

IMPELSYS INTERNSHIP

From 08-07-2024 To 19-07-2024

KIRTHANA PRAVEEN

FULL STACK DEVELOPMENT

Full Stack Development refers to the development of both **front end** (client side) and **back end** (server side) portions of web applications. Full Stack web developers have the ability to design complete web applications and websites. They work on the frontend, backend, database, and debugging of web applications or websites. Here's a breakdown of the roles:

Front-End Developer:

- Also known as a **client-side** developer.
- Focuses on creating the visible parts of a website that users interact with.
- Skills include **HTML**, **CSS**, and **JavaScript**.
- Think of them as the decorators at a wedding reception – setting up the ambiance for guests to enjoy!

Back-End Developer:

- Also known as a **server-side** developer.
- Works behind the scenes, handling data processing, server communication, and database management.
- Creates pathways to deliver information to users.
- Imagine them as the chefs in the kitchen – preparing the delicious meal that guests don't see directly!

Full-Stack Developer:

- The superhero of development!
- Proficient in both front-end and back-end development.
- Can build entire applications independently.
- Skills include a mix of front-end technologies (like HTML, CSS, and JavaScript) and back-end technologies (such as server frameworks and databases).

Some popular full-stack technology stacks:

1. LAMP Stack:

- Components: Linux, Apache, MySQL, PHP.
- Description: LAMP is a widely used stack for web development. It includes the Linux operating system, the Apache web server, the MySQL database, and PHP for server-side scripting.

2. LEMP Stack:

- Components: Linux, Nginx, MySQL, PHP.
- Description: Similar to LAMP but with Nginx as web server instead of Apache.

3. MEAN Stack:

- Components: MongoDB, Express, AngularJS, Node.js.
- Description: MEAN is a JavaScript-based stack. It uses MongoDB for the database, Express for the backend framework, AngularJS for the frontend, and Node.js for server-side scripting

4. MERN Stack:

- Components: MongoDB, Express, ReactJS, Node.js.
- Description: MERN is another JavaScript-based stack. It replaces AngularJS with ReactJS for frontend development.

FRONT END

The front end is the part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc.

Languages used for the front end are HTML, CSS, and JavaScript

Responsiveness and performance are the two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

Frontend Frameworks and Libraries:

- **AngularJS**
- **React.js**
- **jQuery**
- **SASS**
- **Flutter.**
- Some other libraries and frameworks are Semantic-UI, Foundation, Materialize, Backbone.js, Ember.js, etc.

BACK END

Backend is how everything works behind the scenes. Languages used for the back end include Java, Ruby, Python, and .Net.

The backend is the server side of the website. It manages data and ensures everything on the frontend works properly. Users don't see or interact directly with the backend; it's the behind-the-scenes functionality. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application.

Back-End Frameworks:

- **Express**
- **Django**
- **Ruby on Rails**
- **Laravel**
- **Spring**
- Some more back-end programming/scripting languages are C#, Ruby, GO, etc.

MY LEARNINGS - FRONT END DEVELOPMENT

HTML

- HTML stands for Hyper Text Markup Language
 - HTML is the standard markup language for creating Web pages
 - HTML describes the structure of a Web page
 - HTML elements tell the browser how to display the content
-
- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
 - The `<html>` element is the root element of an HTML page
 - The `<head>` element contains meta information about the HTML page
 - The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
 - The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
 - The `<h1>` element defines a large heading
 - The `<p>` element defines a paragraph

An HTML element is defined by a start tag, some content, and an end tag:
`<tagname> Content goes here... </tagname>`

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements

- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**
- The href attribute of <a> specifies the URL of the page the link goes to
- The src attribute of specifies the path to the image to be displayed
- The width and height attributes of provide size information for images
- The alt attribute of provides an alternate text for an image
- The style attribute is used to add styles to an element, such as color, font, size, and more
- The lang attribute of the <html> tag declares the language of the Web page
- The title attribute defines some extra information about an element

Tag	Description
<code><html></code>	Defines the root of an HTML document
<code><body></code>	Defines the document's body
<code><h1></code> to <code><h6></code>	Defines HTML headings
<code><p></code>	Defines a paragraph
<code><hr></code>	Defines a thematic change in the content
<code>
</code>	Inserts a single line break
<code><pre></code>	Defines pre-formatted text

HTML STYLES

- Use the style attribute for styling HTML elements
- Use background-color for background color
- Use color for text colors
- Use font-family for text fonts
- Use font-size for text sizes
- Use text-align for text alignment

NAME- KIRTHANA PRAVEEN
AGE-19
Qualification Btech

Output:

MY HOBBIES

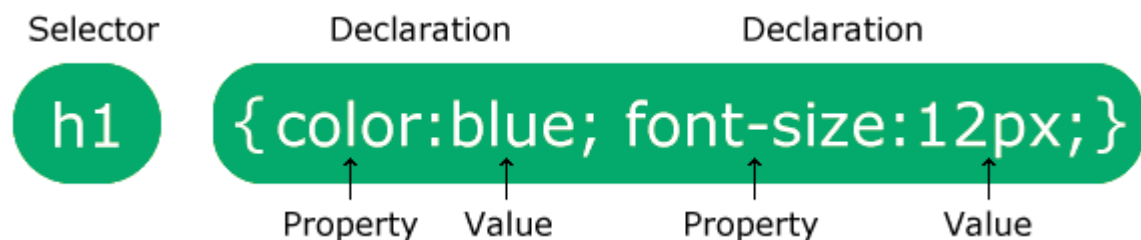
MUSIC, DANCE and drawing

CSS

CSS is the language we use to style a Web page.

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- 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

CSS SYNTAX:



A CSS rule consists of a selector and a declaration block.

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS SELECTORS

Selector	Example	Example description
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>*</u>	*	Selects all elements

THREE WAYS TO INSERT CSS:

- 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.

- 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.
- Inline CSS-- An inline style may be used to apply a unique style for a single element.

CSS BORDER STYLE

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border



[Home](#)

[News](#)

[Contact](#)

[About](#)

Content

Mary had a little lamb.

Twinkle Twinkle little star.

Humpty Dumpty sat on a wall.

[Help me](#)

JAVASCRIPT

JavaScript is the world's most popular programming language.

It is the programming language of the Web.

It is easy to learn.

JavaScript is one of the **3 languages** all web developers **must** learn:

1. [HTML](#) to define the content of web pages
2. [CSS](#) to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages

JavaScript can "display" data in different ways:

- Writing into an HTML element, using `innerHTML`.
- Writing into the HTML output using `document.write()`.
- Writing into an alert box, using `window.alert()`.
- Writing into the browser console, using `console.log()`.

JAVASCRIPT KEYWORDS

Keyword	Description
var	Declares a variable
let	Declares a block variable
const	Declares a block constant
if	Marks a block of statements to be executed on a condition
switch	Marks a block of statements to be executed in different cases
for	Marks a block of statements to be executed in a loop
function	Declares a function
return	Exits a function
try	Implements error handling to a block of statements

JavaScript keywords are reserved words. Reserved words cannot be used as names for variables.

JavaScript uses the keywords `var`, `let` and `const` to **declare** variables.
An **equal sign** is used to **assign values** to variables.

JAVASCRIPT HAS 8 DATATYPES:

String
Number
Bigint
Boolean
Undefined
Null
Symbol
Object

JavaScript objects are written with curly braces `{}`.

Object properties are written as `name:value` pairs, separated by commas.

Ex:

```
const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

JavaScript arrays are written with square brackets.

Array items are separated by commas.

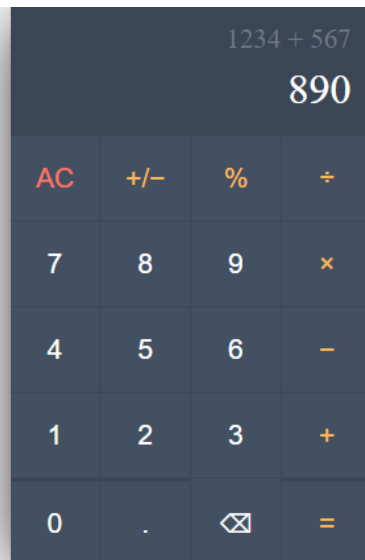
Ex:

```
const cars = ["Saab", "Volvo", "BMW"];
```

JAVASCRIPT FUNCTION:

SYNTAX:

```
function name(parameter1, parameter2, parameter3) {  
  // code to be executed  
}
```



JSON

JSON stands for JavaScript Object Notation

JSON is a text format for storing and transporting data

JSON is "self-describing" and easy to understand

JSON syntax is derived from JavaScript object notation syntax:

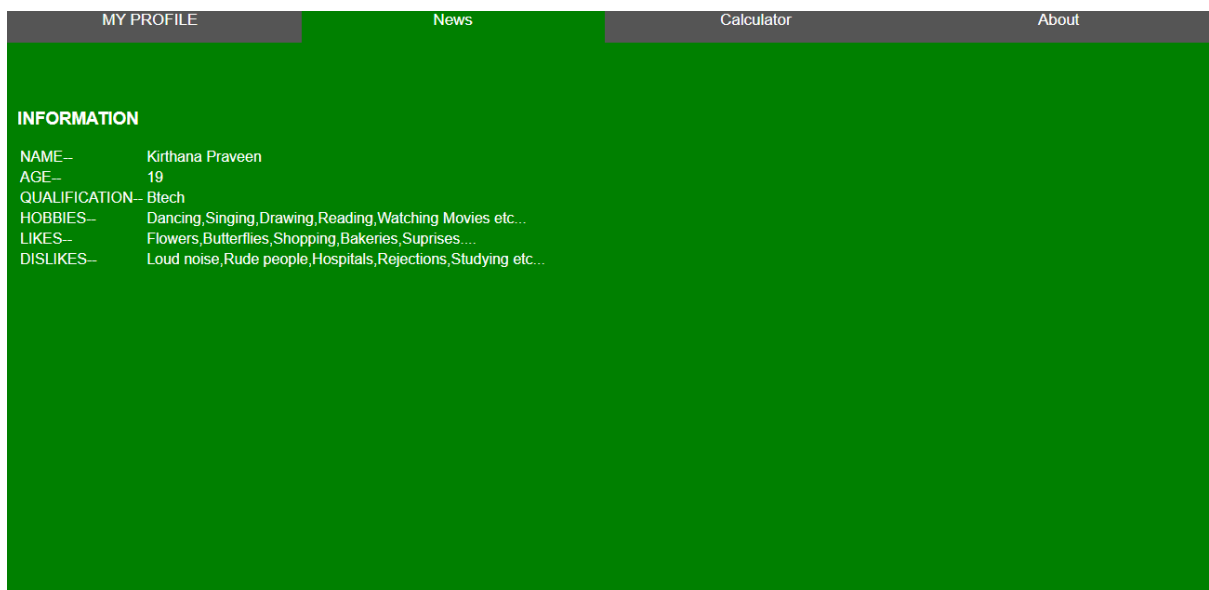
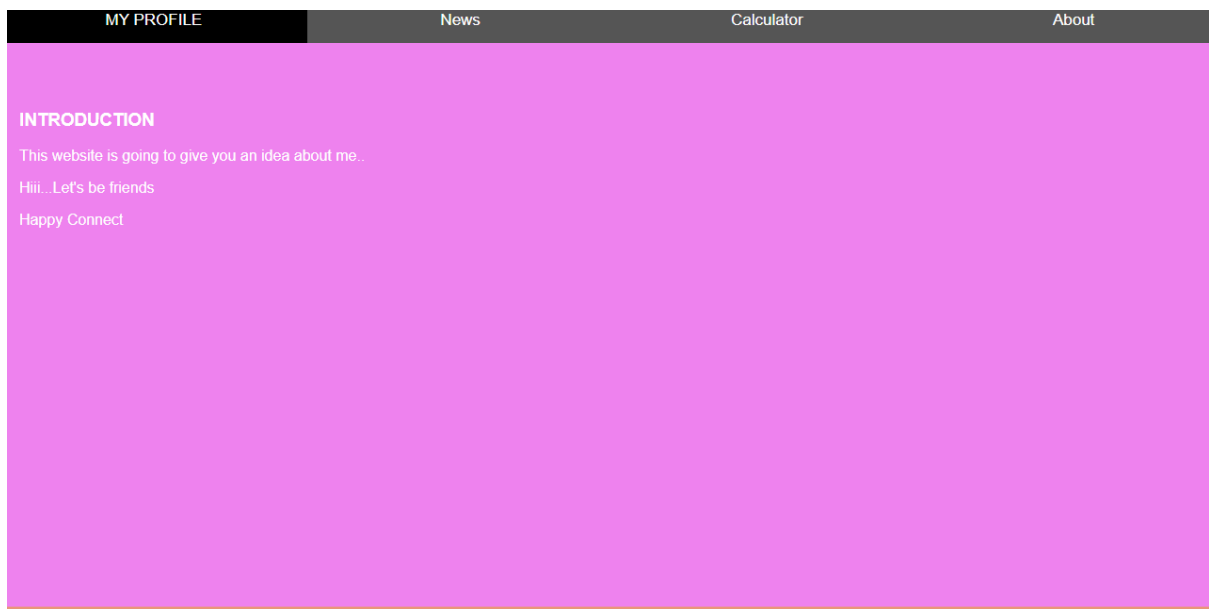
- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

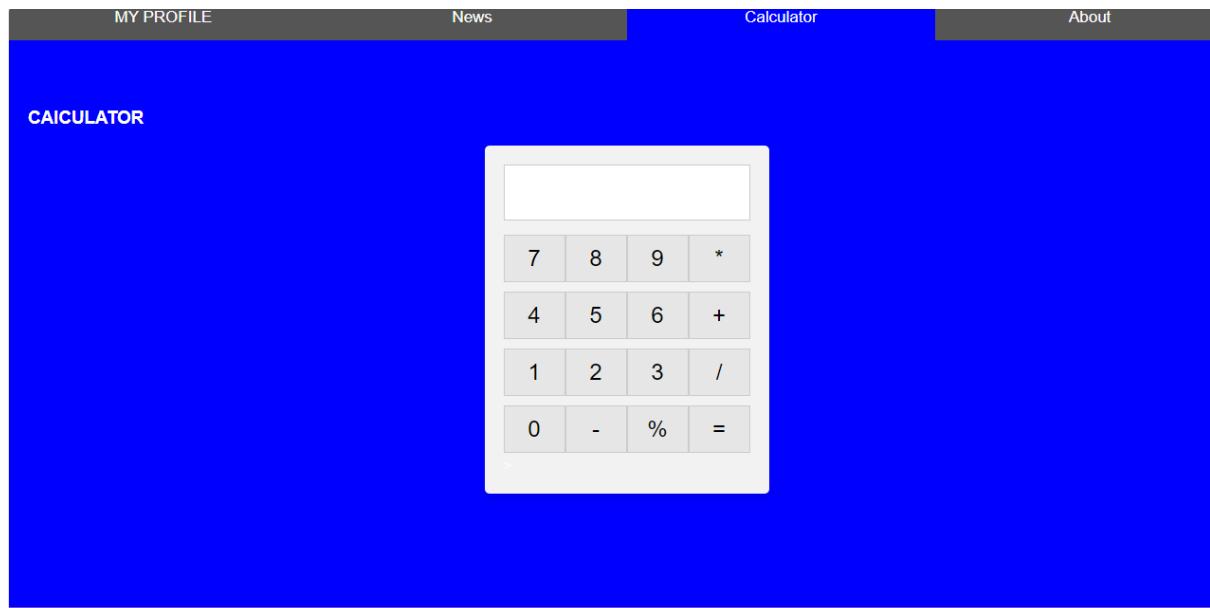
JavaScript Strings

The concat() Method

seafood

Example website that I created:





About

A blog is an informational website consisting of discrete, often informal diary-style text entries (posts). Posts are typically displayed in reverse chronological order so that the most recent post appears first, at the top of the web page. In the 2000s, blogs were often the work of a single individual, occasionally of a small group, and often covered a single subject or topic.

Author: Kirthana Praveen
kirthana.praveen@impelsys.com

APPLICATION PROGRAMMING INTERFACE(API)

An application programming interface (API) is code that enables two software programs to communicate. An API defines how a developer should request services from an operating system (OS) or other application, and expose data within different contexts and across multiple channels.

WHAT IS CRUD?

- CRUD refers to the four basic operations a software application should be able to perform – Create, Read, Update, and Delete.
- In such apps, users must be able to create data, have access to the data in the UI by reading the data, update or edit the data, and delete the data.
- In full-fledged applications, CRUD apps consist of 3 parts: an API (or server), a database, and a user interface (UI).
- The API contains the code and methods, the database stores and helps the user retrieve the information, while the user interface helps users interact with the app.
- You can make a CRUD app with any of the programming languages out there. And the app doesn't have to be full stack – you can make a CRUD app with client-side JavaScript.
- In fact, the app with which I will be showing you how create, read, update and delete operations work is made with client-side JavaScript.
- Each letter in the CRUD acronym has a corresponding HTTP request method.

CRUD OPERATION	HTTP REQUEST METHOD
Create	POST
Read	GET
Update	PUT or PATCH
Delete	DELETE

1. HTTP GET

Use *GET* requests **to retrieve resource representation/information only** – and not modify it in any way. As GET requests do not change the resource's state, these are said to be **safe methods**.

Example URIs:

HTTP GET <https://dummy.restapiexample.com/api/v1/employees>

Route	Method	Sample Json	Results
/employees	GET	-	<pre>{ "status": "success", "data": [{ "id": "1", "employee_name": "Tiger Nixon", "employee_salary": "320800", "employee_age": "61", "profile_image": "" }, ] }</pre>

2. HTTP POST

Use POST APIs **to create new subordinate resources**, e.g., a file is subordinate to a directory containing it or a row is subordinate to a database table.

Example URIs:

HTTP POST <https://dummy.restapiexample.com/api/v1/create>

Route	Method	Sample Json	Results
/create	POST	<code>{"name":"test","salary":"123","age":"23"}</code>	<pre>{ "status": "success", "data": { "name": "test", "salary": "123", "age": "23", "id": 25 } }</pre>

3. HTTP PUT

Use PUT APIs primarily **to update an existing resource (if the resource does not exist, then API may decide to create a new resource or not)**.

Example URIs:

HTTP PUT <https://dummy.restapiexample.com/api/v1/update/21>

Route	Method	Sample Json	Results
/update/{id}	PUT	<code>{"name":"test","salary":"123","age":"23"}</code>	<pre>{ "status": "success", "data": { "name": "test", "salary": "123", "age": "23", "id": 25 } }</pre>

4. HTTP DELETE

As the name applies, DELETE APIs **delete the resources** (identified by the Request-URI).

Example URIs:

HTTP DELETE <https://dummy.restapiexample.com/api/v1/delete/2>

Route	Method	Sample Json	Results
/delete/719	GET	-	<pre>{ "status": "success", "message": "successfully! deleted Records" }</pre>

Github Link: [Kirthanapraveen/MYINTERNSHIP \(github.com\)](https://github.com/Kirthanapraveen/MYINTERNSHIP)