Name:Rakib Rary,Shailesh Ghale, Djiedjom Gbonkou, Hang Lin

HTML

* <nav>
* <ul>
* <li><a href="Home.html">Home</a></li>
* <li><a href="About.html">About</a></li>
* <li><a href="Services.html">Services</a></li>
* <li><a href="Contact.html">Contact</a></li>
* </ul>
* </nav>
* Navigation bar connecting 4 webpage files together implemented using an unordered list <ul>.
* <li><a href="https://www.thingiverse.com/" target="\_blank">Thingiverse</a></li>
* The code is for links and when pressed opens up into a new tab<head>
* <meta charset="UTF-8">
* <title>My Website</title>
* <link rel="stylesheet" href="About\_Style.css">
* <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.3.0/css/all.min.css">
* <script src="script.js"></script>
* </head>
* This is the <head> section of the About.html page. It contains the website title and the links to the CSS and JS.

<div style="display:flex;">

<img src="3D\_Print\_Photos/3drobot.JPG" alt="Image" style="width:50%;height:auto;">

<p style="margin-left:2%; padding-top: 80px; font-family: 'Courier New', Courier, monospace;font-size: 24px;"><b>

Welcome to my 3D print webpage! I am excited to share you our 3d printing technology. You can find on our website

many educational resource links, 3d print files links, list of our prices, and much more! We do custom pieces and

money back guarantee if you're not satisfied. If you have any questions please feel free to reach out and thank you

for visiting your best 3d print shop page!

</b></p>

</div>

* This <div> section has a few things in it. There’s a photo inserted and also a paragraph giving a summary of what the website is about. We also have an inline CSS for the <p> and <div> sections.

<div class ="search">

<input type="text" class="input" placeholder="Search...">

<button class="btn">

<i class = "fas fa-search"></i>

</button>

</div>

* This sets up a search box along with a button and an input field.

<footer>

<p>Copyright © 2023 My Website<br> Rakib Rary, Shailesh Ghale, Djiedjom Gbonkou, Hang Lin</p>

</footer>

* The <footer> section defines a footer along with the creators names and the copyright symbol.

<header>

<h1>Contact</h1>

</header>

* The <header> section has “Contact” in big letters for the Contact.html page.

<div style="text-align:center;">

<form>

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label>Gender:</label>

<input type="radio" id="male" name="gender" value="male" required>

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="female" required>

<label for="female">Female</label><br><br>

<label>Subject:</label>

<input type="checkbox" id="sales" name="subject" value="sales">

<label for="sales">Sales</label>

<input type="checkbox" id="support" name="subject" value="support">

<label for="support">Support</label><br><br>

<label for="message">Message:</label>

<textarea id="message" name="message" rows="5" required></textarea><br><br>

<input type="submit" value="Submit">

<input type="reset" value="Reset">

</form>

<form>

<p>Welcome to the newsletter subscribe form!!!</p>

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" name="email" required><br><br>

<input type="submit" value="Subscribe">

<input type="reset" value="Reset">

</form>

</div>

* This generates a div element with a style tag that uses the text-align CSS property to center the information within it. There are two forms in the <div> section.
* The first form has a label and input element for the user's name, radio buttons for the user's gender, checkboxes for the message's subject, a label and textarea for the user to enter their message, and input buttons when they want to submit and reset the form.
* The second form is a newsletter subscription form, which has a <p> element including a welcome message, a label and input elements for the user's name, a label and email address. There are also input buttons for subscribing or resetting it the form.

<script src="Contact\_js.js"></script>

<br></br>

<br></br>

<br></br>

<br></br>

<footer>

<p>Copyright © 2023 My Website<br> Rakib Rary, Shailesh Ghale, Djiedjom Gbonkou, Hang Lin</p>

</footer>

* This contains the link to the JS for the Contact.html page along with line breaks and a footer.

<div class="faq-container">

<div class="faq active">

<h3 class="faq-title">

What is 3D printing?

</h3>

<p class="faq-text">

3D printing is the automated process of building a three-dimensional object by adding material rather than taking material away (as in drilling or machining).

</p>

<button class="faq-toggle">

<i class="fas fa-chevron-down"></i>

<i class="fas fa-times"></i>

</button>

</div>

* This generates a container that will display commonly asked questions (FAQs) about a specific topic. The first FAQ asks, "What is 3D printing?" and then answers, "3D printing is the automated technique of producing a three-dimensional item by adding material rather than taking material away (as in drilling or machining)". The FAQ is placed inside a div element with the class faq-container. Each FAQ is enclosed within another div element with the class name faq inside this container. The active class has been assigned to the first FAQ, indicating that it is now displaying on the page. The FAQ question is in a <h3> element with the class name faq-title, and the response is in a <p> element with the class name faq-text.
* Lastly, underneath the response is a button with the class name faq-toggle. The icons inside the button suggest that the user may control the visibility of the response, with a down-facing chevron indicating that the answer can be extended, and a fa times symbol indicating that the answer can be closed.

<div class="faq">

<h3 class="faq-title">

How does 3D printing work?

</h3>

<p class="faq-text">

To create a 3D printed object, you use an “additive process”.

The three-dimensional object is created by laying down successive layers of material until the object is finished.

</p>

<button class="faq-toggle">

<i class="fas fa-chevron-down"></i>

<i class="fas fa-times"></i>

</button>

</div>

* This defines a frequently asked question (FAQ) container. A <div> element with the class "faq" defines the structure of the container. Within this <div>, there is a <h3> element with the class "faq-title" that shows the question "How does 3D printing work?". A <p> element with the class "faq-text" displays the answer to the question. After that, there is a button element with the class "faq-toggle" that, when pressed, reveals or conceals the response text. The Font Awesome library provides the icons within the <button> element, which vary based on whether the response is currently displayed or hidden. While the answer is concealed, the fa-chevron-down icon appears, and when the answer is shown, the fa-times icon appears. By pressing the button, the user may quickly alter the visibility of the response text.

<div class="faq">

<h3 class="faq-title">

What are the benefits of 3D printing?

</h3>

<p class="faq-text">

Like most emerging technologies, 3D printing offers benefits in a lot of areas. These include improvements in financial, logistical, healthcare, creative and environmental areas.

</p>

<button class="faq-toggle">

<i class="fas fa-chevron-down"></i>

<i class="fas fa-times"></i>

</button>

</div>

<div class="faq">

<h3 class="faq-title">

Is there a difference between 3D printing and additive manufacturing?

</h3>

<p class="faq-text">

The short answer is no. The term “3D printing” comes from the use of inkjet printer heads (in the first 3D printers) to deposit, either layers of UV-curable photopolymer resin or a binding material onto a layer of powder in a powder bed process. However, the term now universally encompasses all additive manufacturing technologies.

</p>

<button class="faq-toggle">

<i class="fas fa-chevron-down"></i>

<i class="fas fa-times"></i>

</button>

</div>

<div class="faq">

<h3 class="faq-title">

Can I try 3D printing before buying a 3D printer?

</h3>

<p class="faq-text">

Yes. This is a great place to try out and use 3D printers without having to buy one.

Just look up the nearest location near you and buy a membership.

A membership for 3D printing costs around $5 per month which is a lot cheaper than buying a 3D printer yourself.

</p>

<button class="faq-toggle">

<i class="fas fa-chevron-down"></i>

<i class="fas fa-times"></i>

</button>

</div>

* These <div> elements follow the same structure as the first and one above answering frequently asked questions.

<div class ="search">

<input type="text" class="input" placeholder="Search...">

<button class="btn">

<i class = "fas fa-search"></i>

</button>

</div>

* This creates a search bar interface with a text input field and a button to start the search implemented using structures from the Udemy course. The <div> element with the class "search" acts as a container for the search bar interface. There are two components within the <div> section:
  + A text input element with the class name "input" and the placeholder text "Search...". This is the area where the user will enter their search phrase.
  + A button element with the class "btn" and an i element within with the class "fas fa-search". This button is used to initiate the search activity. When you click it, a JavaScript code that handles the search query is executed.

The i element with the class name "fas fa-search" displays a magnifying glass, which is a Font Awesome icon. This is simply decorative and may be changed with any other symbol.

<div id="myModal" class="modal">

<span class="close">&times;</span>

<img class="modal-content" id="img01">

<div id="caption"></div>

</div>

* This generates a simple modal window with a picture and a close button. JS may be used to open and close the modal window which is a pop-up window that opens on the top of the primary content and requires interaction from the user before going back to it.

<table>

<thead>

<tr>

<th>3D Print</th>

<th>Price</th>

</tr>

</thead>

<tbody>

<tr>

<td>3D Printed Keychain</td>

<td>$5.99</td>

</tr>

<tr>

<td>3D Printed Phone Case</td>

<td>$19.99</td>

</tr>

<tr>

<td>3D Printed Figurine</td>

<td>$34.99</td>

</tr>

<tr>

<td>3D Printed Jewelry</td>

<td>$24.99</td>

</tr>

<tr>

<td>3D Printed Replacement Part</td>

<td>$9.99</td>

</tr>

<tr>

<td>3D Print Personals</td>

<td>$9.99 - $$$</td>

</tr>

</tbody>

</table>

* This is from the Services.html page which as the <table> tag indicates creates a table displaying the different pricing options for our 3D printing services. <tr> creates the rows and <td>, the data that goes with those rows.

<ul class="links" style="padding-left: 150px;">

<li><a href="https://www.thingiverse.com/" target="\_blank">Thingiverse</a></li>

<li><a href="https://www.myminifactory.com/" target="\_blank">MyMiniFactory</a></li>

<li><a href="https://www.cults3d.com/" target="\_blank">Cults</a></li>

<li><a href="https://pinshape.com/" target="\_blank">Pinshape</a></li>

<li><a href="https://www.tinkercad.com/" target="\_blank">Tinkercad</a></li>

</ul>

* Also from the Services.html page this <ul> unordered list displays a variety of links to check out but also 3D designs uusing the <a> tag. The target in the <a> tag is set to "\_blank," which means that the link will open in a new tab.

CSS

body {

font-family: Arial, sans-serif;

font-size: 16px;

line-height: 1.5;

color: #333;

background-color: #f5f5f5;

}

header {

display: flex;

font-family: 'Courier New', Courier, monospace;

justify-content:center;

align-items: center;

}

header h1 {

font-size: 42px;

font-weight: bold;

color: #333;

}

nav ul {

font-family: 'Courier New', Courier, monospace;

font-size: 24px;

display: flex;

justify-content:center;

align-items: center;

list-style: none;

background-color: grey;

margin:0;

padding:0;

}

nav ul li {

margin: 0 20px;

}

nav ul li a {

text-decoration: none;

color: #333;

}

nav ul li a:hover {

color: purple;

}

main {

display: flex;

flex-wrap: wrap;

justify-content: space-between;

align-items: flex-start;

margin: 20px 0;

}

footer {

font-family: 'Courier New', Courier, monospace;

text-align: center;

font-size: 18px;

padding: 20px 0;

background-color: #333;

color: #fff;

position:fixed;

bottom:0;

left:0;

width:100%;

height: 70px;

}

.search {

position: relative;

height:50px;

}

.search .input{

background-color:#fff;

border:0;

font-size:18px;

padding:15px;

height:20px;

width:20px;

transition:width 0.3s ease;

}

.btn{

background-color:#fff;

border:0;

cursor:pointer;

font-size:24px;

position: absolute;

top:0;

left:0;

height:50px;

width:50px;

transition:transform 0.3s ease;

}

.btn:focus,

.input:focus{

outline:none;

}

.search.active .input{

width:200px;

}

.search.active .btn{

transform:translateX(198px);

}

* CSS for About.html page
* All body elements are assigned specific font family and size, color, line height and background color. The header is configured to show as a flex container, with its contents centered horizontally and vertically by justify-content and align-items. The font style is 'Courier New,' Courier, monospace. The h1 element in the header has a bigger size for the font, a bold font weight, and a distinct text color.
* The nav ul element is designed to show as a flex container, with each of its list elements (nav ul li) having a specific defined margin. Grey is set to be the background color. The text decoration and color of the nav ul li an element are distinct, and the color changes on hover. The main element is set up to appear as a flex container, and its elements (not defined in this CSS) are given the flex-wrap: wrap attribute, enabling them to wrap to several lines. The container has space between its components and is aligned at the top using align-items.
* The.search class has a fixed height, and its components (.search,.input, and .btn) have explicit layouts. The.input component has a white background, without borders, one font size, and padding. While the .search component is active, it also gets a transition effect added to its width. The .btn component just like the .input component also has a white background, without borders, a particular font size, and is positioned at the top left of the.search container. While the .search element is active, it also gets a transition effect added to its transform attribute.Whenever the .search element is active, the .search.active class is added to it, and it changes the width and position of its child components to produce a search bar that extends when clicked. When one clicks on the.btn and.input elements, the:focus "class" is also called upon to delete the outlines surrounding them.

form {

border: 1px solid black;

padding: 10px;

margin: 0 auto;

float: left;

margin-left: 100px;

}

* Part of CSS for Contact.html
* CSS for Contact.html has about the same styling attributes as the About.html page
* The form border is set to a solid black 1 pixel with a 10 pixel of padding on all sides. float: left is used to slide the form to the left of the container. The form has a left margin of 100 pixels

.faq-container {

max-width: 600px;

margin: 0 auto;

}

.faq {

background-color: transparent;

border: 1px solid #9fa4a8;

border-radius: 10px;

margin: 20px 0;

padding: 30px;

position: relative;

overflow: hidden;

transition: 0.3s ease;

}

.faq.active {

background-color: #fff;

box-shadow: 0 3px 6px rgba(0, 0, 0, 0.1), 0 3px 6px rgba(0, 0, 0, 0.1);

}

.faq.active::before,

.faq.active::after {

content: '\f075';

font-family: 'Font Awesome 5 Free';

color: #2ecc71;

font-size: 7rem;

position: absolute;

opacity: 0.2;

top: 20px;

left: 20px;

z-index: 0;

}

.faq.active::before {

color: #3498db;

top: -10px;

left: -30px;

transform: rotateY(180deg);

}

.faq-title {

margin: 0 35px 0 0;

}

.faq-text {

display: none;

margin: 30px 0 0;

}

.faq.active .faq-text {

display: block;

}

.faq-toggle {

background-color: transparent;

border: 0;

border-radius: 50%;

cursor: pointer;

display: flex;

align-items: center;

justify-content: center;

font-size: 16px;

padding: 0;

position: absolute;

top: 30px;

right: 30px;

height: 30px;

width: 30px;

}

.faq-toggle:focus {

outline: 0;

}

.faq-toggle .fa-times {

display: none;

}

.faq.active .faq-toggle .fa-times {

color: #fff;

display: block;

}

.faq.active .faq-toggle .fa-chevron-down {

display: none;

}

.faq.active .faq-toggle {

background-color: #9fa4a8;

}

* Part of CSS for faq.html
* It provides styling for the FAQ container as well as each individual FAQ article. The FAQ item has a border, a background color, and rounded corners.
* Whenever you activate (click on it) a FAQ item, it modifies the background color and adds a box shadow.
* It displays an icon before and after the question, adjusts the hue of the icon and background, and reveals the answer also.
* Last but not least it styles a toggle button that reveals or conceals the question's response. When the FAQ item is active, the toggle button in the upper right corner of the FAQ item changes its icon which is from Font Awesome 5 Free.

JAVASCRIPT

const search = document.querySelector('.search')

const btn = document.querySelector('.btn')

const input = document.querySelector('.input')

btn.addEventListener('click',()=>{

search.classList.toggle('active')

input.focus()

})

* This is the JS for the About.html page
* It selects elements from the About.html page and adds an event listener.
* The const term is used to create three variables (search, button, and input) and assign values to them using the querySelector method, which is used to pick HTML components relying on their class names (.search, .btn, and .input).
* With the addEventListener method, the btn element is looking for a click event. When you click the button, a nameless arrow function is invoked. The classList.toggle method is used within the arrow function to switch the active class on the search element. This indicates that if the active class is not already existent, it will be added; otherwise, it will be deleted.
* Lastly, on the input element, the focus method is called, which provides it focus (— in other words, it'll become the component that may receive input from the user/visitor)

var modal = document.getElementById("myModal");

var img = document.getElementsByTagName("img");

var modalImg = document.getElementById("img01");

var captionText = document.getElementById("caption");

for(var i=0; i<img.length; i++){

img[i].onclick = function(){

modal.style.display = "block";

modalImg.src = this.src;

captionText.innerHTML = this.alt;

}

}

var span = document.getElementsByClassName("close")[0];

span.onclick = function() {

modal.style.display = "none";

}

var images = document.getElementsByTagName('img');

for (var i = 0; i < images.length; i++) {

images[i].addEventListener('click', function() {

this.classList.toggle('enlarged');

});

}

const search = document.querySelector('.search');

const btn = document.querySelector('.btn');

const input = document.querySelector('.input');

btn.addEventListener('click',()=>{

search.classList.toggle('active');

input.focus();

})

const loveMe = document.querySelector(".container");

const times = document.querySelector("#times");

let clickTime = 0;

const createHeart = (e) => {

const heart = document.createElement("i");

heart.classList.add("fas");

heart.classList.add("fa-heart");

const x = e.clientX;

const y = e.clientY;

const leftOffset = e.target.offsetLeft;

const topOffset = e.target.offsetTop;

const xInside = x - leftOffset;

const yInside = y - topOffset;

heart.style.top = `${yInside}px`;

heart.style.left = `${xInside}px`;

loveMe.appendChild(heart);

times.innerHTML = ++timesClicked;

setTimeout(() => heart.remove(), 1000);

};

loveMe.addEventListener("click", (e) => {

if (clickTime === 0) clickTime = new Date().getTime();

else {

if (new Date().getTime() - clickTime < 800) {

createHeart(e);

clickTime = 0;

} else clickTime = new Date().getTime();

}

});

const textEl= document.getElementById('autoText')

const text = 'CLICK ON THE IMAGES!!!'

let idx=1;

let speed = 300

writeText()

function writeText(){

textEl.innerText= text.slice(0,idx)

idx++

if(idx>text.length){

idx=1

}

setTimeout(writeText,speed)

* This is the JS for the Home.html page

Here is how it works:

* First, when a picture is selected, a modal window opens up. The code retrieves all of the photos on the page and assigns an event listener to each of them. When you click on a photo, a modal window appears and shows the photo with a description. The alt text is the one displayed as a description/caption.
* Then a search bar that shows up when the magnifier button is clicked is created. A search button event listener that toggles a "active" attribute on a search bar, enabling it to expand or collapse is added.
* After that a functionality wherein double-clicking on a container adds a heart icon to the place of the click is created. The container component, the heart symbol, and a counter element are then inserted. Whenever one clicks on the container, it is assessed and determined whether or not it is a double click. If this is the case, the heart icon is added to the click location, and the counter is incremented. If not, the timestamp of the click is recorded.
* At last an auto-typing capability for a type element on the page is created by the JS. It retrieves the text component, a text string, and a speed. It also uses recursion to add one text character at a time to the element, with a latency determined by the speed. When the function reaches the end of the text, it restarts. This simply means that it creates a text animation that shows a message on the webpage by displaying each letter one at a time.