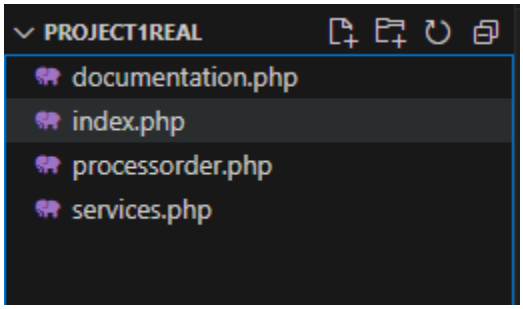
Iman Talukder

A multi-page, fully functional website that gives users information about the hospital and its services, as well as a contact platform, was created as part of the Iman Hospital project. The user-friendly interface of this website was intended to make it simple for patients, employees, and guests to navigate. The platform uses PHP for form processing, HTML, and CSS for structure and styling. A thorough project analysis is provided below, covering the steps taken, the difficulties faced, and the solutions used.

Initial Setup: Configuring the file structure was the initial stage in building the website. Each page (Home, Services, Documentation, and Contact) has its own.html file in the project, along with a style.css file for styling and a processorder.php file for managing form submissions. Every file has the proper, , , and sections in accordance with the fundamental HTML structure.



Header and Navigation: A logo and navigation bar are shared by every page. As a result, users can move between pages with ease. Each list item () in the unordered list (

) contained in the element links to a different website. All parts offer a cohesive user experience thanks to the consistent navigation.

```
<nav>
    <ul>
        <li><a href="index.php">Home</a></li>
        <li><a href="services.php">Services</a></li>
        <li><a href="documentation.php">Documentation</a></l
        <li><a href="processorder.php">Contact</a></li>
    </ul>
</nav>
```

Main Content: The tag contains each page's primary content. The content on each page varies based on its intended use:

Home page: Welcomes visitors to Iman Hospital with a hero section.

Services page: Provides a list of available medical services.

Information about the project and its development process is available on the documentation page.

A form for user inquiries is located on the contact page.

Form Creation: On the contact page (processorder.php), the contact form was positioned inside the section. The user's name, email address, and message are all necessary fields on the form. In order to guarantee that the PHP script processes the form input, the action attribute points to processorder.php.

```
<form action="processorder.php" method="POST">
    <label for="name">Your Name:</label>
    <input type="text" id="name" name="name" required>

    <label for="email">Your Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="message">Your Message:</label>
    <textarea id="message" name="message" rows="6" required></textarea>

    <input type="submit" value="Send Message">
</form>
```

PHP Logic: To make sure the email is legitimate, the PHP script uses the filter_var() function to validate the input once the form is submitted. In order to stop code injection attempts, it also cleans the input. The mail() function is used to send an email if validation is successful. Error messages are shown otherwise.

Important PHP functions that are utilized:

HTMLspecialchars() and trim(): Clean user input; filter_var(): Verifies the email format.

```
if (!empty($name) && !empty($email) && !empty($message) && filter_var($email, FILTER_VALIDATE_EMAIL)) {
    if (mail($to, $subject, $body, $headers)) {
        echo "<p class='success-message'>Thank you, your message has been sent successfully!</p>";
    } else {
        echo "<p class='error-message'>Sorry, there was an issue sending your message. Please try again later.</p>";
    }
} else {
    echo "<p class='error-message'>Please fill out all fields and provide a valid email address.</p>";
}
```

Course Materials and Adaptations

Course Materials, Chapters 1–7: Several of the ideas covered in the course were used, including:

Using semantic HTML elements (header, nav, main, and footer) correctly is known as HTML structure.

CSS Styling: Best practices for responsive styling and layout design.

PHP Form Handling: Email sending capabilities, sanitization methods, and basic form validation.

By including PHP for dynamic form handling and interaction, these fundamental ideas were expanded upon to create a user-friendly, useful website.

Form Validation: Ensuring proper form validation was a major challenge. Initially, I encountered issues with incorrect email formats being accepted. To resolve this, I implemented PHP's filter_var() function for email validation, ensuring only valid emails were processed.

Responsive Design: Creating a responsive layout that looked good on both large and small screens required adjustments to margins and image sizes. This was achieved using percentages for layout dimensions and ensuring flexible image sizes via the width and height: auto properties in CSS.

Error Handling: Another challenge was error handling for form submission. I implemented condition checks to provide user feedback in case of incomplete forms or incorrect email inputs.

The Iman Hospital project serves as a comprehensive example of integrating front-end and back-end technologies to create a functional, user-friendly website. By leveraging HTML, CSS, and PHP, I successfully built a platform that allows users to navigate through hospital services, contact the hospital, and interact with the platform in a meaningful way. The experience provided valuable insight into web development, problem-solving, and ensuring a positive user experience.