js and those browsers supported that way. Generally speaking programmers do not program in WebAssembly (or asm.js) directly, but use languages such as Rust, C or C++ or in theory any language, that compile to it.

## What Is a Front-End Developer?

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

## FEATURES



The basic toolset for a front-end is well-defined: HTML, CSS, and JavaScript. The frontend development technologies may be however extended with package managers, CSS preprocessors, frameworks, and many more.

## The key front-end technology

HTML (or Hypertext Markup Language) is a computer language designed to create websites that later can be explored by anybody who accesses the Internet. HTML is normally employed to structure a web document. It defines such elements as headlines or paragraphs and enables embedding images, video, and other media.

**How HTML works.** HTML is composed of a series of shortcodes called *tags*, normalized into a text-file by a site creator. The text is then stored as an HTML file and observed through a browser. The browser scans the file and interprets the text into a visible form, and in the best case, renders the page as the designer had planned.

- *HyperText* is the way by which we travel across the web by clicking hyperlinks – specific texts taking you to other pages. Hyper means it's non-linear, which allows for moving to any other place, as there is no predefined order to do so.
- *Markup* determines the qualities that HTML tags apply to the text inside them. Tags mark it as a particular type of text.
- As a *Language*, it holds code words and syntax like any other language.

For example:



```
index.html
1 <!DOCTYPE html>
2
3 <html>
4   <body>
5     <h1>My First Heading</h1>
6     <p>My first paragraph.</p>
7   </body>
8 </html>
```

The text between <html> and </html> specifies the web page, while the text between <body> and </body> determines the visible content.

<!DOCTYPE html> on the top is the Document Type Declaration for HTML5. If it's not included, different browsers will render it in their way.

This is how we gradually move to HTML5.

## **HTML5**

Since its initial release in 1991, HTML has undergone many updates. HTML5 released in 2014. It has added such features as offline media storage support, more precise content elements (i.e. header, footer, navigation), and support for audio and video embedding.

## **CSS: Styling your facade**

CSS (or Cascading Style Sheets) is a style sheet language. It's applied to define how HTML elements are supposed to be presented on a webpage in terms of design, layout, and variations for diverse devices with different screen sizes. CSS masters the layout of numerous different web pages at a time.

**How it works.** CSS interacts with HTML elements, the components of a web page.

- To communicate with HTML, CSS uses **selectors**. A selector is the part of CSS code defining which HTML piece the CSS styling will impact.
- A **declaration** contains properties and values that are employed by the selector.
- **Properties** define font size, color, and margins. Values are the settings for these properties.

If we take an example,

```
86
87  p {
88    font-size:24px;
89    color:blue;
90 }
```

*The example of CSS code*

P (for paragraph) is selector, { font-size:24px; color:blue; } is a declaration, font-size: and color: are properties, while 24px; and blue; are values.

CSS is written in plain text over a text editor or word processor on a computer. If you were to examine how the CSS code is implemented to HTML content, there are three ways to do so:

- External style sheets are stored as .css files and can be applied to define the appearance of a whole website via a single file, instead of putting extra instances of CSS code to every HTML element which has to be modified. To use an external style sheet, .html files need to contain a header section that connects to the external style sheet.
- Internal style sheets are CSS directions put straight into the header of a particular .html page.
- Inline styles are snippets of CSS recorded into HTML code itself.

**CSS frameworks.** A CSS framework is a set of default CSS and HTML files. It extends a front-end developer's capabilities for website design. In addition to assisting while building a responsive design, CSS frameworks also present distinct and symmetric layouts, saving developers from writing code from ground zero at every occasion. They are usually considered a good choice to fit diverse platforms and screen sizes. With common user interface components, grid systems, layouts, and many other features, CSS frameworks considerably accelerate development workflow. Many frameworks exist in the CSS Universe:

- Full featured (Bootstrap, Foundation, Semantic UI, and more),
- Aimed at Material Design: (Materialize and Material Design Lite), and
- Lightweight (Pure).

We won't describe them all. You can examine our comparison table instead.

## COMPARATIVE TABLE OF THE MOST POPULAR CSS FRAMEWORKS

	Full-featured					Material-based		Very Lightweight
	Bootstrap	Foundation	Semantic UI	UIkit	Bulma	Materialize	Material Design Lite	Pure
Principles	RWD, Mobile first	RWD, Mobile first Semantic	Semantic Tag, Ambivalence, Responsive	RWD, Mobile first	RWD, Mobile first, Modern free	Responsive, Web design, UX-focused	Cross-device use	SMACSS, Minimalism
Size(zip folder)	592 KB	233 KB	1.8 MB	347 KB	113 KB	162 KB	205 KB	72 KB
Preprocessors	Sass	Sass	Less	Less, Sass	Sass	Sass	Sass	-
Unique Features	Jumbotron, Card component, Responsive navbar	Icon Bar, Clearing Lightbox, Flex video, Keystrokes, Joyride, Pricing tables	Divider, Flag, Rail, Reveal, Step, Advertisement, Card, Feed, Item, Statistic, Dimmer, Rating,	Article, Comments, Placeholder Flex, Cover, HTML editor	-	ScrollFire, ScrollSpy, Wave behaviors, Slide-out drawer menu, Toast, Parallax	Tooltips, Spinners	-
Flexbox	+	- (some key components can be converted)	+	+(The Flex component)	+	-	+	-
Grid	up to 12 columns	XY12-column grid: float grid empowered by Flexbox features	default theme: 16 columns	components: grid, flex and width; offers a border between grid columns	Simple building of columns layout	a standard 12-column fluid responsive grid system	12 - desktop; 8 - tablet; 4 - phone	a 5ths and 24ths unit-based grid
Documentation	Excellent	Good	Excellent	Good	Good	Good	Excellent	Good
JS skills required	+ -	+	+	-	-	+	-	-
Learning curve	Mild	Steep	Moderate	Moderate	Mild	Mild	Mild	Mild



*Most popular CSS frameworks*

In case you want to explore these CSS frameworks in detail, check out our article on Most Popular Responsive CSS Frameworks.

**Pre-processors – Sass and LESS.** Drafting CSS is routine, while short tasks such as looking up colour values, closing tags, or any other repetitive operations take a lot of time. That's where a pre-processor comes in handy. A CSS pre-processor is a scripting language that expands CSS and assembles it into general CSS.

Sass and LESS are the most common pre-processors. They share some common basics, such as:

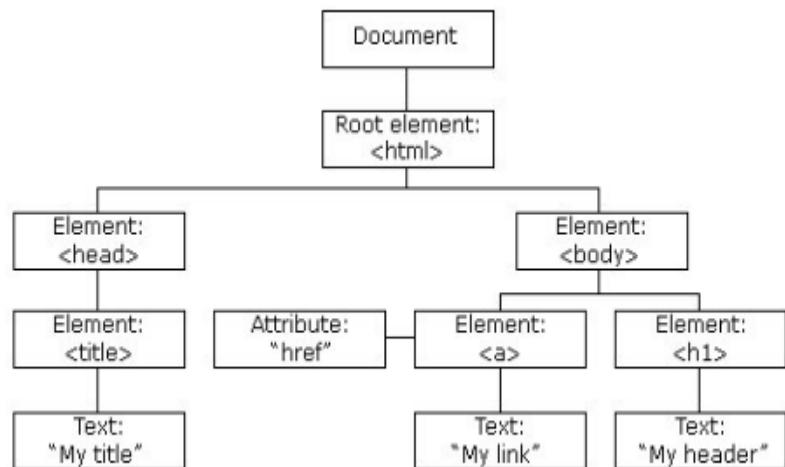
- Syntax elements and
- Backward compatibility with normal CSS files.

However, there are also a lot of differences between them.

- Sass means Syntactically Awesome Style Sheets. Sass runs on Ruby and is rendered on the server side. Due to its Ruby language origin, the installation is provided via so-called gems (multiple Ruby/Rails libraries).
- LESS stands for Leaner Style Sheets. It's a JavaScript library and is rendered on the client side in the browser. Developers choose LESS much more regularly when using JavaScript with style sheets. It resembles composing regular CSS. The technology allows for reusing pieces of CSS within LESS files.

## What is the basic Structure of a Front End?

The Document Object Model (DOM) is defined as a programming interface for HTML and XML documents. It interprets the page so that programs can modify the document structure, style, and content. The DOM renders the document as nodes and objects, enabling programming languages to connect to the page.



*The HTML DOM tree of objects*

*Source: W3Schools*

**How it works.** Consider a web page as a document that can be either presented in the browser window or as the HTML source code. The Document Object Model (DOM) represents this document so it can be modified. The DOM is an object-oriented representation of the web page, which can be altered with a scripting language like JavaScript.

The DOM must adhere to W3C and WHATWG standards that are executed in most current browsers. The modern DOM is built using multiple APIs that work together. The core DOM specifies the objects that entirely explain a document and the objects within it.

## **JavaScript: making the Web alive**

JavaScript (JS) is one of the most popular scripting languages. It's mostly renowned for providing a full stack of technologies for both front-end and back-end development. As we are talking about the first one, it's applied to make web pages dynamic.

**How JavaScript works.** JS boosts a website's overall interactivity. It enables modeling animated UI components such as image sliders, pop-ups, extensive site navigation menus, and more. JavaScript provides a website with extended functionality that isn't otherwise achievable with HTML and CSS alone. JavaScript allows web pages to reply to users' actions and dynamically refresh themselves. Thanks to JavaScript, this process doesn't demand a page reload to alter its representation.

## **JavaScript frameworks and libraries and why we need them**

Those who are just entering this field of expertise may think that frameworks and libraries do the same thing, allowing different visual elements to interact with each other. And let's be honest: It's not too far from the truth. However, there are a few distinguishing peculiarities. So, let's define what makes a JS framework, what a JS library is, and what purpose they both serve.

**Frameworks** are templates to create a website or a web application. They provide a structure (such as a skeleton or scaffolding) on which to arrange the whole project. While the framework sets the page templates, they build the structure with particular allocated areas to embed a framework code in.

So, JavaScript frameworks are complete sets of tools to form and arrange a website or web application.

**Libraries** are sets of prewritten code snippets used and reused to implement core features of JavaScript. The snippet can be easily integrated into an existing project code when necessary.

So, libraries are specialized tools for particular coding needs, not an all-purpose machine for grooming the whole existing project.

## **Main frameworks and libraries to know**

Let's start with frameworks:

**Angular** is a front-end framework powered by Google that is compatible with most of the common code editors. Angular is intended to create dynamic, single-page web applications (SPAs), and progressive web-apps. Upon its initial release, the framework was most appreciated for its ability to convert HTML-based documents into dynamic content. Angular is among the most popular front-end frameworks. You can explore this topic in our article on the pros and cons of Angular development.

**Vue.js** is yet another open-source framework for SPAs. It employs a component-based development model and enables attaching components to a project. Vue.js is an example of a library that is more of a framework. To utilize Vue.js, knowledge of HTML and CSS is an absolute requirement. It suggests a whole bunch of templates and patterns applied in development. Vue is primarily recognized for the small size of documents and HTML-based syntax. Again, to immerse yourself in Vue.js advantages and disadvantages click the link.

**Ember.js** is a framework for SPAs, mobile, and desktop applications. It uses the model-view-view-model (MVVM) pattern. Ember tools enable designing a development environment, while its command-line interface gives tools for script automation.

And now let's outline a few libraries:

**React** is an open-source library for building dynamic user interfaces, born and bred by Facebook. The framework is applied to create web-applications with multiple dynamic components. It relies on JavaScript and JSX, a Facebook PHP language extension. React enables building reusable HTML elements for the front end. React also includes React Native, a dedicated cross-platform mobile development framework. To discover more about the pros and cons of React and React Native, consider exploring our article on this subject.

**jQuery**, in turn, is aimed at controlling HTML documents. It has a simple API to control events and design animation in browsers. Aside from that, jQuery is applied to manipulate the Document Object Model (DOM) and serves as a plugin development tool as well. It also comes with a lighter cross-browser library, jQuery UI for a mobile framework jQuery Mobile and for graphic interface construction.

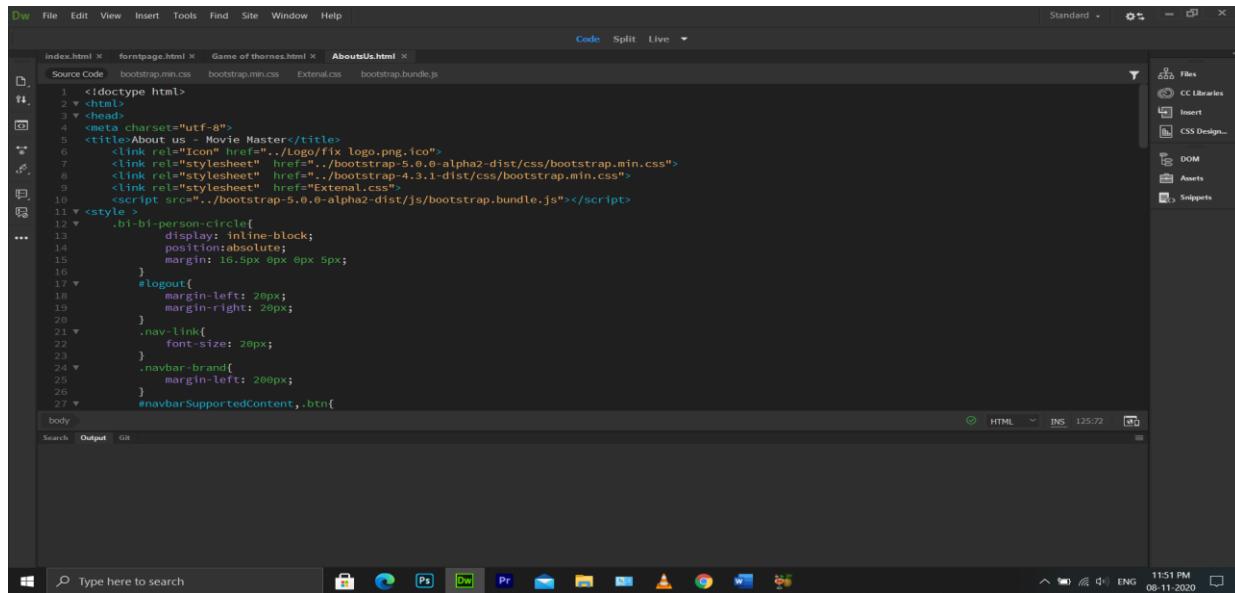
**D3.js** is a data-driven library for data visualization. By tying temporary data to DOM and implementing data-driven changes to the document, the library enables managing data and making dynamic data visualizations. It can uphold and process large datasets and dynamic responses for interaction and animation. D3's functional style permits code reusing and runs with CSV and HTML.

As we see, JavaScript's extensive libraries have broad functionality, being a one-stop solution for front-end developers. But the thing is, it's not a matter of substitution or favoring libraries over frameworks or vice versa. There's always a place for both a library and a framework.

If you're starting with the front end, it would be reasonable to begin with libraries since they have a milder learning curve. Upon capturing the basics from a few of them, learners can proceed to frameworks, as they require a greater understanding of JS fundamentals.

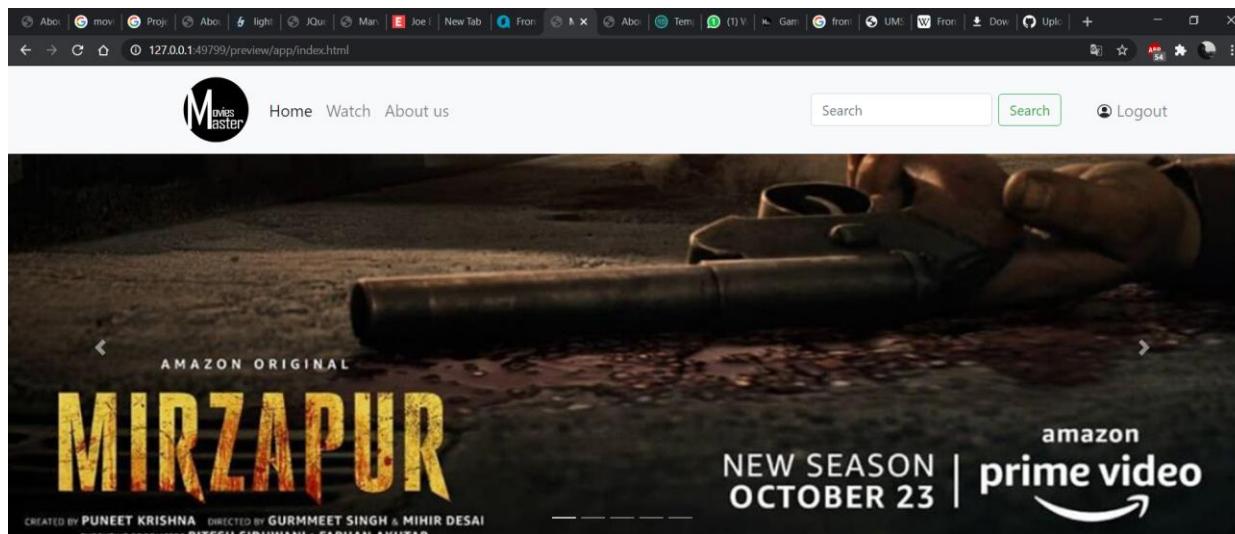
# CODING APART

## Usage of Bootstrap



The screenshot shows the Adobe Dreamweaver interface with the code editor open. The file being edited is 'AboutUs.html'. The code includes standard HTML structure, a meta charset declaration, and a title. It also includes Bootstrap's CSS imports for icons and styles, and a script tag for the Bootstrap bundle. A sidebar on the right contains various tools like CC Libraries, Insert, CSS Design, DOM, Assets, and Snippets.

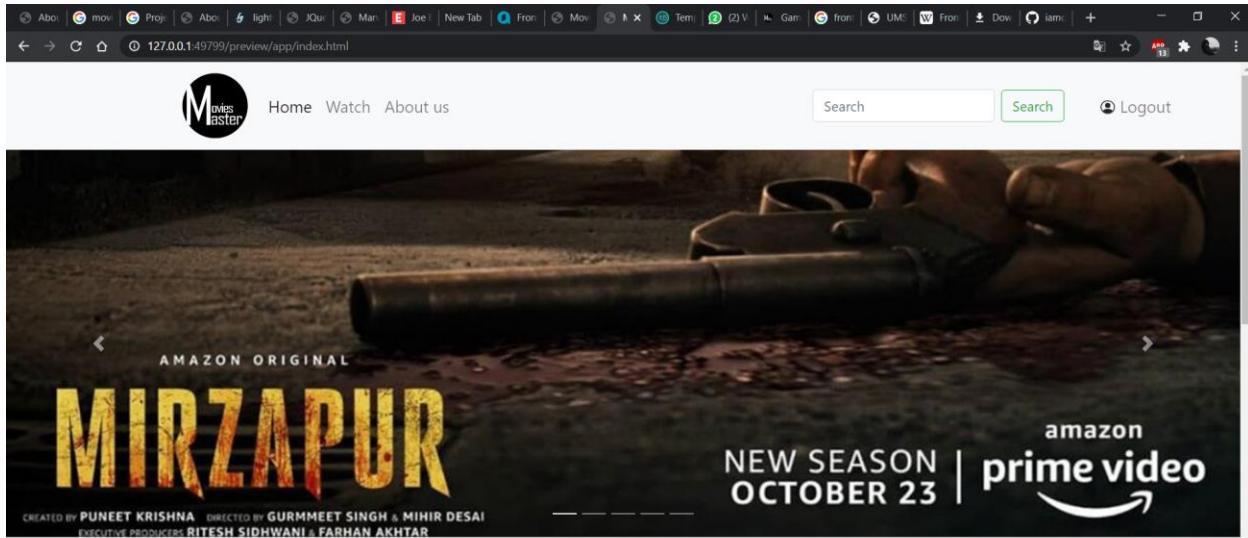
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>About Us - Movie Master</title>
    <link rel="icon" href="../../Logo/favicon/logo.png.ico">
    <link rel="stylesheet" href="../../bootstrap-5.0.0-alpha2-dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="../../bootstrap-4.3.1-dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="External.css">
    <script src="../../bootstrap-5.0.0-alpha2-dist/js/bootstrap.bundle.js"></script>
  </head>
  <body>
    <.bi.bi-person-circle>
      display: inline-block;
      position: absolute;
      margin: 16.5px 0px 0px 0px;
    </>
    <.logout>
      margin-left: 20px;
      margin-right: 20px;
    </>
    <.nav-link>
      font-size: 20px;
    </>
    <.navbar-brand>
      margin-left: 200px;
    </>
    <.navbarSupportedContent>, <button>
  </body>
```



### Trending News



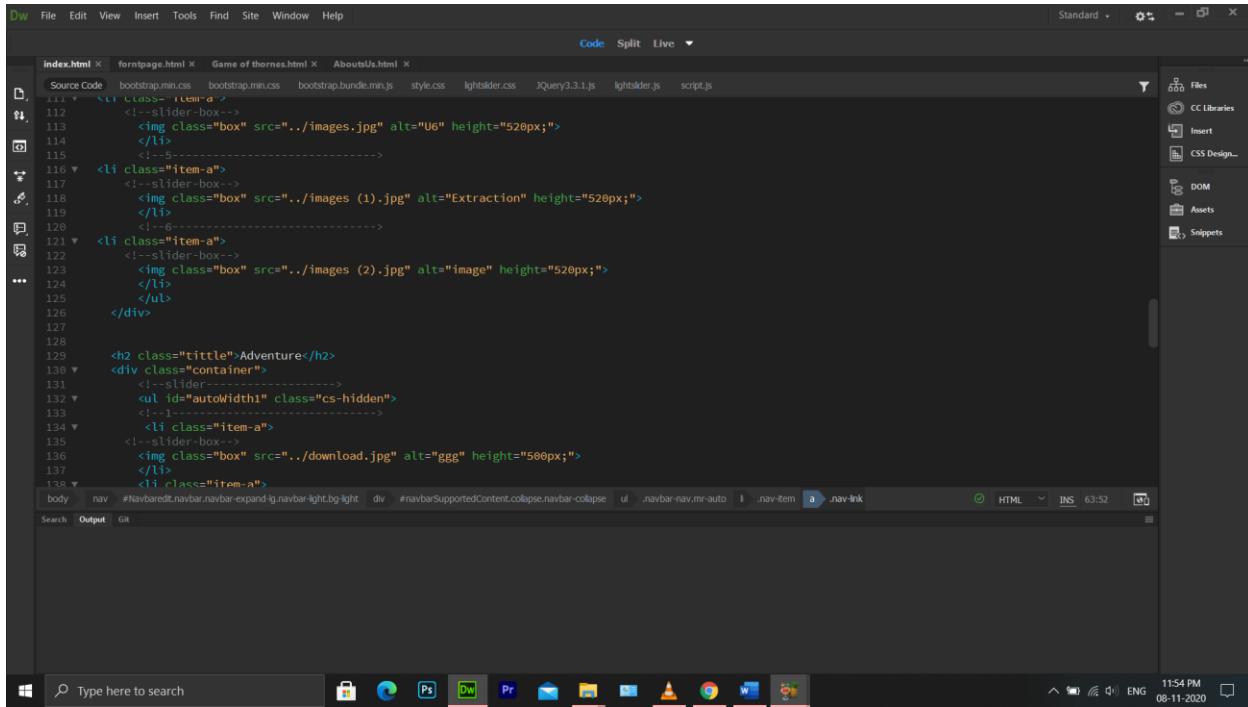
## **HOME PAGE**



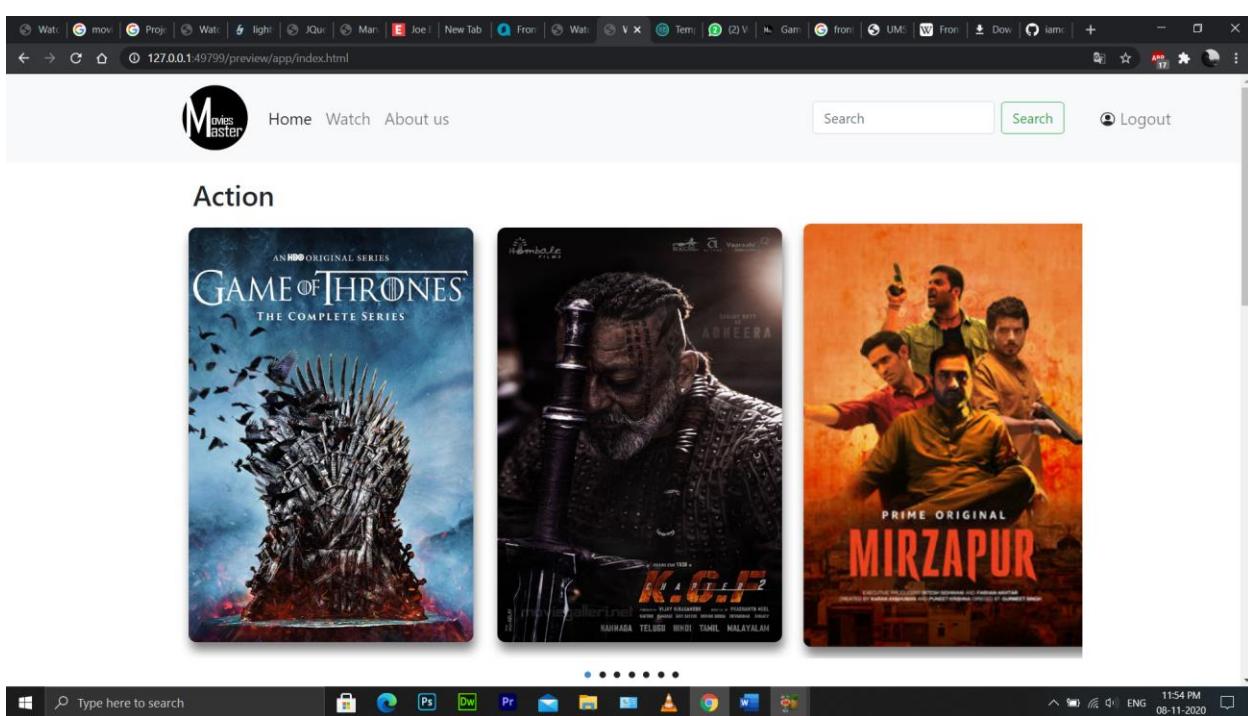
## Trending News



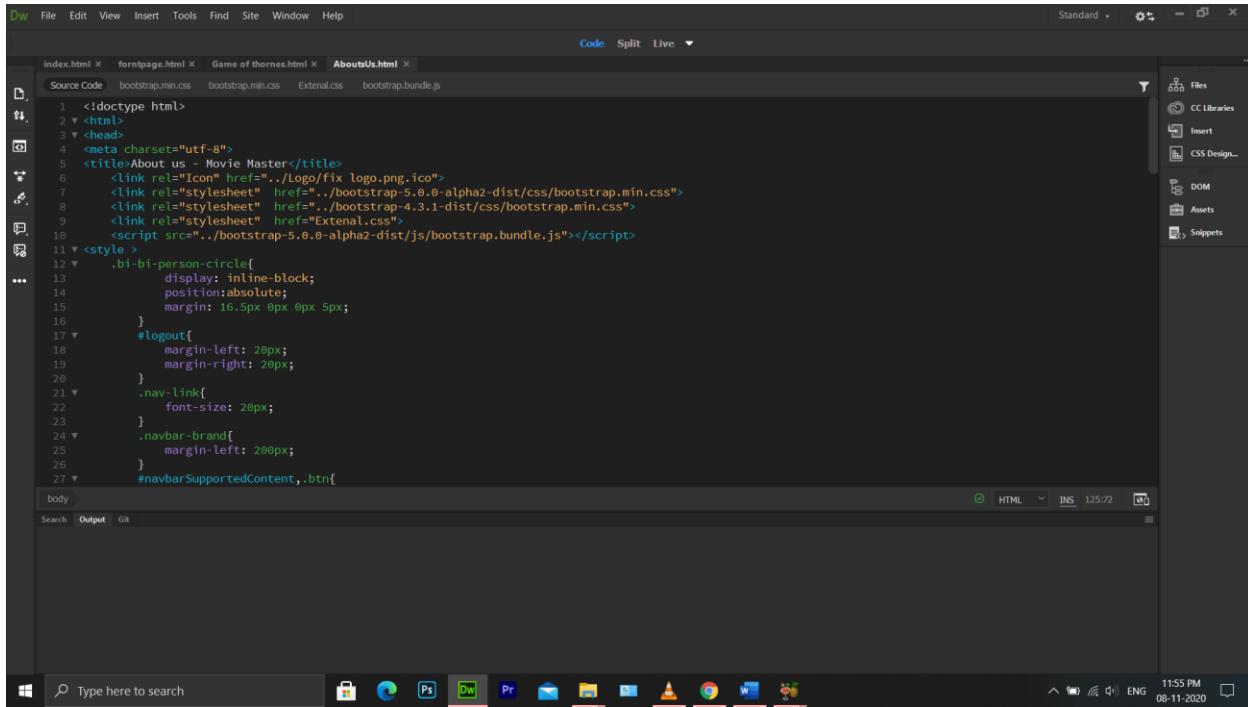
## Watch Page



```
index.html x formpage.html x Game of thrones.html x AboutsUs.html x
Source Code bootstrap.min.css bootstrap.bundle.min.css style.css lightslider.css JQuery3.3.1.js lightslider.js script.js
111 <li class="item-a">
112   <!--slider-box-->
113   
114   </li>
115   <!--5----->
116 <li class="item-a">
117   <!--slider-box-->
118   
119   </li>
120   <!--6----->
121 <li class="item-a">
122   <!--slider-box-->
123   
124   </li>
125 </ul>
126 </div>
127
128 <h2 class="tittle">Adventure</h2>
129 <div class="container">
130   <!--slider-->
131   <ul id="autoWidth1" class="cs-hidden">
132     <li>
133       <!--1----->
134     <li class="item-a">
135       <!--slider-box-->
136       
137       </li>
138     <li class="item-a">
```

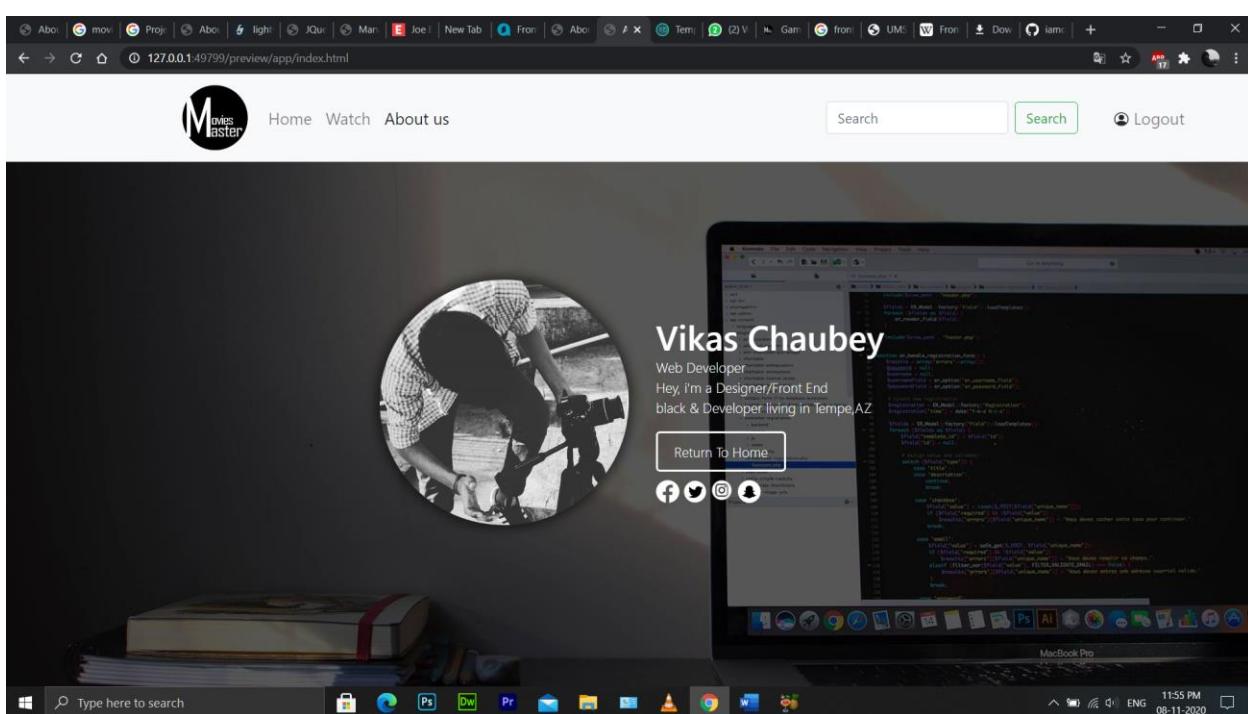


## About us Page



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>About us - Movie Master</title>
<link rel="stylesheet" href="../Logo/fix logo.png.ico">
<link rel="stylesheet" href="../../bootstrap-5.0.0-alpha2-dist/css/bootstrap.min.css">
<link rel="stylesheet" href="../../bootstrap-4.3.1-dist/css/bootstrap.min.css">
<link rel="stylesheet" href="External.css">
<script src="../../bootstrap-5.0.0-alpha2-dist/js/bootstrap.bundle.js"></script>
<style>
.bi.bi-person-circle{
    display: inline-block;
    position: absolute;
    margin: 16.5px 0px 0px 5px;
}
#logout{
    margin-left: 20px;
    margin-right: 20px;
}
.nav-link{
    font-size: 20px;
}
.navbar-brand{
    margin-left: 200px;
}
#navbarSupportedContent, .btn{
    border-radius: 50px;
}
</style>

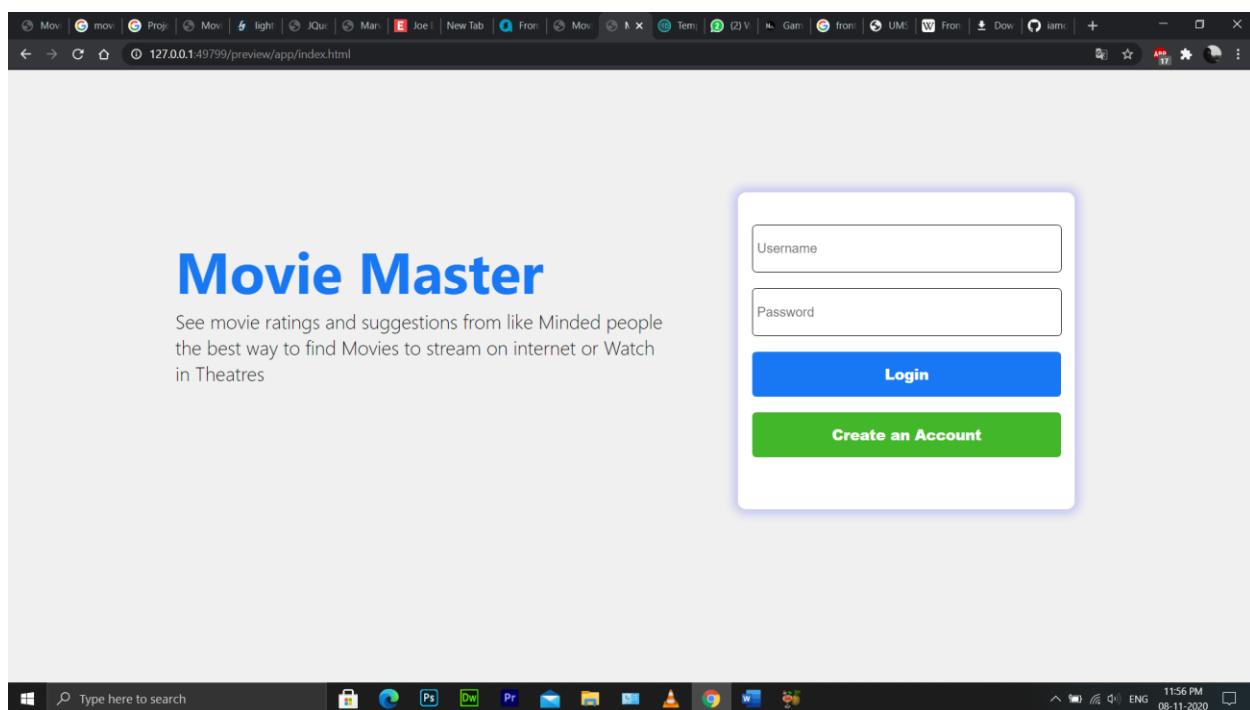
```



## Login page

The screenshot shows the Adobe Dreamweaver interface with the code editor open. The code is written in JavaScript and handles user input for a login form. It checks if the username is 'vikash' and the password is '12345'. If correct, it redirects to 'forntpage.html'. If incorrect or fields are empty, it alerts the user and clears the fields. The code includes comments explaining the logic.

```
index.html x forntpage.html x Game of thrones.html x AboutUs.html x Login.html x
107     border-radius: 5px;
108 }
109 </style>
110 <script>
111 function ValidateLogin() {
112     var username = document.getElementById('username');
113     var password = document.getElementById('password');
114     if (username.value != '' && password.value != '') {
115         if (username.value == 'vikash' && password.value == '12345') {
116             window.location = "forntpage.html";
117         } else {
118             alert('Invalid credentials. Please try again.');
119             username.value = '';
120             password.value = '';
121             return false;
122         }
123     } else {
124         if (username.value == '') {
125             alert('Username is required field.');
126             username.focus();
127             return false;
128         }
129         if (password.value == '') {
130             alert('Password is required field.');
131             password.focus();
132             return false;
133         }
134     }
135 }
body
```



## **REFERENCES**

[https://en.wikipedia.org/wiki/Front-end\\_web\\_development](https://en.wikipedia.org/wiki/Front-end_web_development)

<https://cdnjs.com/libraries/OwlCarousel2>

<https://v5.getbootstrap.com/>

[Project like](#)

<https://github.com/iamoxygen/MovieMaster.git>