



NATIONAL UNIVERSITY
of Computer & Emerging Sciences

HTML Newspaper

Recreation - Code Explanation

Student Name: Saud Nasir

Roll Number: 22P-9227

Course: CS-4070 Web Technologies | Assignment 1

1. Code Structure Overview

The HTML document follows a standard HTML5 structure with a complete DOCTYPE declaration, head section containing metadata and title, and a body section housing all visible content. The overall architecture consists of three main structural components: a masthead area (currently in title only), a table-based main layout containing three distinct columns, and a footer section with copyright information. The document uses approximately 80 lines of code organized hierarchically with consistent 4-space indentation for readability.

2. Key Sections Explanation

Main Content Area (<main> tag)

The core newspaper layout is wrapped in a <main> tag containing a full-width table (width: 100%) with three columns in a single row. This table uses `table-layout: fixed` to ensure consistent column widths (25% - 50% - 25%) regardless of content length. The table approach provides precise control over column alignment and creates natural vertical dividers with border styling.

Left Column - Secondary News

The first table cell (<td>) contains an <article> tag featuring the Mozambique air disaster story with an accompanying image, headline, and body text. Below this is a <section> element containing local briefs displayed as an unordered list (). This column uses `vertical-align: top` to ensure content aligns at the top of the column and includes a right border (1px solid) to separate it from adjacent content.

Center Column - Lead Story

The middle column houses the main headline article about European immigration policy. It uses nested elements including a large <h2> heading (36px), a bold subheading paragraph, a centered image with caption, and two body paragraphs. This column occupies 50% of the total width and features the most prominent typography to draw reader attention to the lead story.

Right Column - Sidebar (<aside> tag)

The rightmost column uses the semantic <aside> tag to contain supplementary content: a newspaper logo image, a special notice box with distinct styling (2px border, white background), and a secondary article about a cultural event. This semantic choice clearly indicates that the content is related but tangential to the main news stories.

Footer Section

A `<footer>` element at the bottom provides copyright information with top border separation (3px solid), centered text alignment, and smaller font size (12px) to visually distinguish it from body content.

3. Design Decisions

Tables vs. Divs: I chose to use an HTML table for the three-column layout because the assignment specifically requires "at least ONE `<table>` for newspaper column layout," and tables provide inherent advantages for newspaper-style columns: automatic height matching across columns, precise width control with percentages, and natural vertical borders between cells. While modern web design typically favors CSS Grid or Flexbox, tables are semantically appropriate for tabular data presentation and were the standard for multi-column layouts in the 1980s era being recreated.

Semantic Tags: I implemented semantic HTML5 tags (`<article>`, `<section>`, `<aside>`, `<main>`, `<footer>`) to provide meaningful structure. Each news story is wrapped in `<article>` tags to indicate self-contained content, while `<section>` groups related brief items. The `<aside>` tag semantically identifies sidebar content as complementary rather than primary, which aids accessibility and SEO.

Inline Styling Strategy: Since external CSS files are forbidden, I used inline styles extensively for formatting. I applied consistent styling patterns across similar elements (e.g., all article headings have black backgrounds with white text) and used the vintage color palette (`#f4f1ea` background, `#1a1a1a` text) with Times New Roman serif font to authentically recreate 1980s newspaper aesthetics.

4. Challenges and Solutions

Challenge 1 - Column Height Consistency: The three columns contained varying amounts of content, which initially created visual imbalance. **Solution:** I used `vertical-align: top` on all table cells to align content at the top, and relied on the table's natural behavior to extend all cells to match the tallest column's height.

Challenge 2 - Visual Hierarchy Without CSS Classes: Without external stylesheets or reusable classes, maintaining consistent styling across multiple elements required repetitive inline code. **Solution:** I established a styling pattern early (black headers with white text, specific font sizes for different heading levels) and carefully replicated these patterns in each section's inline styles to maintain visual consistency.

Challenge 3 - Responsive Image Sizing: Images needed to fit within their respective columns without breaking the layout. **Solution:** I used `width: 100%` on all images to make them responsive to their container width, combined with `border: 1px solid #000` for authentic newspaper photo framing.

Total Lines of Code: ~90 lines

Validation Status: HTML5 Valid (W3C)

Browser Tested: Chrome, Firefox, Safari