abc.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Rabbit Mart</title>

    <link rel="stylesheet" href="abc.css">

</head>

<body>

    <nav>

        <div class="nav-bar">

            <i class='bx bx-menu sidebarOpen' ></i>

            <span class="logo navLogo"><img class="logo" src="./logo.png" alt="Rabbit Mart Logo"/> </span>

            <div class="menu">

                <div class="logo-toggle">

                    <span class="logo"><a href="#">Rabbit</a></span>

                    <i class='bx bx-x siderbarClose'></i>

                </div>

                <ul class="nav-links">

                    <li><a href="#">Home</a></li>

                    <li><a href="#">Product</a></li>

                    <li><a href="#">Track Shipping</a></li>

                    <li><a href="#">Contact</a></li>

                </ul>

            </div>

            <div class="darkLight-searchBox">

                <div class="searchBox">

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

                    <i class='bx bx-search'></i>

                </div>

            </div>

        </div>

    </nav>

    <div class="wrapper">

        <img class="logo" src="./logo.png" alt="Rabbit Mart Logo"/>

        <div class="form-container">

            <p class="title">Login with your email</p>

            <form action="#" method="post"> <input type="email" name="email" id="email" placeholder="Email" required>

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

                <button onclick="checkAndNavigate()" type="button">Login</button>

                <a href="#" class="forgot-password">Forgot Password?</a>

            </form>

            <p class="signup-message">New to Rabbit Mart? <a href="#">Sign Up</a></p>

        </div>

    </div>

    <script>

        function checkAndNavigate() {

            var email = document.getElementById("email").value.toLowerCase();

            var password = document.getElementById("password").value;

            if (email.endsWith("@gmail.com") && password === "suraj123") {

                window.location.href = "thank.html";

            } else {

                alert("Incorrect Email or Password");

            }

        }

    </script>

</body>

</html>

thank.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Rabbit Mart</title>

    <link rel="stylesheet" href="abc.css">

</head>

<body>

    <nav>

        <div class="nav-bar">

            <i class='bx bx-menu sidebarOpen' ></i>

            <span class="logo navLogo"><img class="logo" src="./logo.png" alt="Rabbit Mart Logo"/> </span>

            <div class="menu">

                <div class="logo-toggle">

                    <span class="logo"><a href="#">Rabbit</a></span>

                    <i class='bx bx-x siderbarClose'></i>

                </div>

                <ul class="nav-links">

                    <li><a href="#">Home</a></li>

                    <li><a href="#">Product</a></li>

                    <li><a href="#">Track Shipping</a></li>

                    <li><a href="#">Contact</a></li>

                </ul>

            </div>

            <div class="darkLight-searchBox">

                <div class="searchBox">

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

                    <i class='bx bx-search'></i>

                </div>

            </div>

        </div>

    </nav>

    <div class="wrapper">

        <div class="th">

        <span> Successfully Logged In!</span>

    </div>

    </div>

</body>

abc.css

\*{

  margin: 0;

  padding: 0;

}

body {

  font-family: sans-serif;

  margin: 0;

  padding: 0;

  display: flex;

  justify-content: center;

  align-items: center;

  min-height: 100vh;

  background-image: url('./background.png');

  background-size: cover;

  background-repeat: no-repeat;

  background-position: center;

}

.wrapper {

  background-color: white;

  margin-top: 60px;

  padding: 3em;

  border-radius: 5px;

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

  max-width: 400px;

  width: 100%;

  display: flex;

  flex-direction: column;

  align-items: center;

}

.logo {

  width: 100px;

  margin-bottom: 2em;

}

.form-container {

  display: flex;

  flex-direction: column;

  gap: 1em;

  width: 100%;

}

.title {

  font-size: 1.1rem;

  font-weight: 500;

  text-align: center;

}

form {

  display: flex;

  flex-direction: column;

  gap: 1em;

  width: 100%;

}

input[type="email"], input[type="password"] {

  padding: 1em;

  border: 1px solid #ccc;

  border-radius: 4px;

}

button[type="button"] {

  background-color: #04AA6D; /\* Adjust to desired primary color \*/

  color: white;

  font-weight: 500;

  padding: 1em;

  width: 100%;

  border: none;

  border-radius: 4px;

  cursor: pointer;

}

.forgot-password {

  font-size: .8rem;

  text-align: right;

  color: #999;

}

.signup-message {

  text-align: center;

  font-size: .8rem;

  color: #999;

}

.signup-message a {

  color: var(--PRIMARY); /\* Replace with desired link color \*/

  font-weight: 500;

}

nav{

  position: fixed;

  top: 0;

  left: 0;

  height: 70px;

  width: 100%;

  background-color: var(--nav-color);

  z-index: 100;

}

body.dark nav{

  border: 1px solid #393838;

}

nav .nav-bar{

  position: relative;

  height: 100%;

  width: 100%;

  background-color: rgb(19, 216, 29);

  color: white;

  margin: 0 auto;

  padding: 0 30px;

  display: flex;

  align-items: center;

  justify-content: space-between;

}

nav .nav-bar .sidebarOpen{

  color: var(--text-color);

  font-size: 25px;

  padding: 5px;

  cursor: pointer;

  display: none;

}

nav .nav-bar .logo{

  margin-top: 30px;

}

nav .nav-bar .logo a{

  font-size: 25px;

  margin-top: 30px;

  font-weight: 500;

  color: var(--text-color);

  text-decoration: none;

}

.menu .logo-toggle{

  display: none;

}

.nav-bar .nav-links{

  display: flex;

  align-items: center;

}

.nav-bar .nav-links li{

  margin: 0 5px;

  list-style: none;

  color: white;

}

.nav-links li a{

  position: relative;

  font-size: 17px;

  font-weight: 400;

  color: var(--text-color);

  text-decoration: none;

  padding: 10px;

}

.nav-links li a::before{

  content: '';

  position: absolute;

  left: 50%;

  bottom: 0;

  transform: translateX(-50%);

  height: 6px;

  width: 6px;

  border-radius: 50%;

  background-color: var(--text-color);

  opacity: 0;

  transition: all 0.3s ease;

}

.nav-links li:hover a::before{

  opacity: 1;

}

nav .nav-bar .darkLight-searchBox .searchBox{

  margin-right: 50px;

  align-items: center;

}

.wrapper .th{

  font-size: 30px;

}

.new\_class {

  text-decoration: none;

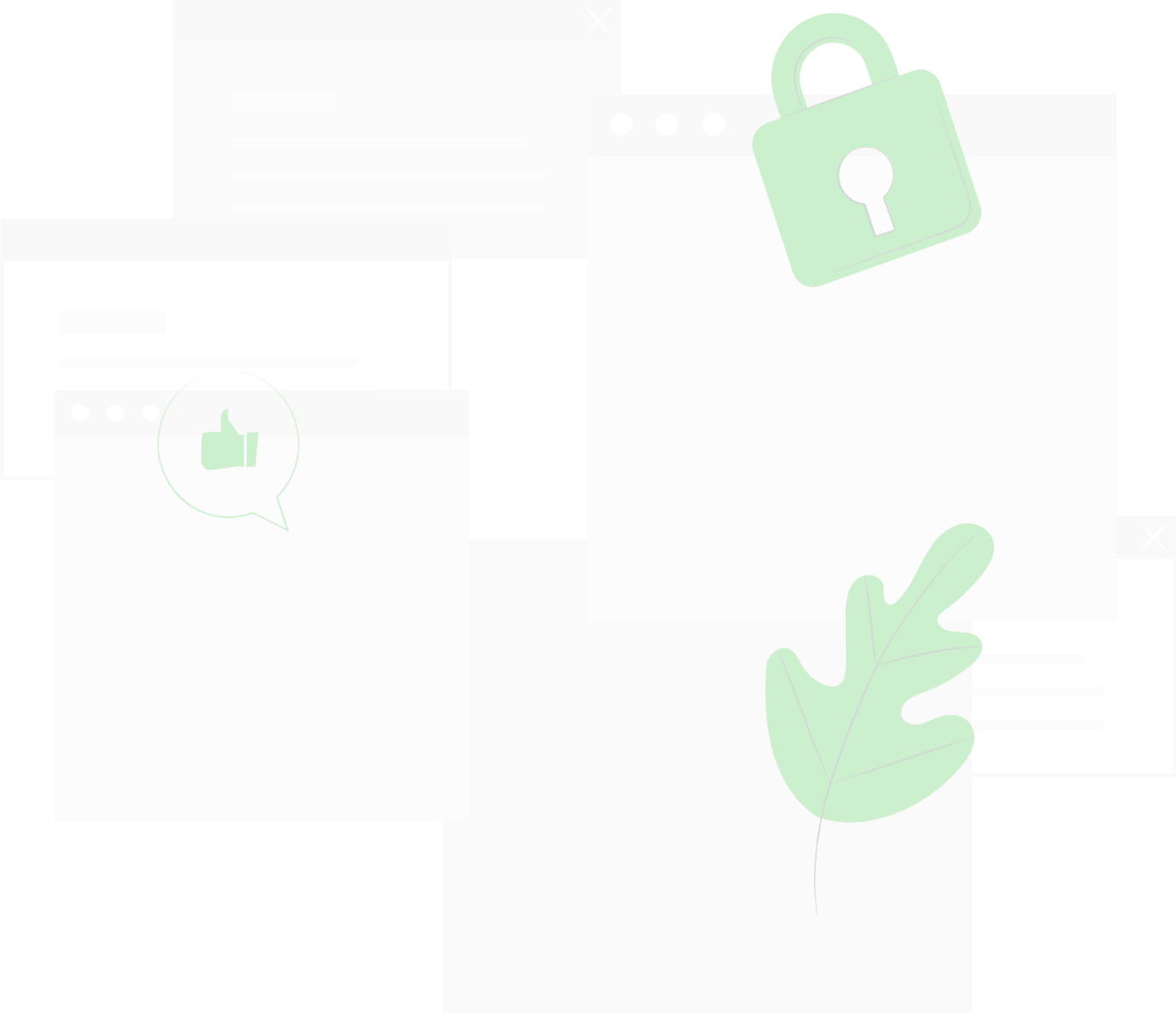
  display: flex;

  justify-content: center;

  align-items: center;

}

background.png



logo.png



Algo

-------------------------------------------------------------------------------------------------------------------------------------

**1. abc.html Algorithm:**

**1.1. Load Document:**

* The browser sends a request to the server for the "abc.html" file.
* The server receives the request and sends back the HTML content.
* The browser begins parsing the received HTML code.

**1.2. Build Document Structure (DOM Tree):**

* The browser interprets the HTML code, creating a Document Object Model (DOM) tree. This tree represents the hierarchical structure of the webpage, with each element (like <html>, <body>, <div>, etc.) being a node in the tree.
* Attributes associated with each HTML element (like id, class, etc.) are also stored in the DOM tree.

**1.3. Render Page:**

* The browser traverses (walks through) the DOM tree, starting from the root (<html>) node.
* For each element encountered in the DOM tree:
  + The element's content is displayed on the screen (e.g., text, images, etc.).
* The browser looks for a linked CSS file (likely "abc.css" in this case) and fetches it.

**1.4. Apply CSS Styles (after CSS is loaded):**

* The browser parses the CSS code, identifying selectors (like .wrapper, .logo, etc.) and corresponding styles (background color, font size, etc.).
* The browser matches these selectors to elements in the DOM tree based on their IDs, classes, or element types.
* For each matched element, the browser applies the corresponding styles, altering the appearance of the webpage.

**1.5. JavaScript Interaction (when the user clicks the login button):**

* When the user clicks the login button, a JavaScript event listener triggers the checkAndNavigate function defined in the <script> section of the HTML.

**2. thank.html Algorithm:**

**2.1. Similar to abc.html (Steps 1.1 - 1.3):**

* The process of loading, parsing, and building the DOM tree for "thank.html" is very similar to steps 1.1 - 1.3 in abc.html.

**2.2. Simpler Structure:**

* "thank.html" likely has a less complex structure compared to "abc.html". It might primarily contain a heading or message congratulating the user on successful login.

**3. abc.css Algorithm:**

**3.1. Load Stylesheet:**

* After parsing the HTML and encountering a <link> element referencing "abc.css", the browser sends a request to the server for this file.

**3.2. Parse CSS Code:**

* The browser receives the CSS content and begins parsing it.
* The browser identifies selectors (like .wrapper, .logo, etc.) and their corresponding style definitions (e.g., background color, font size, etc.).

**3.3. Match Selectors to DOM Elements (after DOM is built):**

* Once the DOM tree is built (from abc.html), the browser can match the CSS selectors to their corresponding elements in the DOM.
* This matching happens based on element IDs, classes, or element types.

**3.4. Apply Styles:**

* For each matched element, the browser applies the corresponding styles defined in the CSS, affecting the visual presentation of the elements and ultimately the webpage.

**4. Overall Login and Success Algorithm:**

**4.1. User Loads Login Page (abc.html):**

* The user's browser follows the steps outlined in 1.1 - 1.3 of the abc.html algorithm, resulting in the login page being displayed.

**4.2. User Enters Credentials:**

* The user types their email address into the email field and their password into the password field of the login form on the webpage.

**4.3. Login Button Click:**

* When the user clicks the login button, a JavaScript event listener triggers the checkAndNavigate function defined in the HTML.

**4.4. JavaScript Validation (checkAndNavigate function):**

* The checkAndNavigate function retrieves the values entered by the user in the email and password fields.
* It performs validation on these values, likely checking if the email ends with "@gmail.com" and the password matches a specific value (e.g., "suraj123" in this example, but you'd use a more secure authentication mechanism in reality).

**4.5. Navigation Based on Validation:**

* **Valid Credentials:**
* If the email and password validation in checkAndNavigate is successful (meaning they meet the criteria you defined), the function redirects the user's browser to the "thank.html" page.
  + This is typically achieved using JavaScript's window.location.href property. Assigning a new URL to this property instructs the browser to load a new webpage.

**4.6. Invalid Credentials:**

* If the email and password validation fails, the checkAndNavigate function likely displays an alert message informing the user of the incorrect credentials.
  + This is often done using JavaScript's alert() function, which pops up a modal dialog with a message.

**4.7. Success Page Display (thank.html):**

* If the validation in checkAndNavigate is successful, the browser follows the steps outlined in 2.1 - 2.2 of the thank.html algorithm. This results in the "thank.html" page being loaded and displayed, congratulating the user on a successful login.

**In essence:**

* The HTML files define the structure and content of the login page ("abc.html") and the success page ("thank.html").
* The CSS stylesheet ("abc.css") provides visual presentation for the HTML elements.
* JavaScript adds interactivity, in this case, handling login form submission, validation, and redirecting to the success page upon successful login.

I hope this detailed breakdown clarifies the algorithms involved in each file and their combined functionality for the login and success process.