REPORT ON MULTI-PAGE RESPONSIVE WEBSITE

RESTAURANT MENU

HTML Structure:

1. Document Type Declaration (DOCTYPE):

- The document starts with `<!DOCTYPE html>`, indicating that it is an HTML5 document.

2. HTML Tag:

- The `<html>` tag encloses the entire HTML document.

- The `lang` attribute is set to "en" for English.

3. Head Section:

- Contains metadata about the document.

- Title: "Full website Design"

- External CSS files are linked, including a style.css file and the Boxicons library.

- Google Fonts API is used to import the "Poppins" font.

4. Body Section:

- The main content of the webpage is enclosed within the `<body>` tags.

5. Header Section:

- Contains a logo and a navigation bar.

- Navigation links include Home, About, Menu, Service, and Contact.

6. Home Section:

- Banner section with a heading, subheading, and a button.
- Includes an image.

7. About Section:

- Contains an image and a text section introducing the website.
- Includes a "Learn more" button.

8. Menu Section:

- Displays a heading and a list of food items with images, names, descriptions, prices, and a cart icon.

9. Services Section:

- Lists three services with images, titles, and descriptions.

10. Call to Action Section:

- Encourages users to engage with a message and a button.

11. Footer Section:

- Divided into columns with menu links, services, information, and contact details.
- Includes social media icons.

12. JavaScript:

- The script.js file is linked at the end of the document for JavaScript functionality.

Improvements:

- The `id` attribute for the menu icon div has a typo. It should be `id="menu-icon"` instead of `id=""menu-icon""`.

- Semantic HTML elements like `<nav>`, `<main>`, `<section>`, and `<article>` can be used for better structure.

- Consider adding alt attributes to the `<img>` tags for accessibility.

- It's advisable to add responsive design elements for better display on different devices.

Overall, the HTML code provides a foundation for a restaurant website. Depending on your needs, you may want to enhance it further with additional styling and functionality.

CSS STRUCTURE:

1. Global Reset and Styling:

- The `*` selector is used for a global reset, setting padding, margin, and box-sizing to border-box for all elements.

- `scroll-behavior: smooth;` is applied to enable smooth scrolling.

2. Root Variables:

- CSS custom properties (variables) are defined using `:root` to set global colors, font sizes, and other reusable values.

3. Selection Styling:

- `::selection` is styled to change the background and text color of selected text.

4. Body Styling:

- `body` is styled with the chosen text color and background color.

5. Header Styling:

- The `header` is fixed at the top, with a background color, padding, and a flex layout.

- Logo and navigation links are styled accordingly.

6. Media Queries:

- Media queries are used to make the website responsive for different screen sizes.

- Adjustments are made for smaller screens in terms of padding, font sizes, and the appearance of the menu icon.

7. Sections Styling:

- Each section (`home`, `about`, `menu`, `services`, `cta`, `footer`) is styled with specific padding, background, and layout.

8. Home Section Styling:

- Utilizes CSS Grid for a two-column layout and responsive font sizes.

- Buttons have a hover effect.

9. About Section Styling:

- Uses CSS Grid for a two-column layout.

- Adds styling to images, headings, and paragraphs.

10. Menu Section Styling:

- Utilizes CSS Grid for a responsive grid layout.

- Each menu item is enclosed in a box with specific styling.

11. Services Section Styling:

- Displays services in a grid layout with specific padding and margins.

12. Call to Action (CTA) Section Styling:

- Styled with a specific background, padding, and text alignment.

13. Footer Styling:

- Divided into columns with specific widths.
- Social media icons have a hover effect.

14. Responsive Design:

- Media queries adjust styling for smaller screens, ensuring a responsive layout.

Suggestions:

- Consider using semantic HTML elements like `<nav>`, `<main>`, `<section>`, and `<article>` for better document structure.
- Ensure images have appropriate alt attributes for accessibility.
- Review the typo in the `:before` pseudo-element of `.col h4 ::before`. It should be a single colon, not double.
- The transition property is used for hover effects; consider adding transitions for other interactive elements for consistency.
- Test the website on various browsers and devices to ensure cross-browser compatibility.

This CSS code provides a clean and responsive styling approach for a restaurant website. It effectively uses CSS Grid and Flexbox for layout and adjusts styles based on the screen size through media queries.

JavaScript Code Analysis:

1. Menu Toggle:

- The code selects the menu icon and the navbar elements using `document.querySelector`.

- It adds a click event listener to the menu icon.

- On a click event, it toggles the "active" class on the navbar, providing a mechanism to show/hide the navigation menu.

2. Navbar Behavior on Scroll:

- It uses the `window.onscroll` event to detect when the user scrolls.

- If a scroll is detected, it removes the "active" class from the navbar, ensuring the navigation menu is hidden when the user scrolls.

Suggestions:

- The JavaScript code is concise and accomplishes its intended functionality.

- Ensure that the class toggling behavior is consistent with your CSS styles and that the "active" class is defined appropriately in your CSS.

- Consider adding comments to explain the purpose of the code or any complex logic for better code readability.

- Test the code thoroughly to ensure it works as expected in various scenarios and browsers.

- If your website includes other interactive elements or complex functionality, modularize the code or use a more organized structure for better maintainability.

Overall, the JavaScript code provided effectively handles the toggle functionality for the menu icon and the hiding of the navigation menu on scroll.