BAHRIA UNIVERSITY

Department of Software Engineering

LAB JOURNAL: 04

WORKING WITH CSS - II

Name: Talha Naveed

Enrollment: 01-131232-089

Section: BSE-6A

Instructor: Engr. Subas Bilal

Repository: **Lab 04**

February 17, 2026

1 Objective

The primary goal of this lab is to transition from static layouts to responsive and flexible web design. In today's web development landscape, ensuring that applications render correctly across a multitude of devices from large desktop monitors to tablets and mobile phones is paramount.

2 Introduction

Responsive Web Design (RWD) suggests that design and development should respond to the user's behavior and environment based on screen size, platform, and orientation. This approach uses a mix of flexible grids, layouts, images, and careful use of CSS media queries.

2.1 Media Queries and Responsive Design

The '@media' rule, originally introduced in CSS2, made it possible to define different style rules for specific media types (e.g., 'screen', 'print'). However, CSS3 significantly extended this concept with **Media Queries**. Instead of just looking for a type of device, CSS3 media queries look at the **capability of the device**.

Media queries allow us to check many features, such as:

- Width and height of the viewport.
- Width and height of the device.
- Orientation (is the tablet/phone in landscape or portrait mode?).
- Resolution.

This capability makes media queries a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones.

2.2 The Viewport

The **viewport** is the user's visible area of a web page. It varies with the device; it will be smaller on a mobile phone than on a computer screen. Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.

To ensure a page is optimized for mobile, we must include the meta viewport tag in the '<head>' of our document:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This '<meta>' element gives the browser instructions on how to control the page's dimensions and scaling:

- `width=device-width` sets the width of the page to follow the screen-width of the device.
- `initial-scale=1.0` sets the initial zoom level when the page is first loaded by the browser.

3 CSS Grid Exploration

I practiced the fundamentals of CSS Grid using W3Schools tutorials, focusing on grid containers and items.

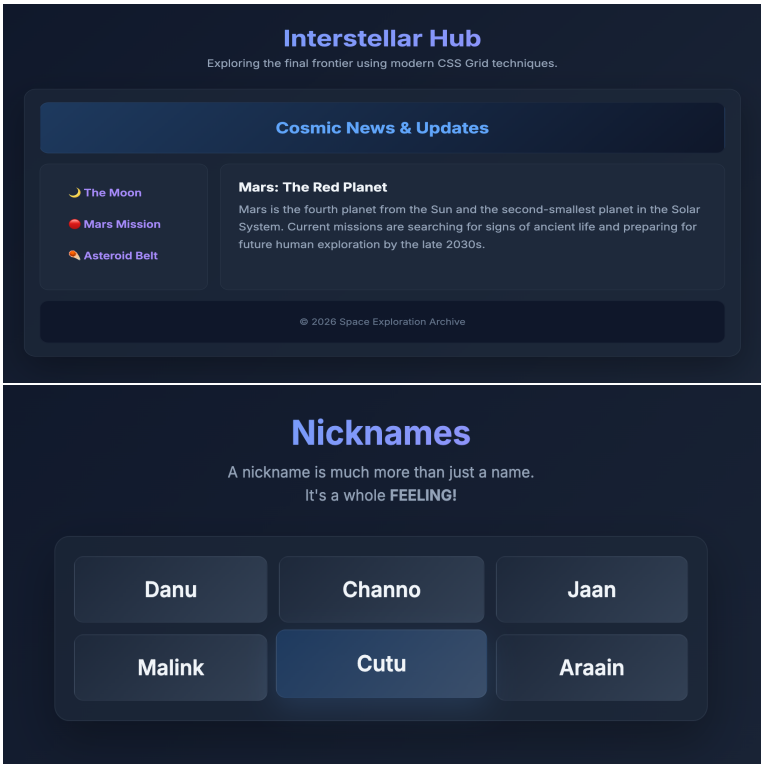

Figure 1: Grid Container and Layout Exercises

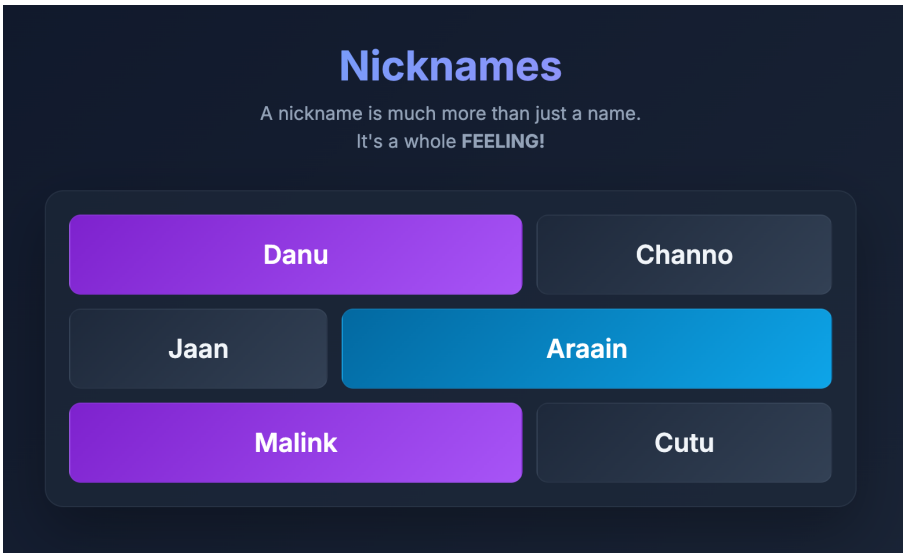

Figure 2: Advanced Grid Item Practice

4 Structural Web Page Layout

The task involved creating a structured web page using HTML semantic elements including a **Header, Article, Aside, and Footer**. The layout uses floats to position elements.

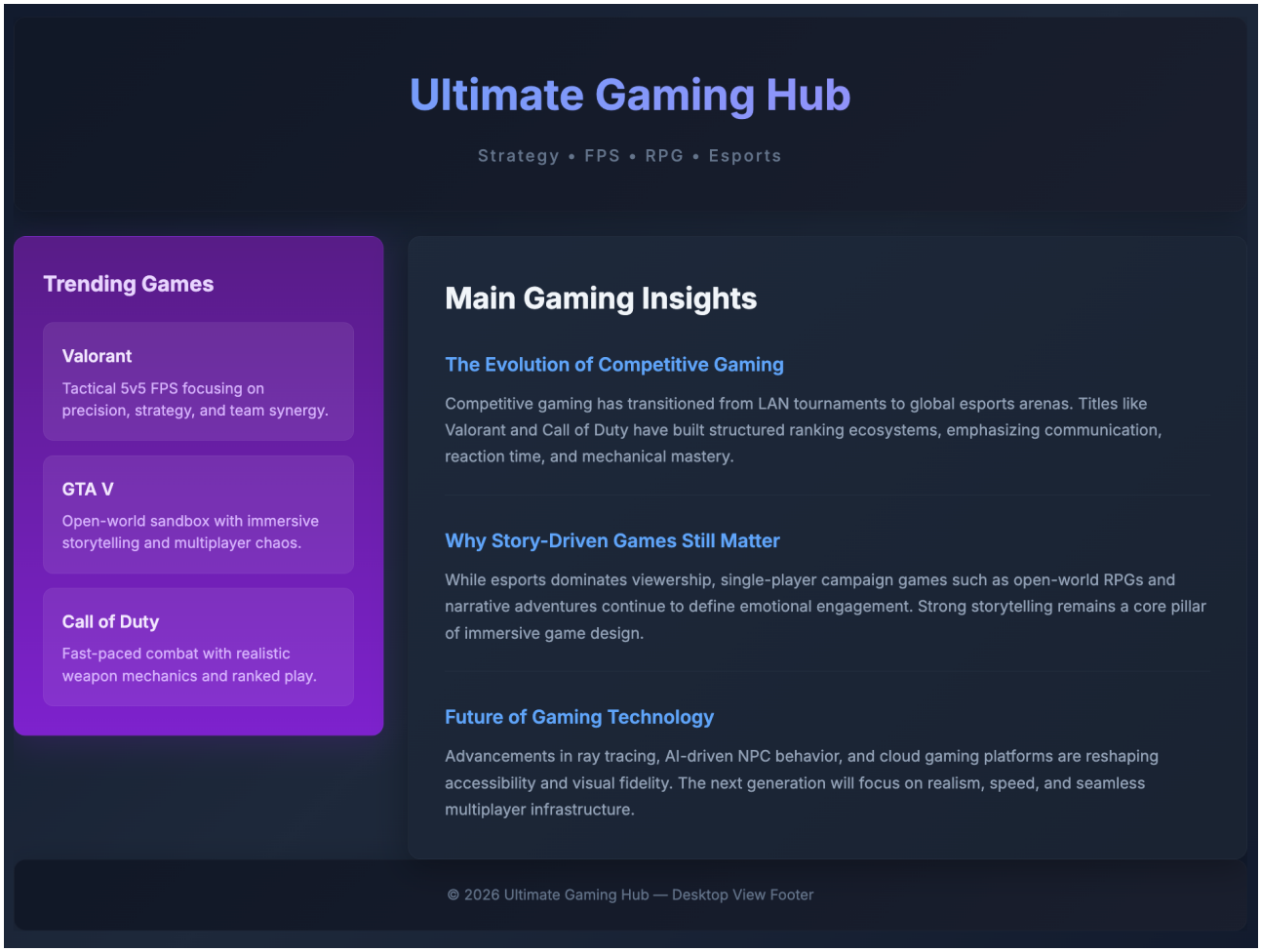

Figure 3: Initial Structural Layout with Floats

5 Responsive Implementation

In this task, I made the previous layout responsive by applying **Media Queries** with a breakpoint of **700px**.

- On screens smaller than 700px, elements stack vertically.
- The footer is hidden on smaller screens to save space.

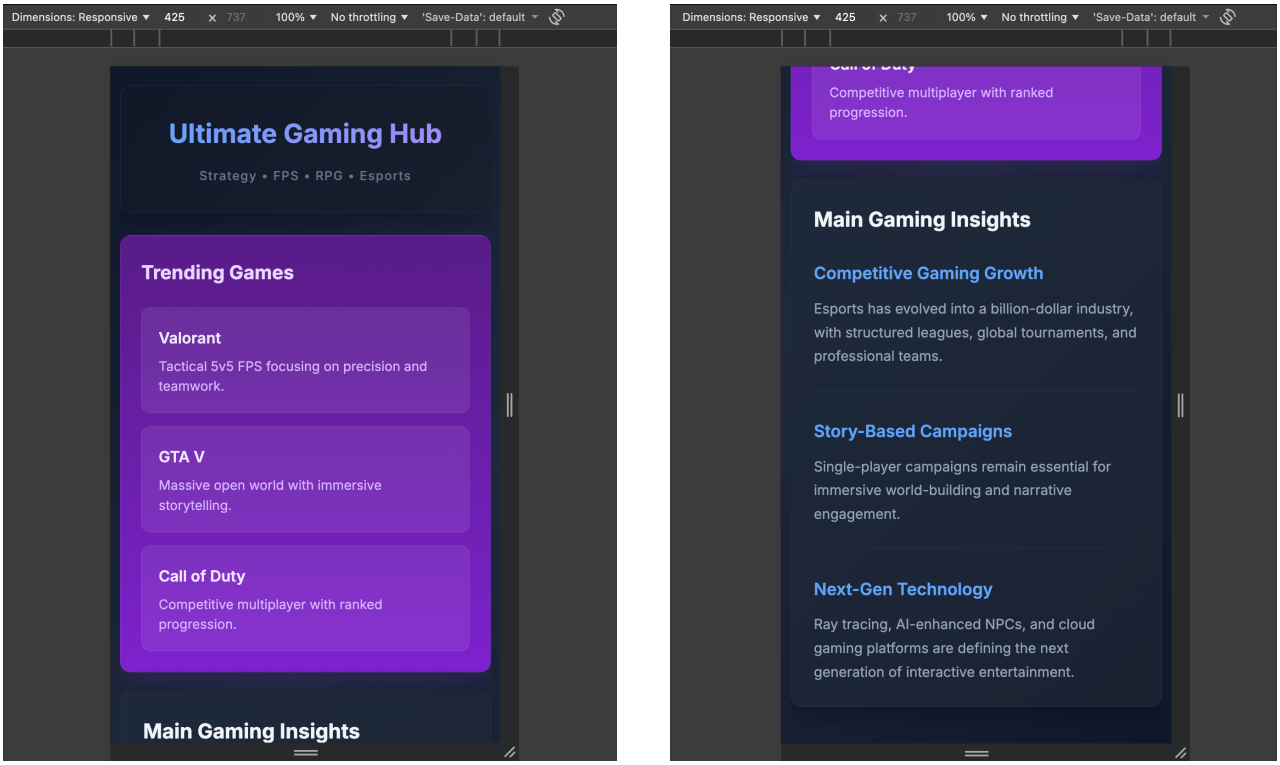

Figure 4: Task 3: Responsive View Comparison

6 Dynamic Image Scaling

The final task required adding images that **scale dynamically** based on the container width. This ensures that images never overflow their parent elements.

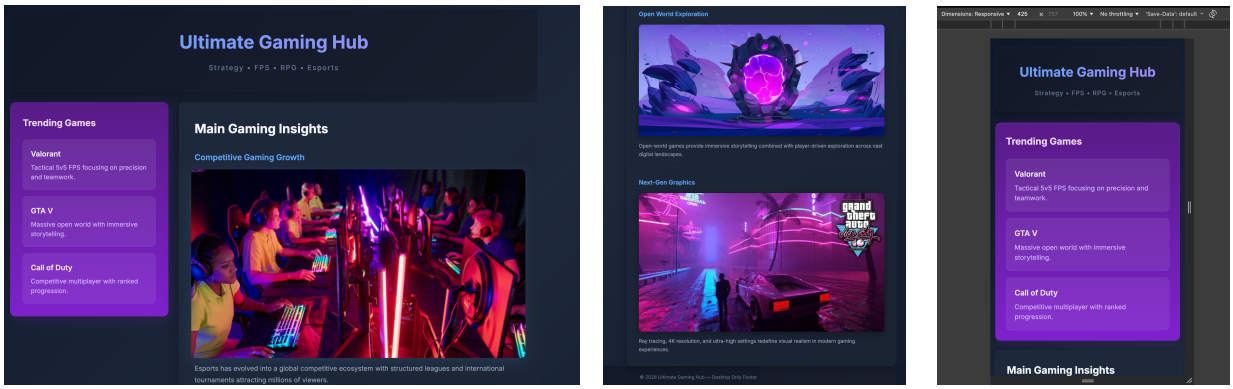

Figure 5: Desktop View

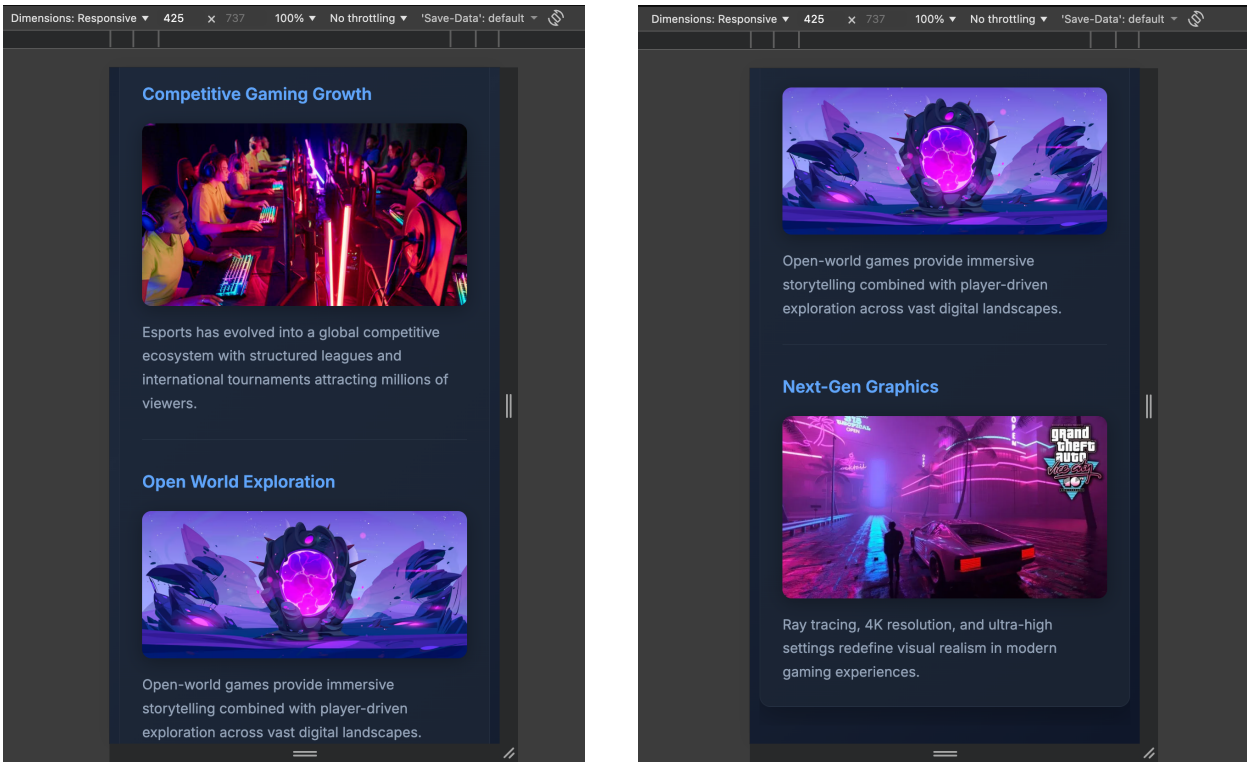

Figure 6: Mobile View

7 Conclusion

In Lab 04, we focused on the transition to **Responsive Web Design (RWD)**. By utilizing media queries and flexible image scaling, I successfully created a layout that adapts to both desktop and mobile environments.