**SONIYA KAMBLE – Assignment 5**

1. **By using CSS properties, create following web Page as shown.**
   * Output



* + home.html - Code

<html lang="en"></html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>ITVEDANT</title>

<link href="style.css" rel="stylesheet" />

</head>

<body>

<div>

<h1>ITVEDANT</h1>

<p>India’s Leading and Trusted IT Training Institute Offering Classroom & Online Training</p>

<img src="itvedant.png" />

</div>

</body>

</html>

* + style.css- Code

div {

display: flex;

flex-direction: column;

justify-content: flex-start;

align-items: center;

height: 100vh;

}

div \*:first-child {

margin-top: 50px;

}

div \* {

padding: 10px;

margin-top: 0px;

box-shadow: 5px 5px 2px rgba(0, 0, 0, 0.5);

}

div h1{

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.6);

}

div p{

font-style: italic;

border: 1px solid black;

}

div img {

width: 50%;

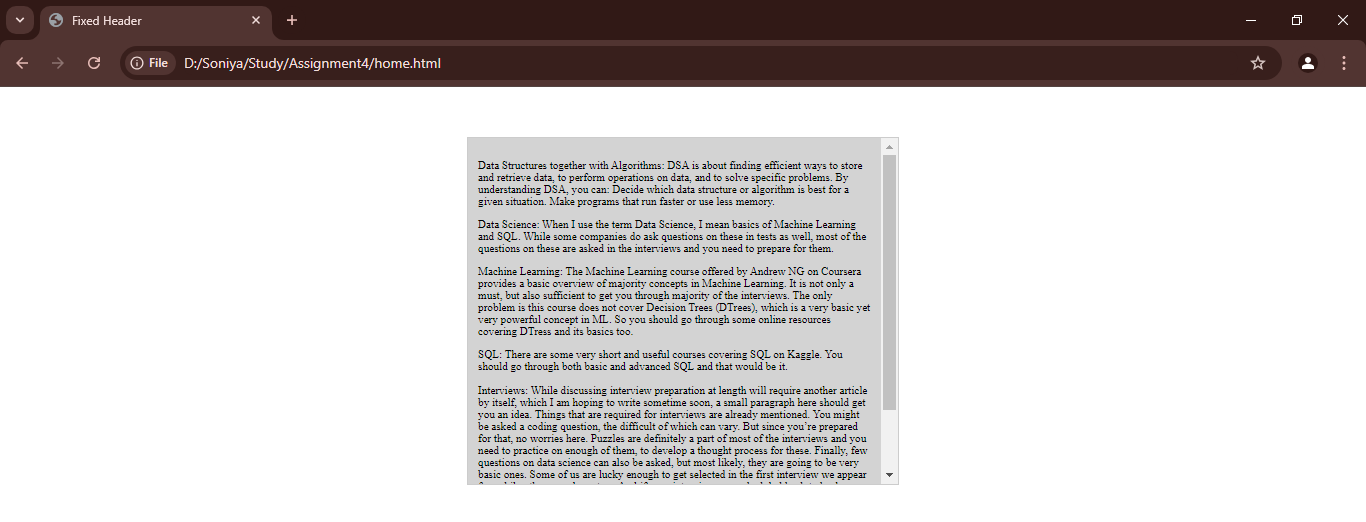
height: auto;

max-width: 500px;

max-height: 500px;

}

1. **Using CSS Position Properties, design the given web page.**
   * Output



* + home.html - Code

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Fixed Header</title>

<link href="newstyle.css" rel="stylesheet" />

</head>

<body>

<div class="scrollable-div">

<p>Data Structures together with Algorithms:

DSA is about finding efficient ways to store and retrieve data, to perform operations on data,

and to solve specific problems. By understanding DSA, you can:

Decide which data structure or algorithm is best for a given situation.

Make programs that run faster or use less memory.

</p>

<p>

Data Science:

When I use the term Data Science, I mean basics of Machine Learning and SQL.

While some companies do ask questions on these in tests as well,

most of the questions on these are asked in the interviews and you need to prepare for them.

</p>

<p>

Machine Learning:

The Machine Learning course offered by Andrew NG on Coursera provides a basic overview

of majority concepts in Machine Learning. It is not only a must, but also sufficient

to get you through majority of the interviews. The only problem is this course does

not cover Decision Trees (DTrees), which is a very basic yet very powerful concept in ML.

So you should go through some online resources covering DTress and its basics too.

</p>

<p>

SQL: There are some very short and useful courses covering SQL on Kaggle.

You should go through both basic and advanced SQL and that would be it.

</p>

<p>

Interviews:

While discussing interview preparation at length will require another article by itself,

which I am hoping to write sometime soon, a small paragraph here should get you an idea.

Things that are required for interviews are already mentioned.

You might be asked a coding question, the difficult of which can vary.

But since you’re prepared for that, no worries here.

Puzzles are definitely a part of most of the interviews and you need to practice on enough of them,

to develop a thought process for these. Finally, few questions on data science can also be asked,

but most likely, they are going to be very basic ones.

Some of us are lucky enough to get selected in the first interview we appear for,

while others maybe not so. And if your interviews are scheduled back to back,

poor performance in one can lead to mental stress which will ensure poorer performance in next interview too,

and the cycle would perpetuate.

Therefore, it is of paramount importance that you don’t let the results of previous

interviews hurt your performance in the present or coming interviews.

</p>

</div>

</body>

</html>

* + newstyle.css- Code

html, body {

height: 100%;

font-family: 'Times New Roman', Times, serif;

font-size: smaller;

margin: 0;

}

body {

display: flex;

flex-direction: column;

justify-content: flex-start;

align-items: center;

height: 100vh;

}

.scrollable-div {

margin-top: 50px;

position: fixed;

width: 30%;

height: 50%;

overflow: auto;

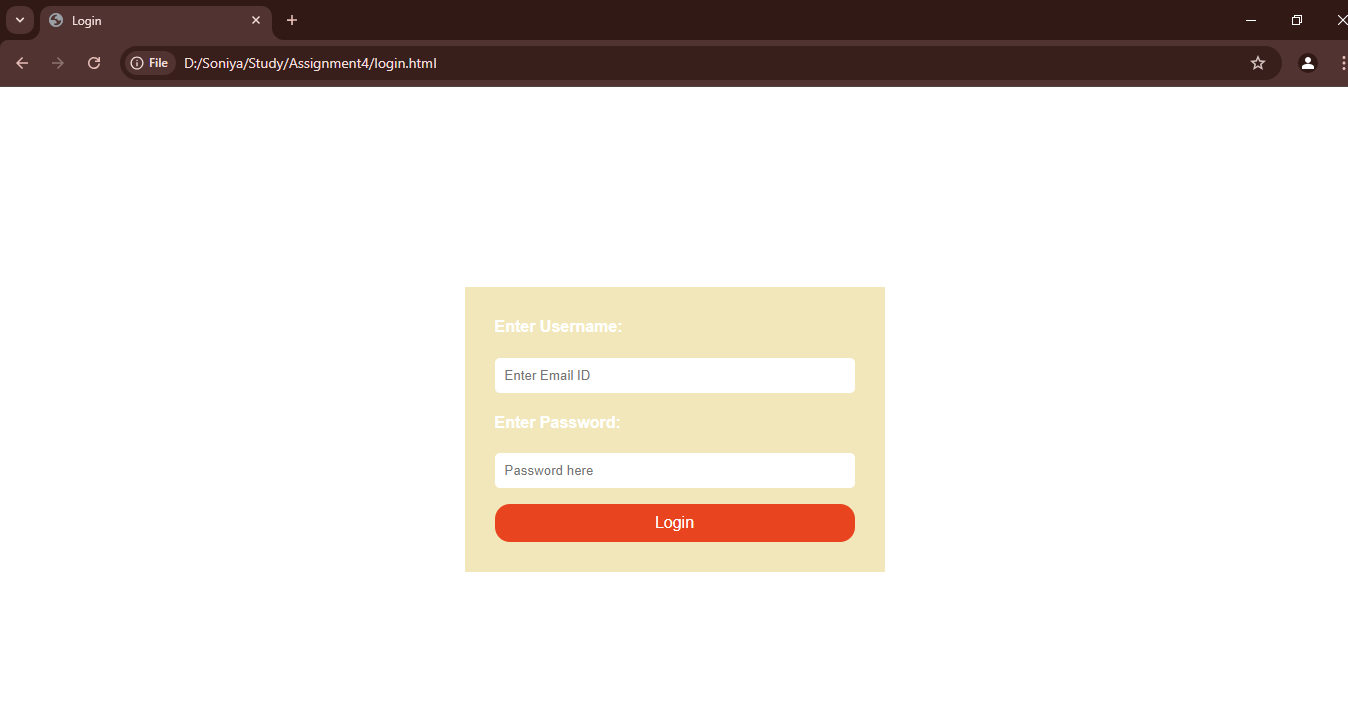
border: 1px solid #ccc;

padding: 10px;

background-color: lightgrey;

}

1. **Design the Login Form as shown below.**
   * Output



* + home.html - Code

<html lang="en"></html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<link href="loginstyle.css" rel="stylesheet" />

</head>

<body>

<form method="POST" enctype="multipart/form-data">

<div>

<h4>

<label for="email">Enter Username: </label>

</h4>

<p>

<input type="email" id="email" placeholder="Enter Email ID" required>

</p>

<h4>

<label for="password"></label>Enter Password: </label>

</h4>

<p>

<input type="password" id="password" placeholder="Password here" required>

</p>

<input type="submit" id="submit" class="styled-button" value="Login">

</div>

</form>

</body>

</html>

* + loginstyle.css- Code

form {

max-width: 400px;

margin: 200px auto;

padding: 10px;

font-family: Arial, Helvetica, sans-serif;

font-size: medium;

background-color: #F1E7BA;

}

h4 {

color: white;

}

div{

margin: 20px;

}

input {

width: 100%;

padding: 10px;

border: none;

border-radius: 5px;

}

.styled-button {

background-color: #E84420;

color: white;

font-size: medium;

border: none;

padding: 10px 20px;

border-radius: 15px;

cursor: pointer;

}

.styled-button:hover {

background-color: #d43d1c;

}

1. **What is difference between margin and padding?**

**Margin** is the space outside the element's border. It creates space between the element and other elements around it.

#### ***Example:***

* If you apply margin: 20px; to an element, there will be 20px of space between that element and any adjacent elements.

**Padding** is the space inside the element, between the content and the element's border. It creates space between the content of the element and its edge.

#### ***Example:***

* If you apply padding: 20px; to an element, the content inside the element will have 20px of space between it and the element's border.

<div style="margin: 20px; padding: 10px; border: 2px solid black;">

This div has 20px margin and 10px padding.

</div>

1. **How does box-sizing property affect the layout.**

The box-sizing property in CSS controls how the total width and height of an element is calculated, affecting how padding and borders are included in the element’s dimensions. It helps determine whether the padding and borders are inside or outside of the specified width and height of an element.

### **box-sizing Values:**

1. content-box (default):

* The width and height properties only apply to the element’s content. The padding and border are added outside the specified width and height.

1. border-box:

* The width and height properties include the element’s content, padding, and border. In this case, padding and border are inside the specified width and height.

#### ***Example:***

\* {

box-sizing: border-box;

}

1. **what is the position property used for?**

The position property in CSS is used to define how an element is positioned in the document. It determines how the element will be placed relative to its containing block, other elements, or the viewport.

#### ***Types of position:***

#### ***static (default):***

* This is the default positioning for all elements. Elements are positioned according to the normal flow of the document.

#### ***relative:***

* The element is positioned relative to its normal position in the document. You can move it using the top, left, right, or bottom properties, but its original space is preserved.

#### ***absolute:***

* The element is removed from the normal document flow and is positioned relative to its nearest positioned ancestor (non-static). If there is no positioned ancestor, it will be positioned relative to the initial containing block (usually the document body).

#### ***fixed:***

* The element is positioned relative to the browser window (viewport), and it stays fixed in place even when the page is scrolled.

#### ***sticky:***

* The element is positioned relative to its normal flow until it reaches a specified position (using top, left, right, or bottom), at which point it becomes fixed in place (sticky) within its parent container.

div {

position: sticky;

top: 0;

}

1. **What is the use of float property.**

The float property in CSS is used to position an element to the left or right within its containing element. It allows text and inline elements (like images) to wrap around it, creating a more complex layout. The float property is commonly used to create column-based layouts, align images, and position elements beside each other.

#### ***Values of float:***

* left, right, none (default), inherit (from parent)

img {

float: left;

margin-right: 10px; /\* Adds space between the image and the surrounding content \*/

}