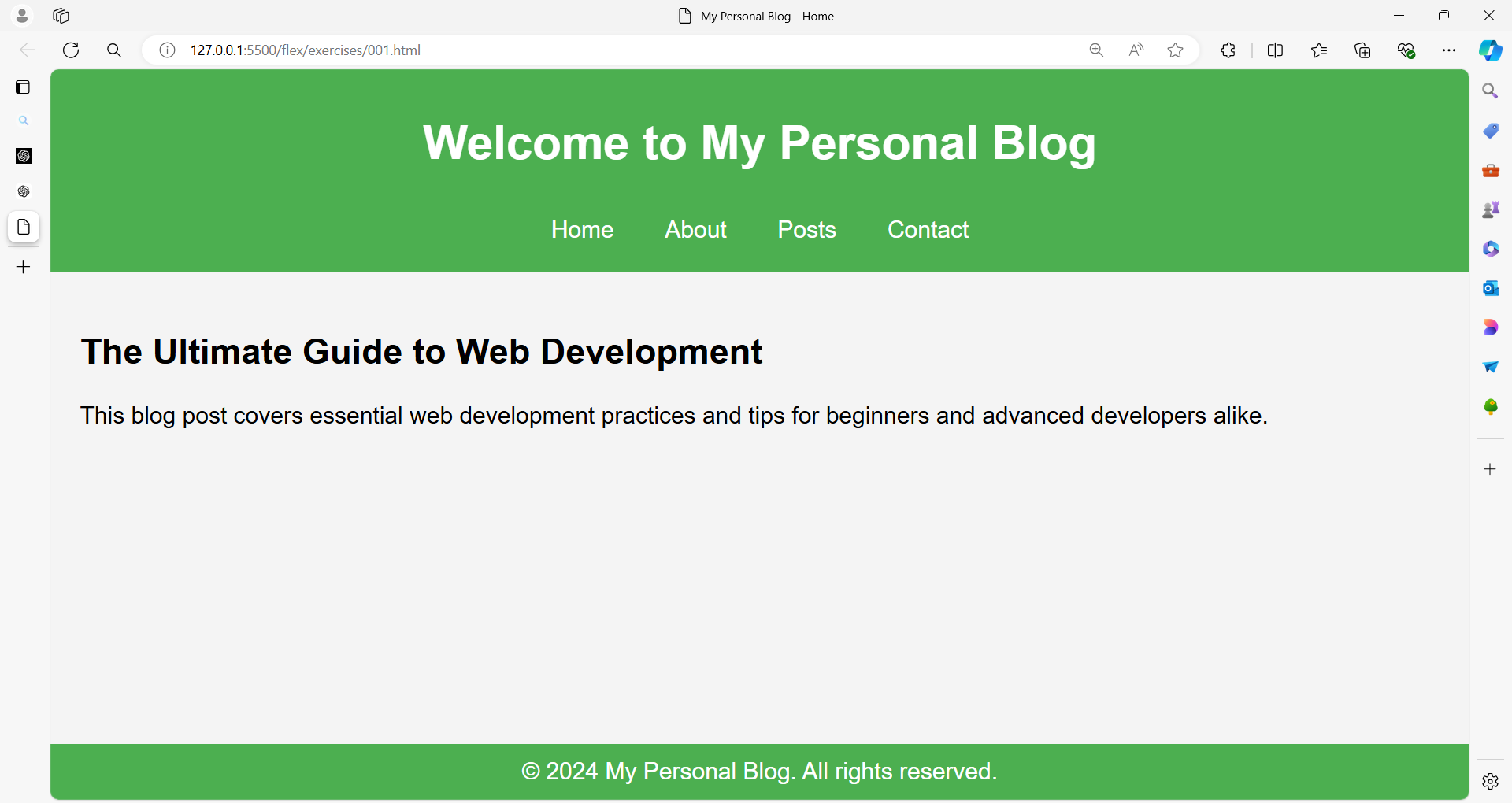
**Exercises on Flexbox Layout**

**Exercise 1: Basic Flexbox Layout**

**Objective**: Create a simple layout with a header, main content area, and footer using Flexbox.



**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>My Personal Blog - Home</title>

<style>

body {

margin: 0;

display: flex;

flex-direction: column;

min-height: 100vh;

font-family: Arial, sans-serif;

}

header {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 10px;

display: flex;

flex-direction: column;

align-items: center;

}

nav {

margin: 10px 0;

}

nav a {

color: white;

text-decoration: none;

margin: 0 15px;

font-size: 16px;

}

nav a:hover {

text-decoration: underline;

}

main {

flex: 1;

background-color: #f4f4f4;

padding: 20px;

}

footer {

background-color: #4CAF50;

color: white;

text-align: center;

padding: 10px;

}

</style>

</head>

<body>

<header>

<h1>Welcome to My Personal Blog</h1>

<nav>

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Posts</a>

<a href="#">Contact</a>

</nav>

</header>

<main>

<h2>The Ultimate Guide to Web Development</h2>

<p>This blog post covers essential web development practices and tips for beginners and advanced developers alike.</p>

</main>

<footer>

&copy; 2024 My Personal Blog. All rights reserved.

</footer>

</body>

</html>

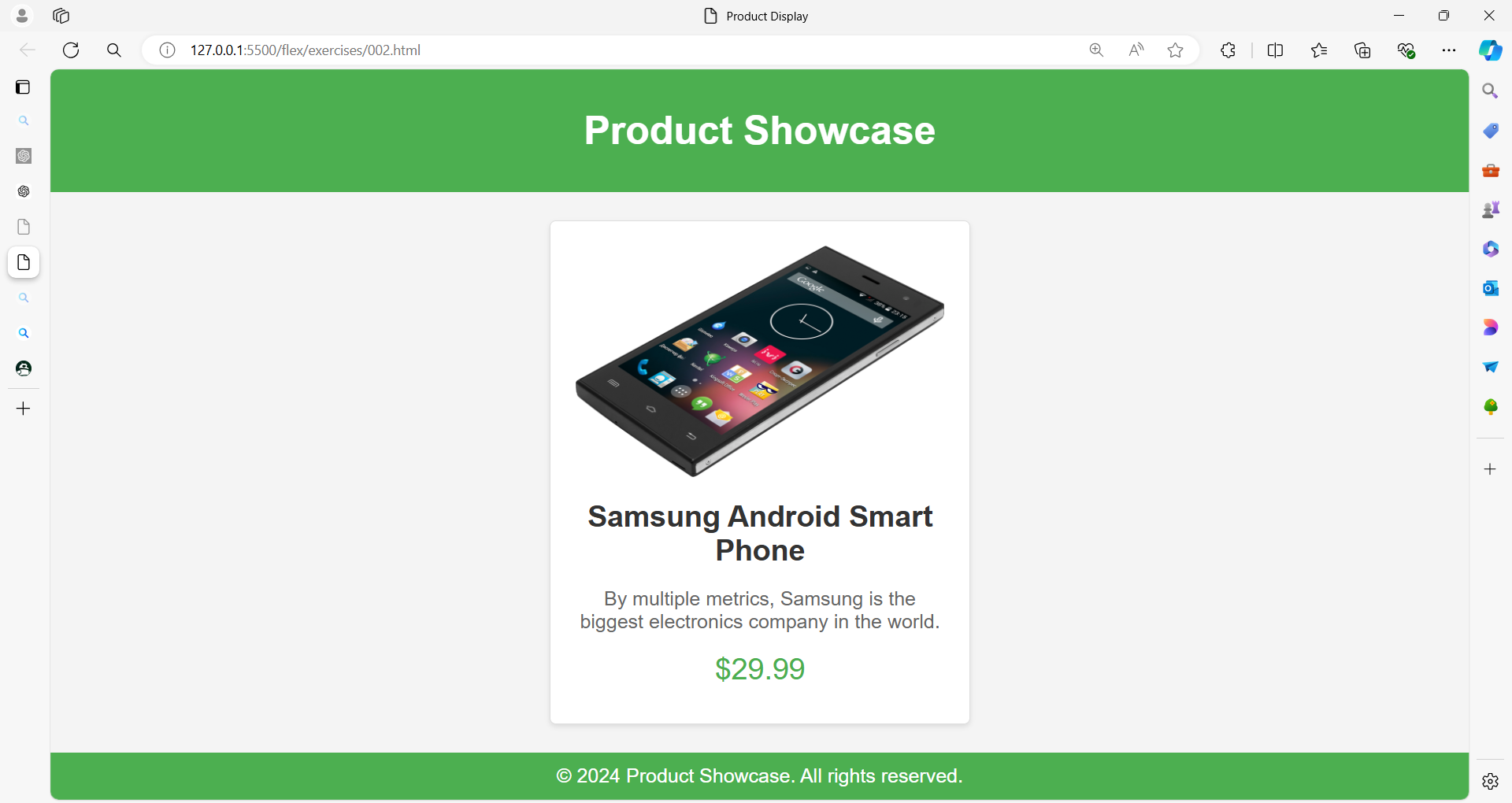
**Explanation**

* **body**: Uses Flexbox to ensure the entire page structure is in a column layout, stretching the main section to fill the remaining space between the header and footer.
* **header**: Provides a green background with white text for the header section.
* **main**: This section uses Flexbox to center its content both horizontally and vertically.
* **.content**: Contains the centered content with a green background, white text, and some padding and border radius for styling.
* **footer**: Provides a green background and white text for the footer section.

By integrating the Flexbox layout into your HTML, you ensure that the content is centered within the main section and the page structure remains consistent with the previous example.

**Exercise 2: Centering Content**

**Objective**: Center a child element both horizontally and vertically within a container.



**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Product Display</title>

    <style>

        body {

            margin: 0;

            display: flex;

            flex-direction: column;

            min-height: 100vh;

            font-family: Arial, sans-serif;

        }

        header {

            background-color: #4CAF50;

            color: white;

            text-align: center;

            padding: 10px;

        }

        main {

            flex: 1;

            display: flex;

            justify-content: center; /\* Center horizontally \*/

            align-items: center; /\* Center vertically \*/

            background-color: #f4f4f4;

        }

        .product {

            background-color: #fff;

            border: 1px solid #ddd;

            border-radius: 5px;

            padding: 20px;

            text-align: center;

            box-shadow: 0 2px 4px rgba(0,0,0,0.1);

            width: 300px;

        }

        .product img {

            max-width: 100%;

            border-radius: 5px;

        }

        .product h2 {

            margin: 15px 0;

            color: #333;

        }

        .product p {

            color: #666;

        }

        .product .price {

            margin: 10px 0;

            font-size: 1.5em;

            color: #4CAF50;

        }

        footer {

            background-color: #4CAF50;

            color: white;

            text-align: center;

            padding: 10px;

        }

    </style>

</head>

<body>

    <header>

        <h1>Product Showcase</h1>

    </header>

    <main>

        <div class="product">

            <img src="image001.png" alt="Product Image">

            <h2>Samsung Android Smart Phone</h2>

            <p>By multiple metrics, Samsung is the biggest electronics company in the world.</p>

            <div class="price">$29.99</div>

        </div>

    </main>

    <footer>

        &copy; 2024 Product Showcase. All rights reserved.

    </footer>

</body>

</html>

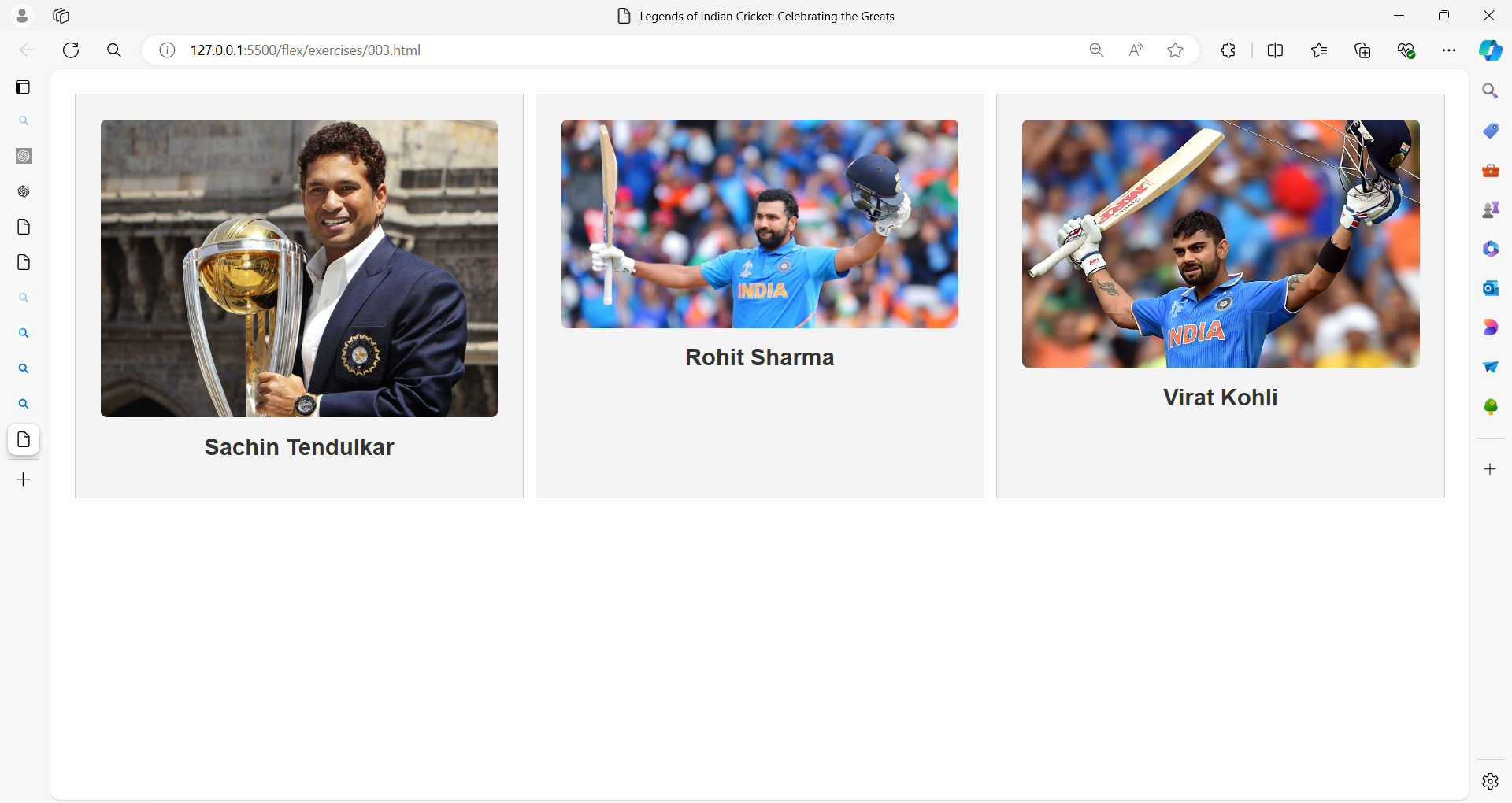
**Explanation**

* **body**: Uses Flexbox to structure the page with a column layout, making sure the main section takes up the space between the header and footer.
* **header**: Provides a green background with white text for the header section.
* **main**: Uses Flexbox to center the .product div both horizontally and vertically.
* **.product**: Contains the product information with a white background, border, border radius, padding, and shadow for a card-like appearance. It also centers its content.
* **.product img**: Ensures the product image is responsive and fits within its container.
* **.product h2**: Displays the product name.
* **.product p**: Contains the product description.
* **.product .price**: Shows the product price in a larger, green font.
* **footer**: Provides a green background and white text for the footer section.

This setup allows you to present a product in a clean and visually appealing way, with all elements centered on the page.

**Exercise 3: Flexbox Columns Layout**

**Objective**: Create a three-column layout where columns have equal width.



**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Legends of Indian Cricket: Celebrating the Greats</title>

    <style>

        body {

            margin: 0;

            font-family: Arial, sans-serif;

        }

        .container {

            display: flex;

            gap: 10px; /\* Space between columns \*/

            padding: 20px;

        }

        .column {

            flex: 1; /\* Equal width for each column \*/

            background-color: #f4f4f4;

            padding: 20px;

            border: 1px solid #ccc;

            text-align: center;

        }

        .column img {

            max-width: 100%; /\* Make image responsive \*/

            height: auto;

            border-radius: 5px;

        }

        .column h3 {

            margin: 10px 0;

            color: #333;

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="column">

            <img src="sachin.jpg" alt="Sachin Tendulkar">

            <h3>Sachin Tendulkar</h3>

        </div>

        <div class="column">

            <img src="rohith.jpg" alt="Rohit Sharma">

            <h3>Rohit Sharma</h3>

        </div>

        <div class="column">

            <img src="kohli.webp" alt="Virat Kohli">

            <h3>Virat Kohli</h3>

        </div>

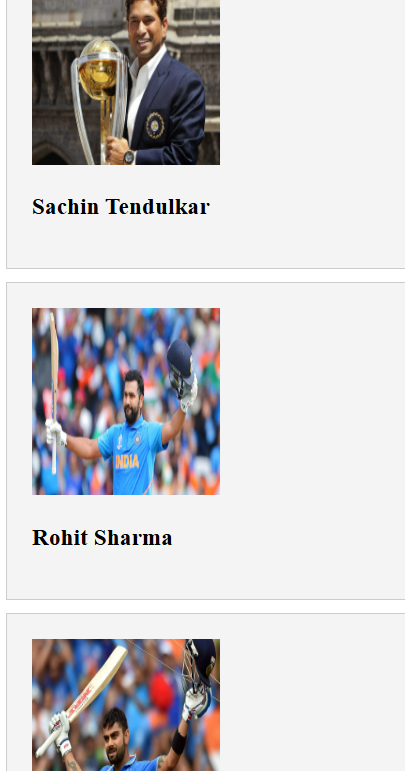
    </div>

</body>

</html>

**Exercise 4: Responsive Flexbox Layout**

**Objective**: Create a responsive layout where columns stack on top of each other on smaller screens.



**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title> Legends of Indian Cricket: Celebrating the Greats </title>

    <style>

        .container {

            display: flex;

            gap: 10px;

            flex-wrap: wrap; /\* Allow wrapping on smaller screens \*/

        }

        .column {

            flex: 1;

            min-width: 200px; /\* Minimum width for each column \*/

            background-color: #f4f4f4;

            padding: 20px;

            border: 1px solid #ccc;

        }

    </style>

</head>

<div class="container">

    <div class="column">

        <img src="sachin.jpg" alt="Sachin Tendulkar" width="150" height="150">

        <h3>Sachin Tendulkar</h3>

    </div>

    <div class="column">

        <img src="rohith.jpg" alt="Rohit Sharma" width="150" height="150">

        <h3>Rohit Sharma</h3>

    </div>

    <div class="column">

        <img src="kohli.webp" alt="Virat Kohli" width="150" height="150">

        <h3>Virat Kohli</h3>

    </div>

</div>

</html>

**Key Points and Enhancements**

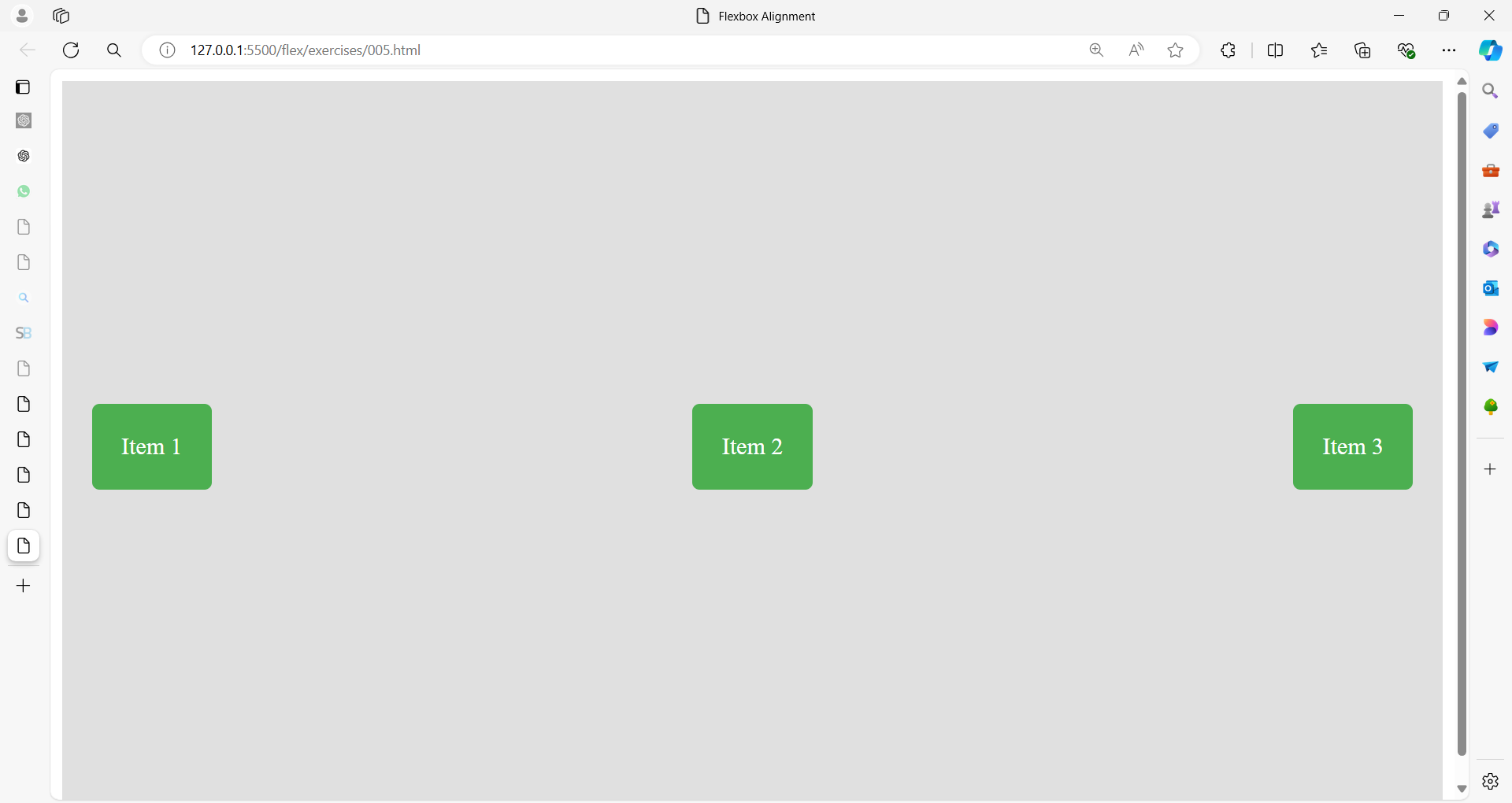
1. **Container Layout**:
   * display: flex; creates a flex container.
   * gap: 10px; adds space between the columns.
   * flex-wrap: wrap; allows columns to wrap onto new lines on smaller screens.
2. **Column Styling**:
   * flex: 1; ensures that each column grows to fill available space equally.
   * min-width: 200px; sets a minimum width for each column to ensure they don’t become too narrow.
   * background-color: #fff; sets a white background for each column.
   * padding: 20px; adds spacing inside each column.
   * border: 1px solid #ddd; provides a light border around each column.
   * border-radius: 8px; rounds the corners of the columns for a softer look.
   * text-align: center; centers the text and images inside the columns.
3. **Image Styling**:
   * border-radius: 50%; makes the images circular.
   * margin-bottom: 10px; adds space below the image.
4. **Heading Styling**:
   * margin: 0; removes default margins from the <h3> element.
   * color: #333; sets a dark gray color for the heading text.

**Result**

This layout will display the images of Sachin Tendulkar, Rohit Sharma, and Virat Kohli in a responsive grid. The images are circular, and each column adjusts its size based on the available screen width, making it suitable for different devices.

**Exercise 5: Flexbox Alignment**

**Objective**: Align items within a container with space between them and centered vertically.



**Code:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Flexbox Alignment</title>

<style>

.container {

display: flex;

justify-content: space-between; /\* Space between items \*/

align-items: center; /\* Center items vertically \*/

height: 100vh; /\* Full viewport height \*/

background-color: #e0e0e0;

padding: 0 20px;

}

.item {

background-color: #4CAF50;

color: white;

padding: 20px;

border-radius: 5px;

}

</style>

</head>

<body>

<div class="container">

<div class="item">Item 1</div>

<div class="item">Item 2</div>

<div class="item">Item 3</div>

</div>

</body>

</html>