**Name:** Md Ali Hasan Jidan

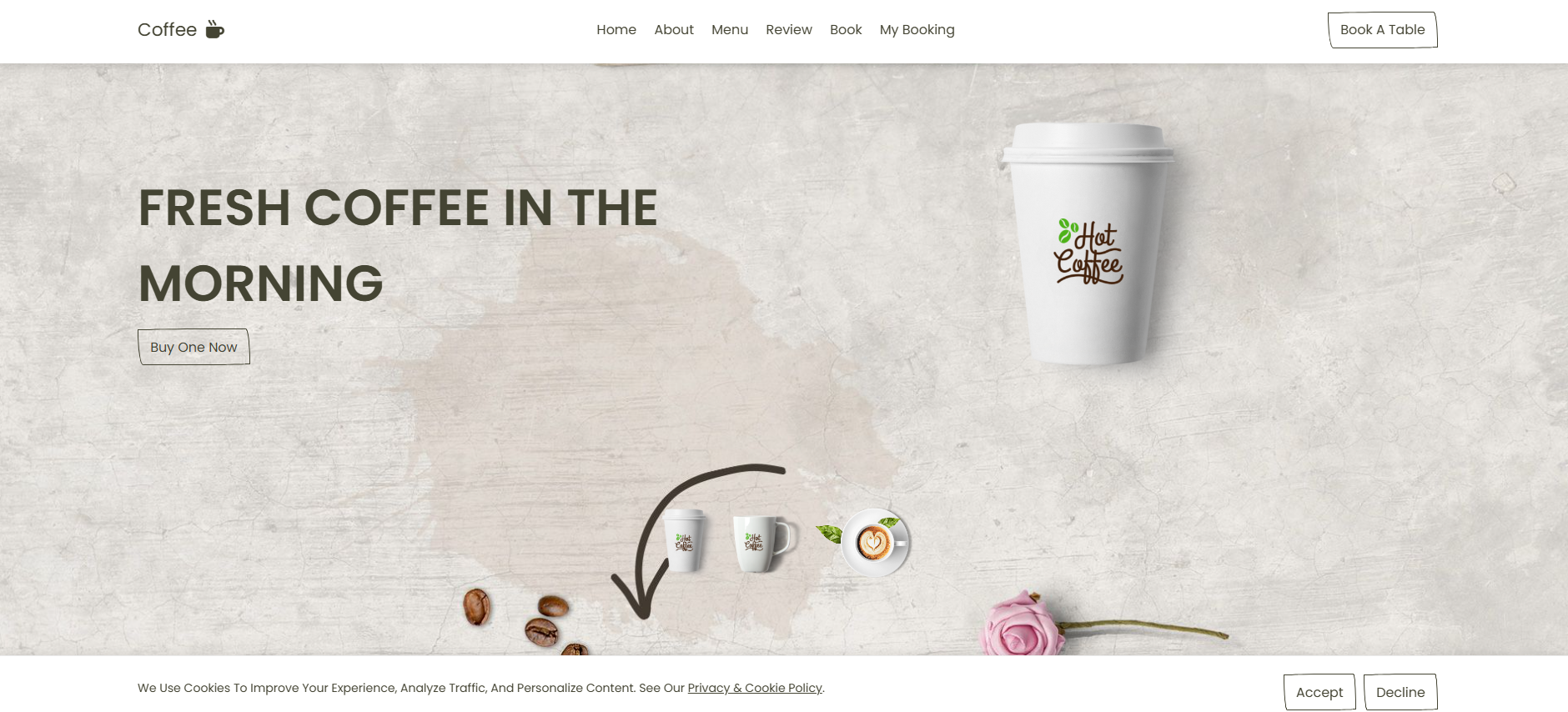
**Student ID:** B01011849

**Assigned Tasks:** Explanation of Home and About pages including HTML, CSS, JavaScript

**Introduction**

As part of my personal contribution to the group coursework, I was responsible for creating and improving the Home and About pages. I built the structure from scratch using HTML, made the website responsive and attractive using CSS, and added interactive features like menus, image sliders, and cookie consent banners using JavaScript. Through all of this work, the website not only looked good, but also created an easy and enjoyable experience for the user.

**Screenshot of the Home Page**



**Explanation of Home Page HTML code**

My HTML file actually creates the “bones” of the entire webpage the structure and basic content (text, images, links). The very first thing is <!DOCTYPE html> it tells the browser, “This document is written in the HTML5 standard.” Right after that, <html lang="en"> tells the browser that the language of the page is English.

The <head> tag contains some information about the webpage that is not directly visible on the screen, but is very useful for browsers and search engines. For example, <meta charset="UTF-8"> this ensures that all Unicode characters Bangla, English, Arabic are displayed correctly. <meta name="viewport" content="width=device-width, initial scale=1.0"> tells the page to be responsive on multiple size screens. <title>Coffee</title> determines the text that is displayed in the browser tab. In this <head>, I also linked a few external files like swipers CSS (used to create the slider), Font Awesome (used to display the icons), and my own css/style.css (the main style of my design). I also added a favicon (a small icon that appears in the tab). All of these helps make the page stylish and responsive.

The actual visible part of the page comes in the <body> tag. At the very top is <header class="header"> this is the top bar or header section of the website. Here, first, there is <div id="menu-btn" class="fas fa-bars"></div> this is a “hamburger” icon on small screens, pressing this button opens/closes the menu (this is done by JavaScript). Then, there is <a href="#" class="logo">coffee <i class="fas fa-mug-hot"></i></a> this is my website’s brand logo. Here, there is a coffee mug icon next to the “coffee” text. Then <nav class="navbar"> here are various links home and about these take you to sections within the same page (#home, #about). Menu, review these take you to another page called habibur.html. Book and the “book a table” button above take you to komal.html and My Booking to another page called myBooking.html.

Then comes the <! --HOME--> section. <section class="home" id="home">—this id="home"—the reason the href="#home" link in the top menu scrolls here. Inside is a <div class="row"> here on the left side, in the content div, there is the title “fresh coffee in the morning” and the “buy one now” button clicking on this button takes you to the habibur.html page. On the right side, in the. image div, there is a large image this is the main coffee cup image (class="main-home-image-1"). Its alt text is nicely described so that even prisoners can use my website.

Below the home section is a <div class="image-slider"> here are three small images (thumbnails). They look small, but the work is big when you click on any of these images, JavaScript changes the src of the main-home-image as a result, the large image changes immediately according to the thumbnail. That is, a very lightweight gallery effect has been created here, that too without any heavy plugins.

**Explanation of Home Page CSS Code**

In this CSS, the “Poppins” font is imported from Google Fonts to give text a clean, modern look. Custom variables defined in :root (such as --main-color, --border-radius, and --border) make the code more consistent and easy to update. A global reset removes default browser margins and padding with \*{margin:0; padding:0; box-sizing: border-box;}.text-transform: capitalize; capitalizes the first letter of each word, while transition: all .2s linear; creates smooth hover effects. Setting html{font-size:62.5%;} simplifies rem-to-px calculations. scroll-behavior:smooth; ensures smooth scrolling, and scroll-padding-top:7rem; prevents content from being hidden behind the fixed header.

.heading , This is the large heading style. Here, a hollow outline title has been created on the outside with color: transparent and -webkit-text-stroke. The span inside it has been brought to the very center (position: absolute; transform: translate (-50%, -50%)) As a result, it looks like a double-layer: large outline text, small filled text inside. This makes the heading look stylish.

The .btn style has your own custom border-radius (different curves on each side) and a dashed border on hover this “organic” button style gives the page a unique feel. .btn: hover changes both the border-style and border-radius, creating a soft, eye-catching animation that tells the user “this is clickable.”

.header fixed meaning it will stay on top even if you scroll. The box-shadow adds a light shadow under the header making the visual stand out. .navbar a has a fixed font-size and color. #menu-btn {display: none} Hiding the hamburger icon on desktop; usually it is displayed: inline-block in media queries for smaller screens and .navbar is hidden and shown with the active class first. Practically, a rule like .navbar.active {display: block } will be needed to work on mobile.

.home section has min-height: 100vh meaning the hero section will span the entire viewport height; background has a large image, background-size: cover that is, it will crop/fit properly across the screen. .home .row { display: flex } where content and images sit side by side in two columns, gap: 1.5rem there is a gap between the columns.

The most interesting part is .home .row .image img{ animation: float 4s linear infinite } where the large image of the coffee cup slowly floats up and down on its own. In the @keyframes below, float at 0% and 100% means the normal position, and at 50%-translateY(0) means a little up in the middle. A cycle completes in 4 seconds and continues indefinitely. This animation is your “floating cup” effect.

.home .image-slider Here are three thumbnails. .image-slider img:hover { transform: translateY(-2rem) } When the mouse is hovered over, the images “jump” up a little, which encourages the user to click.

**Explanation of Home Page JavaScript Code**

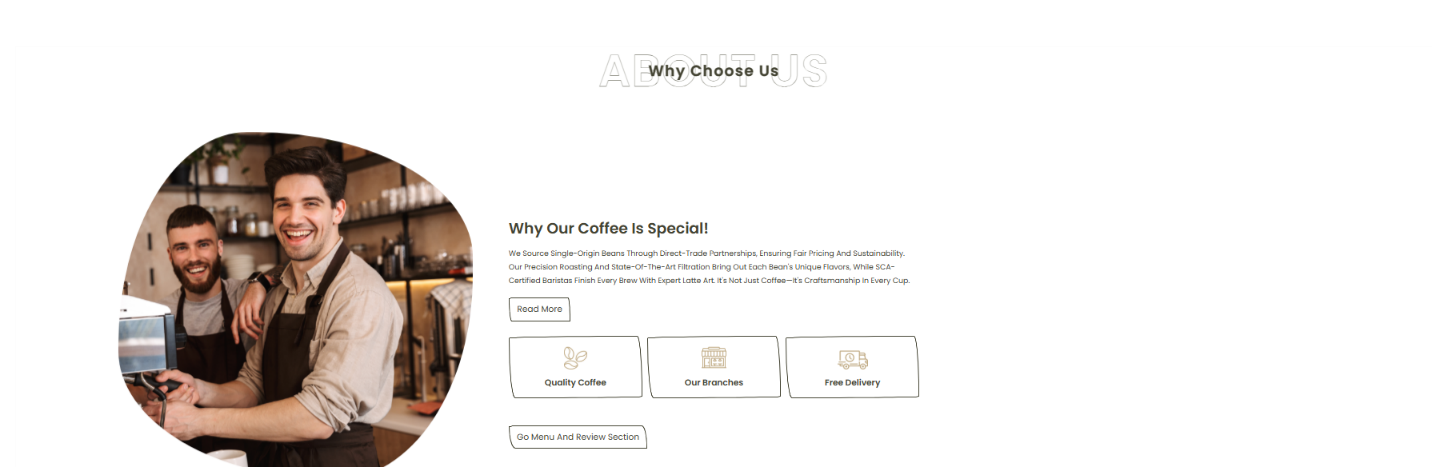
I have added two main functionalities to my JavaScript code navigation menu control and image change mechanism. In the first part, #menu-btn and .navbar have been selected and the condition has been set that the code will only work if both elements are present on the page.

When the menu button is clicked, the icon changes (from hamburger icon to cross icon) using menu.classList.toggle('fa-times'), and the navigation bar is opened or closed using navbar.classList.toggle('active'). At the same time, when the user scrolls, the menu and navigation bar automatically return to their default state. This results in a responsive and user-friendly navigation menu, which is especially useful for mobile devices.

In the second part, I have selected .image-slider img, that is all the thumbnail images of the website slider have been taken. An onclick event has been defined for each image. When a thumbnail is clicked, the main image named .main-home-image is selected and its ‘src’ attribute is changed to the source of the thumbnail image. As a result, when the user clicks on the smaller image, the larger image automatically changes. This makes the website more interactive and provides a dynamic image gallery experience.

In this JavaScript code, I have controlled the logic of the cookie consent banner. First, I have created, read and deleted cookies through the setCookie, getCookie and eraseCookie functions. showCookieBannerIfNeeded() makes the banner visible and in the bindCookieBannerEvents() function, click events have been added to the Accept and Decline buttons, where the banner is hidden when the user clicks. The lines for setting cookies in the code are in comment form, if desired, enabling them can be used to save the user's decision. In addition, a global object named window.CookieConsent has been kept, so that the banner can be shown again by deleting the cookie through the reset() function and the status of the cookie can be known through the status() function. Finally, the initCookieBanner() function is launched after the page loads and activates the banner.

**Screenshot of the About Page**



**Explanation of About Page HTML code**

Next comes the <! -- ABOUT --> section. <section class="about" id="about"> here too, because of the id="about" it goes here when you click on the about link in the menu. The title of this section is <h1 class="heading">about us <span>why choose us</span></h1> here the large text on the outside “about us” and the small text on the inside span “why choose us” the two are displayed in very different styles with CSS. Below is an image on the left side of row an image of a barista at work in a cafe. On the right side, in content, there is an h3 title, a large paragraph where your coffee specialties (single-origin beans, direct-trade, SCA-certified baristas, etc.) are written these give the user an idea of the brand’s value. There is a button called “read more” (currently href="#" meaning it doesn’t take you anywhere, later I will link to the ‘read more’ page). Below that, there is a icons-container here are three icon blocks (quality coffee, our branches, free delivery) that visually show your service points. At the end of this section, I have placed a button in a small content div called “go habibur section” that takes you to habibur.html.

In this code, I create a Cookie Consent Banner that asks the user for permission to use cookies. The main container div uses id="cookie-banner" and class="cookie-banner", along with role="region", aria-label and aria-live to make the banner easy to understand for everyone, including screen readers. Inside, the cookie-inner section displays a message (cookie-text) that describes the purpose of the cookie and provides a link to the Privacy & Cookie Policy. Then, the cookie-actions section contains two buttons—Accept and Decline. The Decline button uses aria-described by, which is associated with a hidden message (sr-only), so that screen reader users know that only a minimal cookie will be saved if they decline.

**Explanation of the About Page CSS code**

.about section .about .row .image img { animation: aboutImage 4s linear infinite } Here the image is scaled up a little (from scale(.9) to scale(.8) to .9) and the border-radius is also changed resulting in a soft pulsing effect. On the content side, there are three icon cards for title, para, and .icons-container; these use your previous --border and --border-radius variables, so the style is consistent across the site.

Then this CSS styles the Cookie Consent Banner to be responsive, accessible, and user-friendly. The .sr-only class hides content visually but keeps it readable by screen readers. The banner is fixed at the bottom, hidden by default, and shown smoothly with the .show class. Flex layout in .cookie-inner aligns text and buttons, while .cookie-link is emphasized with an underline. Buttons are side by side, with spacing adjusted for small screens via media queries. Transitions are disabled for reduced-motion users.