**PART A**

(Part A: TO BE REFFERED BY STUDENTS)

**A.1 AIM:**

Develop a web based Mini Project

**A.2 Pre requisite:**

Basic Knowledge of HTML, CSS, JavaScript, Server Side Scripting, JQuery, Bootstrap, Angular-JS, PHP, Browser.

**A.3 Outcome:**

After successful completion of this experiment students will be able to:

1. Design a website using various web programming languages learnt.

**A.4 Theory:**

A mini project is desirable to be completed by a group of two or three students that cover

following tools.

HTML

DHTML

CSS

Java Script  
 PHP

Bootstrap

JQuery

Angular JS

**A.5 Procedure/Task:**

1. Develop a website using Web technologies learnt.
2. The site should be attractive & interactive.
3. Prepare the document. Save and close the file and name it as RollNo.\_MiniProject.

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

|  |  |
| --- | --- |
| Roll No. : C030, C022, C020 | Name: Raj Kothari, Bhavya Sanghrajka, Moksh Shah |
| Class :B | Batch :B1 |
| Date of Experiment : 11-09-23 | Date/Time of Submission : 30-10-23 |
| Grade : |  |

**B.1 Project Details**

**TITLE: RBM ROBOTICS WEBSITE**

**DESCRIPTION: WITH THE HELP OF HTML,CSS,JS, ETC WE ARE CREATING A ROBOTICS WEBSITE FOR EASIER CAD DRAWING OF METAL PARTS, AND ALL METAL AND PLASTIC PARTS RELATED TO ROBOTICS & DRONES**

***1) Motivation to take the proposed project***

* We are inspired to take on this project because it aligns perfectly with our passion for innovation. The opportunity to work on cutting-edge technology and create something that can truly make a difference in the world is what drives us every day. We have been participating in robotics competition since we were 10. What motivates us to take on this project is the chance to tackle a challenge that others might shy away from. I thrive on pushing boundaries and finding solutions where none seem to exist, and this project offers that exciting prospect.

***2) Working of project***

* I'm motivated to lead this project because it embodies my commitment to continuous learning. Every project I undertake is an opportunity for personal and professional growth, and I'm excited to embrace the steep learning curve this project offers. We have made a website of robotics and drone equipment and tools which are used, primarily, in the national and international robotics competitions worldwide, like FTC, FRC, etc. An individual organization can refer to the manuals, CAD drawings, unique numbers of each equipment and tools, and order the part they require for the students or their company participating in their respective events. We have used html, javascript, css, etc., coding languages to provide an aid to us to develop a magnificent website.

***3) Advantages & disadvantages of project***

Advantages:

* Many Functionalities such as AddToCart buttons
* Carousel for effective Image Motion
* Reaching Out to the Makers of the website is easier
* Google Sheet Linking for Queries and Sin-In or Sign-Up Information
* Interactive with several animations

Disadvantages:

* Not Active
* Only Compatible to Laptops
* Linking only limited to Google Sheets
* Hard-Coded so Time-Consuming

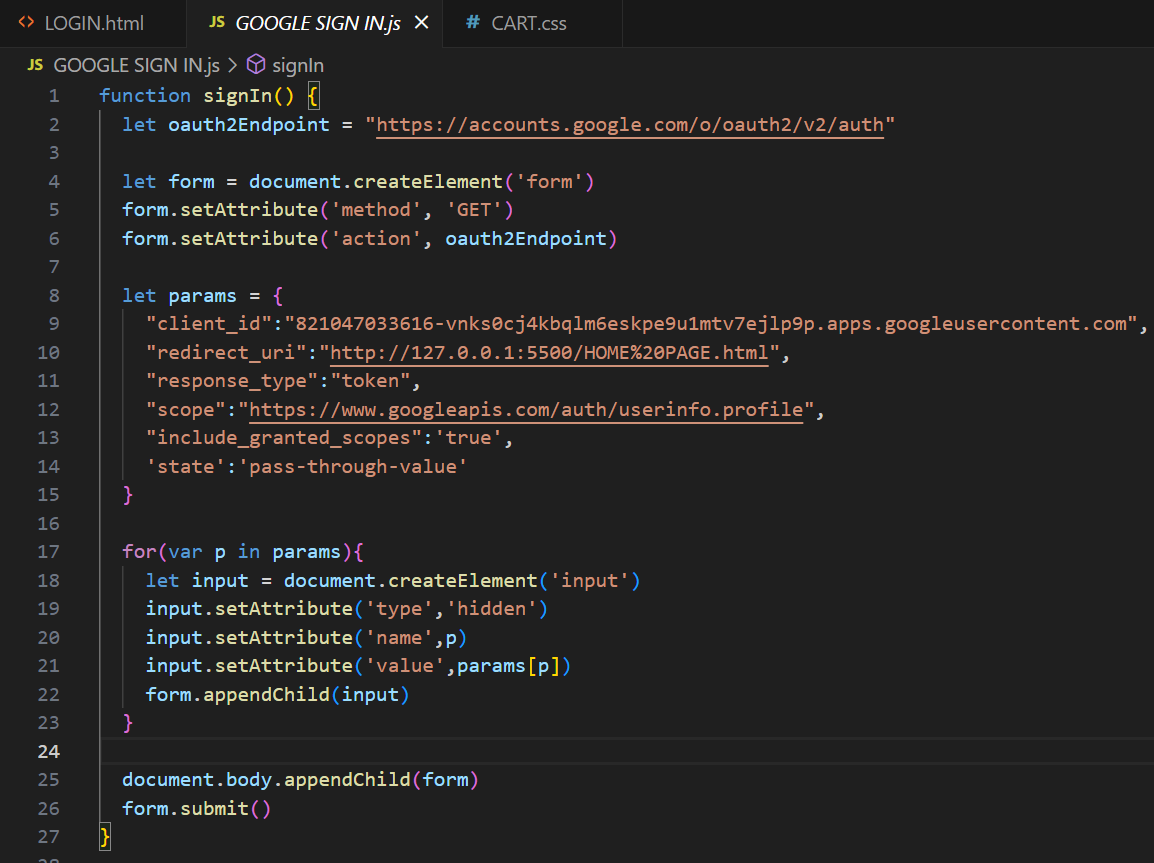
***4) Technologies used for developing the project***

* Web development technologies: We have used html, css, javascript, jquery, etc to make our website look interactive and responsive. In addition, we have used bootstrap so that carousels and other effects can make the user feel blithe after they are redirected to any of the webpages.
* Testing and Continuous Integration: We have ensured that code quality is up to date along with the software, and automation testing is being performed.
* Localization and Internationalization: As our project mainly targets large audiences, we have used tools and libraries for localization to reach as many audience and organizations as possible. Starting with a small base, we have used CAT tools so that the translation quality can be increased and the consistency is maintained.

**B.1 Code**

**Link:** [**https://drive.google.com/drive/folders/1P9D4cxHpcwQTcozr6yL0DwtVGsRa9zSC?usp=sharing**](https://drive.google.com/drive/folders/1P9D4cxHpcwQTcozr6yL0DwtVGsRa9zSC?usp=sharing)

**Google Sign In JS**



EXPLANATION

This JavaScript function `signIn()` is used to initiate the sign-in process with Google using OAuth 2.0. It dynamically creates and submits a form to the Google OAuth endpoint for authentication. Let's break down the code step by step:

1. The `oauth2Endpoint` variable is assigned the value of the Google OAuth 2.0 authentication endpoint.

2. The `form` element is created using the `document.createElement` method.

3. Various attributes are set for the form using `setAttribute` method. These attributes are:

- `method`: Set to 'GET', indicating that the form will be submitted with a GET request.

- `action`: Set to the `oauth2Endpoint`, which is the Google OAuth 2.0 authentication endpoint.

4. The `params` object contains the parameters that will be sent along with the request to the OAuth endpoint. These parameters include:

- `client\_id`: The client ID provided by Google for the application.

- `redirect\_uri`: The URI to which the user will be redirected after authentication. In this case, it is set to "http://127.0.0.1:5500/HOME%20PAGE.html".

- `response\_type`: Set to "token" to indicate that the server should return an access token directly.

- `scope`: The scope of the access request. Here, it is set to "https://www.googleapis.com/auth/userinfo.profile", which specifies the scope for accessing user's basic profile information.

- `include\_granted\_scopes`: Set to 'true' to include any additional scopes that were granted by the user.

- `state`: A custom parameter to maintain state or any other information. In this case, it is set to 'pass-through-value'.

5. The `for` loop iterates through each key-value pair in the `params` object.

6. Inside the loop, a hidden `input` element is created for each parameter using the `document.createElement` method. Attributes such as `type`, `name`, and `value` are set accordingly.

7. Each `input` element is appended as a child to the `form` element using the `appendChild` method.

8. The `form` is appended to the `document.body`.

9. Finally, the form is submitted using the `submit` method.

**Google Sign In & Sign Up Data Storage in Google Sheet**



EXPLANATION

This JavaScript code is responsible for handling the form submission and subsequent redirection upon sign-in. Let's break down the code step by step:

1. The `form1` variable is assigned the form element with the ID `signi` using the `document.getElementById` method.

2. An event listener is attached to the `form1` element for the "submit" event using the `addEventListener` method. This event listener is a function that takes an event parameter, conventionally named `a`.

3. Within the event listener function, `a.preventDefault();` prevents the default form submission behavior, which allows the script to handle the form submission manually.

4. The `fetch` function is used to send a POST request to the URL specified by `form1.action`. The request body contains the data from the form specified by `document.getElementById("signi")`, wrapped in a `FormData` object.

5. The response from the fetch request is processed in two chained `then` functions:

a. The first `then` function converts the response to JSON format using `response.json()`.

b. The second `then` function processes the JSON data received from the server. In this case, it logs the data to the console using `console.log(data)`.

6. After processing the response data, a new page is assigned to the `newPage` variable with the value "HOME PAGE.html".

7. The `window.location.href` is used to redirect the current page to the URL specified in `newPage`, effectively redirecting the user to the "HOME PAGE.html" page.

**Carousel**



EXPLANATION

This code represents a section in an HTML document that creates a carousel of images using the Bootstrap framework. Let's break down the code step by step:

1. The `<section>` tag is used to define a section in the HTML document with the `id` attribute set to "parts" and the class attribute set to "py-5". The class "py-5" is likely used for styling purposes, providing padding to the top and bottom of the section.

2. Within the section, there is a `<div>` element with the class "container" that contains the carousel with the `id` set to "imageCarousel". This `<div>` tag is part of the Bootstrap framework and is used for creating responsive, fixed-width containers.

3. The `imageCarousel` is a Bootstrap carousel component that enables the display of a rotating set of images or content. It has the attribute `data-ride="carousel"`, which enables automatic cycling of the carousel.

4. Inside the carousel, there are three `<div>` elements with the class "carousel-item". Each of these divs contains an `<img>` tag that specifies the image source using the `src` attribute. The images displayed are "extrusion2.png", "extrusion1.png", and "extrusion3.png" respectively.

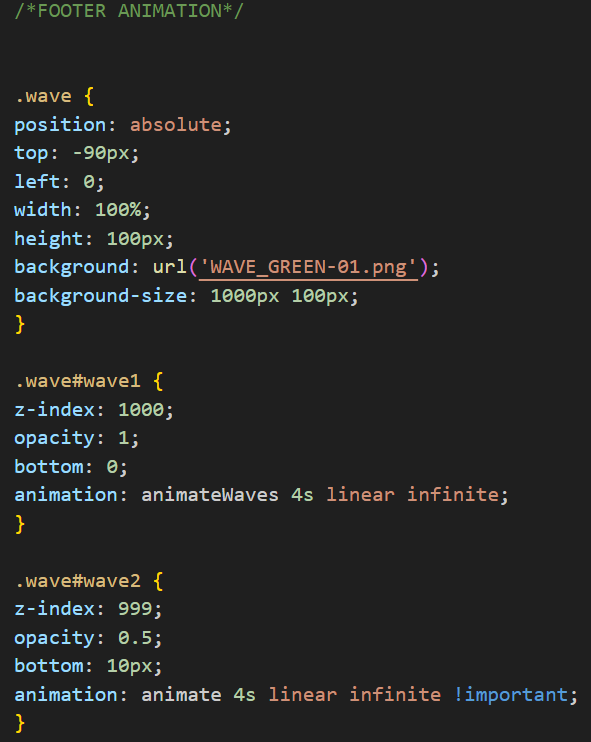
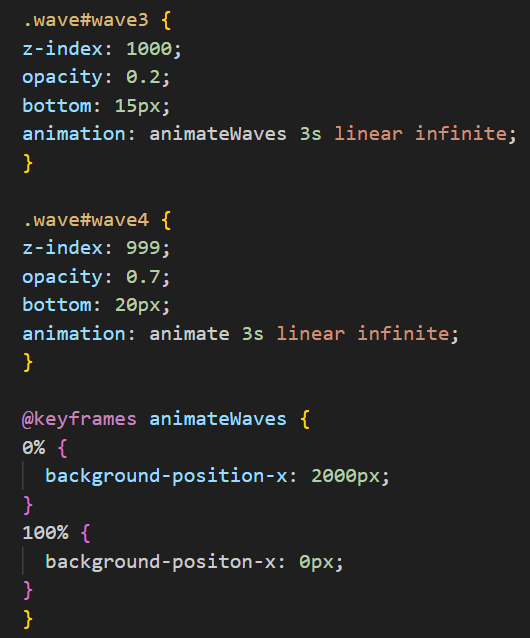
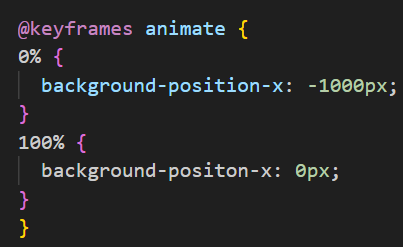
5. The first `<div>` with the class "carousel-item" also has the class "active", indicating that it is the initially active item in the carousel.

6. There are navigation buttons to control the carousel. Two `<a>` tags with the classes "carousel-control-prev" and "carousel-control-next" are used for the previous and next buttons respectively. These buttons link to the carousel via the `href` attribute set to "#imageCarousel".

7. Inside each `<a>` tag, there are `<span>` elements with the classes "carousel-control-prev-icon" and "carousel-control-next-icon", which are responsible for displaying the icons for the previous and next buttons respectively.

8. Additionally, there are `<span>` elements with the class "sr-only" that provide screen reader support for accessibility. These elements contain the text "Previous" and "Next" to provide context to screen reader users.

**Footer Animation**

EXPLANATION

This code represents a CSS animation for creating a wave effect on a webpage. Let's break down the code step by step:

1. The `.wave` class is used to define the basic properties of the wave effect. It is set to have an absolute position, a specific height and width, and a background image specified by the URL 'WAVE\_GREEN-01.png'. The `background-size` property is used to define the size of the background image.

2. There are several variations of the wave effect defined with different IDs, namely `#wave1`, `#wave2`, `#wave3`, and `#wave4`. Each variation has specific settings for `z-index`, `opacity`, `bottom` position, and `animation` properties.

3. The `z-index` property determines the stacking order of the waves. Higher values mean the element will be placed above elements with lower `z-index` values. This allows control over the layering of the wave elements.

4. The `opacity` property sets the transparency level of the waves. Values range from 0 to 1, with 0 being completely transparent and 1 being fully opaque.

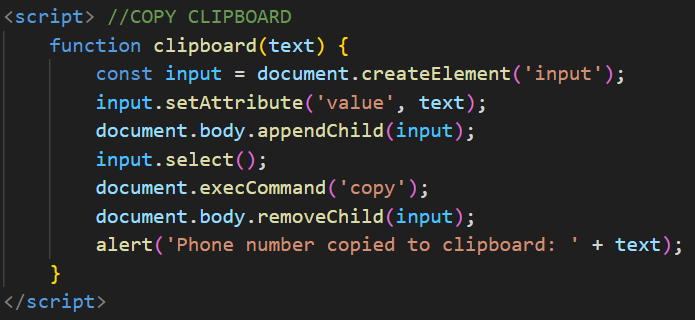
5. The `bottom` property determines the distance between the bottom edge of the element and the bottom edge of its containing block.

6. The `animation` property is used to apply a specified animation to the waves. The animations are defined in the `@keyframes` at-rules.

7. The `@keyframes` at-rules define the animation behavior for the waves. Two animations are defined: `animateWaves` and `animate`. They both specify the change in the `background-position-x` property, creating the effect of the background image shifting horizontally. The `background-position-x` property controls the horizontal position of the background image.

8. The `animateWaves` animation is used for the wave variations with IDs `#wave1` and `#wave3`, while the `animate` animation is used for the wave variations with IDs `#wave2` and `#wave4`.

**Phone Number Copied to Clipboard**



EXPLANATION

This JavaScript code snippet defines a function named `clipboard` that allows copying a specified text to the clipboard. Let's break down the code step by step:

1. The `clipboard` function takes a parameter `text` which represents the text to be copied to the clipboard.

2. Within the function:

- A new `input` element is created using the `document.createElement` method.

- The `value` attribute of the `input` element is set to the `text` parameter.

- The `input` element is appended to the `document.body` using the `appendChild` method.

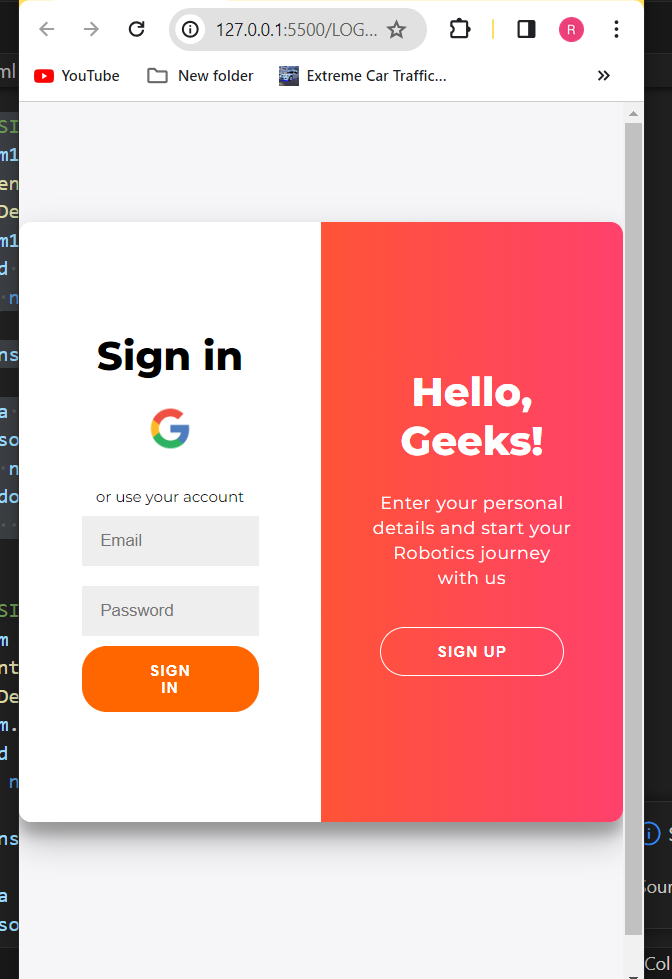
- The `input` element is selected using the `select` method, which highlights the text within the input element.

- The `document.execCommand('copy')` method is used to copy the selected text to the clipboard.

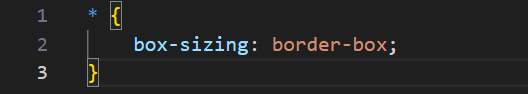
- After the text is copied, the `input` element is removed from the document body using the `removeChild` method.

3. Finally, an `alert` is displayed to the user indicating that the phone number (or the specified text) has been copied to the clipboard.

**RESPONSIVE TO MOBILE**



Code

EXPLANATION

This code is a CSS rule that applies the box-sizing property to all HTML elements.

The `box-sizing` property is used to alter the default CSS box model used to calculate width and height of the elements. The default CSS box model includes content, padding, border, and margin, where the width and height of an element are calculated by adding the content width/height to the padding, border, and margin.

By setting `box-sizing: border-box;`, you are changing the box model so that an element's specified width and height properties include the padding and border, but not the margin. This means that the specified width and height will remain constant, even if padding and border are added to the element.

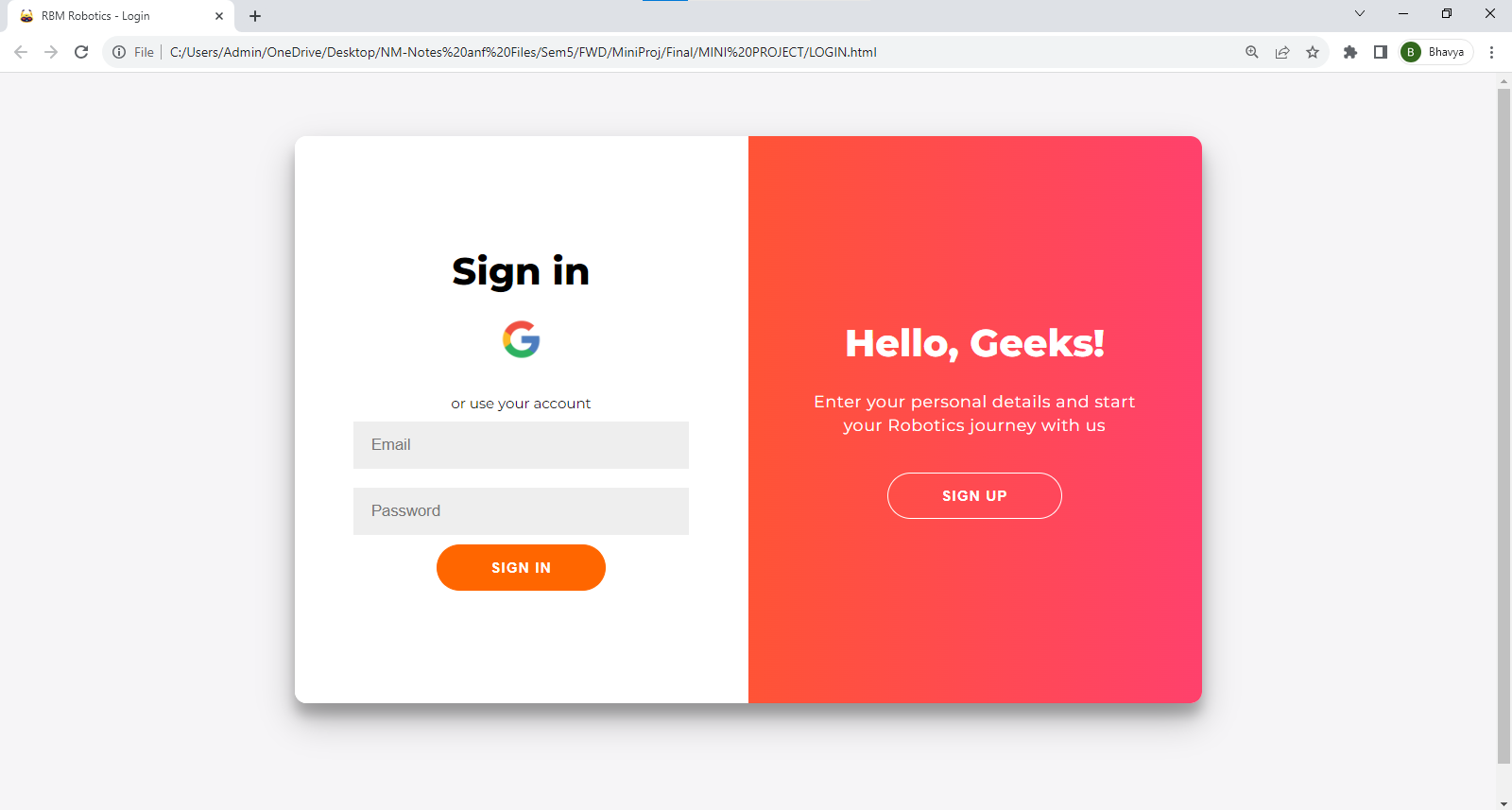
This is particularly useful for creating layouts where you want to precisely control the size of elements, taking into account both their content and their padding and border, without having to recalculate the overall size every time you adjust the padding or border.

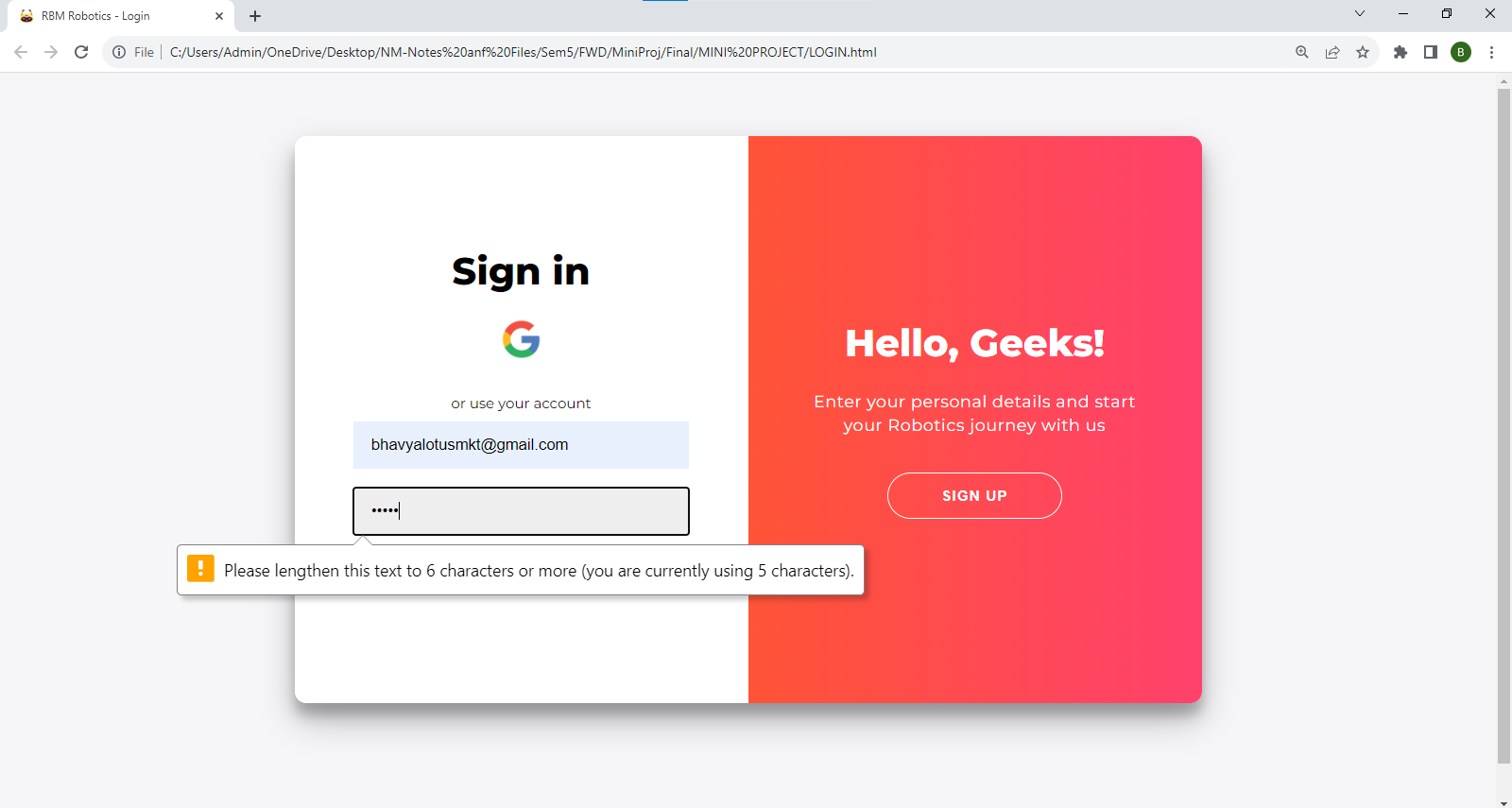
Applying this rule globally using the universal selector `\*` ensures that all elements on the webpage follow this box model, which can help maintain consistency in layout and design.

**B.2 Output**

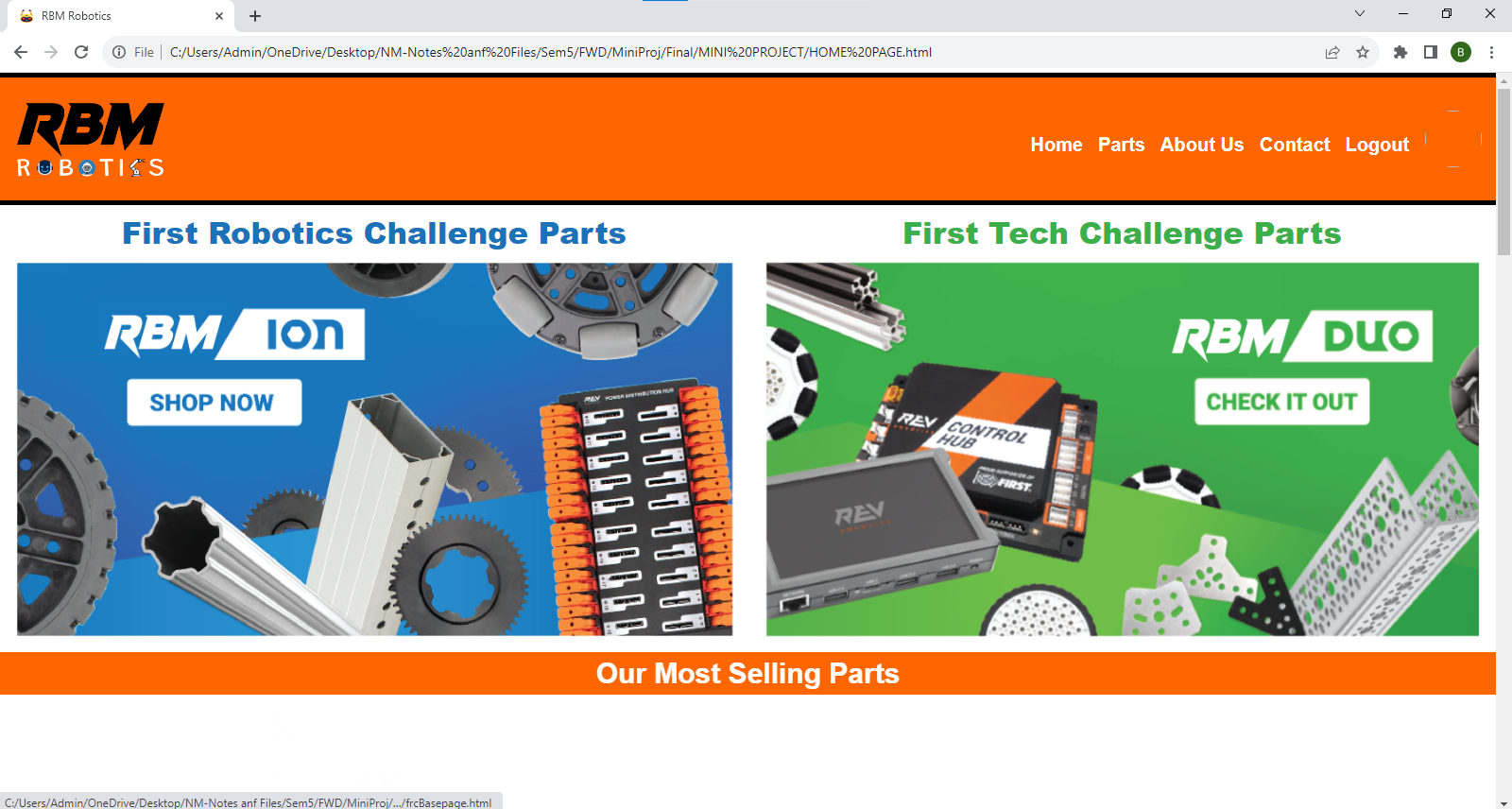
*(Take screen shots of the output at run time and paste it here)*

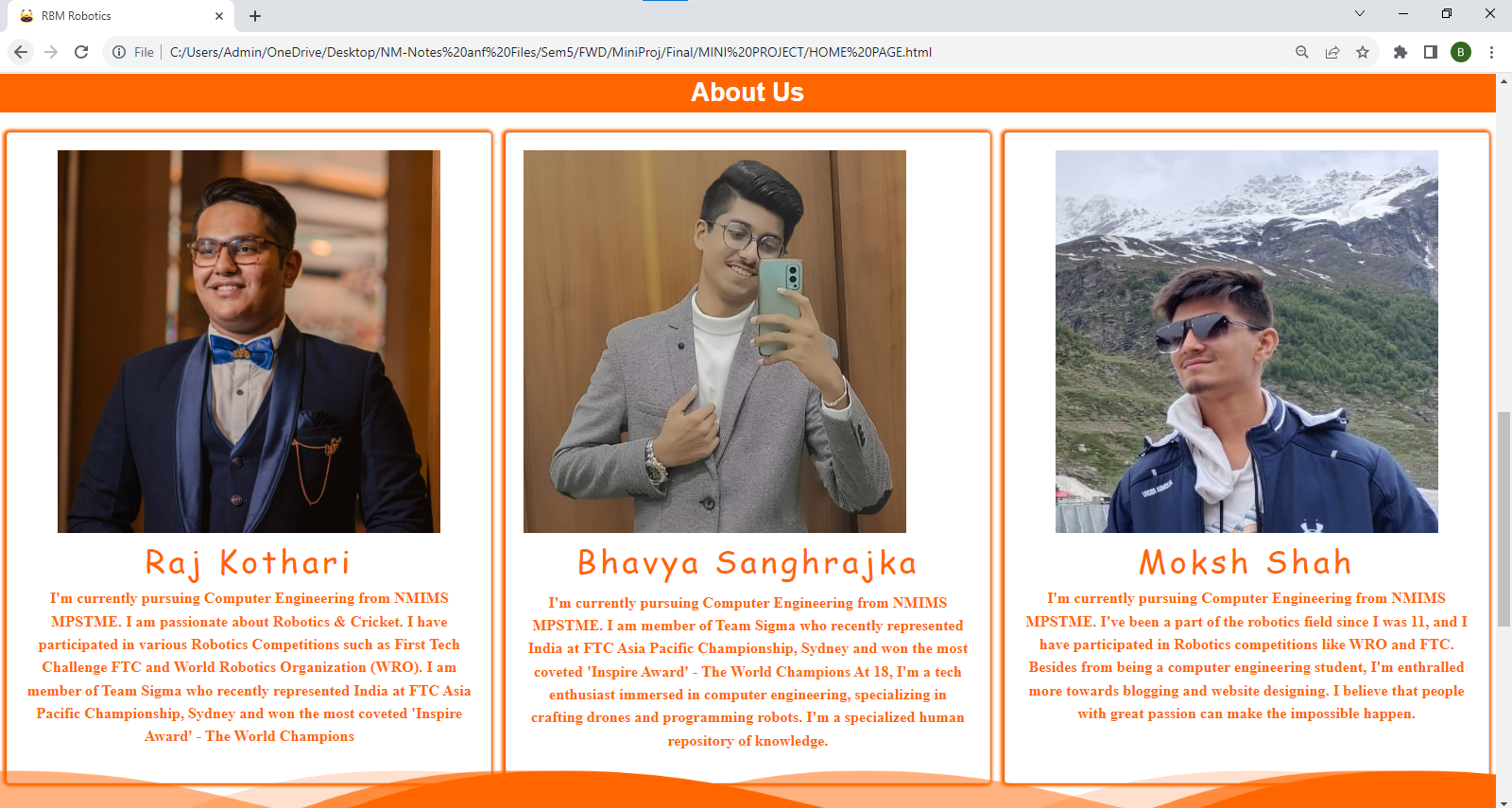
**Login Page**

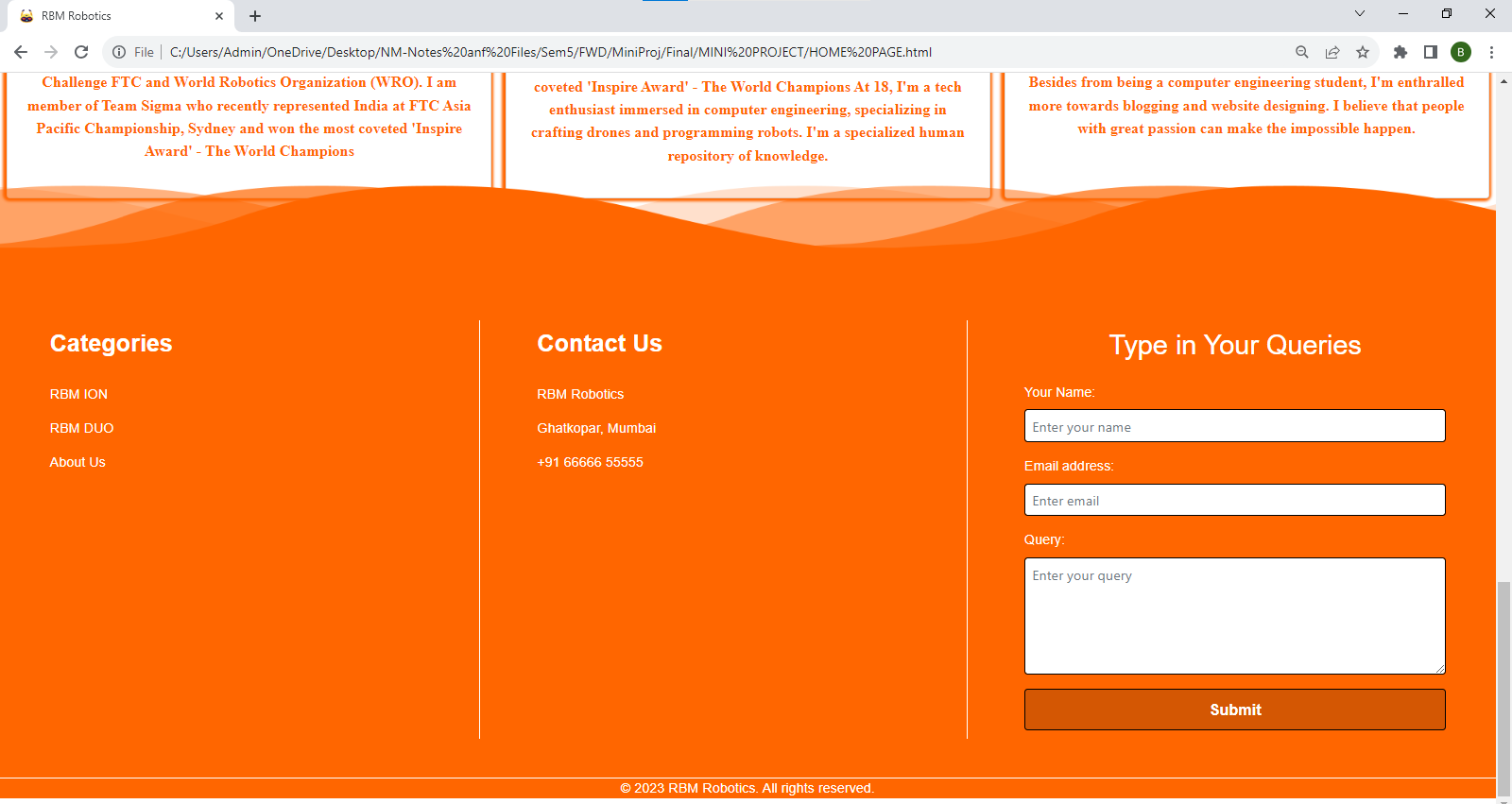
****

****

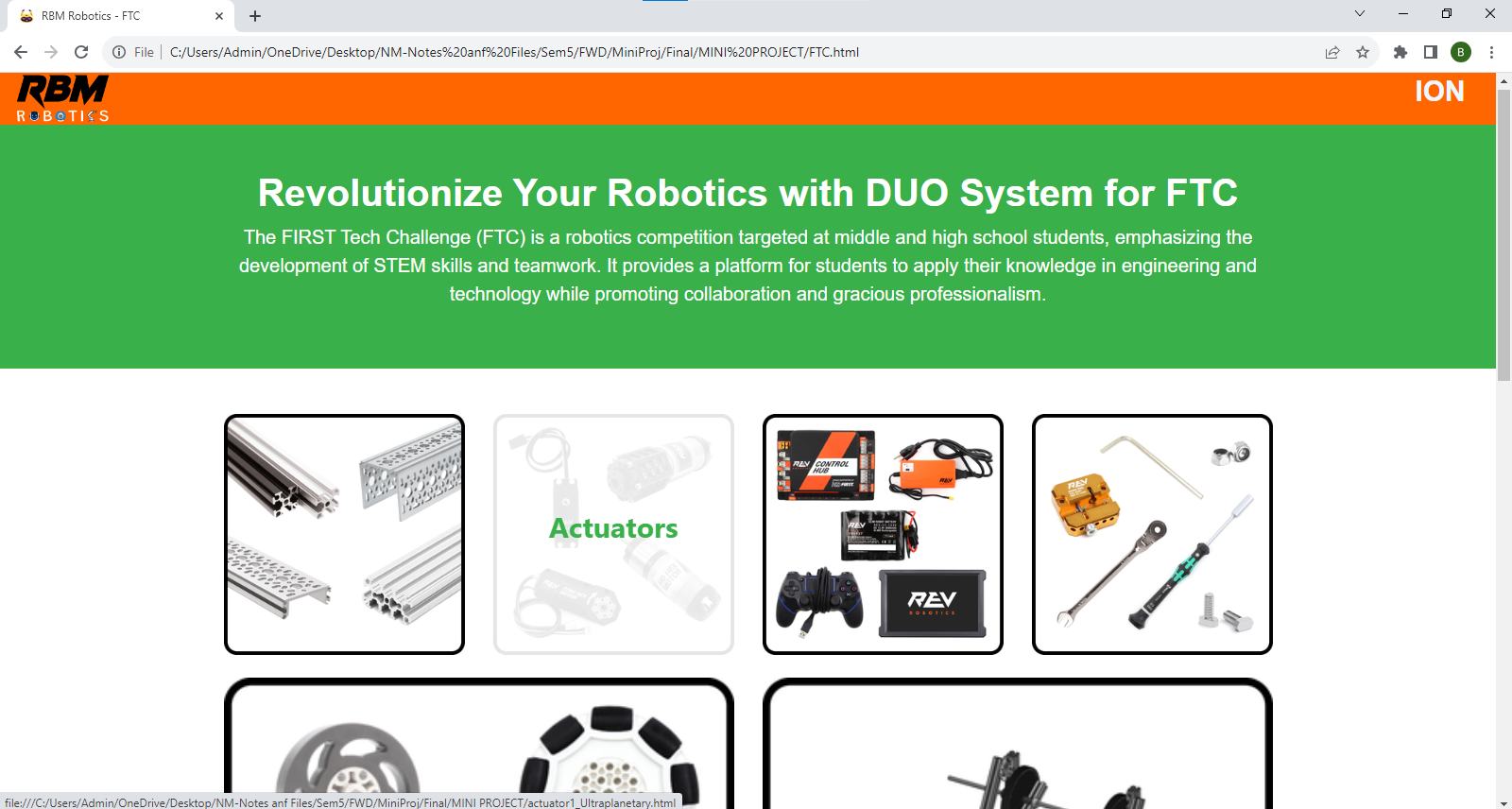
**Home Page**

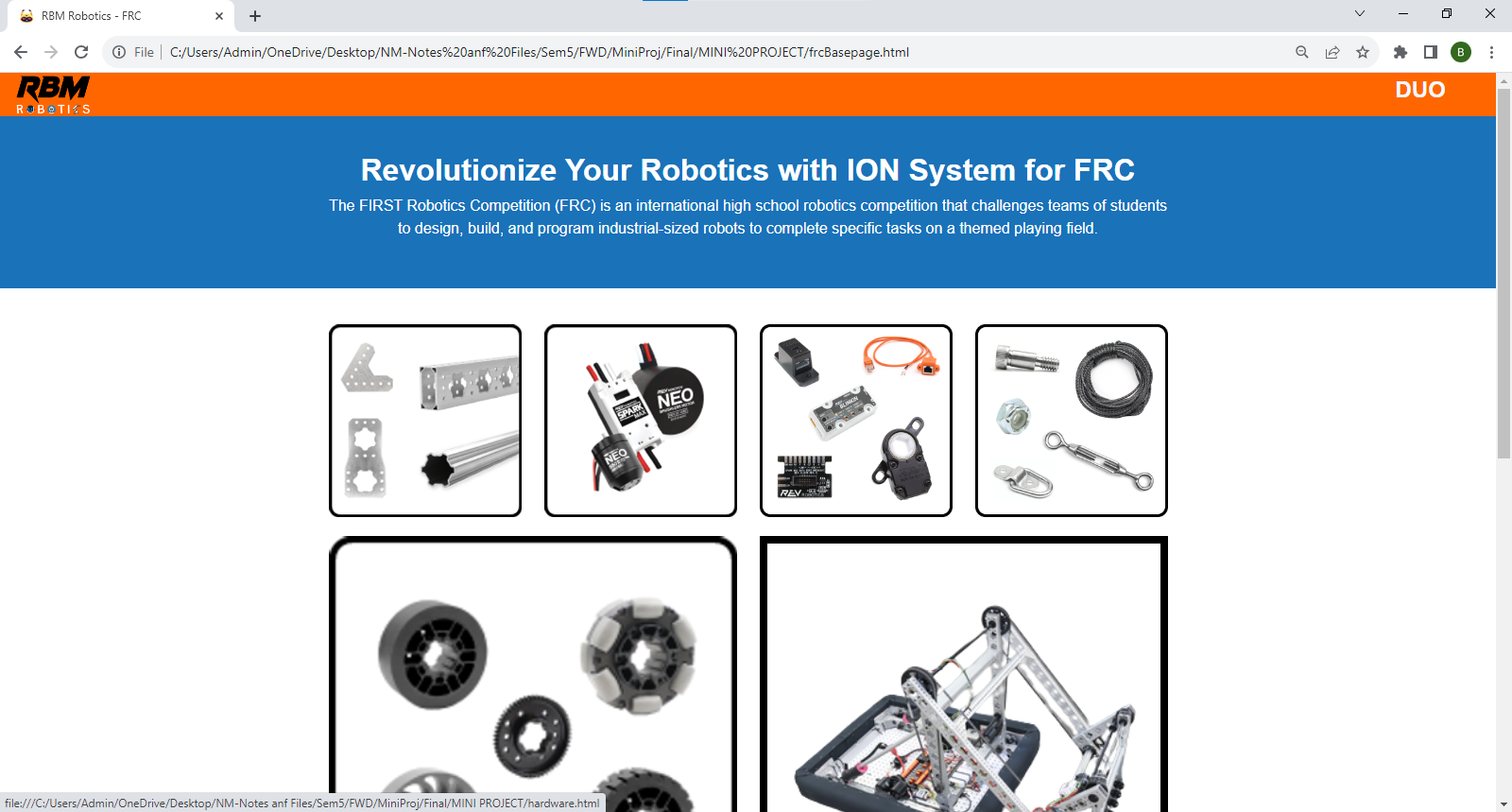
****

****

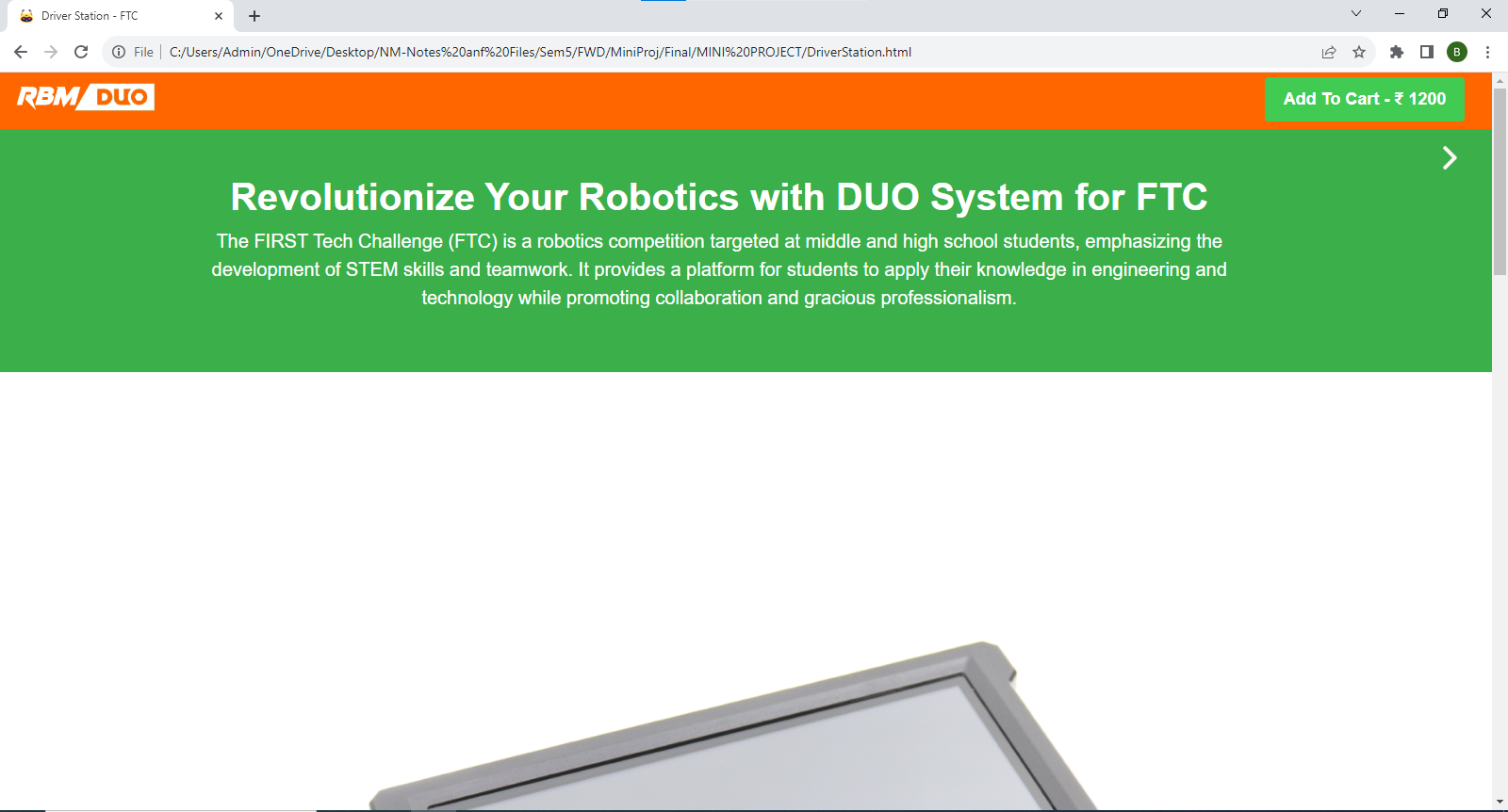
****

**Parts Website Home Page**

****

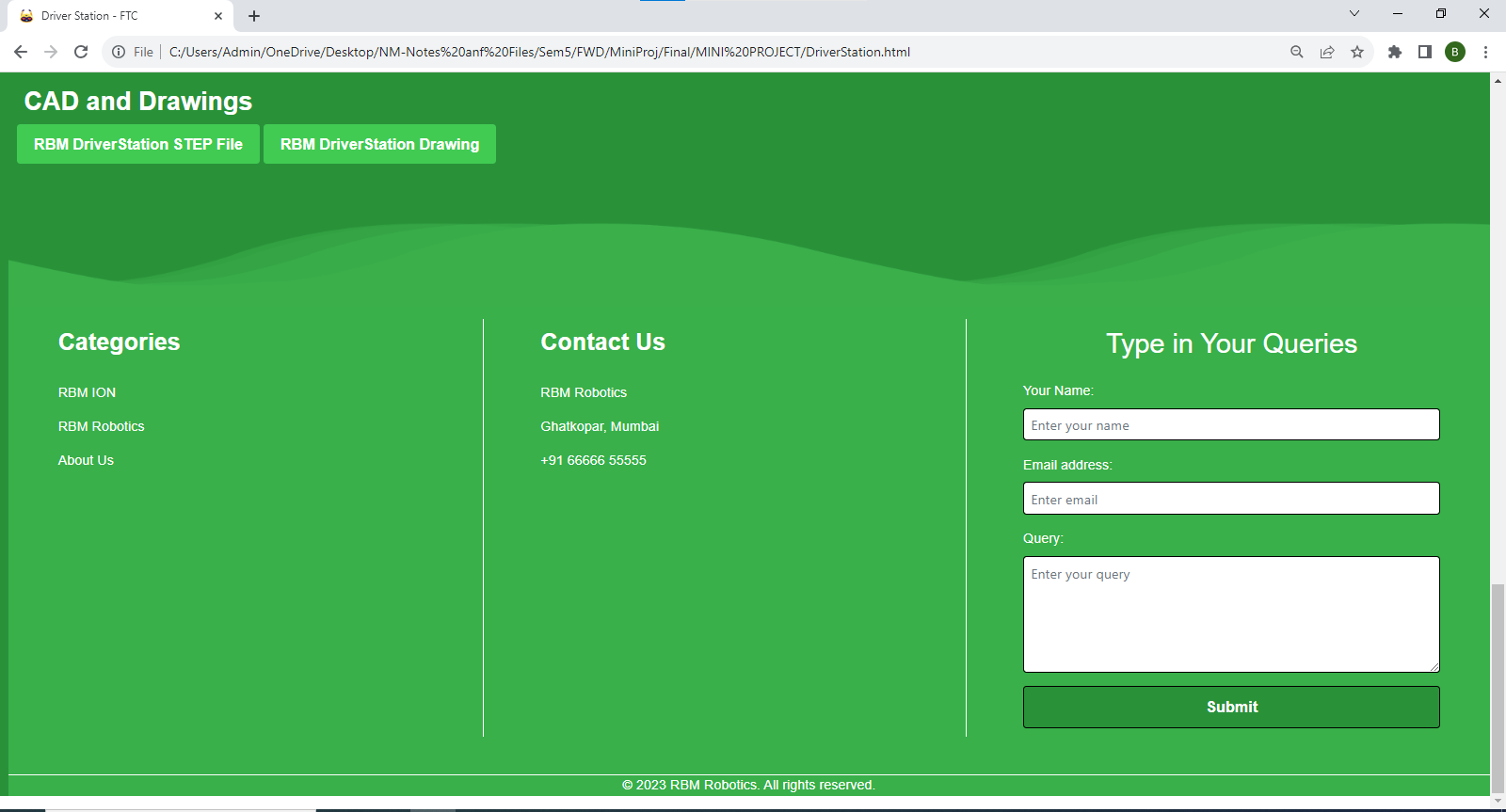
****

**Parts Website Individual Parts**

****

****

****

****

**Add To Cart Page**

****

**B.3 Conclusion:**

From this experiment/ mini project, we got to learn how to compile multiple webpages in a website and run, how multiple programming languages can be used on a single platform and a beautiful website can be made. How local hosting is done is also a part which was interesting to learn, for Google sign in.

**B.4 Observations and Learning:**

Learnt how interactive a website can be using several programming languages. From carousel images to amazing footer animations, working on development of this robotics website from scratch was fun-learning experience. This helped us test the time frame in which we can code each webpage, how interactive an individual can make it, and most importantly, who can bring up the most unique idea to engage audience.