TinkerHub
LBSCEK

Web Designing Bootcamp

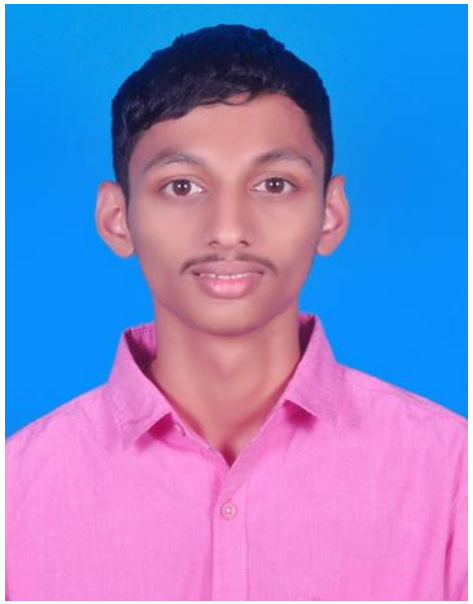
Pratheek Rao K B

Full Stack Web Developer

Contact Me:-

/pratheekraokb | GitHub /pratheekraokb | /pratheekraokb | +91 8078724204 | Blogger My Blogspot | PORTFOLIO My Portfolio

About Me



Hello, I'm Pratheek Rao K B, currently a Computer Science Engineering student in my 3rd Year at ***LBS College of Engineering, Kasaragod***. My educational journey started at Govt Vocational Higher Secondary School, Mulleria, and continued at Navajeevana Higher Secondary School, Peradala, where I completed my 11th and 12th grades.

During my academic journey, I had the opportunity to ***intern at HEXMOS***, gaining hands-on experience that allowed me to create dynamic, full-stack web project such as Poll Applications.

I've also ventured into the world of eCommerce as ***a freelancer***, where I've had the privilege of working independently on eCommerce projects, applying my skills and expertise to create seamless online shopping experiences for clients.

Currently, I hold the position of ***Program Facilitator of Tinkerhub*** at LBS College of Engineering, Kasaragod, where I contribute to the educational experience of my peers by facilitating programs and sharing my knowledge in the field of computer science and engineering.

/pratheekraokb

GitHub /pratheekraokb

/pratheekraokb

+91 8078724204

Blogger My Blogspot

PORTFOLIO My Portfolio

Topics Covered

1. **HTML Recall for the Project**
2. **CSS Recall for the Project**
3. **Introduction to Full Stack Web-Development**
   - **3.1** Frontend Development vs Backend Development
   - **3.2** API and Different Request Methods
   - **3.3** JSON
4. **Resources for Mastering the Web Development**

/pratheekraokb
/pratheekraokb
/pratheekraokb
+91 8078724204
My Blogspot
My Portfolio

LETS BEGIN

HTML

What is HTML?

- HTML stands for ***Hypertext Markup Language***. It is the standard markup language used to create web pages and is an essential part of the World Wide Web. HTML is used to structure the content of a web page, defining elements like headings, paragraphs, lists, links, images, and more.

- HTML documents consist of a series ***of elements or tags*** that are used to define the structure and content of a web page. These elements are ***enclosed in angle brackets***, and most of them come in pairs, with an ***opening tag and a closing tag***. The opening tag tells the browser how to interpret and display the content, while the closing tag marks the end of that content.

What is HTML?

```
<!DOCTYPE html>
<html>
<head>
   <title>My Webpage</title>
</head>
<body>
   <h1>Welcome to My Webpage</h1>
   <p>This is a paragraph of text.</p>
</body>
</html>
```

- **<!DOCTYPE html>** defines the document type and version of HTML being used.
- **<html>** encloses the entire HTML document.
- **<head>** contains metadata about the webpage, such as the title that appears in the browser's tab.
- **<title>** specifies the title of the webpage.
- **<body>** contains the main content of the webpage.
- **<h1>** defines a top-level heading.
- **<p>** defines a paragraph of text.

Basic HTML Tags

- **Div Tag**

- The tag defines a division or a section in an HTML document.
- The tag is easily styled by using the class or id attribute.
- The tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- Any sort of content can be put inside the tag!

- **For more reference goto** - *https://www.w3schools.com/tags/tag_div.ASP*

```
•<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>

</body>
</html>
```

Basic HTML Tags

- **<form> …… </form>**

  Eg:-

```
<form action="/action_page.php" method="get">
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <input type="submit" value="Submit">
</form>
```

For more reference goto - ***https://www.w3schools.com/tags/tag_form.asp***

Basic HTML Tags

- **Input Tag**

- The `<input>` tag specifies an input field where the user can enter data.
- The `<input>` element is the most important form element.
- The `<input>` element can be displayed in several ways, depending on the type attribute.

```
•<input type="button">
•<input type="checkbox">
•<input type="color">
•<input type="date">
•<input type="datetime-local">
•<input type="email">
•<input type="file">
•<input type="hidden">
•<input type="image">
```

- For more reference goto - *https://www.w3schools.com/tags/tag_input.asp*

```
•<input type="month">
•<input type="number">
•<input type="password">
•<input type="radio">
•<input type="range">
•<input type="reset">
•<input type="search">

•<input type="submit">
•<input type="tel">
•<input type="text"> (default value)
•<input type="time">
•<input type="url">
•<input type="week">
```

CSS

What is CSS?

- CSS, which stands ***for Cascading Style Sheets***, is a stylesheet language used for describing the presentation and visual design of web pages written in HTML or XML.
- CSS allows web developers to control how the elements of a web page are displayed, including aspects such ***as layout, colors, fonts, spacing,*** and more.
- It separates the content (defined in HTML) from its presentation (defined in CSS), making it easier to maintain and style web pages consistently across different devices and screen sizes.

What is CSS?

In this example, CSS is used to set the font family and background color for the entire page and to change the text color of the **<h1>** element. This demonstrates how CSS can be used to control the visual style of web content.

```
<!DOCTYPE html>
<html>
<head>
  <title>My Webpage</title>
  <style>
    /* CSS code within a <style> tag */
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
    }
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Welcome to My Webpage</h1>
  <p>This is a paragraph of text.</p>
</body>
</html>
```

```css
/* Reset some default styles for browsers */
body, h1, p {
  margin: 0;
  padding: 0;
}

/* Style the container */
.container {
  display: flex;
  justify-content: center; /* Center content horizontally */
  align-items: center; /* Center content vertically */
  height: 100vh; /* Full viewport height */
  background-color: #f0f0f0; /* Background color */
  font-family: Arial, sans-serif; /* Font family */
}

/* Style the content */
.content {
  text-align: center; /* Center text within the div */
  padding: 20px;
  border: 2px solid #333;
  border-radius: 10px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.2); /* Box shadow for a subtle effect */
}

/* Style the heading */
h1 {
  font-size: 24px; /* Font size for the heading */
  color: #333; /* Text color */
}

/* Style the paragraph */
p {
  font-size: 16px; /* Font size for the paragraph */
  color: #666; /* Text color */
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Div Example</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="content">
      <h1>Welcome to My Website</h1>
      <p>This is a simple example of using CSS to style a div element.</p>
    </div>
  </div>
</body>
</html>
```

JS

What is JS?

- JavaScript, often abbreviated as "JS," is a high-level, versatile, and widely-used programming language primarily known for its role in web development. It enables **interactive and dynamic behavior in web browsers**, making websites more responsive and engaging. JavaScript is not to be confused with Java, as they are distinct programming languages with different purposes and syntax.

Full Stack Web-Development

What is Full Stack Web development?

Full-stack web development refers to ***the practice of designing, developing, and maintaining both the front-end and back-end components*** of a web application. In essence, a full-stack developer is proficient in working with both the client-side (front-end) and server-side (back-end) technologies required to build a complete web application.

What is Full Stack Web development?

Front-End Development

1. **HTML/CSS/JavaScript**: The core technologies for building the front-end of web applications.
2. **React**: A JavaScript library for building user interfaces, developed by Facebook.
3. **Angular**: A front-end framework developed by Google for building dynamic web applications.

Back-End Development

1. **Node.js**: A JavaScript runtime that allows you to run JavaScript on the server side.
2. **Express.js**: A minimal and flexible Node.js web application framework.

What is Full Stack Web development?

Databases

1. **MySQL**: An open-source relational database management system.
2. **PostgreSQL**: An open-source, powerful, and extensible relational database.
3. **MongoDB**: A NoSQL database for handling unstructured or semi-structured data.
4. **SQLite**: A self-contained, serverless, and zero-configuration SQL database engine.
5. **Redis**: An in-memory data store often used for caching and session management.

Web Development Tools and Utilities:

1. **Visual Studio Code**: A popular code editor with extensions for various languages and frameworks.
2. **Git**: A version control system for tracking changes in code.
3. **GitHub/GitLab/Bitbucket**: Platforms for hosting and collaborating on code repositories.
4. **Docker**: A containerization platform for packaging applications and dependencies.
5. **Postman**: A tool for testing and documenting APIs.

What is API?

An API, or Application Programming Interface, is a ***set of rules and protocols that allows different software applications to communicate with each other***. It defines the methods and data formats that applications can use to request and exchange information, enabling them to work together seamlessly.

Examples of APIs include ***social media APIs*** (e.g., Facebook Graph API, Twitter API), ***payment gateway APIs*** (e.g., PayPal API, Stripe API), ***cloud service APIs*** (e.g., AWS API, Google Cloud API), and many others.

Different Request Methods

1.GET:

1. Purpose: ***Retrieve data*** from the server.
2. Idempotent: Yes (Multiple identical GET requests will have the same effect as a single request.)
3. Safe: Yes (It should not have any side effects on the server.)
4. Example: Fetching a web page, requesting an image, or retrieving data from an API.

2.POST:

1. Purpose***: Submit data to the server*** to create or update a resource.
2. Idempotent: No (Repeated POST requests may create multiple resources or have different effects.)
3. Safe: No (It can have side effects on the server, such as database updates.)
4. Example: Submitting a form, creating a new user account, or making a comment.

Different Request Methods

3. DELETE:

1. Purpose: ***Remove a resource from the server***.
2. Idempotent: Yes (Repeating the same DELETE request should have the same result as the initial request.)
3. Safe: No (It deletes the resource from the server.)
4. Example: Deleting a user account, removing a file, or canceling a reservation.

Other Examples:- HEAD, PATCH, PUT,etc

For more reference goto:- ***https://restfulapi.net/http-methods/***

What is JSON?

JSON, which stands for ***JavaScript Object Notation***, is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is often used to ***transmit data between a server and a web application***, or between different parts of an application.

```
{
  "name": "John Doe",
  "age": 30,
  "email": "john.doe@example.com",
  "isSubscribed": true,
  "hobbies": ["reading", "running", "cooking"],
  "address": {
    "street": "123 Main St",
    "city": "Anytown",
    "postalCode": "12345"
  }
}
```

- The data is enclosed in **curly braces {}** to define an object.
- Each key is a string enclosed in double quotes, followed by a colon (:).
- Values can be **strings** (enclosed in double quotes), **numbers, booleans** (true or false), **arrays** (lists of values enclosed in square brackets []), or **nested objects** (objects within objects).

Here's a breakdown of the example:

- "name" is a key with the value "John Doe", which is a string.
- "age" is a key with the value 30, which is a number.
- "email" is a key with the value "john.doe@example.com", which is a string.
- "isSubscribed" is a key with the value true, which is a boolean.
- "hobbies" is a key with an array value containing three strings: "reading", "running", and "cooking".
- "address" is a key with an object value, which contains its own key-value pairs.

Resources to Learn More

1. **W3 Schools**
   - HTML:- https://www.w3schools.com/html/
   - CSS:- https://www.w3schools.com/css/
   - Javascript:- https://www.w3schools.com/html/html_scripts.asp

2. **Documentation**
   - HTML:- https://developer.mozilla.org/en-US/docs/Web/HTML
   - CSS:- https://developer.mozilla.org/en-US/docs/Web/CSS
   - JS:- https://developer.mozilla.org/en-US/docs/Web/JavaScript

You can also refer youtube channels like ***CodeWithHarry*** (Hindi), ***freeCodeCamp***(English), ***brototype***(Malayalam), and many more.

- You can learn ***Django (Framework)*** through their official documentation.(https://docs.djangoproject.com/en/4.2/)

THANKYOU

/pratheekraokb
/pratheekraokb
/pratheekraokb
+91 8078724204
My Blogspot
My Portfolio