**Name:** Habibur Rahman

**Student ID:** B01009639

**Assigned Tasks:** Implementing Menu, Reviews Section respective HTML, CSS, and JavaScript components

**Introduction**

In my personal contribution to the group coursework, I was mainly responsible for the HTML, CSS and JavaScript development of the Menu and Review sections. In the menu section, I used a grid-based design to attractively display each item with an image, name, description and price. In the review section, I created a modern and responsive slider using the Swiper.js library, where users can easily scroll, click or auto play reviews. I also added a cookie consent banner, which made the website user-friendly and legal. These contributions of mine made the website more interactive, professional and an attractive experience for the user.

**Screenshot of Menu section of Coffee Shop website**



**Description of HTML code for Our Menu Section**

I started my HTML code with <!DOCTYPE html>—this basically tells the browser that I'm using the HTML5 standard. This means that the browser will render the page in standards mode instead of running “quirks mode.” Then I wrote <html lang="en"> which defines the language of the page as English. This is useful for search engines, screen readers everything, because setting the language improves pronunciation, spell check, and even search rankings. The <head> section contains things that are not directly visible to the eye, but that give instructions to browsers, devices, and search engines. For example, meta charset="UTF-8" this provides Unicode support throughout the document, so everything from Bengali to Arabic is displayed correctly. I added <meta http-equiv="X-UA-Compatible" content="IE=edge"> to make the old IE browser use the new rendering engine. And <meta name="viewport" content="width=device-width, initial-scale=1.0"> is very important for mobile view this is why responsive layout works properly. <title>Coffee</title> is the tab name and SEO title. I added a favicon for branding. I linked Swiper's CSS so that it is slider-ready, used Font Awesome for the icon, and my own style.css this is where I customize the colors, fonts, layout, animations.

The actual content is in the <body> section. First, start with the Menu section <section class="menu" id="menu">. I put id="menu" here to make internal navigation easy. I gave the title <h1> and a subheading inside with <span> this can be styled separately with CSS. Then <div class="box-container"> here I put each menu item as <a class="box">, meaning that clicking on them will take you to another page. Inside, I put the image, name, short description, and price. I have given alt text to each image this is important for SEO and accessibility. I have written the price with currency symbols so that it is clear to the user. Overall, this section is a grid layout that is easy to scan with image-first, small text, and clear pricing.

**Description of CSS code for Our Menu Section**

In my CSS code, I first designed the Menu section. I gave the .menu class a background image that sits in the center and covers the entire section with background-size: cover;. As a result, the menu section creates a different visual impact, as if the entire block is decorated with a coffee shop theme.

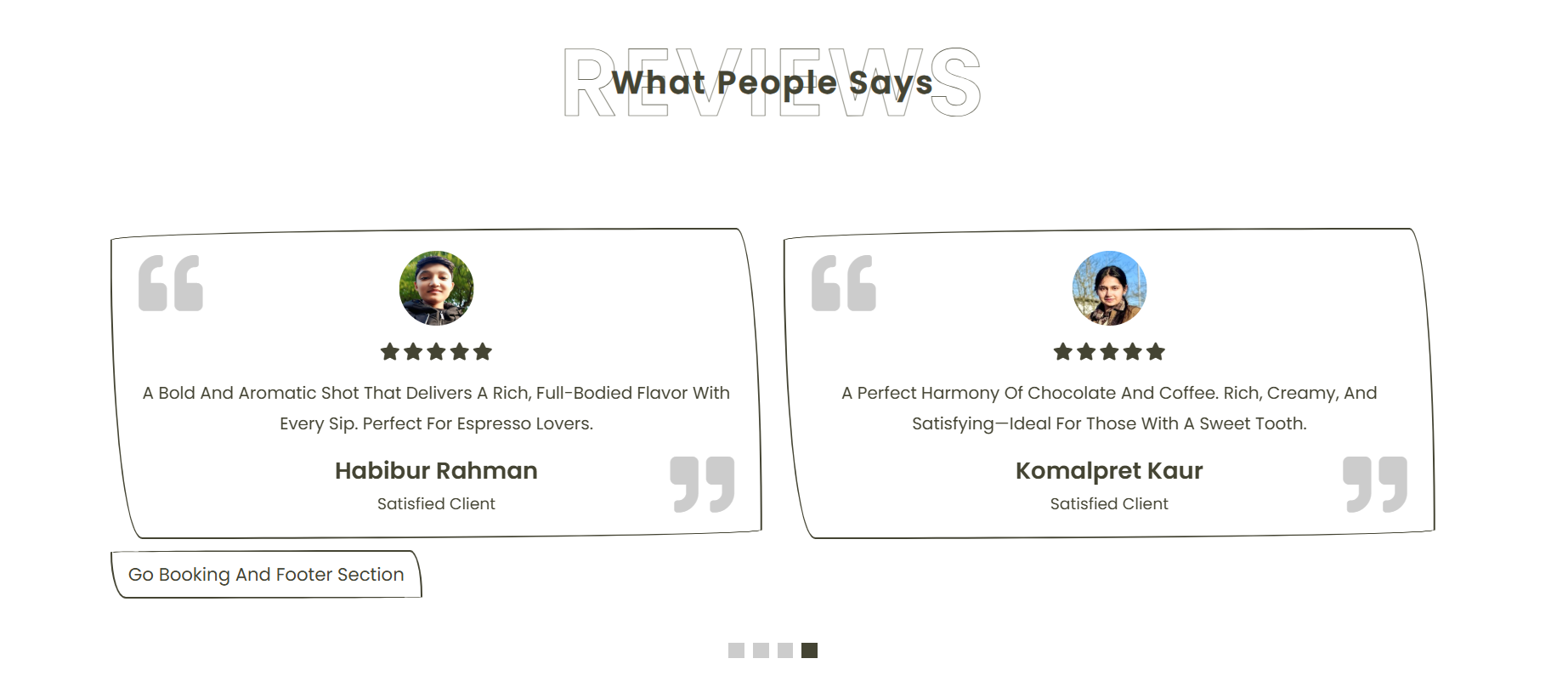
I gave the .menu .box-container a flexbox. Using display: flex; flex-wrap: wrap; all the menu cards are placed side by side, and they move down when the screen is small. With gap: 2rem; equal space is kept between each card, and with justify-content: center; the entire layout is brought to the center. This makes the entire grid layout responsive and beautiful.

Inside each card .box, I again used flex so that the image on the left side and the text on the right side line up together. I set the size of the cards with flex: 1 1 42 rem; which means they are basically 42 rem wide, but can be smaller or larger if necessary. I have arranged the cards as separate blocks using borders and border-radius. When I hover the mouse over them, the borders and radius change, so that the user's focus is on the card.

I have kept the image of each card small (height: 8rem;), but I have pulled the image a little to the left with margin-left: -6 rem;. This makes the image come out of the card a little, creating a kind of "pop-out" effect. I have set the title (h3), description (p) and price (span) to different font sizes inside the card. This visual hierarchy, with the title being the largest, the price being medium and the description being small, helps the user scan quickly.

Overall, Coffee-themed visuals on a center-aligned cover background in the menu section. Responsive flex grid in `.box-container`: `display:flex; flex-wrap:wrap; gap:2rem; justify-content:center;`. Each `.box` flex-row image on the left, text on the right; width controlled with `flex:1 1 42rem`. Borders, hover-transitions, margin-left to pop-out images, and typographic layers increase quick scannability. Cards fall down the row at responsive breakpoints; spacing is maintained.

**Screenshot of Review section of Coffee Shop website**



**Description of HTML code for Review Section**

The second block is the Review section. I have also created this with id="review" for navigation. Inside, I have the structure of Swiper container, wrapper, and slide. Each slide has a quote icon, reviewer's image, five stars, small text, and name/role. These create social proof. Finally, I have added Swiper pagination so that the user can navigate easily. I have also placed a button below the review section which is to take to another page. It is important to note here that Swiper requires JS initialization to work. I have fetched Swiper's JS via CDN, and initialized it in my custom script. Just having a space in the filename can cause problems, so it needs to be fixed.

I put all the scripts at the end of the body because it loads the DOM first, then the script runs. It is also good for performance.

I ensured a few key things throughout the document standards compliance, accessibility, SEO friendly content, social proof, and legal compliance.

**Description of CSS code for the Review Section**

Then I have created the Review section. I have put large padding at the top and bottom of .review .review-slider so that the slider looks like a separate block. I have created a clean frame in each review card (.box) with borders and border-radius, and centered the content inside with text-align: center;.

The most interesting part is the quote icon. I placed fa-quote-left on the left and fa-quote-right on the right. They are positioned with position: absolute; so they look like watermarks in the background rather than on top of the text. When hovered over, they move out of the card this creates a subtle animation that makes the entire UI lively.

I made the reviewer images rounded (border-radius: 50%) so that they look like avatars. I wrote the name (h3) and role (span) at the bottom in different font sizes, and kept the main review text (p) in a medium size so that it is easy to read. I used Font Awesome star icon for the star rating and colored them in the brand color so that the value of the review is visually easy to understand.

I also customized Swiper’s pagination bullets. I made them square (border-radius: 0;) instead of the default rounded ones, and highlighted the active one in the brand color. This makes it easy for the user to understand which slide they are on.

Overall, in this CSS code of mine, both the menu and review sections have different visual styles. Flexbox makes the entire layout responsive, the hover effect interactive, the background and border styles visually appealing, and the star rating/quote icon makes the reviews more credible and beautiful.

**Description of JavaScript code for Review Section**

The purpose of this JavaScript code is to create an interactive slider in the review section of the website. At the beginning, I used a condition if (document.querySelector('.review .review-slider')) { ... }. This means that first I am checking whether there is an element with the class name .review .review-slider on the page. If there is, then the code in the next part will be executed. I have given this as a safety check, because if the review section is not on the page, the code will not give an error for no reason. This way my code has become more reliable.

Inside this condition, I have used new Swiper(). Swiper is a very popular library that can be used to easily create responsive and beautiful sliders. Here I have targeted the .review .review-slider selector, meaning that my slider will only work inside the review section. Then I have added a few configurations or settings, which actually determine the behavior and appearance of the slider.

First, I set space Between: 20. These leaves 20 pixels between each slide. This makes each review box feel separate and the design is cleaner and more attractive. Then I set loop: true. This causes the slider to loop infinitely. This means that even if the user scrolls to the last slide, the slide will start again from the beginning. This means that the slider never feels “finished”, but rather keeps moving smoothly. I have enhanced the user experience by setting grabCursor: true. When this is set, when the mouse is moved over the slider, the cursor changes to a hand icon, indicating that the slider can be dragged and moved.

Next comes pagination. Here I have targeted .review .swiper-pagination . This means that small dots will appear below the slider that represent each slide. Again, I have set clickable: true, so that the user can click on those dots directly to go to a specific slide. This makes navigation easier.

Then I used the autoplay configuration. Here I gave delay: 7500, which means that each slide will change automatically after 7.5 seconds. And by giving disableOnInteraction: false, even if the user drags the slide manually, the auto play will not stop. The slide will start moving again automatically after a certain time. This keeps the slider always running and makes the website look more professional to the users.

Finally, I used breakpoints. Through this, I made the slider responsive. For example, on small screens (mobile) from 0px to 767px, only 1 slide can be displayed at a time. And on large screens of 768px or more (such as tablets or desktops), 2 slides can be displayed at a time. As a result, the slider adapts differently on mobile, tablets and desktops, which improves the user experience.

All in all, this code has created a modern and interactive slider for my review section. In this, the user can drag the slide if they want, change the slide by clicking on the dot below, and the slide can also move by itself. In addition, the slider works beautifully on all devices due to the loop mode and responsive design. This makes the review section of my website look more attractive, user-friendly, and professional.