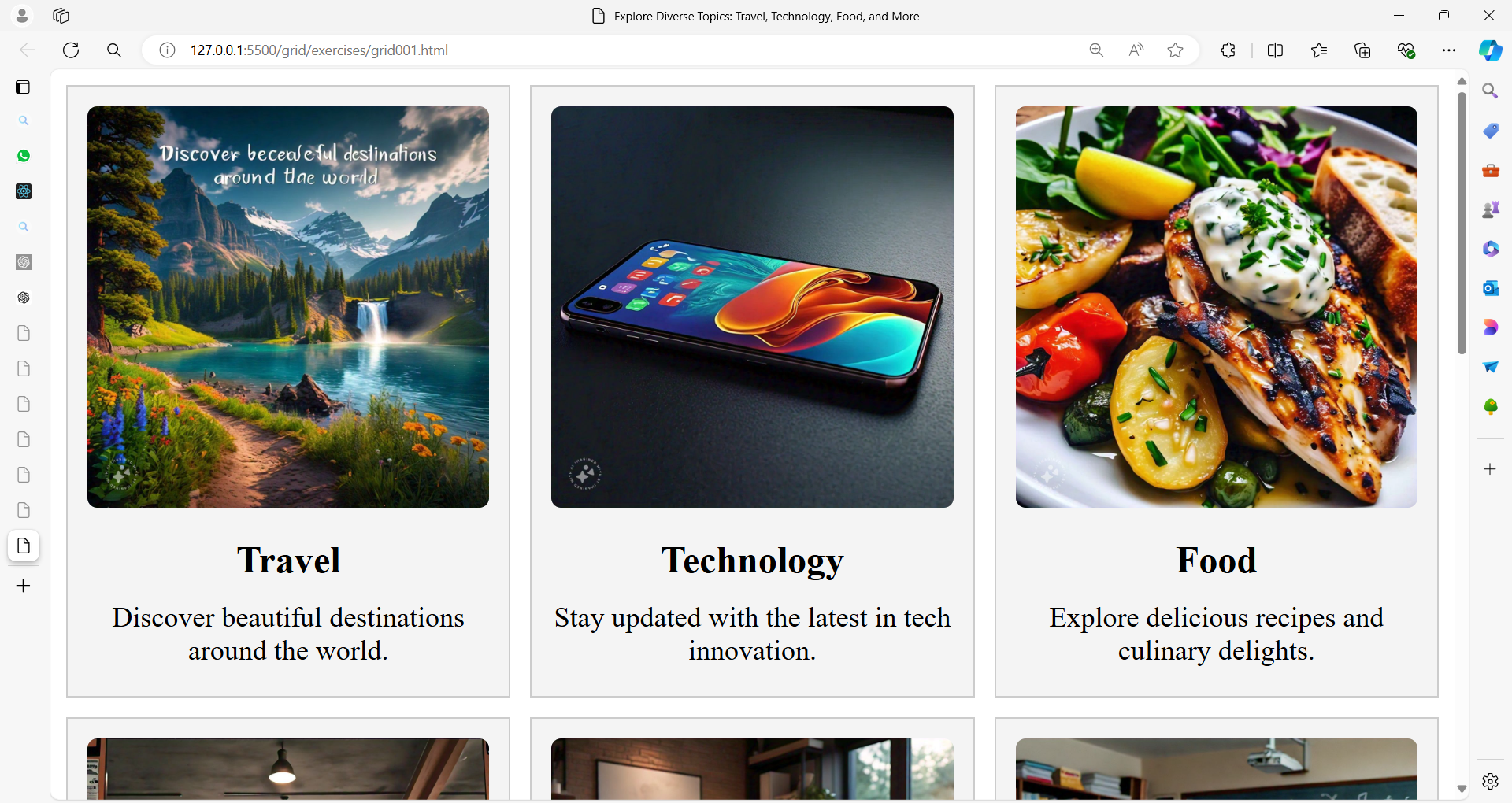
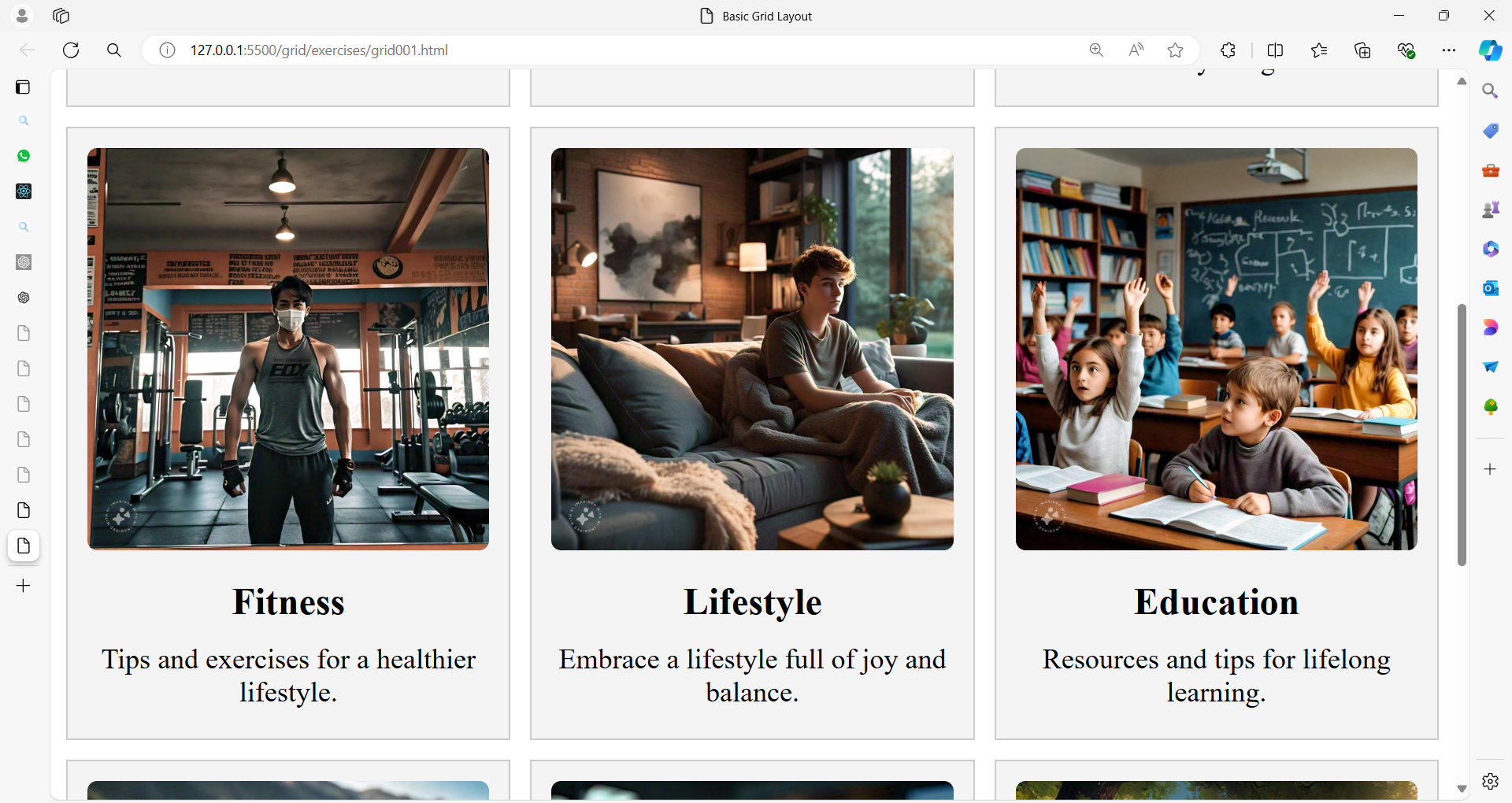
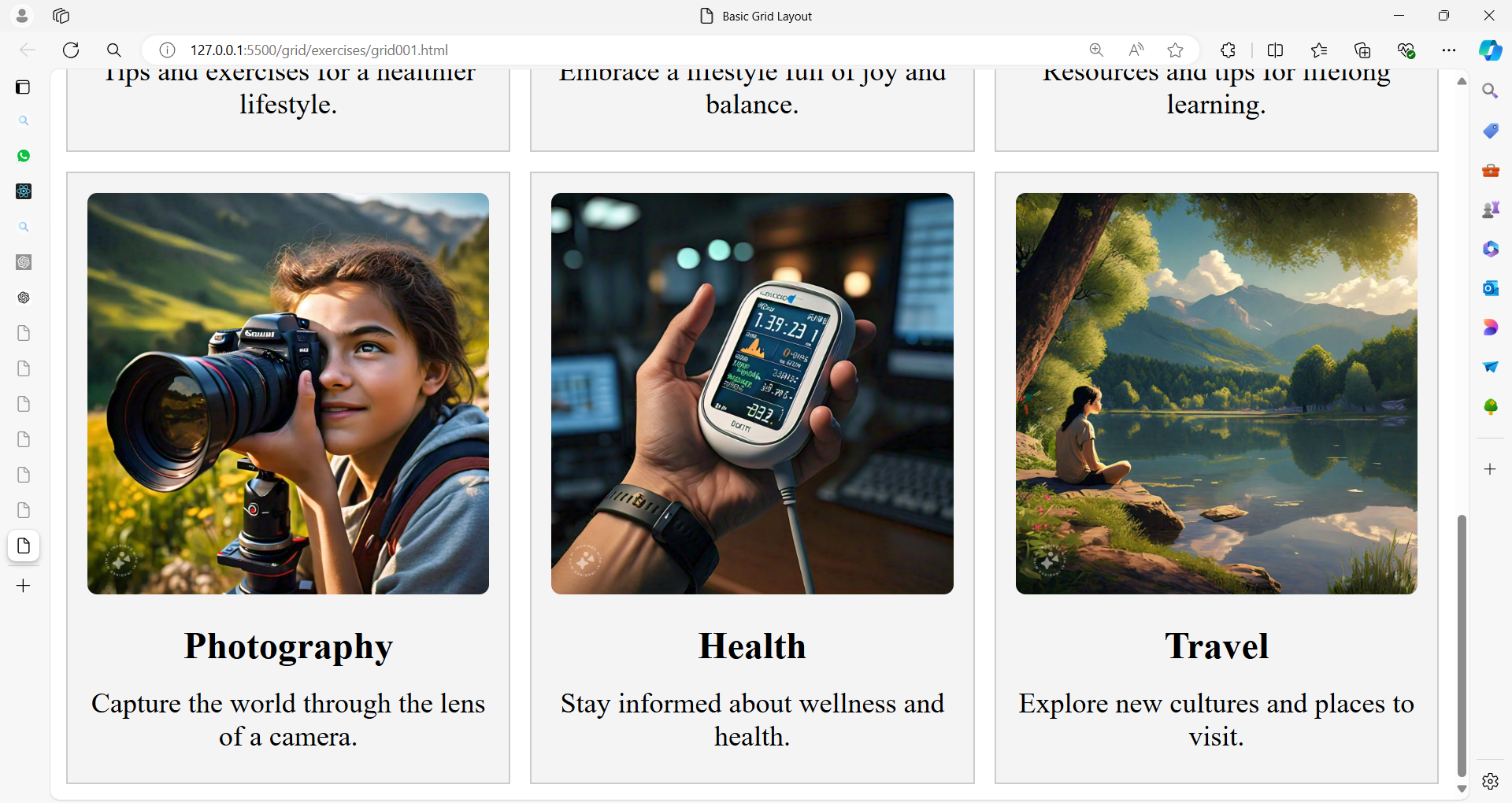
**Exercises on Grid Layout**

**Exercise 1: Basic Grid Setup**

**Exercise:** Create a simple 3x3 grid layout where each cell is evenly spaced and has a border. Each cell should contain image and text indicating its position as shown in the below pic.







**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Explore Diverse Topics: Travel, Technology, Food, and More</title>

    <style>

        .grid {

            display: grid;

            grid-template-columns: repeat(3, 1fr);

            grid-template-rows: repeat(3, auto);

            gap: 10px;

        }

        .cell {

            background-color: #f4f4f4;

            border: 1px solid #ccc;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            padding: 10px;

            box-sizing: border-box;

            text-align: center;

        }

        .cell img {

            max-width: 100%;

            height: auto;

            object-fit: contain;

            border-radius: 5px;

            margin-bottom: 10px;

        }

        .cell h2 {

            margin: 5px 0;

            font-size: 1.2em;

        }

        .cell p {

            margin: 5px 0;

            font-size: 0.9em;

        }

    </style>

</head>

<body>

    <div class="grid">

        <div class="cell">

            <img src="travel.jpg" alt="Travel" />

            <h2>Travel</h2>

            <p>Discover beautiful destinations around the world.</p>

        </div>

        <div class="cell">

            <img src="tech.jpg" alt="Technology" />

            <h2>Technology</h2>

            <p>Stay updated with the latest in tech innovation.</p>

        </div>

        <div class="cell">

            <img src="food.jpg" alt="Food" />

            <h2>Food</h2>

            <p>Explore delicious recipes and culinary delights.</p>

        </div>

        <div class="cell">

            <img src="fitness.jpg" alt="Fitness" />

            <h2>Fitness</h2>

            <p>Tips and exercises for a healthier lifestyle.</p>

        </div>

        <div class="cell">

            <img src="lifestyle.jpg" alt="Lifestyle" />

            <h2>Lifestyle</h2>

            <p>Embrace a lifestyle full of joy and balance.</p>

        </div>

        <div class="cell">

            <img src="education.jpg" alt="Education" />

            <h2>Education</h2>

            <p>Resources and tips for lifelong learning.</p>

        </div>

        <div class="cell">

            <img src="photography.jpg" alt="Photography" />

            <h2>Photography</h2>

            <p>Capture the world through the lens of a camera.</p>

        </div>

        <div class="cell">

            <img src="health.jpg" alt="Health" />

            <h2>Health</h2>

            <p>Stay informed about wellness and health.</p>

        </div>

        <div class="cell">

            <img src="travel2.jpg" alt="Travel" />

            <h2>Travel</h2>

            <p>Explore new cultures and places to visit.</p>

        </div>

    </div>

</body>

</html>

**Exercise 2: Asymmetric Grid Layout**

**Exercise:** Create a grid layout with three columns and two rows. The first row should have a full-width cell that spans all three columns, and the second row should have three equally-sized cells.



**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Asymmetric Grid Layout</title>

    <style>

        .grid {

            display: grid;

            grid-template-columns: repeat(3, 1fr);

            grid-template-rows: auto auto;

            gap: 10px;

        }

        .header {

            grid-column: span 3;

            background-color: #2196F3;

            color: white;

            padding: 20px;

            text-align: center;

            grid-row: 1 / 2; /\* Ensure the header occupies the first row \*/

        }

        .cell {

            background-color: #f4f4f4;

            border: 1px solid #ccc;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            padding: 10px;

            box-sizing: border-box;

            text-align: center;

        }

        .cell img {

            max-width: 80%;

            height: auto;

            margin-bottom: 10px;

        }

        .cell h3 {

            margin: 5px 0;

            font-size: 1.2em;

        }

        .cell p {

            margin: 5px 0;

            font-size: 0.9em;

        }

    </style>

</head>

<body>

    <div class="grid">

        <div class="header">Product Showcase</div>

        <div class="cell">

            <img src="smartphone.jpg" alt="Smartphone">

            <h3>Smartphone</h3>

            <p>Latest model with high-resolution camera and long battery life.</p>

        </div>

        <div class="cell">

            <img src="laptop.jpg" alt="Laptop">

            <h3>Laptop</h3>

            <p>Powerful laptop with a fast processor and large storage capacity.</p>

        </div>

        <div class="cell">

            <img src="headphones.jpg" alt="Headphones">

            <h3>Headphones</h3>

            <p>Noise-cancelling headphones with superior sound quality.</p>

        </div>

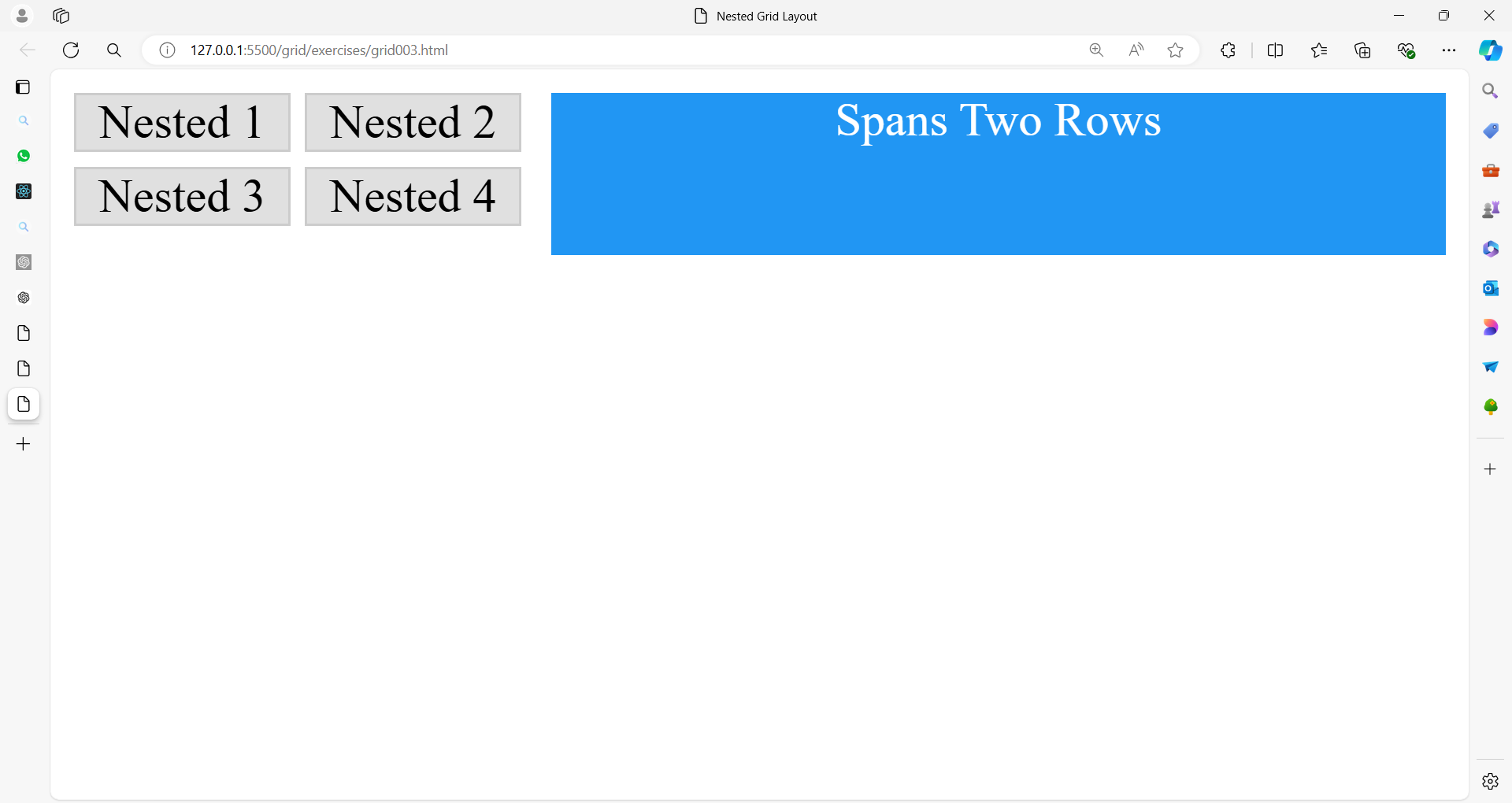
    </div>

</body>

</html>

**Exercise 3: Nested Grid Layout**

**Exercise:** Create a grid layout where the main grid has two columns. The first column contains a nested grid with two rows and two columns. The second column contains a single cell that spans both rows of the nested grid.



**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Nested Grid Layout</title>

<style>

.main-grid {

display: grid;

grid-template-columns: 1fr 2fr;

gap: 10px;

}

.nested-grid {

display: grid;

grid-template-columns: 1fr 1fr;

grid-template-rows: 1fr 1fr;

gap: 5px;

}

.cell, .nested-cell {

background-color: #e0e0e0;

border: 1px solid #ccc;

display: flex;

align-items: center;

justify-content: center;

}

.main-cell {

grid-row: span 2;

background-color: #2196F3;

color: white;

text-align: center;

}

</style>

</head>

<body>

<div class="main-grid">

<div class="nested-grid">

<div class="nested-cell">Nested 1</div>

<div class="nested-cell">Nested 2</div>

<div class="nested-cell">Nested 3</div>

<div class="nested-cell">Nested 4</div>

</div>

<div class="main-cell">Spans Two Rows</div>

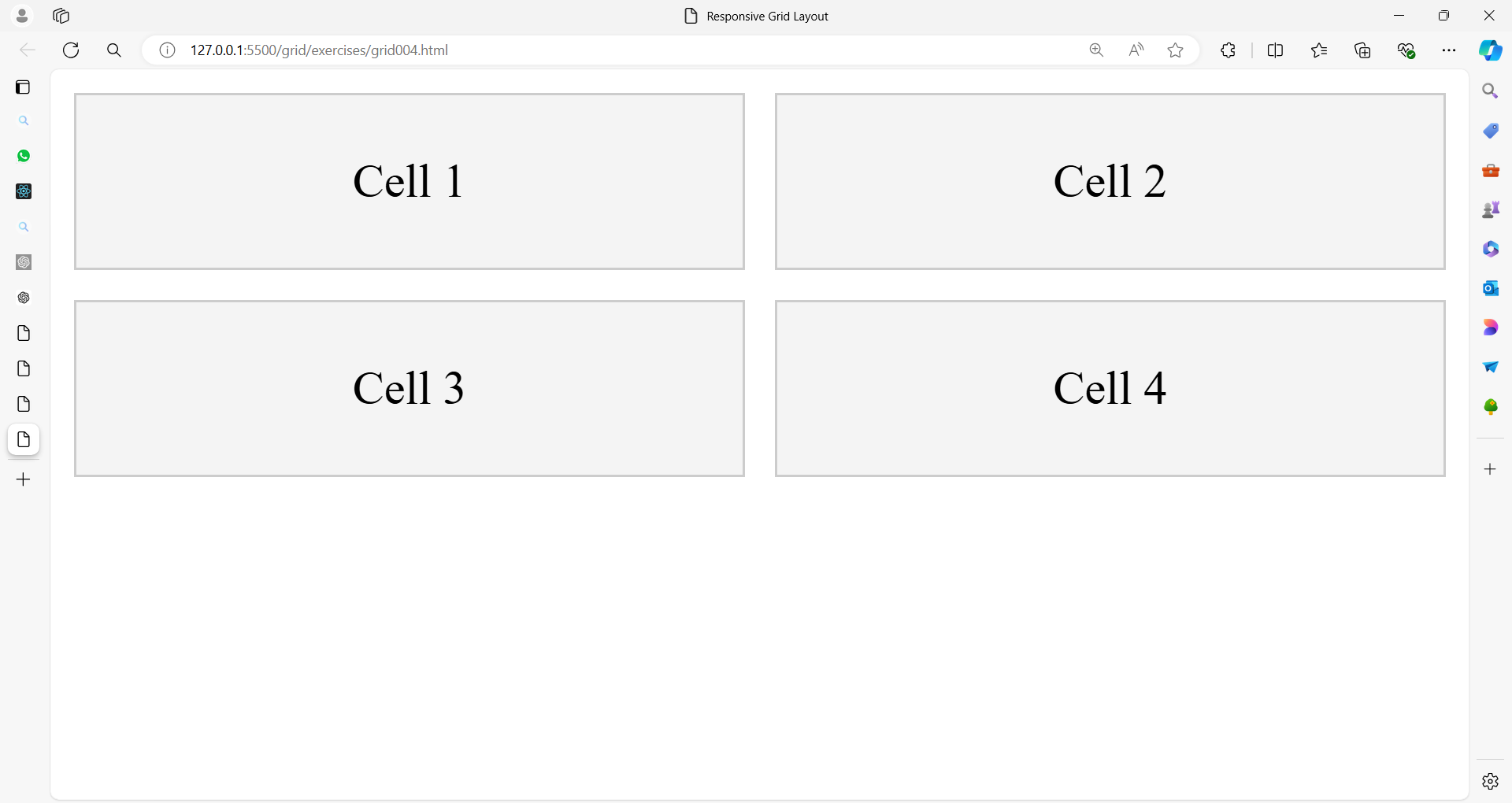
</div>

</body>

</html>

**Exercise 4: Responsive Grid Layout**

**Exercise:** Create a responsive grid layout with 4 columns on large screens, 2 columns on medium screens, and 1 column on small screens. Use media queries to adjust the layout.



**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Responsive Grid Layout</title>

<style>

.grid {

display: grid;

grid-template-columns: repeat(4, 1fr);

gap: 10px;

}

.cell {

background-color: #f4f4f4;

border: 1px solid #ccc;

padding: 20px;

text-align: center;

}

@media (max-width: 768px) {

.grid {

grid-template-columns: repeat(2, 1fr);

}

}

@media (max-width: 480px) {

.grid {

grid-template-columns: 1fr;

}

}

</style>

</head>

<body>

<div class="grid">

<div class="cell">Cell 1</div>

<div class="cell">Cell 2</div>

<div class="cell">Cell 3</div>

<div class="cell">Cell 4</div>

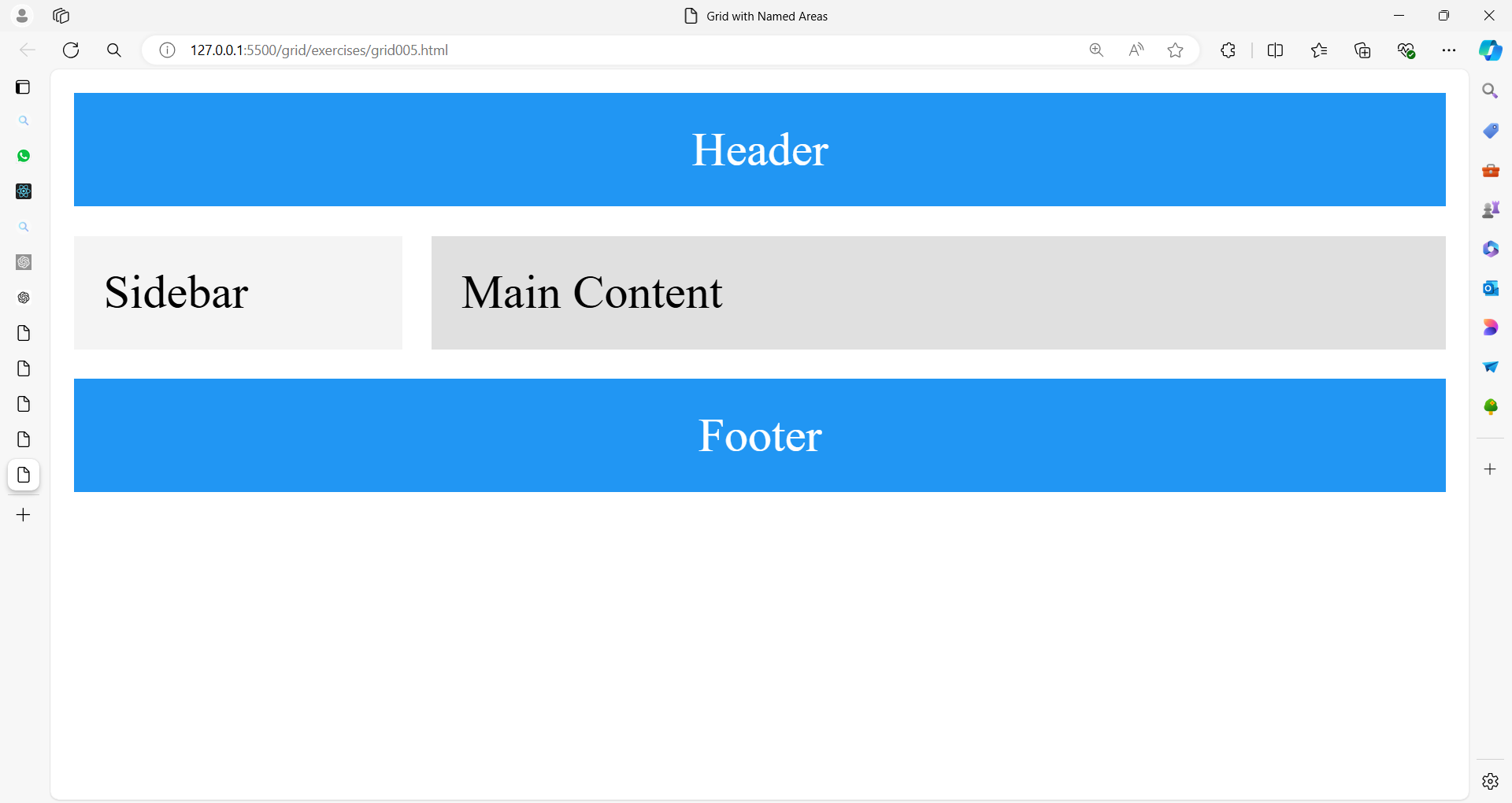
</div>

</body>

</html>

**Exercise 5: Grid with Named Areas**

**Exercise:** Create a grid layout with named areas. The layout should have a header, a sidebar, a main content area, and a footer.



**Solution:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Grid with Named Areas</title>

<style>

.grid {

display: grid;

grid-template-areas:

"header header header"

"sidebar main main"

"footer footer footer";

grid-template-columns: 1fr 3fr;

grid-template-rows: auto 1fr auto;

gap: 10px;

}

header {

grid-area: header;

background-color: #2196F3;

color: white;

padding: 10px;

text-align: center;

}

.sidebar {

grid-area: sidebar;

background-color: #f4f4f4;

padding: 10px;

}

.main {

grid-area: main;

background-color: #e0e0e0;

padding: 10px;

}

footer {

grid-area: footer;

background-color: #2196F3;

color: white;

padding: 10px;

text-align: center;

}

</style>

</head>

<body>

<div class="grid">

<header>Header</header>

<div class="sidebar">Sidebar</div>

<div class="main">Main Content</div>

<footer>Footer</footer>

</div>

</body>

</html>

Feel free to adjust the exercises based on your needs and try experimenting with different properties and values to better understand how CSS Grid Layout works.