

Web Development – Project Descriptions

V5 – Section

Course Instructor: Ammar Haider

(CLO-2, CLO-3, CLO-4)

Instruction:

- Submit your project in *.zip folder*
 - Include all project files, such as *`index.html`, `style.css`, and `app.js`*.
 - Ensure your code is well-structured, commented, and easy to understand.
 - This project is in grouped assignment. Do not collaborate with others or copy their work.
 - All work must be your own. *Plagiarism will result in F Grade.*
 - The deadline for submitting your project is *June – 11- 2024 on Google Classroom.*
 - Late submissions are not allowed and marked *zero*, so submit your project on time.
 - *Do not cheat or plagiarize.* Academic dishonesty will not be tolerated.
 - If you have any questions about what constitutes cheating or plagiarism, ask your instructor for clarification.
 - If you have any questions or need support with the project, *visit my office CB1-708 or email: ammar.haider@umt.edu.pk*
-

1. Blog Website Project Description

1. Objective:

The goal of this project is to create a simple blog website that allows users read the blog posts. This project will help students understand the basics of web development, including HTML, CSS, JavaScript and php.

2. Project Overview:

The blog website will have several core features that allow users to interact with blog content seamlessly. The main functionalities will include a homepage to display blog posts, a form to create new posts (user fill the form and developers will add the blog on website later), the ability to view detailed blog posts, and options to like and share blog post. Highlights of popular blogs or writers to attract visitors' attention.

Features:

3. Homepage:

- A welcoming landing page that lists all blog posts.
- Each post will display a title, a brief summary, the author's name, and a publication date.
- A navigation bar to access the homepage and other sections like creating new posts.

4. Create Post Page:

- A user-friendly form for creating new blog posts.
- Input fields for the post title, content, author name, and optional tags.
- Basic validation to ensure all required fields are filled out before submission.

5. Read Post Page:

- A detailed view of a blog post, displaying the full content along with the title, author, and publication date.

- A link or button to navigate back to the homepage for easy browsing.

6. Like and Share Post Page:

- Allows users to like and share the post.

7. Delete Post Functionality:

- An option to delete a blog post from the system via form filling option.
- if any user wants to delete their blog, they will fill the form and share the request with website owners.
- A confirmation prompt to prevent accidental deletions.

8. Styling and Responsiveness:

- CSS will be used to create a clean, modern, and responsive design.
- Ensures the website looks good and functions well on various devices, including desktops, tablets, and smartphones.

9. Website Pages:

Add these pages in your website

- About us
- Contact Us Form
- Privacy Policy

10. Learning Outcomes:

- By completing this project, students will gain practical experience in:
- Structuring HTML for content organization.
- Styling web pages using CSS for aesthetics and responsiveness.
- Adding interactivity with JavaScript for dynamic content management.

2. Restaurant Website Project Description

1. Objective:

The goal of this project is to create a simple and attractive restaurant website. This project will help students understand the basics of web development, including HTML, CSS, and JavaScript, by building a functional website that showcases a restaurant's offerings, location, and contact information.

2. Project Overview:

The restaurant website will have several essential sections that provide visitors with information about the restaurant, its menu, and ways to contact or visit the establishment. The main functionalities will include a homepage, a menu page, an about page, and a contact page.

Features:

3. Homepage:

- A welcoming landing page with an eye-catching banner image and a brief introduction to the restaurant.

- Navigation links to other sections of the website, such as the menu, about, and contact pages.
- Highlights of popular dishes or special offers to attract visitors' attention.

4. Menu Page:

- A detailed menu displaying different categories of food and beverages offered by the restaurant.
- Each menu item should include a name, description, and price.
- Use CSS to style the menu for a visually appealing and organized layout.

5. About Page:

- A section providing background information about the restaurant, including its history, mission, and values.
- Information about the chef or key staff members to add a personal touch.
- Images of the restaurant's interior or key events to create a connection with visitors.

6. Contact Page:

- Contact information including the restaurant's address, phone number, and email address.
- A simple contact form allowing visitors to send messages or make reservations online.
- A Map shows the restaurant's location and help visitors find directions.

7. Styling and Responsiveness:

- Use CSS to create a clean, modern, and responsive design that works well on various devices, including desktops, tablets, and smartphones.
- Ensure the website looks professional and provides a pleasant user experience.

8. Detailed Steps:

Step 1: Setup the Project Structure

Create a project directory and add the following files:

- ``index.html`` (for the homepage)
- ``menu.html`` (for the menu page)
- ``about.html`` (for the about page)
- ``contact.html`` (for the contact page)
- ``style.css`` (for styling)
- ``app.js`` (for handling any JavaScript logic)

Step 2: Create the Homepage

HTML (``index.html``):

- Include a banner image, a brief introduction to the restaurant, and navigation links to other pages.

CSS (``style.css``):

- Style the homepage with an attractive layout, using colors and fonts that match the restaurant's theme.

Step 3: Create the Menu Page

HTML (``menu.html``):

- Organize the menu items into categories such as appetizers, main courses, desserts, and beverages.
- Include names, descriptions, and prices for each item.

CSS (`style.css`):

- Use CSS to create a visually appealing menu layout.

Step 4: Create the About Page

HTML (`about.html`):

- Provide a brief history of the restaurant, its mission, and values.
- Include images and information about key staff members.

CSS (`style.css`):

- Style the about page with appropriate typography and layout to make it engaging and informative.

Step 5: Create the Contact Page

HTML (`contact.html`):

- Include the restaurant's contact information and a simple contact form and Reservation Form (Name, Email, Contact, Persons, Date and Time, Payment Method etc).
- Map to show the restaurant's location.

CSS (`style.css`):

- Style the contact page and form for ease of use and visual appeal.

Step 6: Add Interactivity with JavaScript

JavaScript (`app.js`):

- Enhance the website with JavaScript, such as form validation or interactive elements on the homepage.

9. Learning Outcomes:

- By completing this project, students will gain practical experience in:
- Structuring HTML for different sections of a website.
- Styling web pages using CSS for aesthetics and responsiveness.
- Adding basic interactivity with JavaScript for enhanced user experience.

3. E-commerce Website Project Description

1. Objective:

The goal of this project is to create a simple e-commerce website that lists products for sale. This project will help to understand the basics of web development, including HTML, CSS, and JavaScript, by building a functional website that showcases products without implementing a checkout process.

2. Project Overview:

The e-commerce website will have several essential sections that provide visitors with information about available products, including product images, descriptions, and prices. The main functionalities will include a homepage, a product listing page, and a contact page.

Features:

3. Homepage:

- A welcoming landing page with a featured products section to attract visitors' attention.
- Navigation links to other sections of the website, such as the product listing and contact pages.
- Highlights of popular or new products to engage visitors.

4. Product Listing Page:

- A detailed listing of products available for sale.
- Each product should include an image, name, description, and price.
- Use CSS to style the product listings for a visually appealing and organized layout.

5. Contact Page:

- Contact information including the business's address, phone number, and email address.
- A simple contact form allowing visitors to send messages or inquiries.

6. Styling and Responsiveness:

- Use CSS to create a clean, modern, and responsive design that works well on various devices, including desktops, tablets, and smartphones.
- Ensure the website looks professional and provides a pleasant user experience.

7. Detailed Steps:

Step 1: Setup the Project Structure

- Create a project directory and add the following files:
- `index.html` (for the homepage)
- `products.html` (for the product listing page)
- `contact.html` (for the contact page)
- `style.css` (for styling)
- `app.js` (for handling any JavaScript logic)

Step 2: Create the Homepage

HTML (`index.html`):

- Include a banner image, a brief introduction to the business, and navigation links to other pages.
- Highlight featured or popular products.

CSS (`style.css`):

- Style the homepage with an attractive layout, using colors and fonts that match the business's theme.

Step 3: Create the Product Listing Page

HTML (`products.html`):

- Organize the products into categories or display them in a grid format.
- Include images, names, descriptions, and prices for each product.

CSS (`style.css`):

- Use CSS to create a visually appealing product listing layout.

Step 4: Create the Contact Page

HTML (`contact.html`):

- Include the business's contact information and a simple contact form.

CSS (`style.css`):

- Style the contact page and form for ease of use and visual appeal.

Step 5: Add Interactivity with JavaScript**JavaScript (`app.js`):**

- Enhance the website with JavaScript, such as dynamic updates to the product listings or interactive elements on the homepage.

8. Learning Outcomes:

- By completing this project, students will gain practical experience in:
- Structuring HTML for different sections of a website.
- Styling web pages using CSS for aesthetics and responsiveness.
- Adding basic interactivity with JavaScript for enhanced user experience.