**WEB TECHNOLOGIES – PART B**

**Weekly Tasks**

**Name:-Krupal Solanki**

**Id:-413976**

# Unit 1 Task 1

Code:

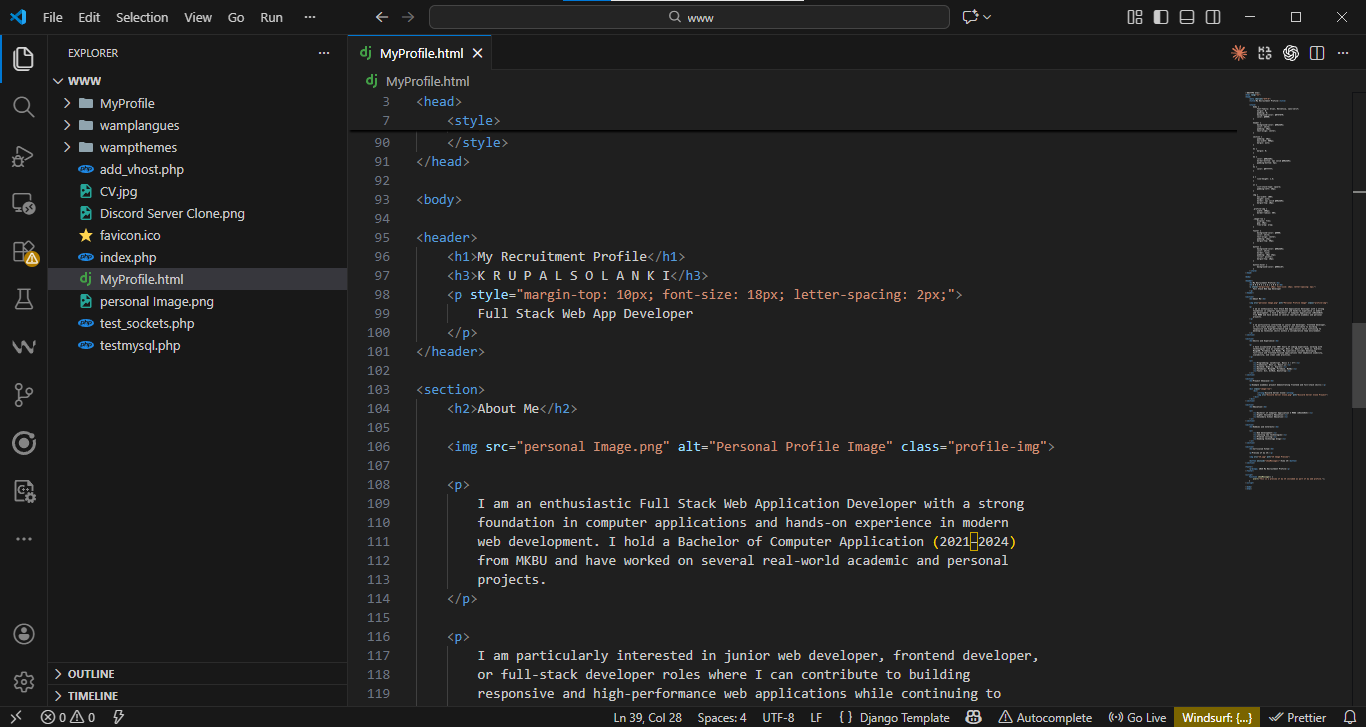


Figure 1: HTML, CSS, and JavaScript Code for Personal Recruitment Profile Web Page (Unit 1 – Task 1)

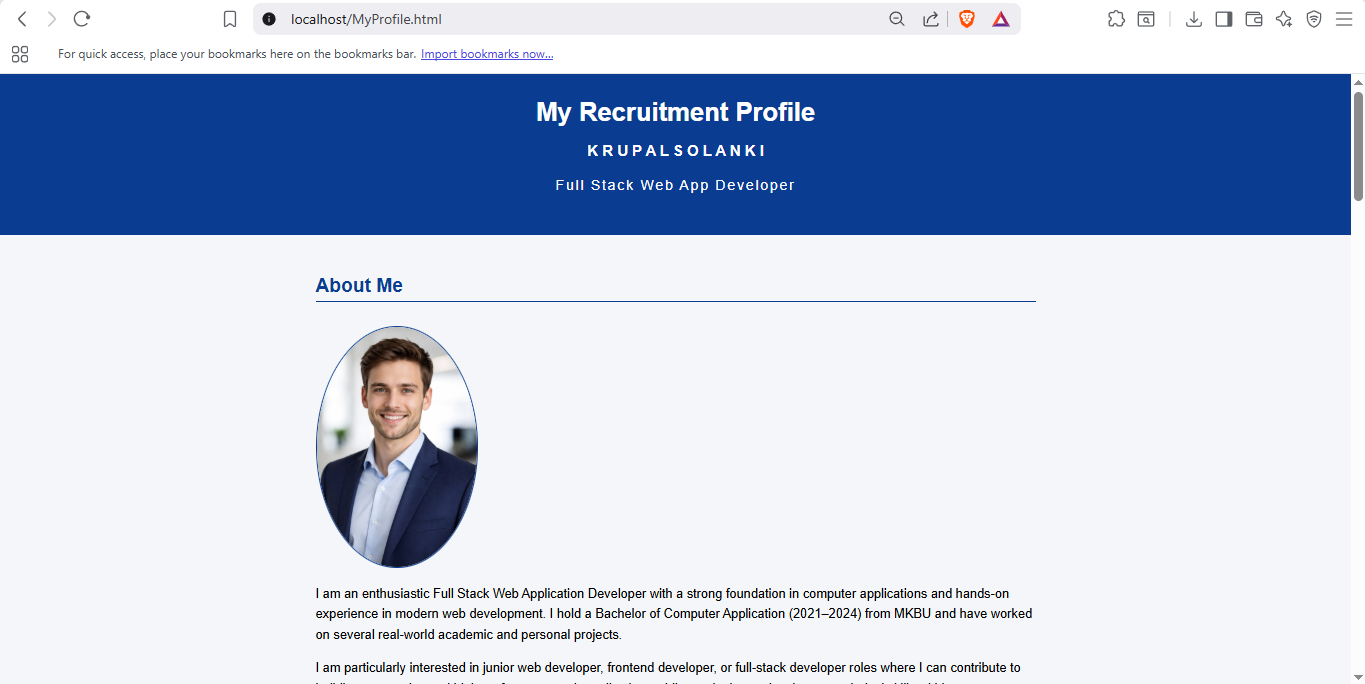


Figure 2: Rendered Output of Personal Recruitment Profile Web Page in Web Browser (Unit 1 – Task 1)

# Unit 2 Task 1

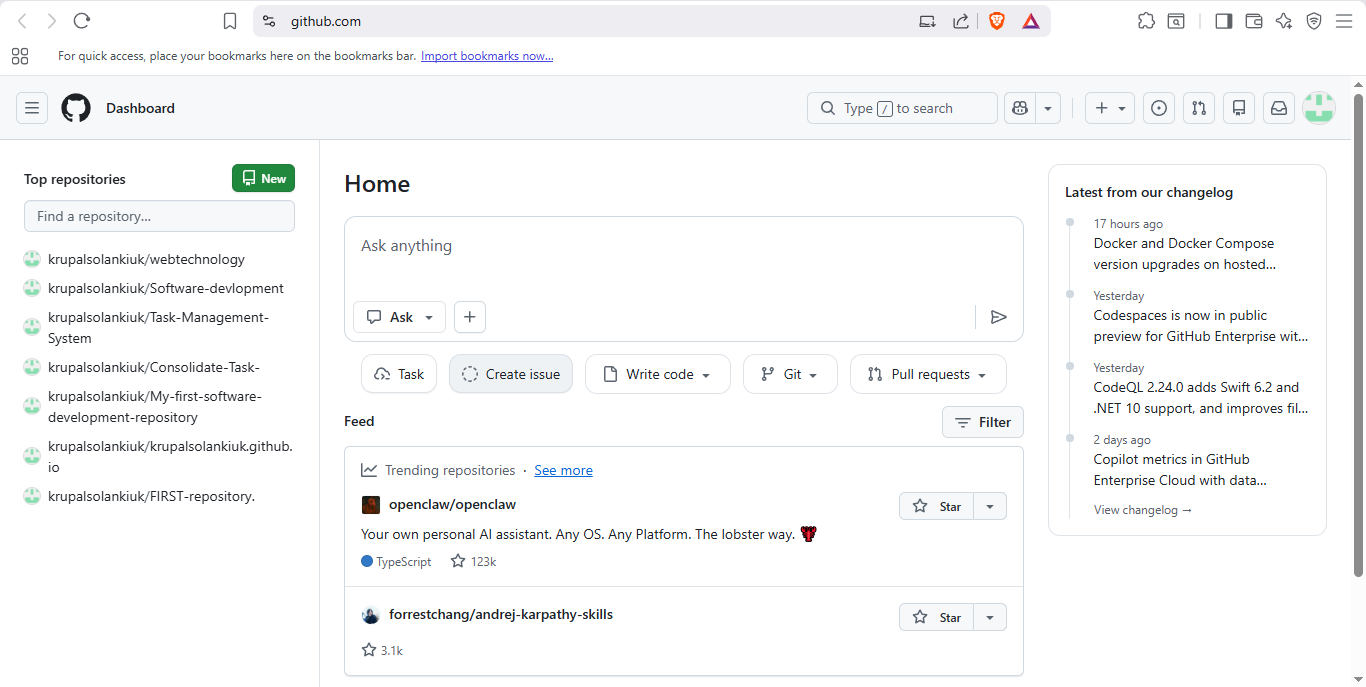


Figure 1: GitHub Dashboard After Successful Login (Unit 2 – Task 1)

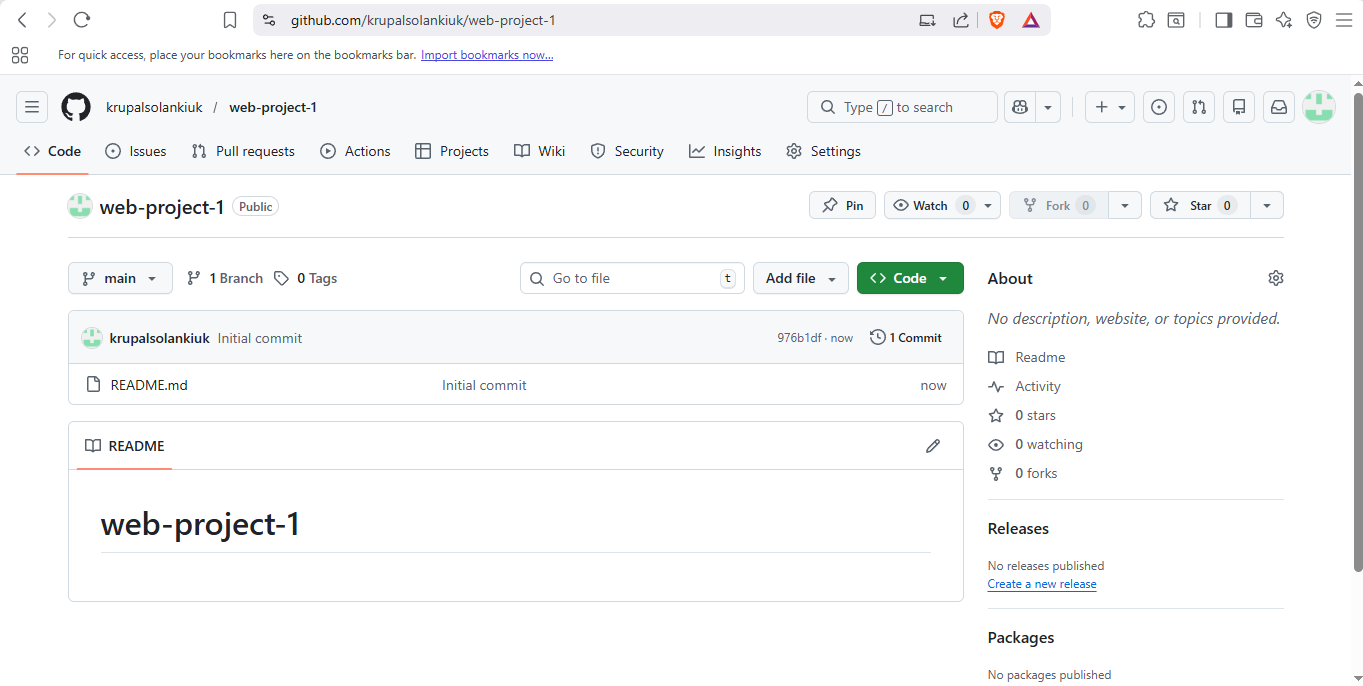


Figure 2: Web Project Repository Homepage on GitHub (Unit 2 – Task 1)

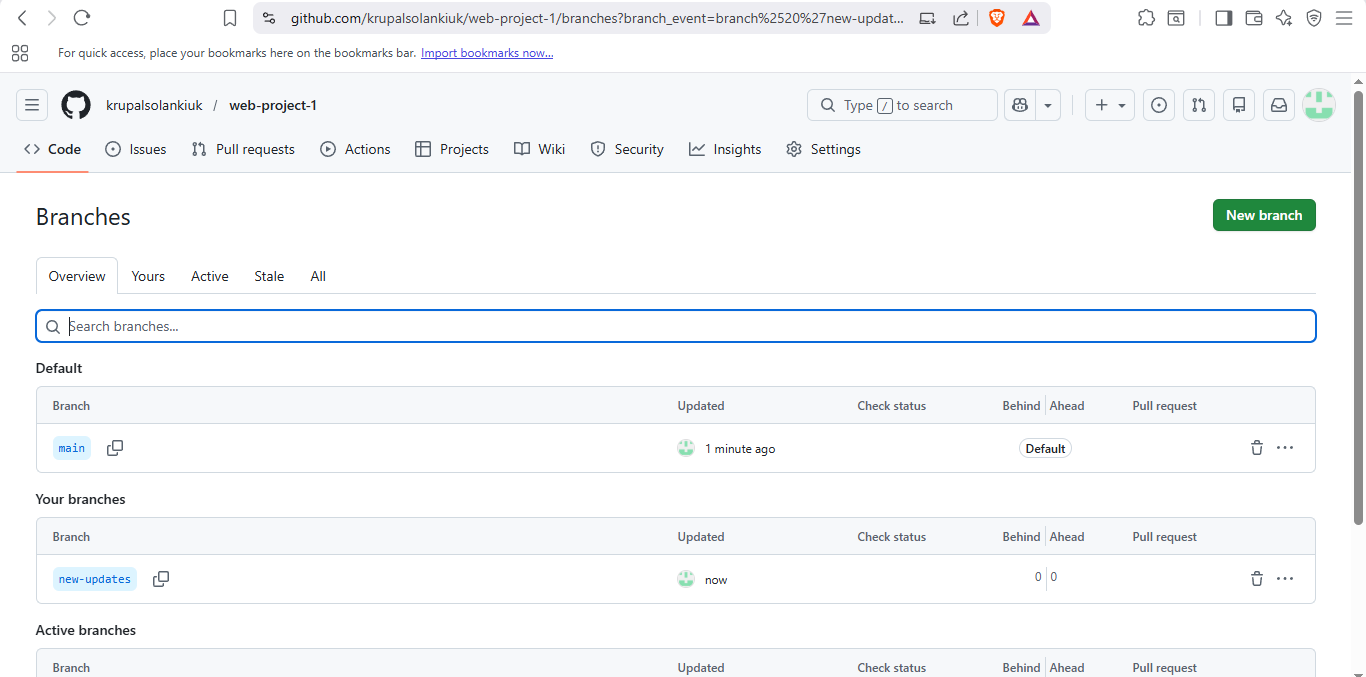


Figure 3: Creation of “new-updates” Branch in GitHub Repository (Unit 2 – Task 1)

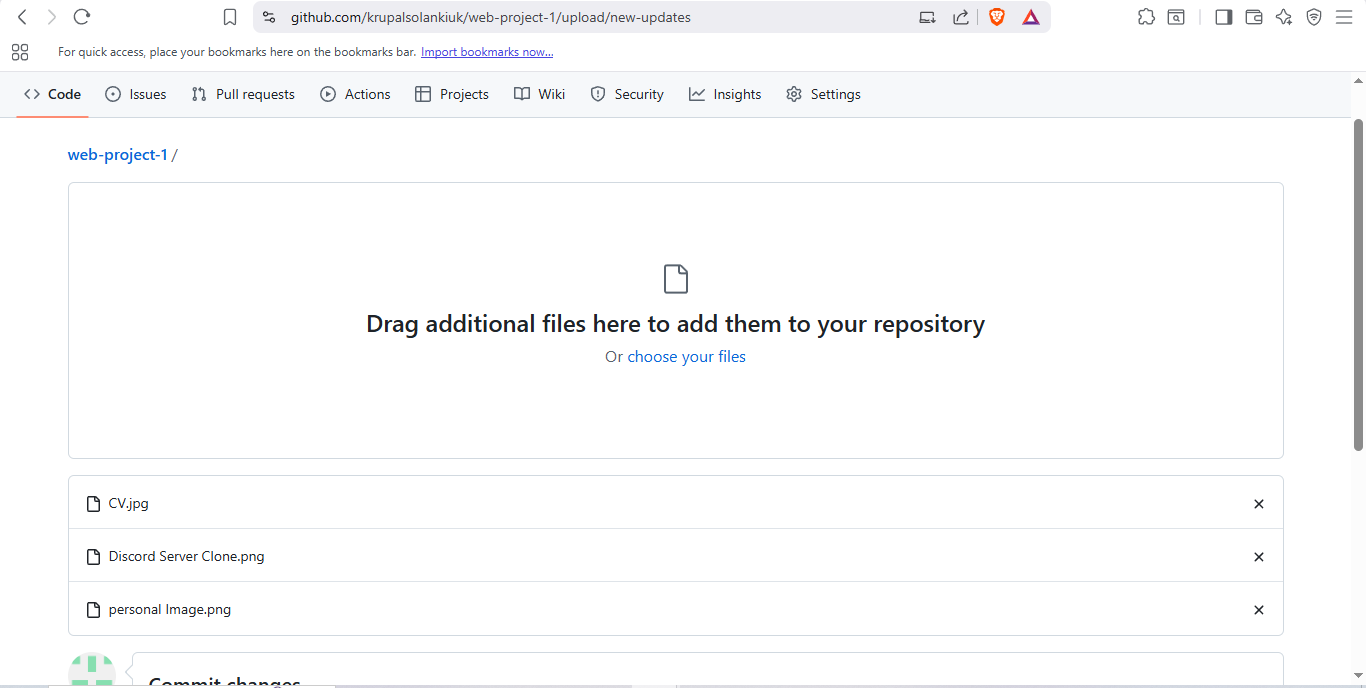


Figure 4: Uploading Image Files to the New Branch (Unit 2 – Task 1)

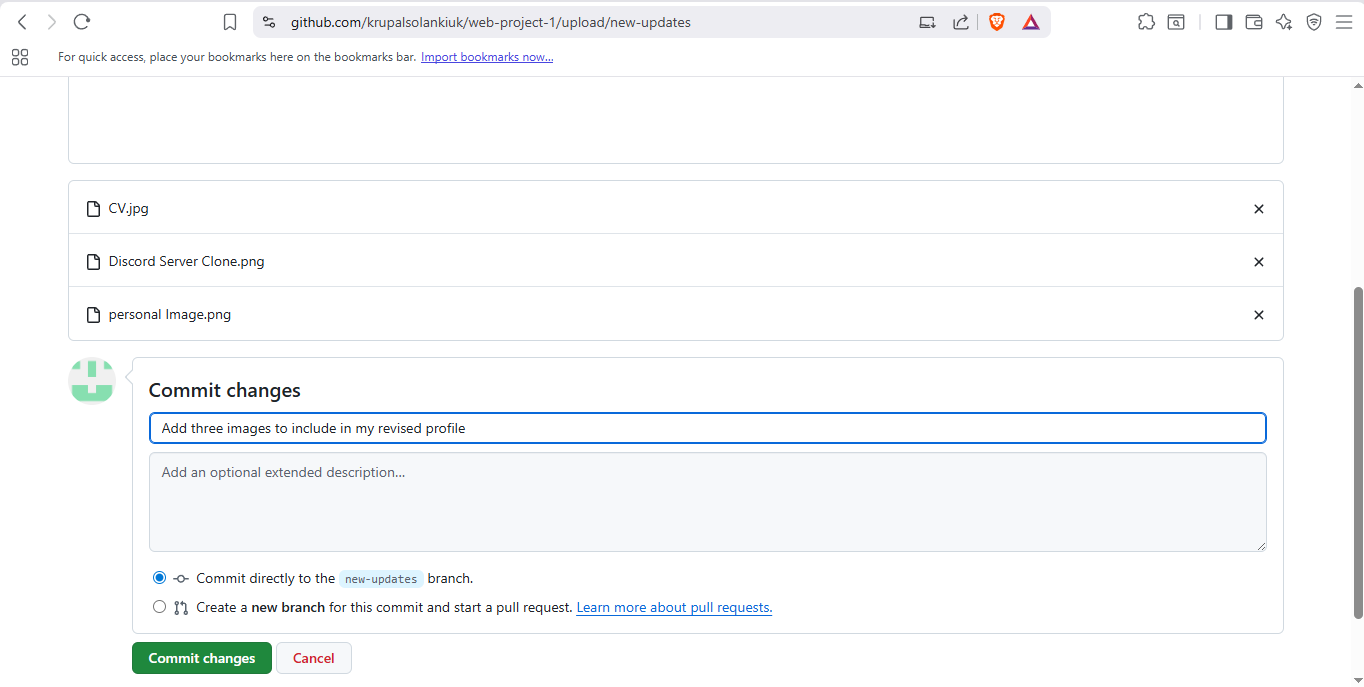


Figure 5: Committing Image Files to the “new-updates” Branch (Unit 2 – Task 1)

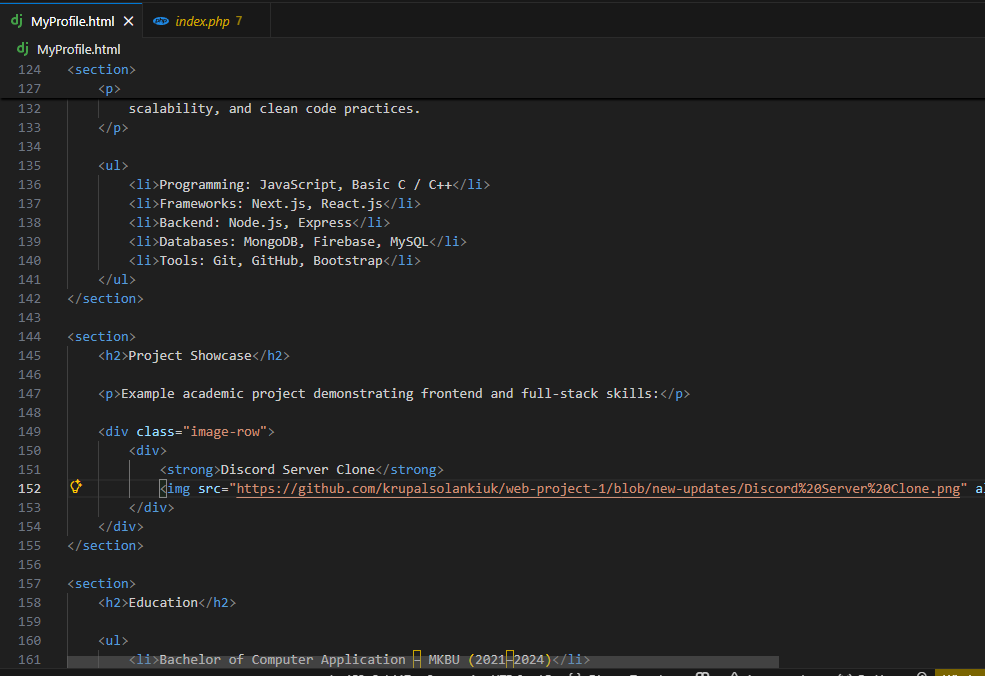


Figure 6: Updating HTML File with Image Placeholders (Unit 2 – Task 1)

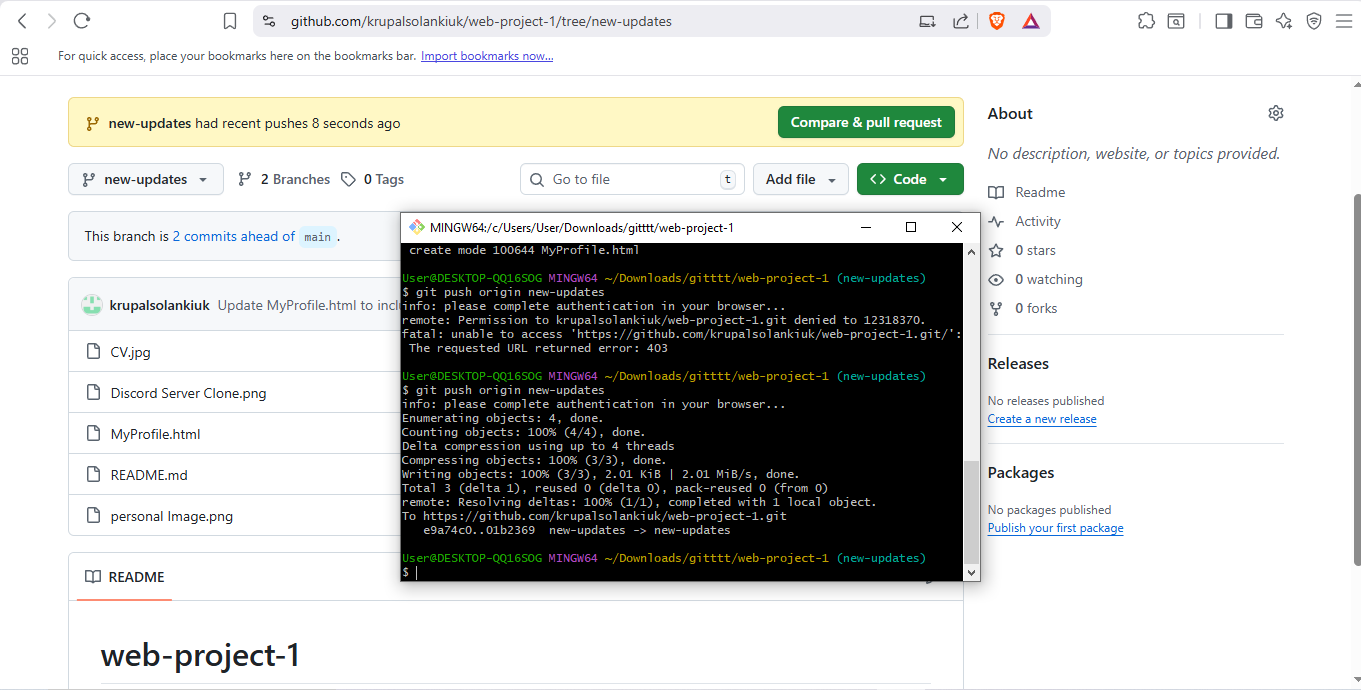


Figure 7: Updated Repository Showing Commits on New Branch (Unit 2 – Task 1)

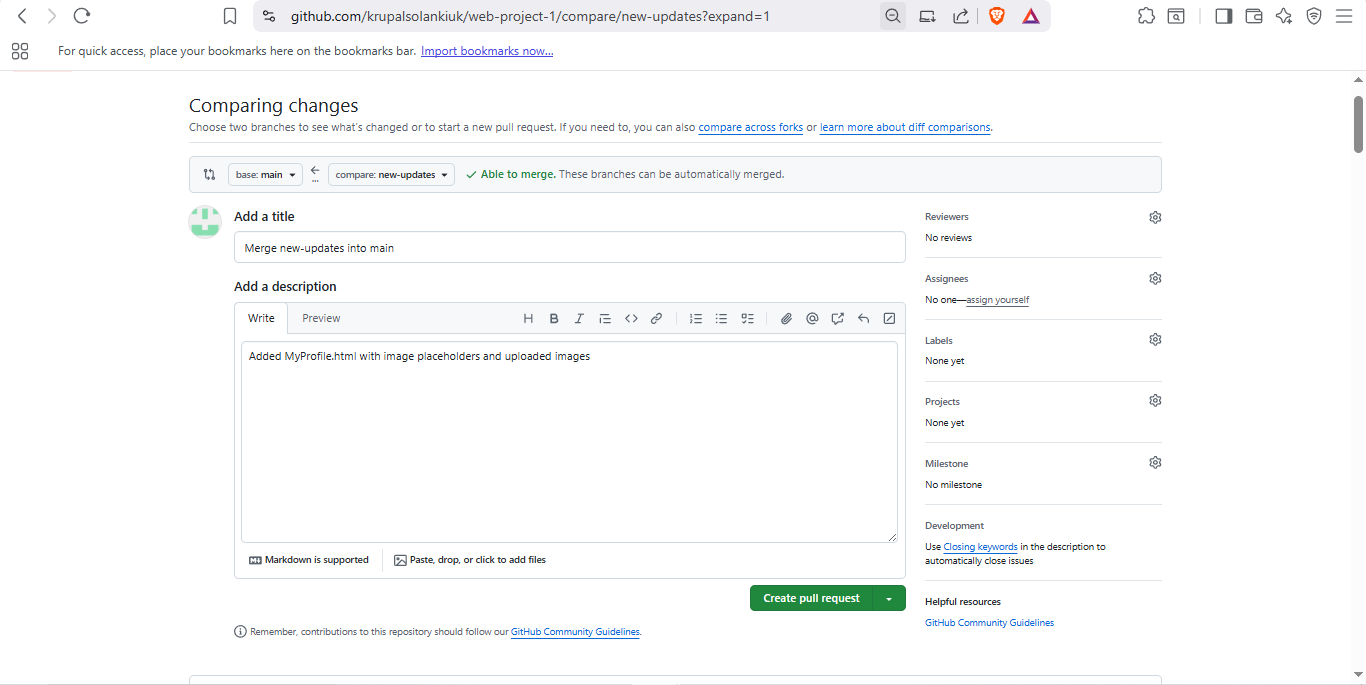


Figure 8: Pull Request Created to Merge “new-updates” into Main Branch (Unit 2 – Task 1)

GitHub Link: <https://github.com/krupalsolankiuk/web-project-1.git>

# Unit 3 Task 1

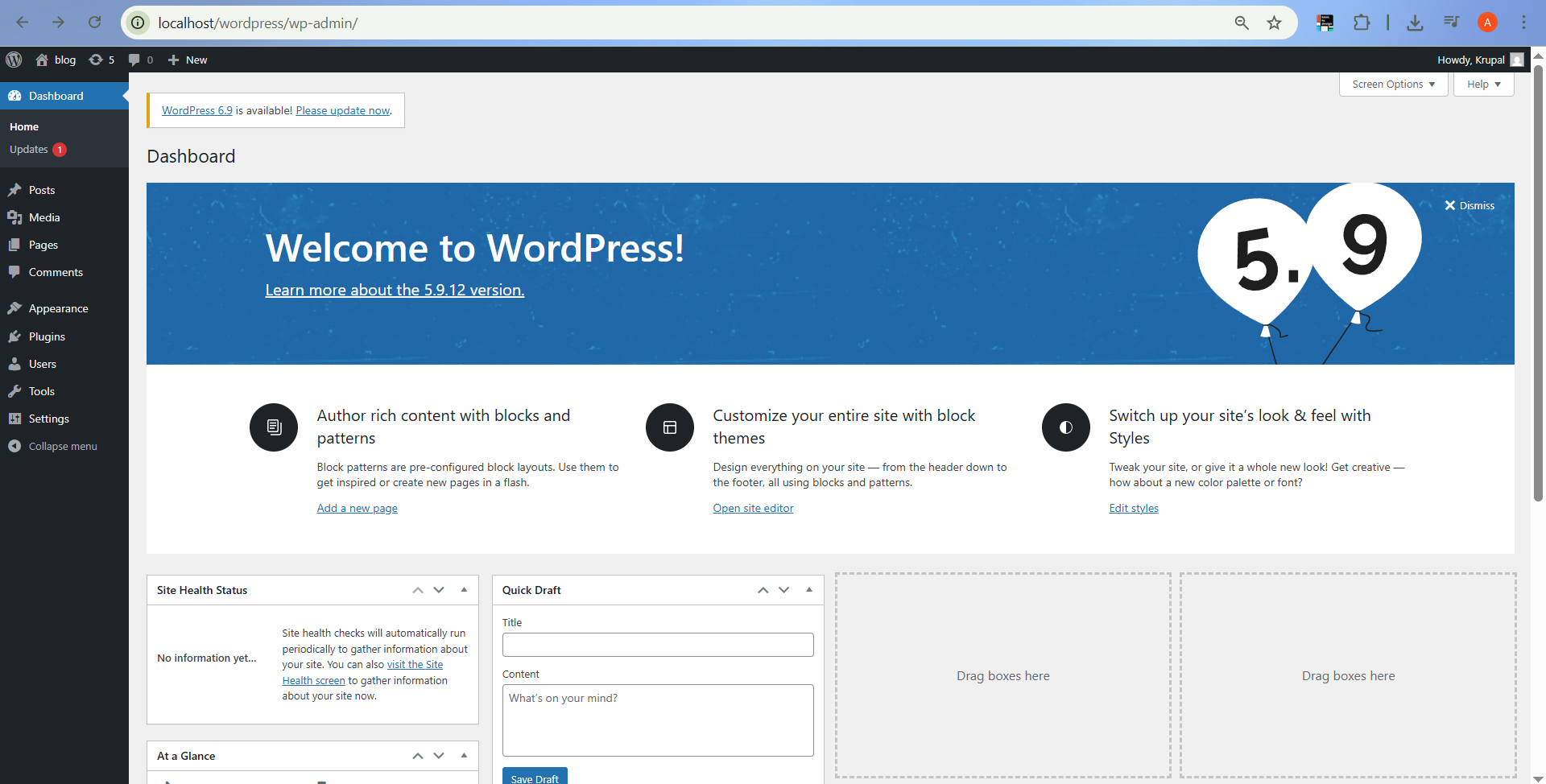


Figure 1: WordPress Dashboard Interface (Unit 3 – Task 1)

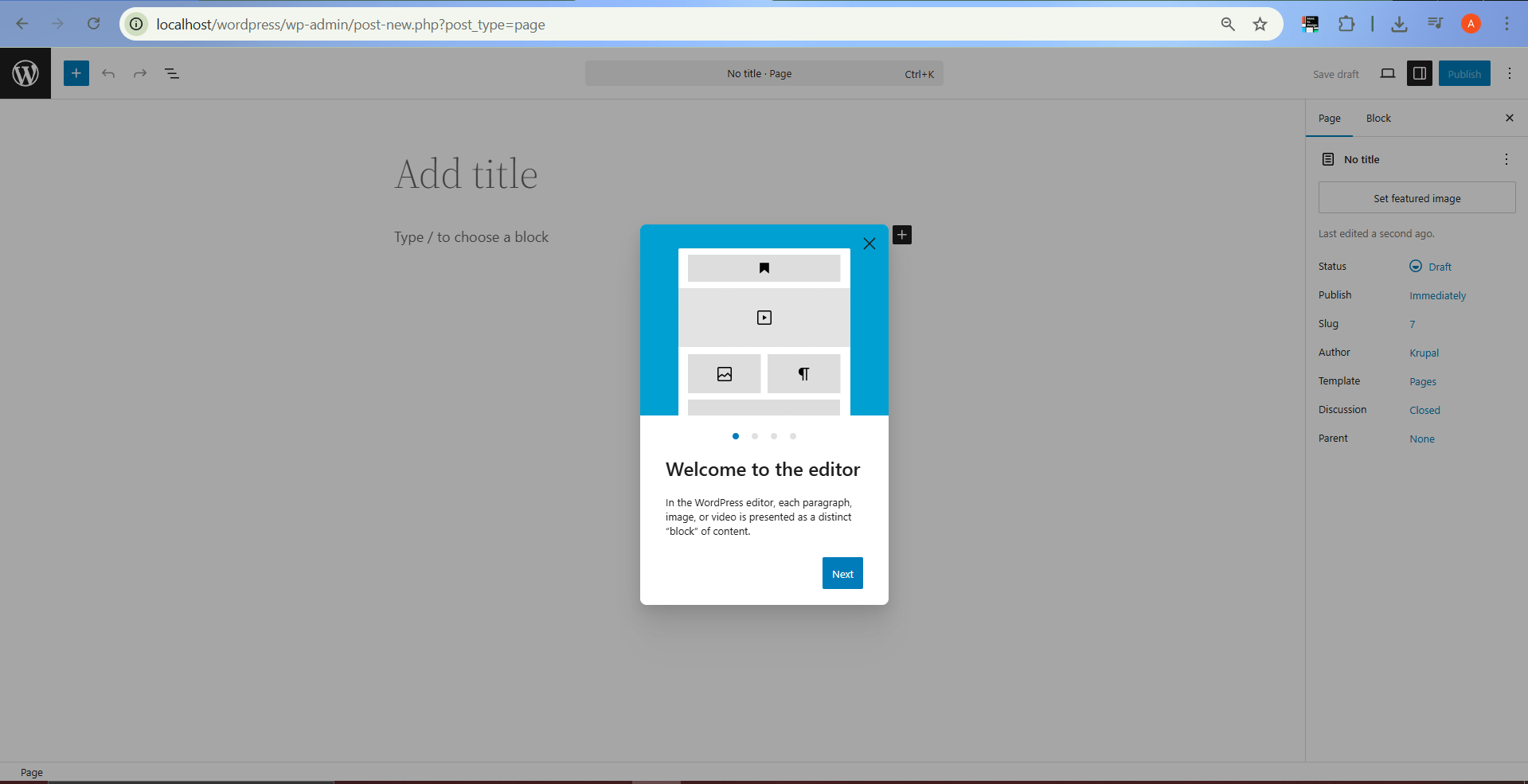


Figure 2: Creating a New Blog Post Using WordPress Editor (Unit 3 – Task 1)

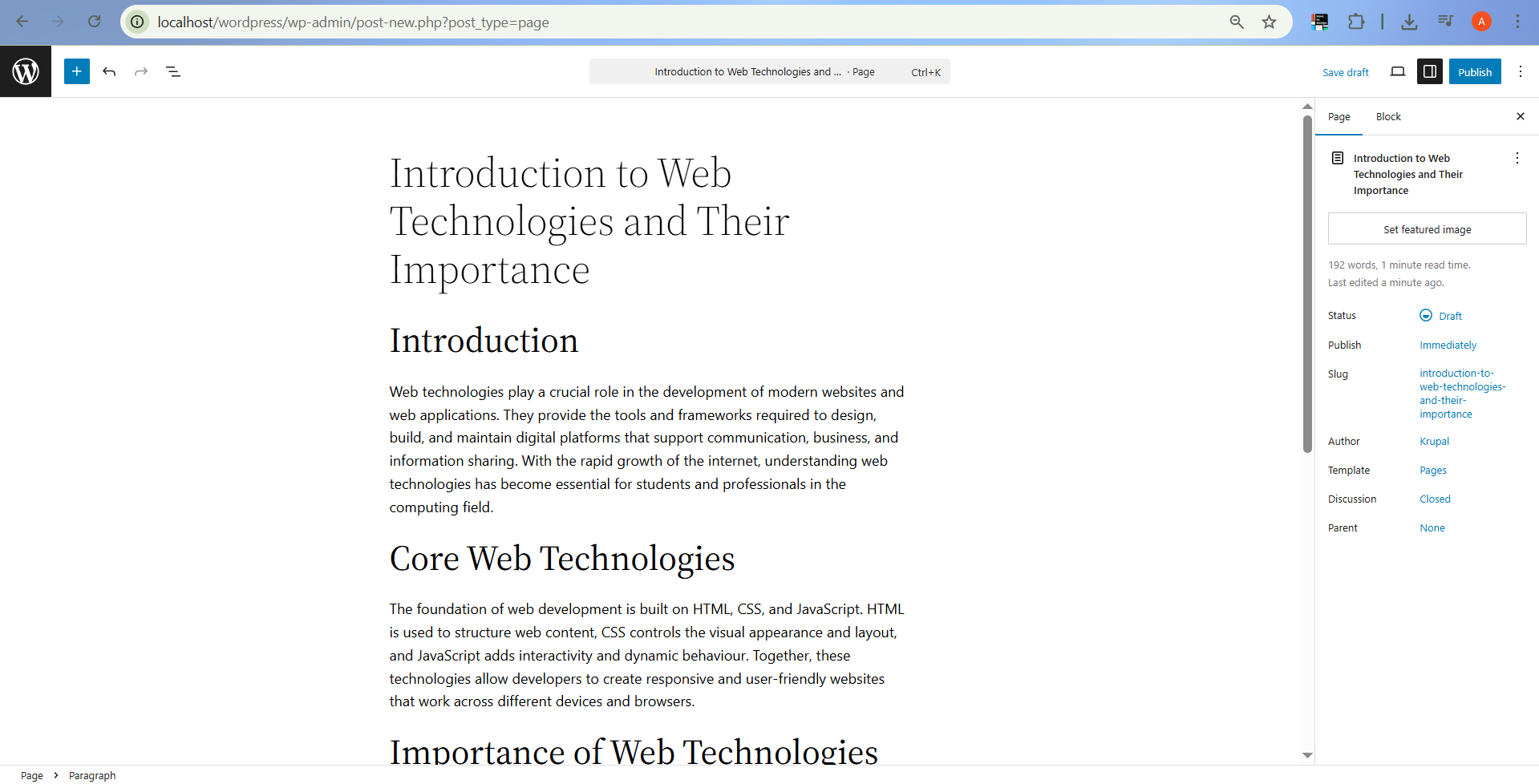


Figure 3: Blog Post Content Added Using WordPress Visual Editor (Unit 3 – Task 1)

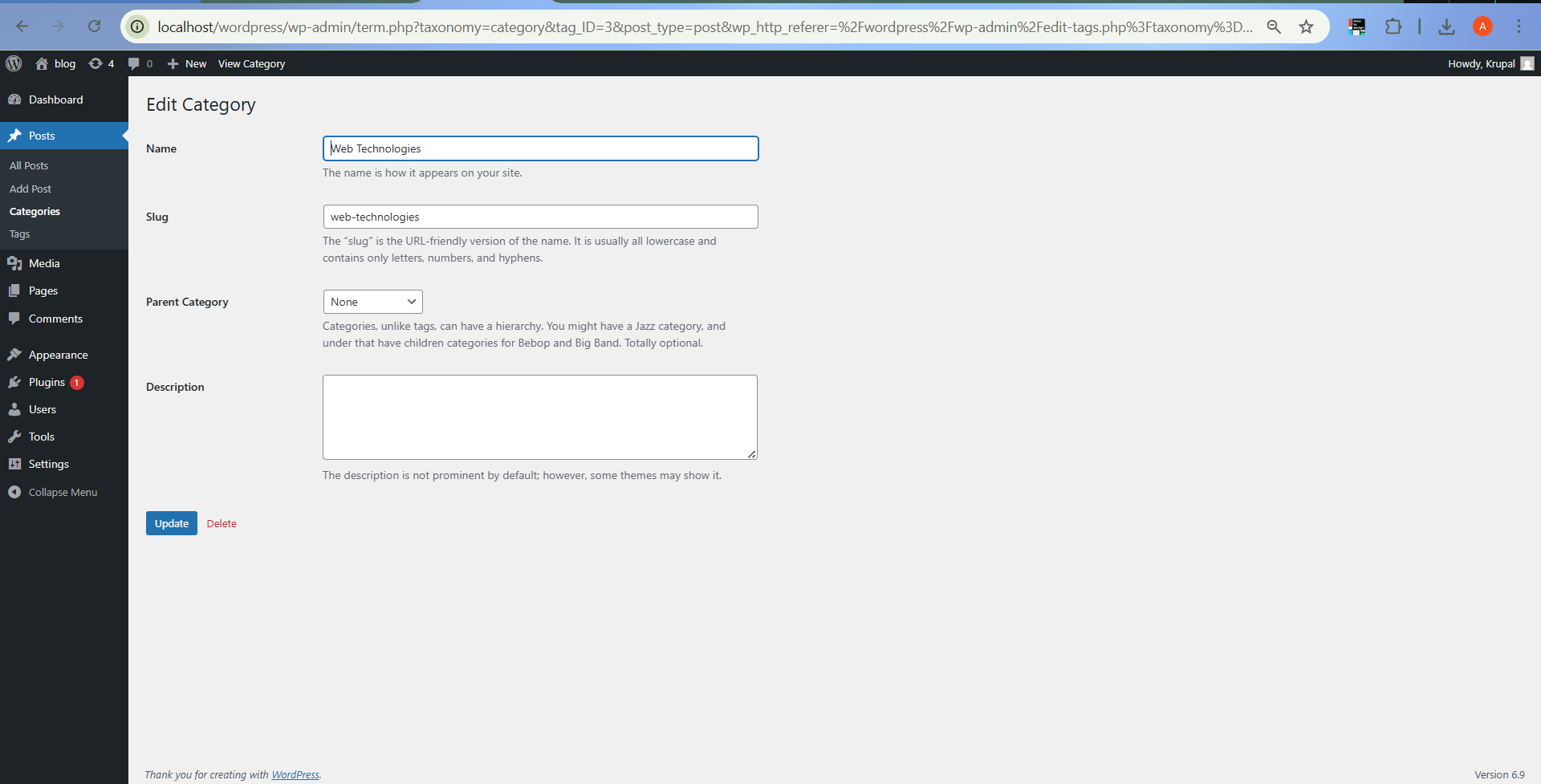


Figure 4: Assigning Categories and Tags to Blog Post (Unit 3 – Task 1)

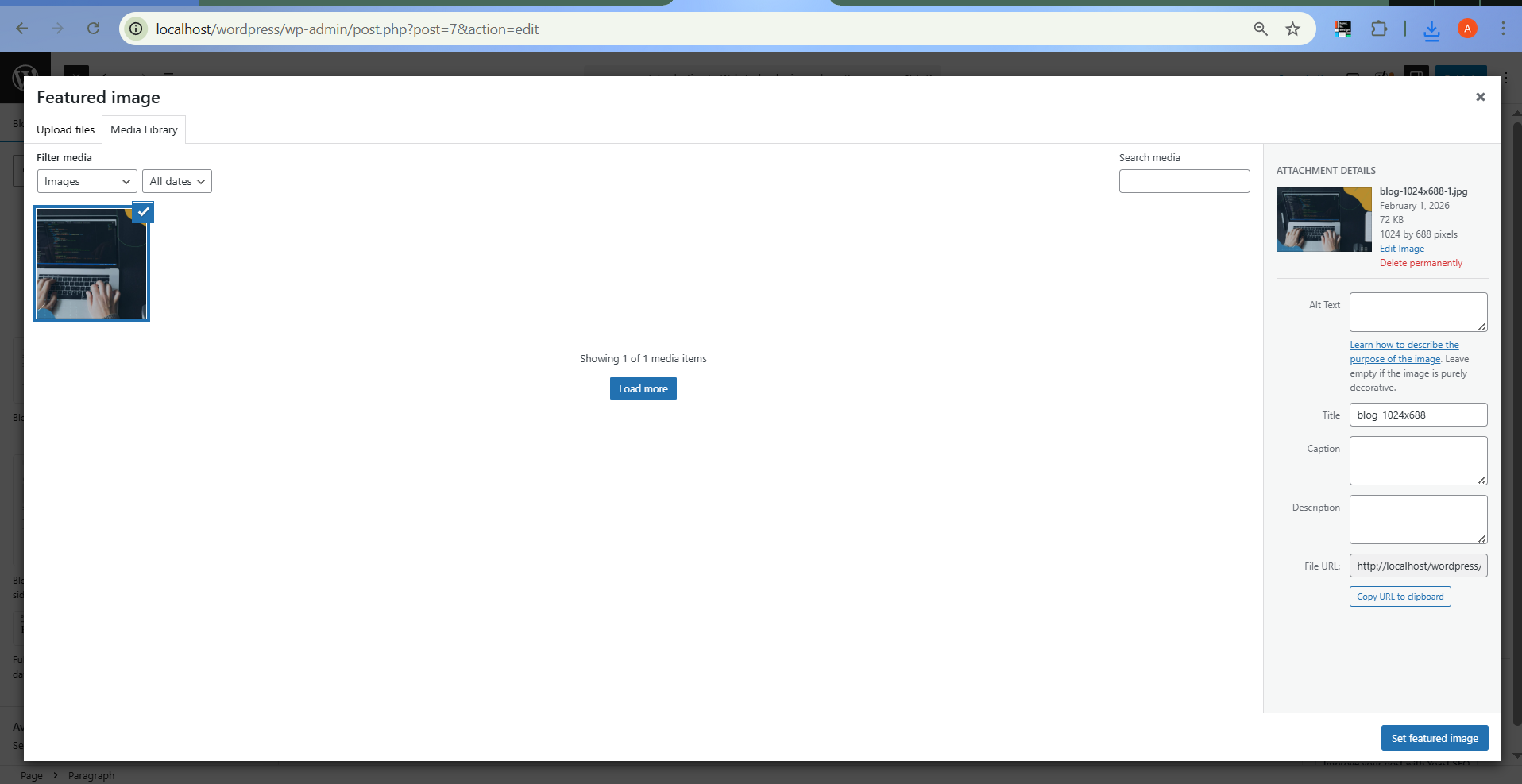


Figure 5: Featured Image Added to Blog Post (Unit 3 – Task 1)

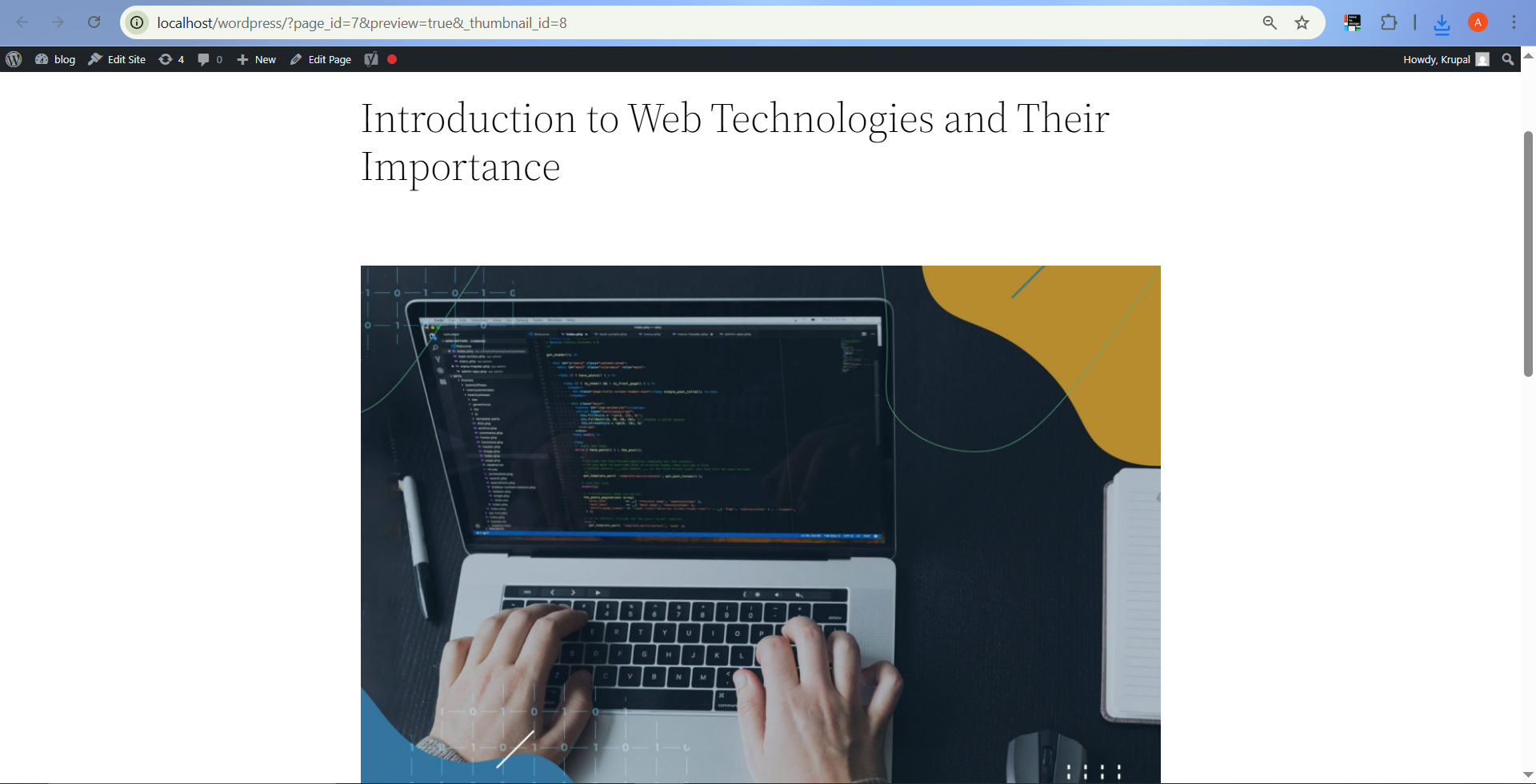


Figure 6: Previewing Blog Post Before Publishing (Unit 3 – Task 1)

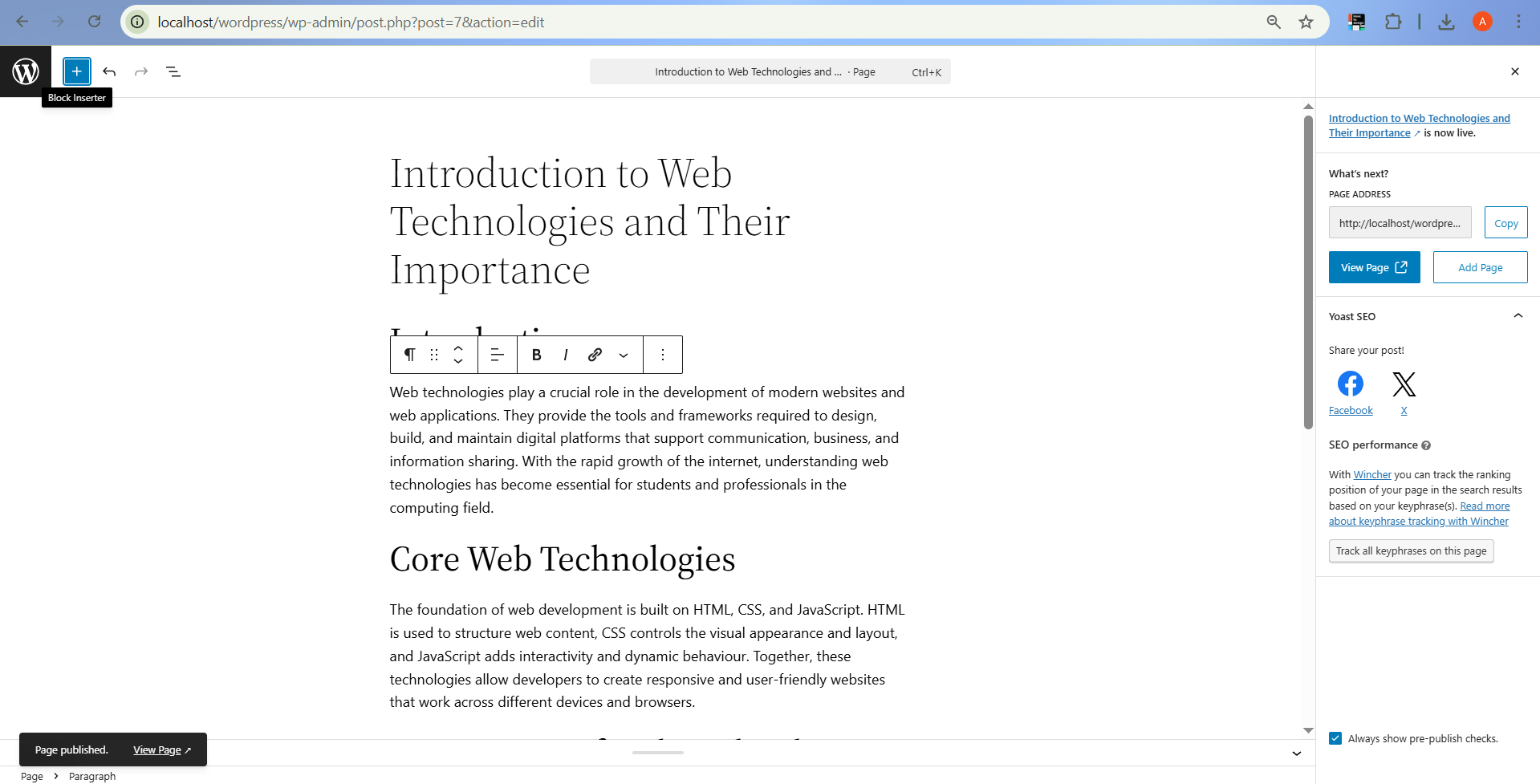


Figure 7: Blog Post Successfully Published on WordPress Website (Unit 3 – Task 1)

Unit 3 Task 1 focused on creating and publishing a blog post using the WordPress CMS. This task enhanced my understanding of content management systems and demonstrated how web content can be efficiently created, organised, and published without extensive coding knowledge. I gained practical experience in using the WordPress block editor, assigning categories and tags, adding featured images, and publishing live content. Figures 1–7 illustrate the key stages involved in creating and publishing the blog post.

# Unit 4 Task 1

Code

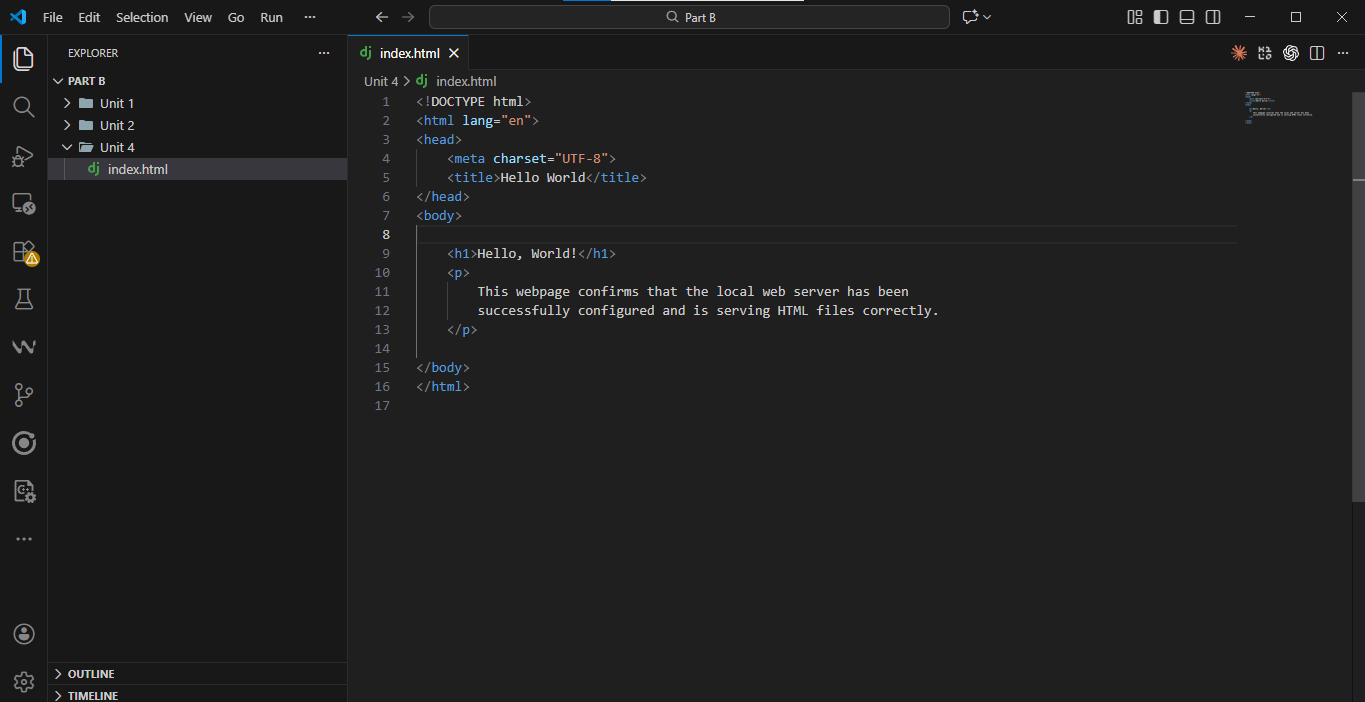


Figure 1: HTML Code for “Hello, World!” Webpage Used to Test Local Server Configuration (Unit 4 – Task 1)

Output



Figure 2: Rendered “Hello, World!” Webpage Displayed via Localhost Web Server (Unit 4 – Task 1)

Unit 4 Task 1 involved configuring a local web server to serve basic HTML files. A simple “Hello, World!” webpage was created and placed in the web server root directory. The successful display of the page via localhost confirmed that the server was correctly configured and functioning as intended.

# Unit 5 Task 1

**Code**

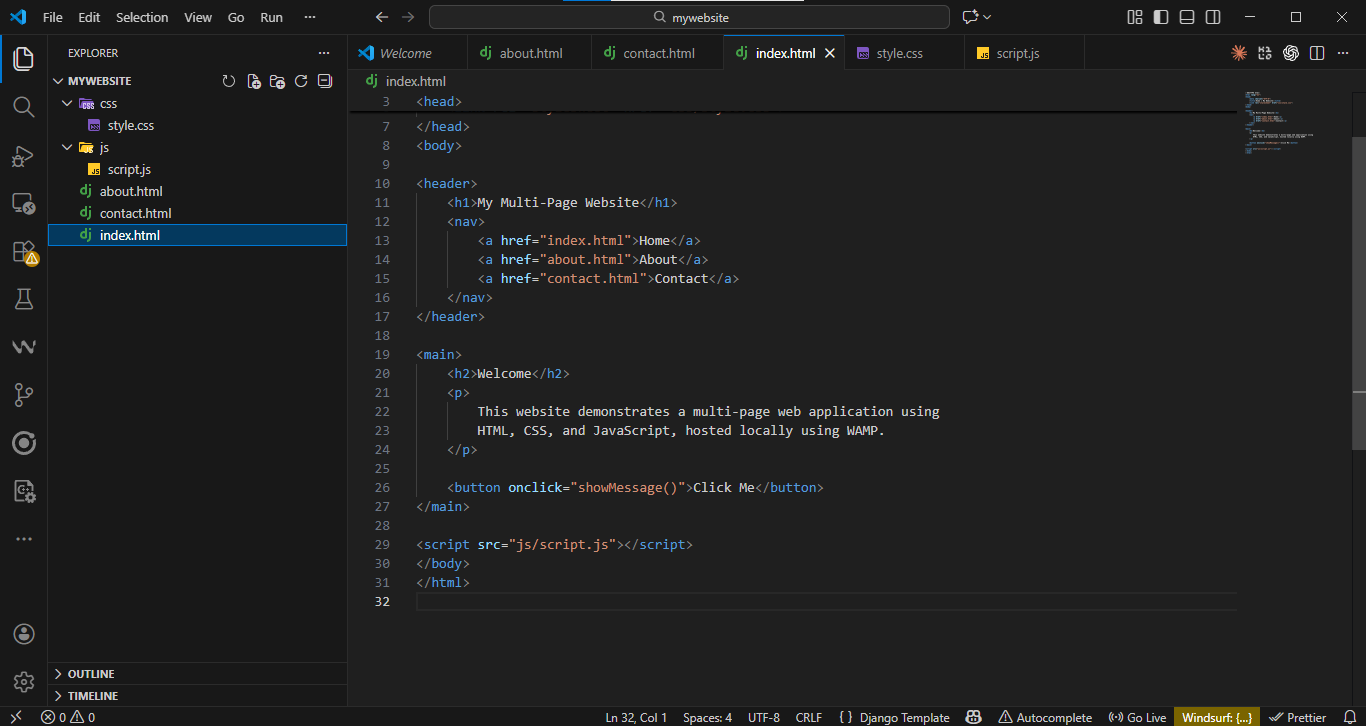


Figure 1: HTML Code for Multi-Page Website Structure (Unit 5 – Task 1)

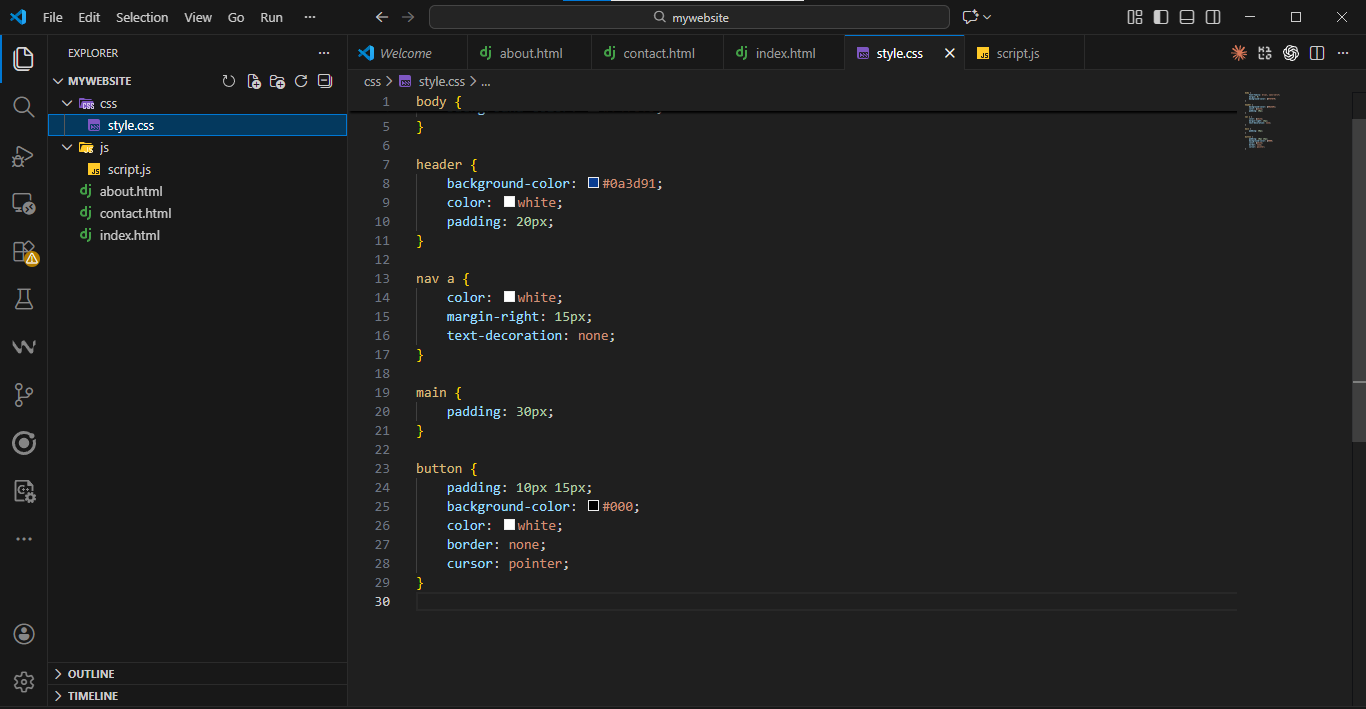


Figure 2: CSS Code for Styling the Multi-Page Website (Unit 5 – Task 1)



Figure 3: JavaScript Code Implementing Dynamic Website Functionality (Unit 5 – Task 1)

**Output**

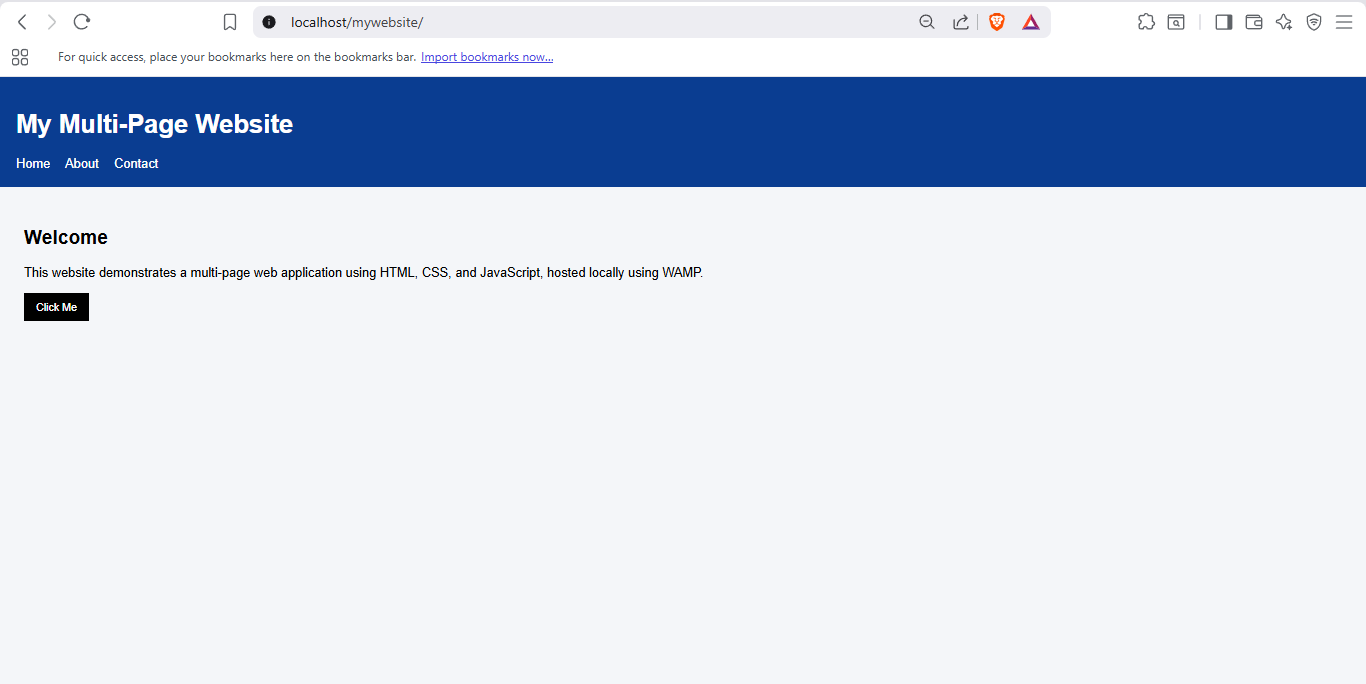


Figure 4: Home Page of Multi-Page Website Displayed via Localhost (Unit 5 – Task 1)



Figure 5: About Page Navigation and Content Displayed in Browser (Unit 5 – Task 1)

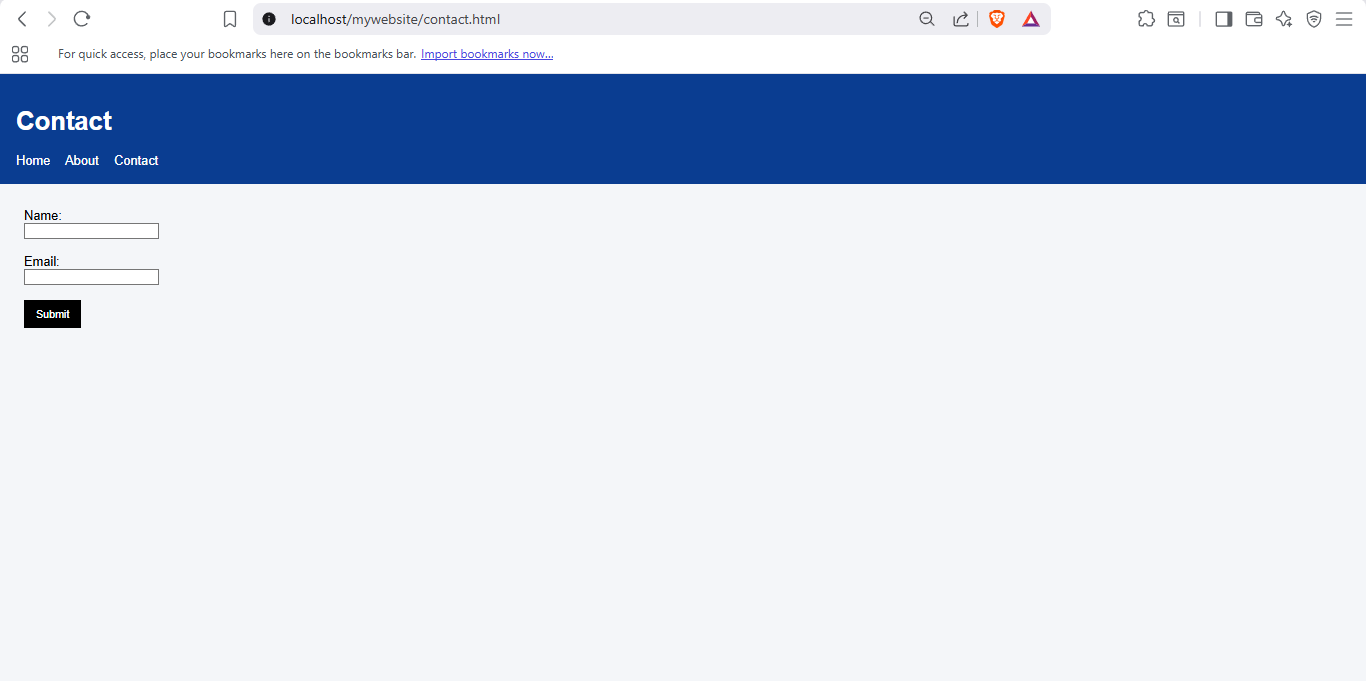
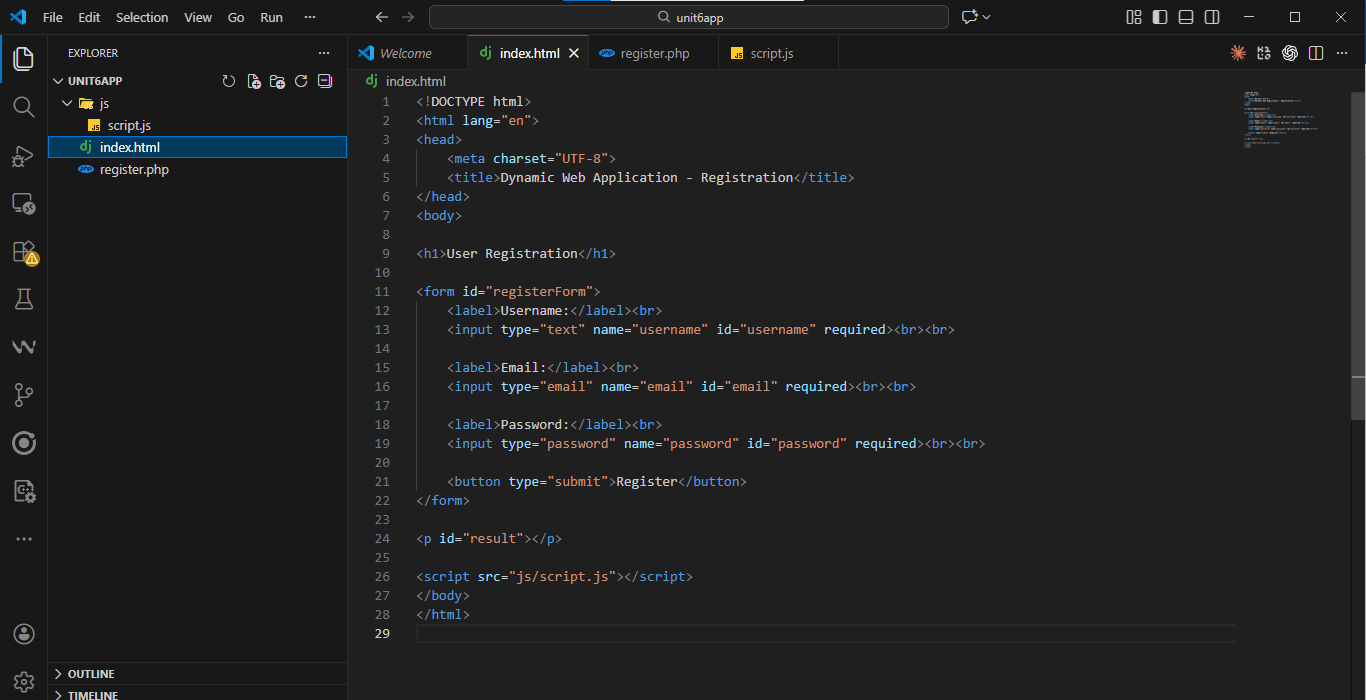


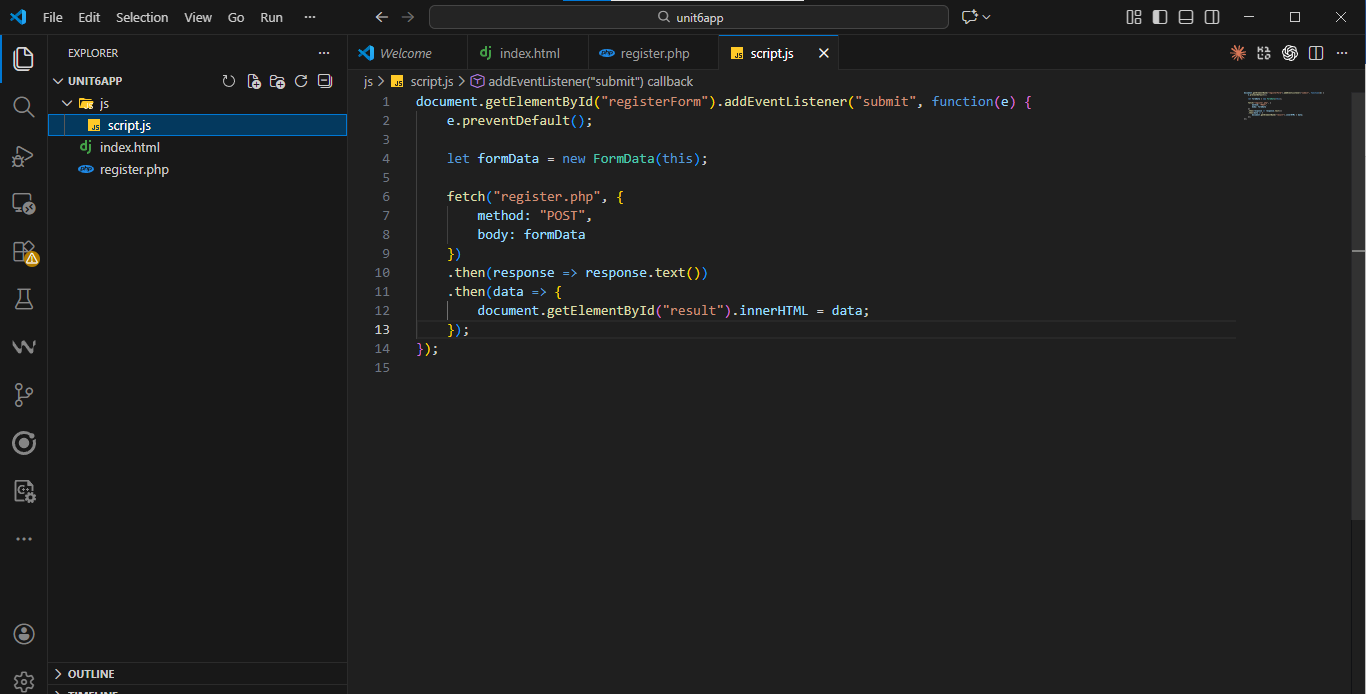
Figure 6: Contact Page Form Validation Using JavaScript (Unit 5 – Task 1)

# Unit 6 Task 1

Code

****

**Figure 1: HTML Code for User Registration Form (Unit 6 – Task 1)**

****

**Figure 2: JavaScript Code for Client-Side Validation and AJAX Form Submission (Unit 6 – Task 1)**

****

**Figure 3: PHP Server-Side Script for Processing Registration Data (Unit 6 – Task 1)**

Output

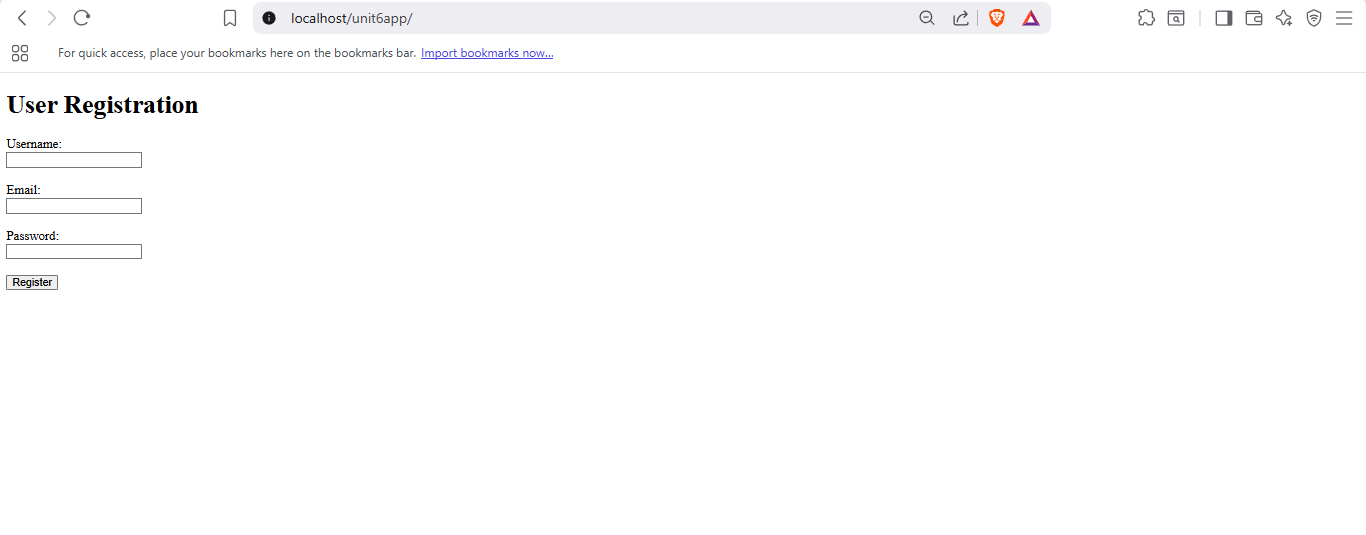


Figure 4: User Registration Form Interface Displayed in Browser (Unit 6 – Task 1)

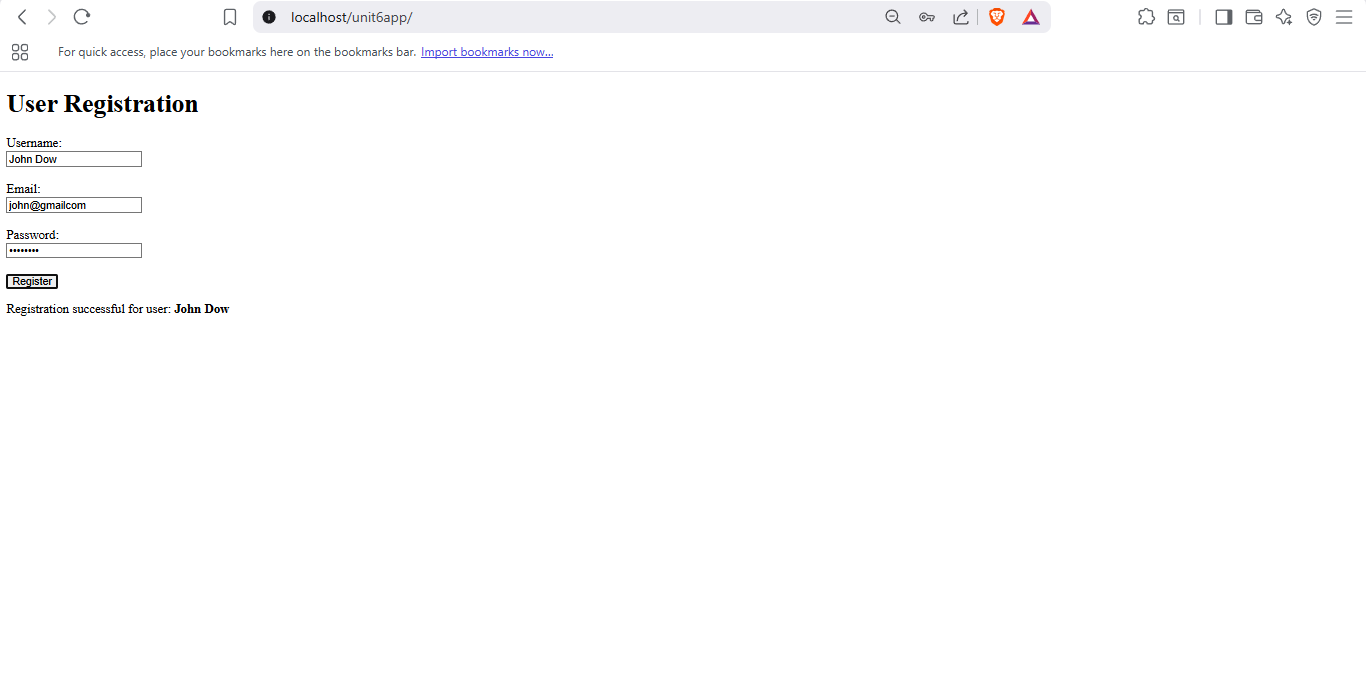
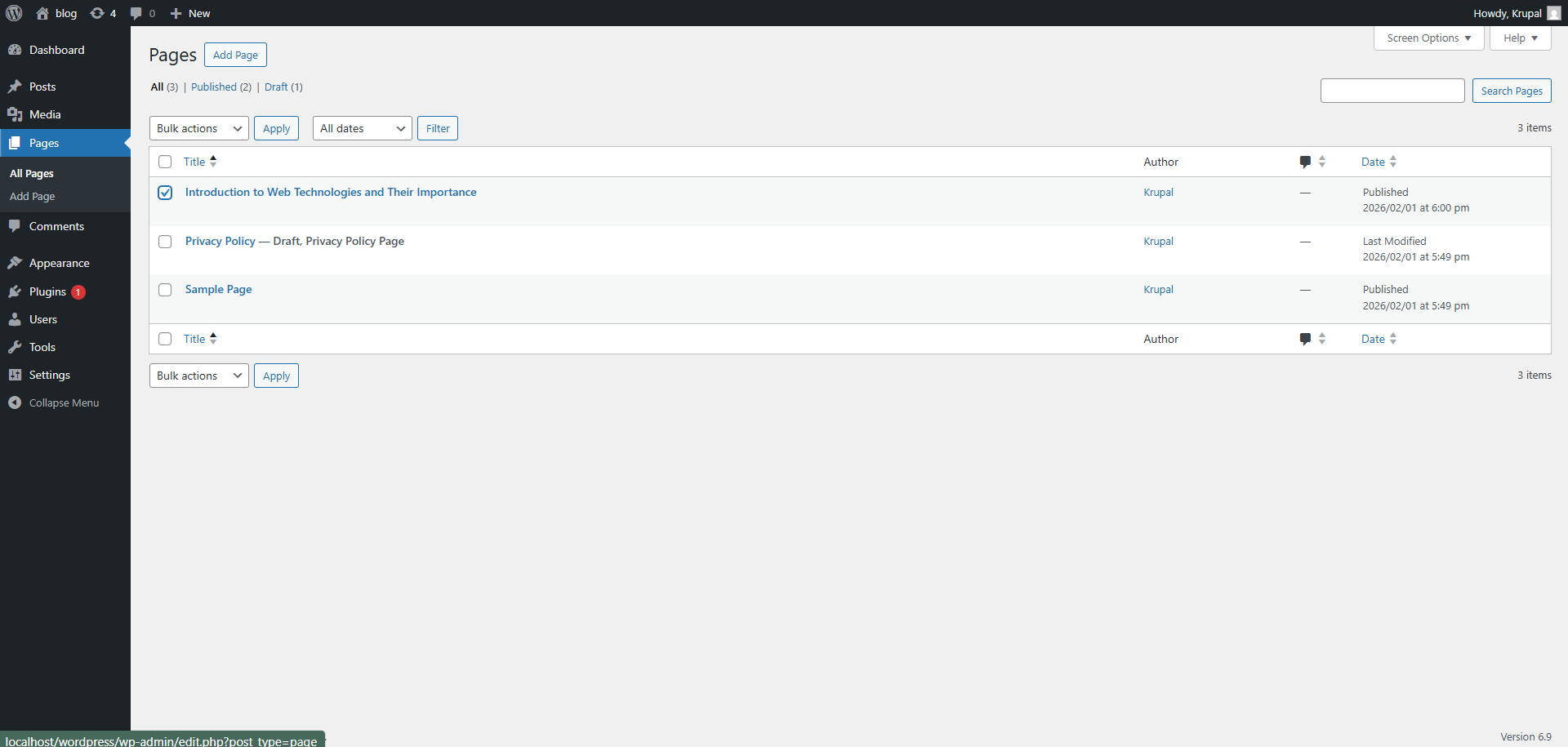


Figure 5: Dynamic Registration Confirmation Displayed Without Page Reload (Unit 6 – Task 1)

Unit 6 Task 1 involved designing a dynamic web application using PHP as a server-side technology. An interactive user registration form was created using HTML and enhanced with JavaScript for client-side validation and AJAX-based submission. The PHP script processed user input and dynamically returned a response without refreshing the page. This task demonstrated the integration of client-side and server-side technologies and highlighted the advantages of dynamic web applications over static HTML solutions.

# Unit 7 Task 1



**Figure 1: WordPress Admin Dashboard Interface (Unit 7 – Task 1)**

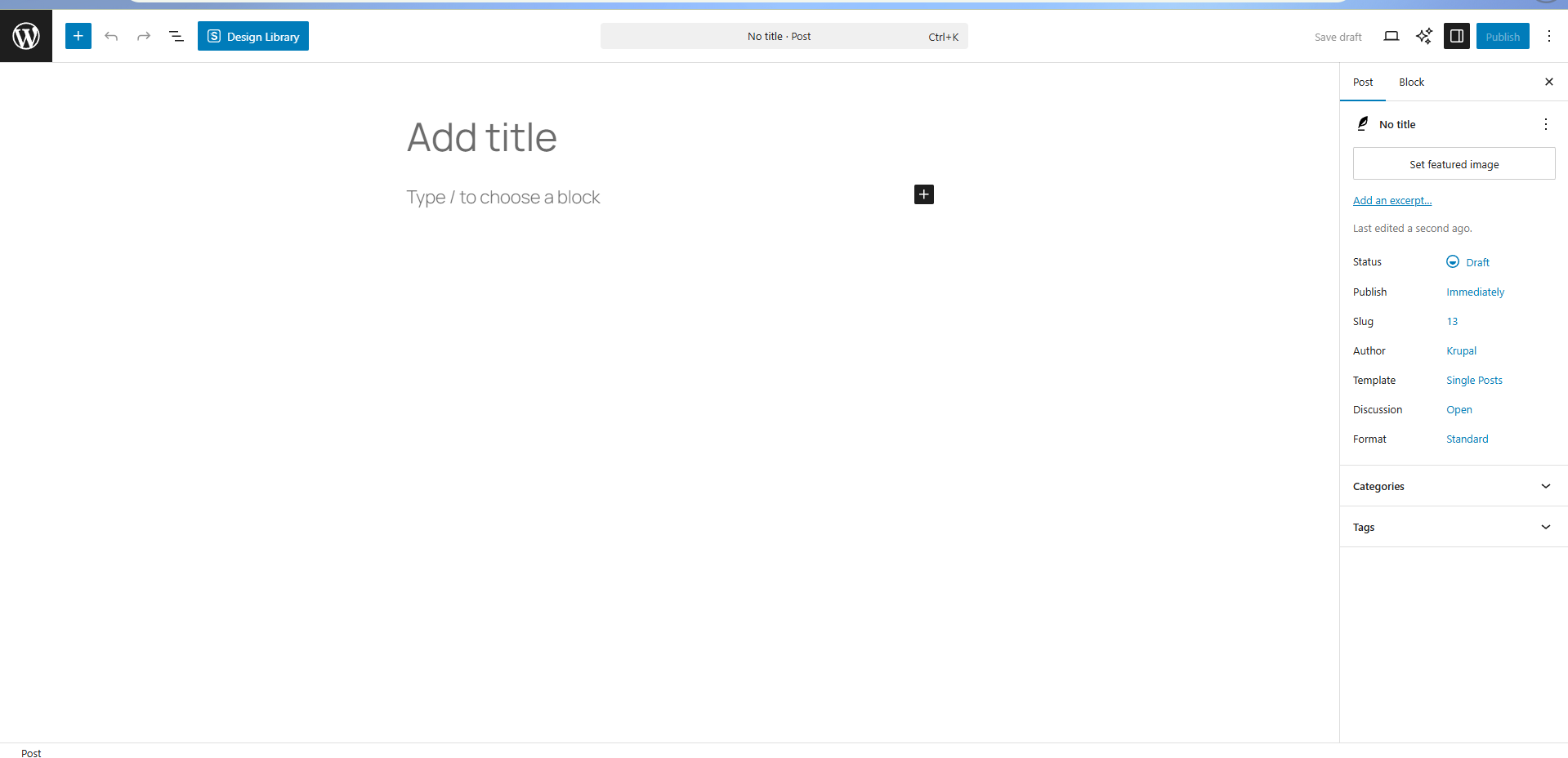


Figure 2: Creating a New Blog Post Using WordPress Editor (Unit 7 – Task 1)

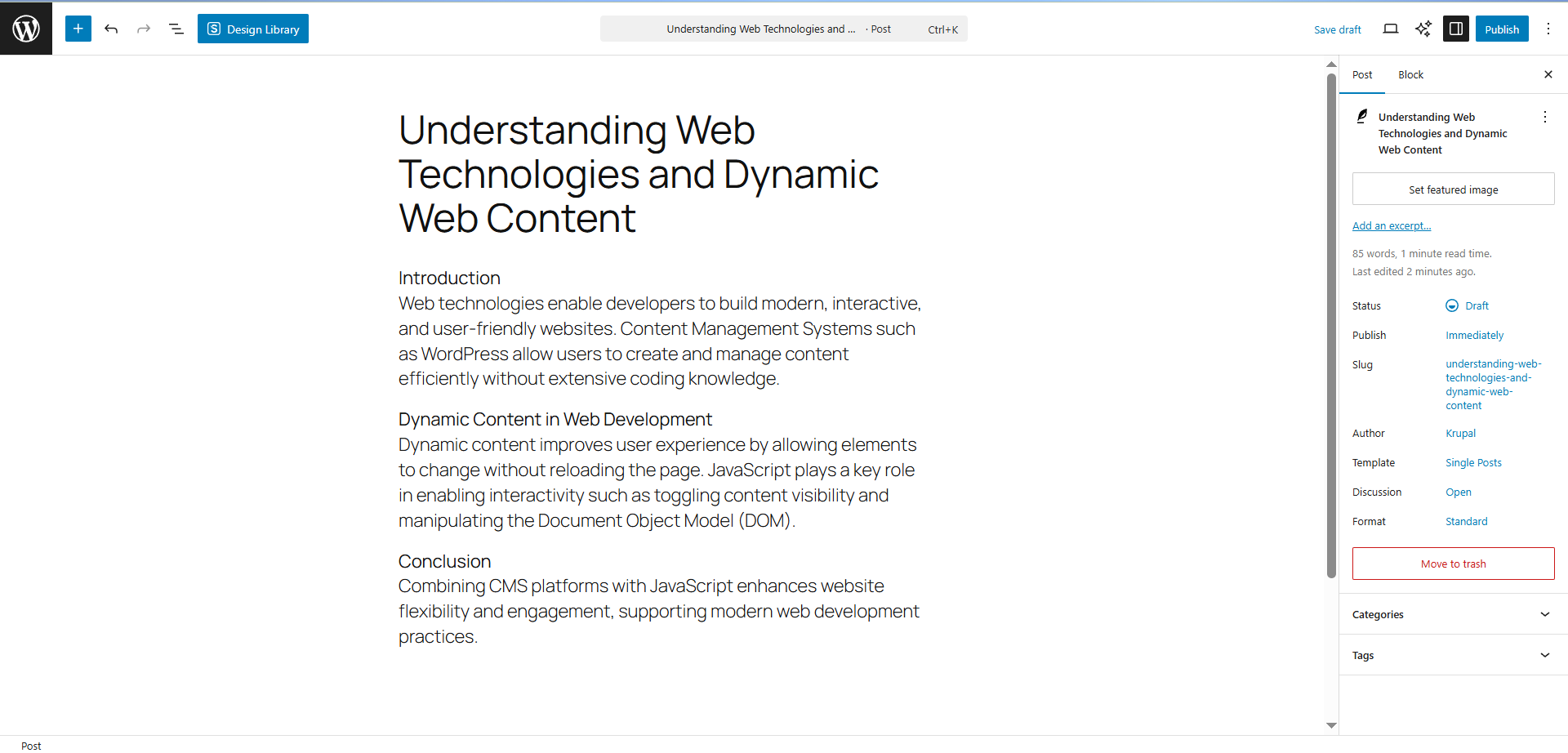


Figure 3: Blog Content Added Using WordPress Visual Editor (Unit 7 – Task 1)



Figure 4: Categories and Tags Assigned to Blog Post (Unit 7 – Task 1)

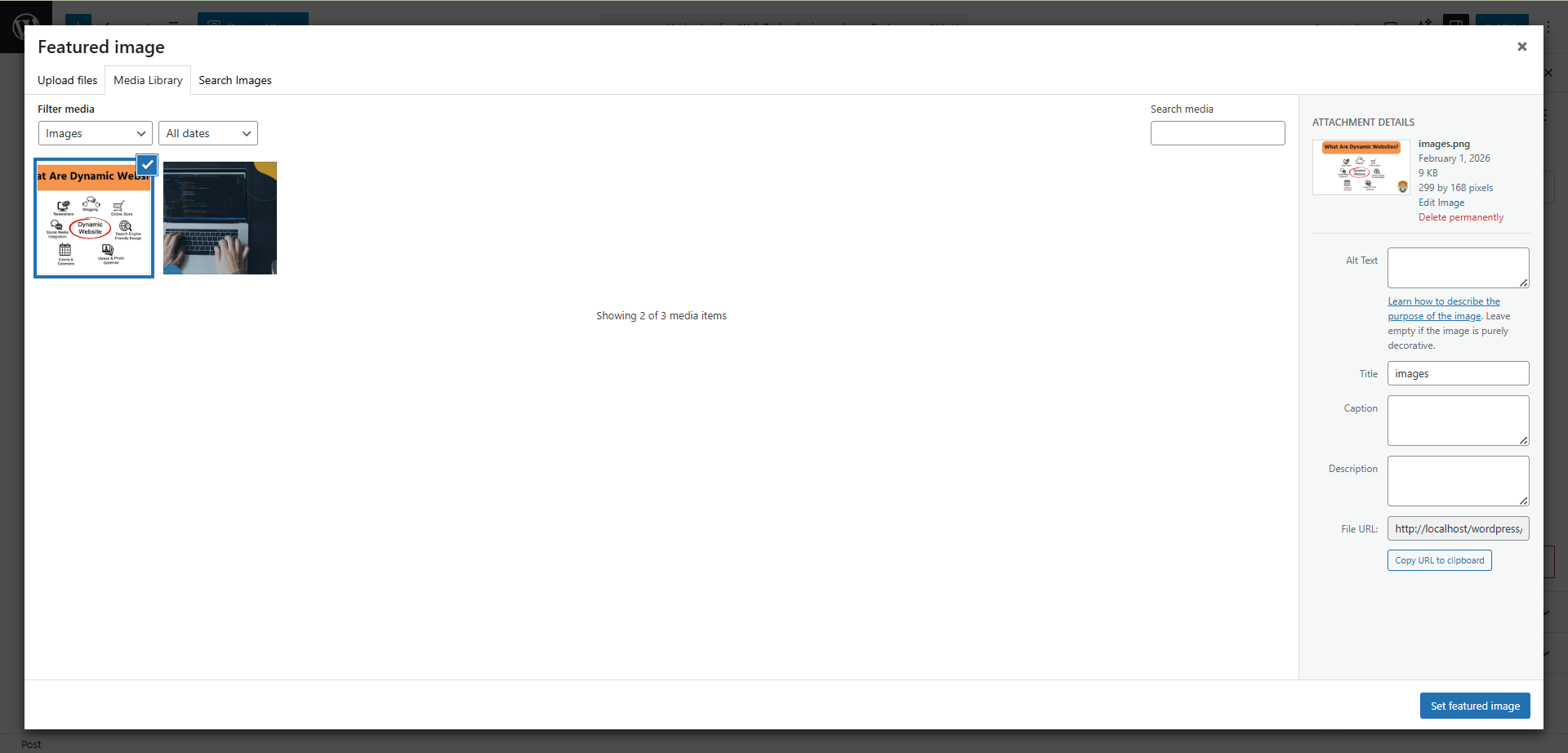


Figure 5: Featured Image Added to Blog Post (Unit 7 – Task 1)



Figure 6: Previewing Blog Post with JavaScript Dynamic Element (Unit 7 – Task 1)

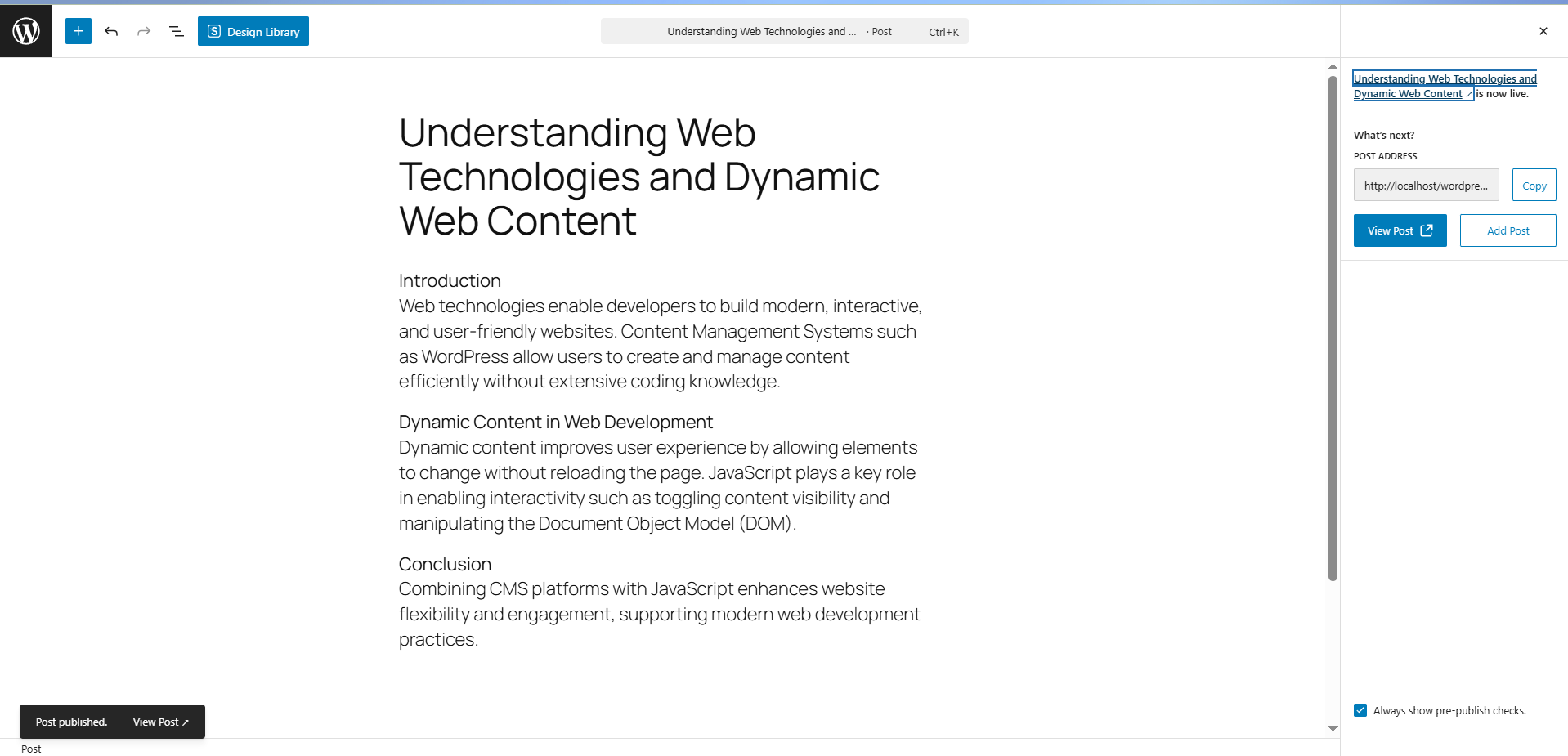


Figure 7: Blog Post Successfully Published (Unit 7 – Task 1)

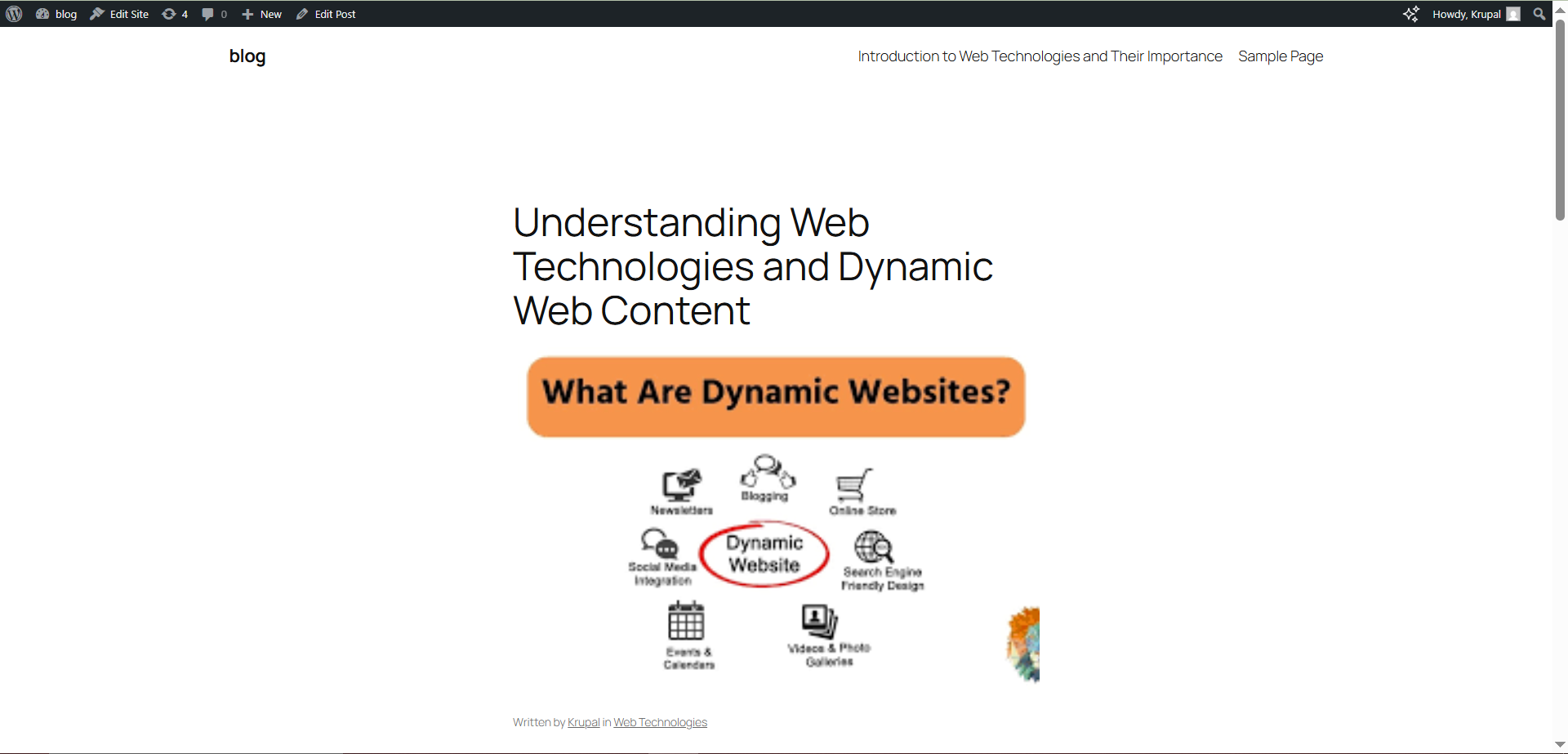
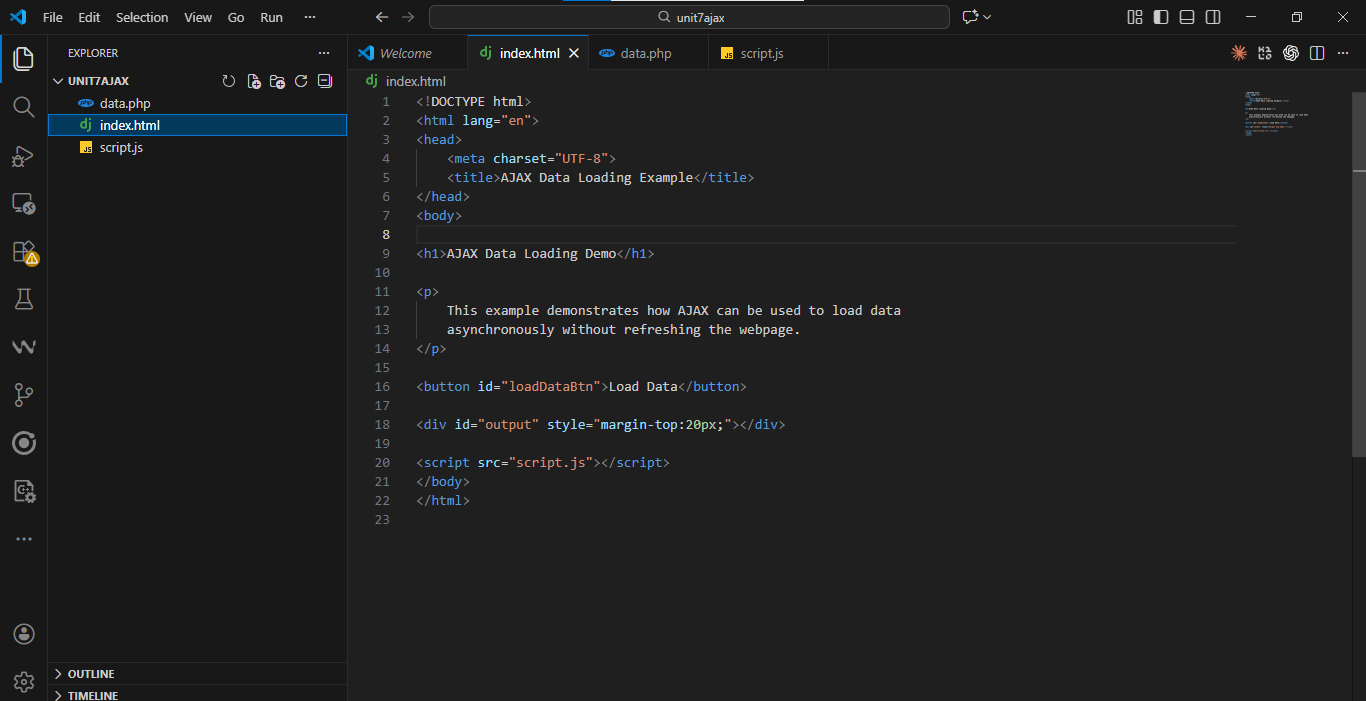
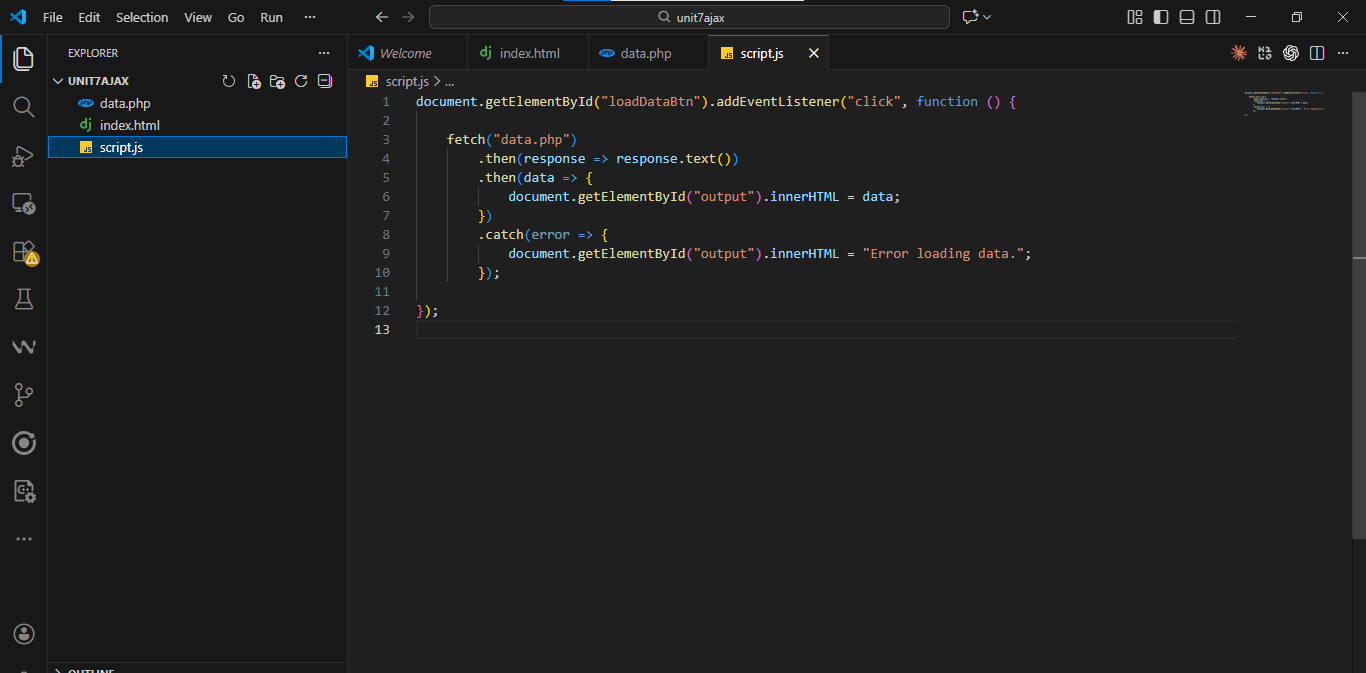


Figure 8: Live Blog Post Showing JavaScript Interactivity (Unit 7 – Task 1)

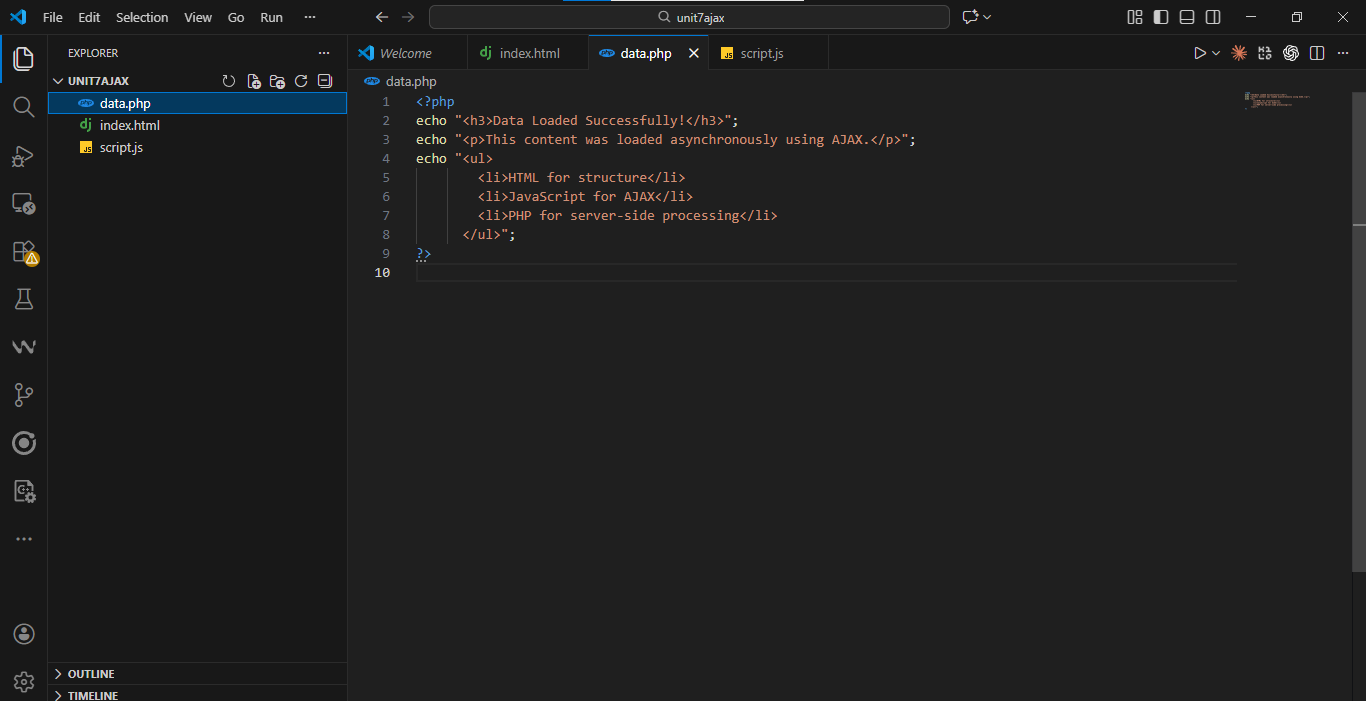
# Unit 7 Task 2

****

**Figure 1: HTML Structure for AJAX-Based Asynchronous Data Loading (Unit 7 – Task 2)**

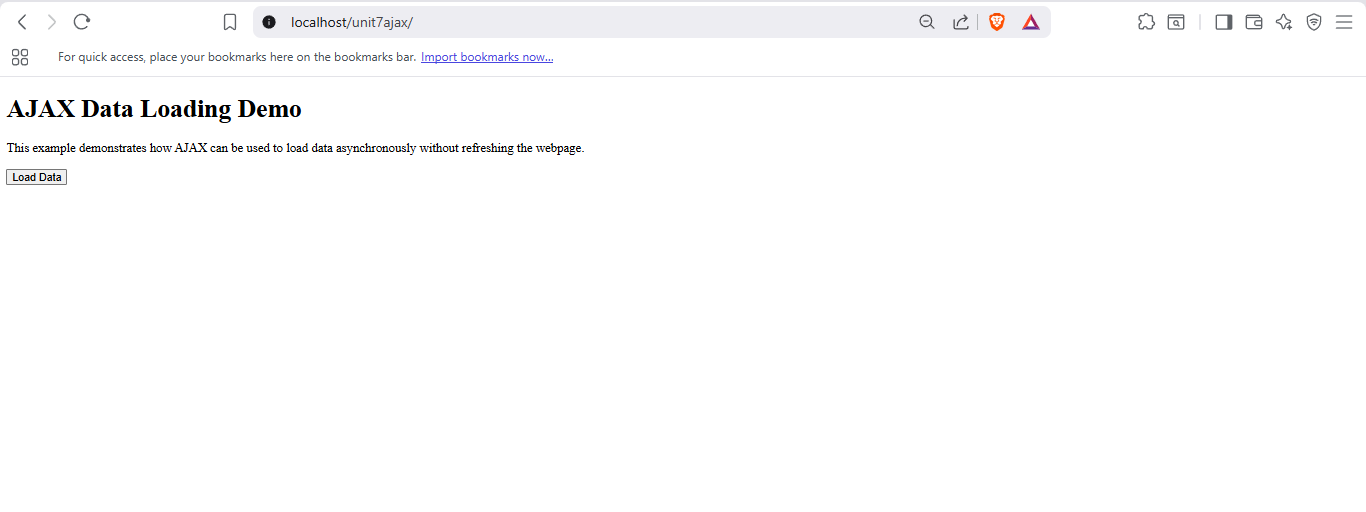
****

**Figure 2: JavaScript AJAX Code for Fetching Data Asynchronously (Unit 7 – Task 2)**

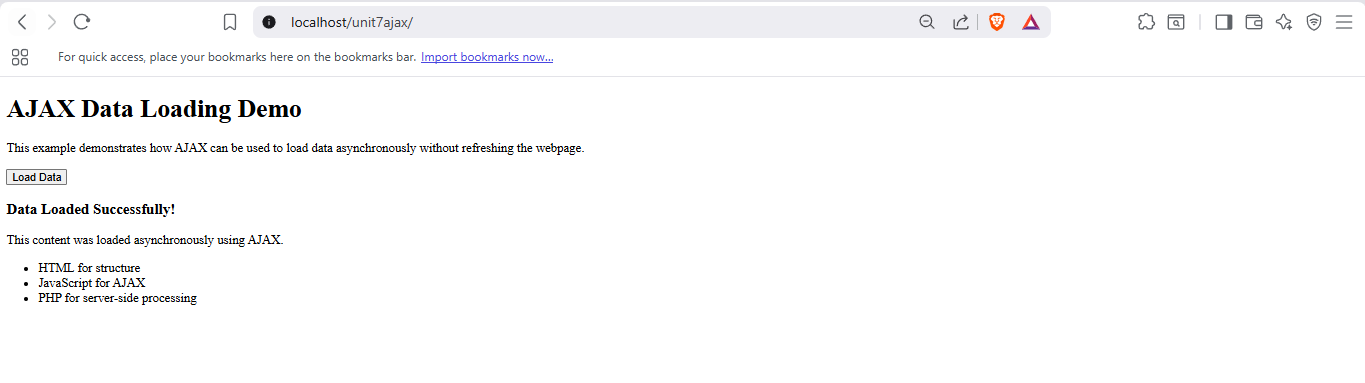
****

**Figure 3: PHP Server-Side Script Providing Data for AJAX Request (Unit 7 – Task 2)**

Output

****

**Figure 4: Initial Webpage Before AJAX Data Load (Unit 7 – Task 2)**

****

**Figure 5: Dynamically Loaded Content Displayed Using AJAX Without Page Reload (Unit 7 – Task 2)**

# Unit 8 Task 1

Code

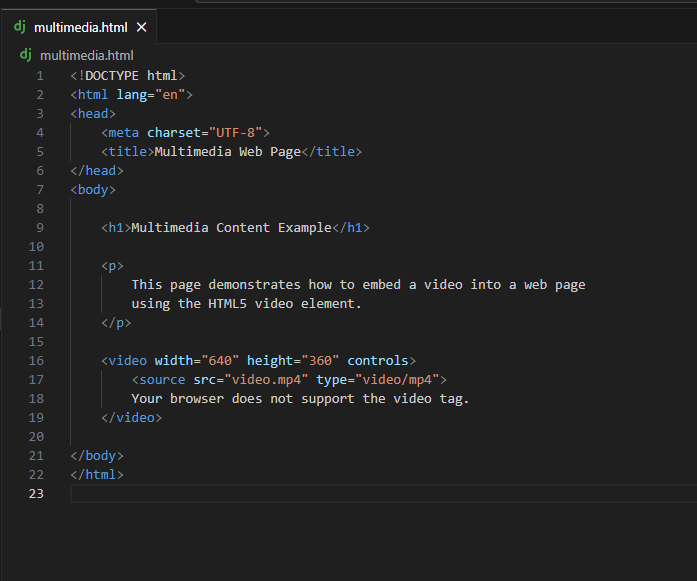


Figure 1: HTML Code for Embedding Video Using the HTML5 <video> Element (Unit 8 – Task 1)

Output

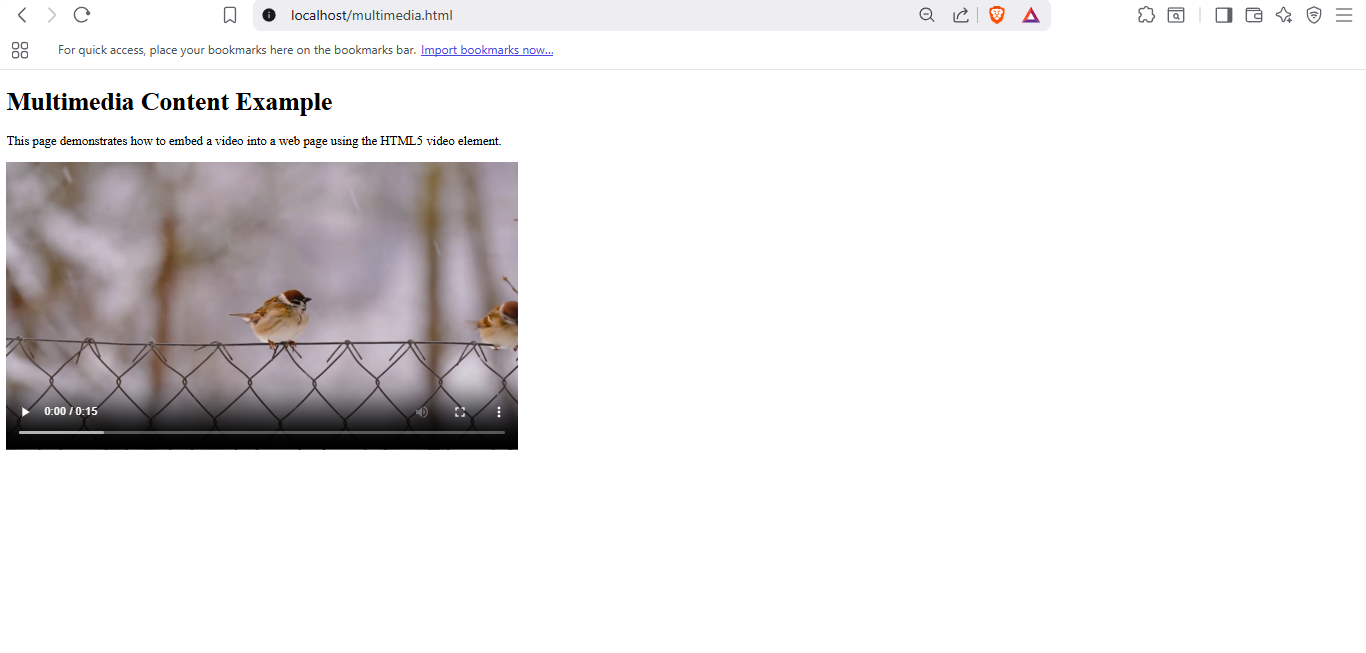


Figure 2: Embedded MP4 Video Displayed with Playback Controls in Web Browser (Unit 8 – Task 1)

# Unit 8 Task 2

Code

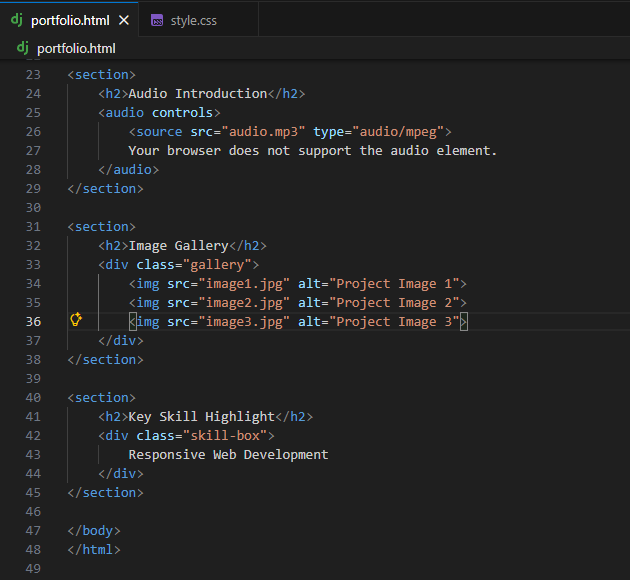


Figure 1: HTML Code for Multimedia Portfolio Webpage Structure (Unit 8 – Task 2)

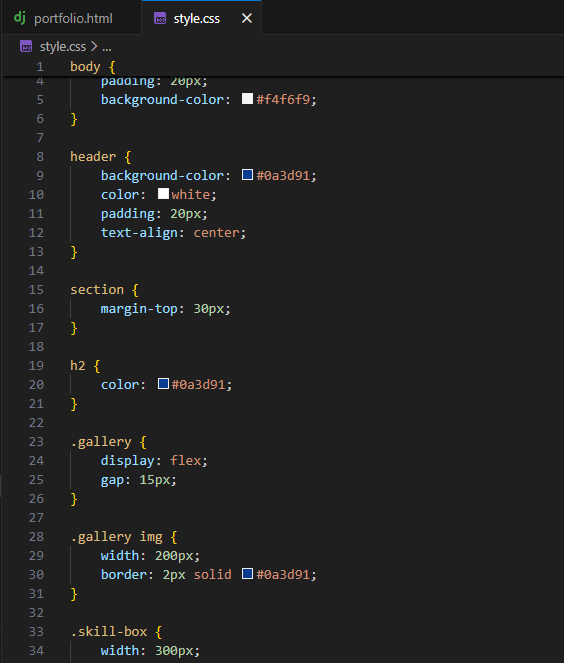


Figure 2: CSS Styling and Animation Code for Multimedia Portfolio Webpage (Unit 8 – Task 2)

Output

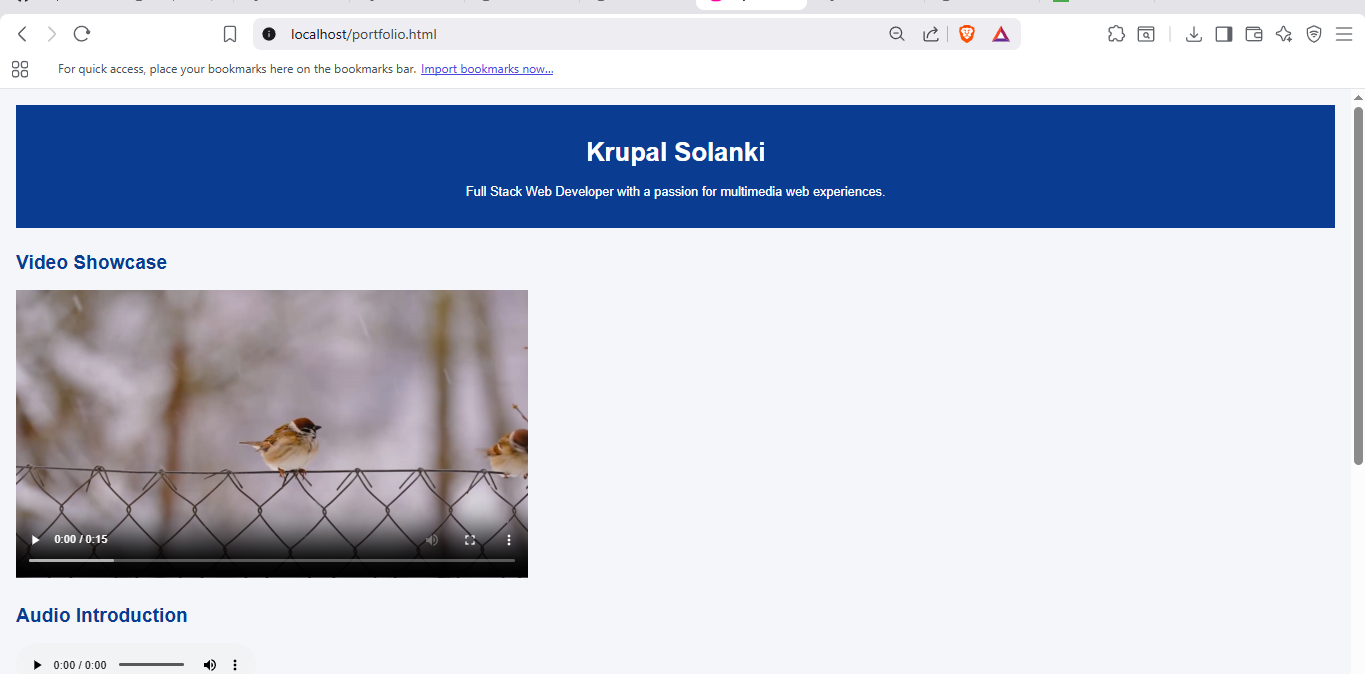


Figure 3: Multimedia Portfolio Webpage Displaying Video, Audio, and Image Gallery (Unit 8 – Task 2)

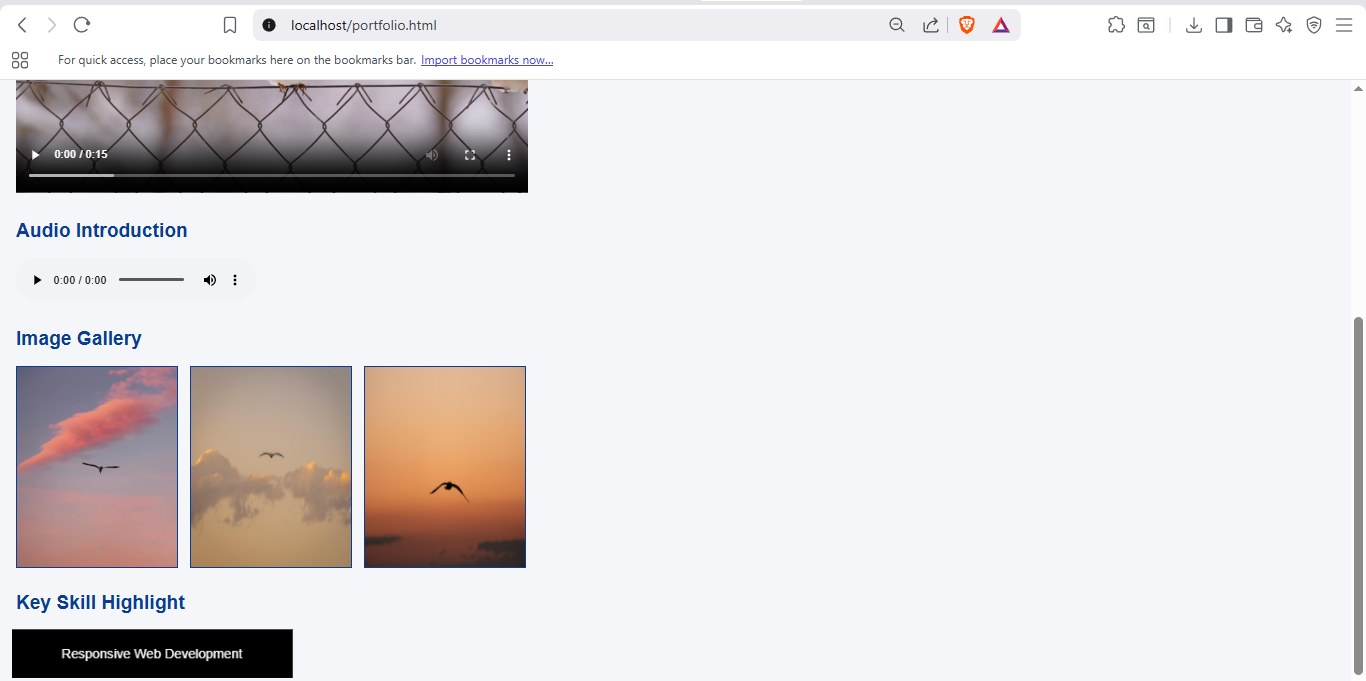


Figure 4: CSS Animation Highlighting Key Skill on Portfolio Webpage (Unit 8 – Task 2)

# Unit 9 Task 1

**Question 1: Identifying Problems**

**List five problems with the registration form**

1. The form does not specify any validation rules for input fields such as email, password, or age.
2. There is no password confirmation field to prevent typing errors.
3. The password field has no strength requirements (length, complexity).
4. There is no privacy policy or consent checkbox for data usage.
5. Sensitive information (passwords) appears to be stored in plain text, which is insecure.

**Question 2: Password Security**

**a) Why is the password system weak in this form?**

The system of passwords is also a weak feature as the user can make a simple password with no rules or restrictions. Also, the form implies that the passwords stored are retrieved as typed, implying that the passwords are not encrypted/hashed.

**b) Suggest two rules to make passwords more secure**

1. Passwords should be at least 8 characters long and include uppercase letters, lowercase letters, numbers, and symbols.
2. Passwords should be hashed and salted before being stored in the system.

**Question 3: Validation**

**a) What is meant by validation?**

Validation refers to the process of verifying user input to verify that the data that is being inputted is correct, complete and that it is taken according to established rules before acceptance or processing.

**b) Give two examples of validation for this form**

1. Email validation to ensure the email address follows a valid format.
2. Age validation to ensure the user enters a numeric value within an acceptable range.

**Question 4: Privacy and Consent**

**a) What important privacy feature is missing from this form?**

A privacy policy agreement or user consent checkbox is missing.

**b) Why is this feature important?**

It is a significant aspect as it will inform the users of the way their data will be utilized and makes sure that the data protection laws like GDPR are followed.

**Question 5: Plain-Text Password Storage**

**a) What is plain-text password storage?**

Plain-text password storage refers to the storage of user passwords in their original form, i.e., as typed.

**b) Explain one risk of storing passwords this way**

In case of a security breach in the system, attackers can easily access and intercept all passwords of users causing identity theft and account breach.

**Question 6: Improving the Form**

**Suggest three improvements**

1. Add client-side and server-side validation for all fields.
2. Implement password strength rules and confirmation fields.
3. Include privacy consent, secure password storage, and clear error messages.

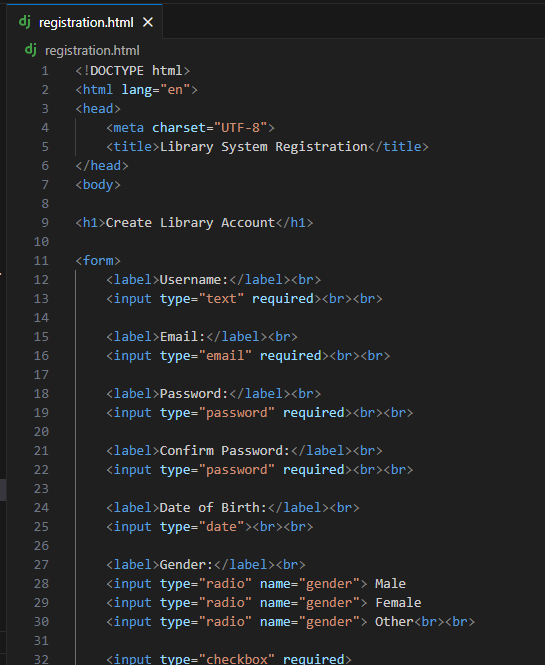


Figure 1: HTML Mock-Up of Secure User Registration Form for Fictional Library System (Unit 9 – Task 1)



Figure 1: HTML Mock-Up of Secure User Registration Form for Fictional Library System (Unit 9 – Task 1)

# Unit 9 Task 2

Admin Privileges in a User Management System

A user management system will usually grant the administrator the ability to create, modify, suspend, or delete user accounts, change passwords, assign user roles, and access permissions. The administrators can also track the user activity, filter the content and set the system settings. These privileges are valuable as they guarantee the security of the functioning of the system, users comply with the correct rules, and confidential information is secured. Decent administration control measures against abuse, allows one to reclaim accounts and makes the system generally more trustworthy and dependable.

# Unit 10 Task 1

Code

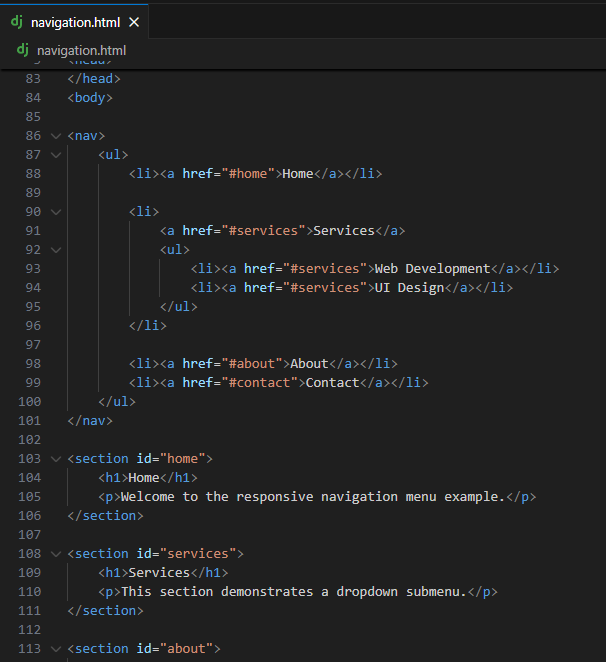


Figure 1: HTML, CSS, and JavaScript Code for Responsive Navigation Menu with Dropdown and Smooth Scrolling (Unit 10 – Task 1)

Output

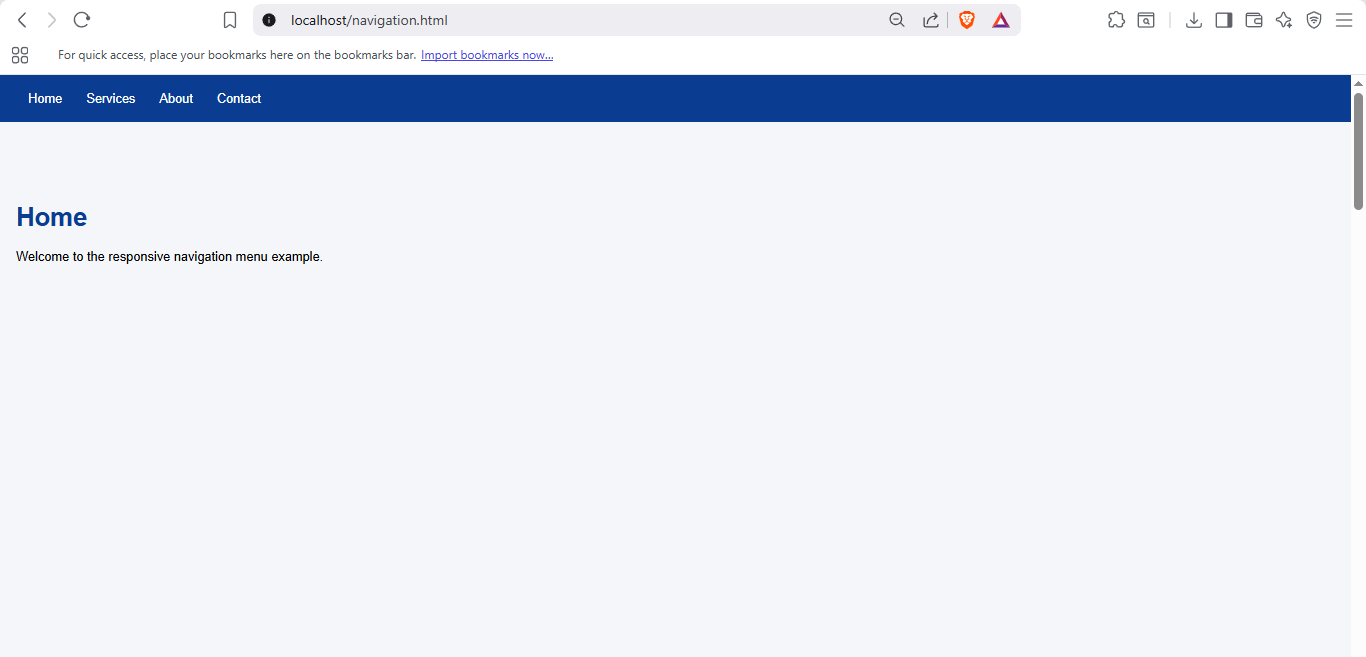


Figure 2: Responsive Navigation Menu with Dropdown Submenu and Smooth Scrolling Displayed in Browser (Unit 10 – Task 1)

# Unit 10 Task 2

1. Two Key Security Measures to Protect a Website

Input validation and sanitisation is another important security technique since it helps in mitigating the attacks like SQL injection and cross-site scripting (XSS). The execution and storage of malicious code inside the system is blocked by not allowing malicious code to be run or kept by authenticating the user input and escaping special characters. The other significant control is the application of secure authentication and access control such as use of good passwords and access control roles. This guarantees that sensitive components of a web site can not be accessed by unauthorized users and minimize the chances of unauthorised access of data, or loss of accounts.

1. Importance of Keeping Website Software Up to Date

It is vital to update the software of the websites including the CMS, themes and other plug-ins to ensure the security and performance. Patches are usually provided as updates to the vulnerabilities which are known to be prone to exploitation by any attacker. The old software poses a threat of malware attacks, data attacks, and defacing of websites. Frequent updates also enhance compatibility, functionality and stability so that the website remains in business running and operating in the ever changing threat environment.

1. Security Plugin for WordPress

Wordfence Security is one of the effective security plugins that can be used in wordpress. Wordfence offers the following features: firewall protection, malware scanning, login security and real-time threat detection. It assists in securing WordPress websites by filtering out bad traffic, recognizing vulnerabilities as well as warning administrators on possible attacks. Wordfence ensures the improvement of the overall security and reliability of a WordPress site by actively checking and protecting against typical attacks.