.**Project Report: Financial Portfolio Pie Chart Website (Investment Version)**

1. Introduction

This Financial Portfolio Pie Chart Website is a simple interactive web tool designed to help users visually understand how their funds are distributed across four different account types: Wallet, Bank, Investment, and Crypto/Stocks. Users enter the amount they hold in each category, and the website instantly calculates and shows a color-coded pie chart representing the proportions of their total portfolio.

The project is built using HTML, CSS, and a small piece of inline JavaScript for live calculation and display updates—without needing any external libraries or backend.

---

2. Features Summary

- User Inputs: Four numeric fields accept the amounts for Wallet, Bank, Investment, and Crypto/Stocks.
- Live Pie Chart: The pie chart updates instantly as the user enters or changes values, using CSS conic-gradient to visually represent proportions.
- Colour Coding: Each account type is represented by a distinct colour in the pie chart and a legend.
- Responsive and Simple Design: Clean layout with intuitive input and clear legends for easy understanding.

---

3. Important Code Explanation

A. HTML Structure

- The webpage contains a .container div that centers and styles the content.
- A form with four <input type="number"> fields collects amounts for each financial category.
- A displays the pie chart.
- A .legend section links each colour with its account.

B. CSS Styling Highlights

- The pie chart is a circle created by setting border-radius: 50%.
- The pie chart background is a CSS custom property named --chart, initially set to a gray conic-gradient.
- Colour classes .color1 to .color4 define the colours for each segment and legend box:
    - Wallet: Red (#f94144)
    - Bank: Blue (#277da1)
    - Investment: Orange (#f3722c)

    - Crypto/Stocks: Green (#90be6d)
- The layout uses grid and flexbox for neat input alignment and legend display.

C. JavaScript (Inline in oninput)

The dynamic pie chart update logic is in the form's oninput attribute:

javascript

```
let w = +wallet.value;
let b = +bank.value;
let r = +investment.value;
let c = +crypto.value;
let total = w + b + r + c;
if (total === 0) total = 1;
let p1 = (w / total) * 100;
let p2 = p1 + (b / total) * 100;
let p3 = p2 + (r / total) * 100;
document.querySelector('.pie-chart').style.setProperty('--chart',
  `conic-gradient(
    #f94144 0% ${p1}%,
    #277da1 ${p1}% ${p2}%,
    #f3722c ${p2}% ${p3}%,
    #90be6d ${p3}% 100%
  )`
);
```

- The variables w, b, r, c hold the numeric input values for Wallet, Bank, Investment, and Crypto.
- total sums the four amounts, with a fallback to 1 to prevent division by zero.
- p1, p2, and p3 are cumulative percentage cutoffs for pie segments.
- The CSS custom property --chart is updated with a conic-gradient specifying each segment's start and end percentages with the assigned colors.
- This redraws the pie chart dynamically to reflect the current data.

---

4. User Instructions

- Enter your amounts for Wallet, Bank, Investment, and Crypto/Stocks in the input fields.

- The pie chart will immediately update to show the relative share of each category.
- Use the color-coded legend below the chart to identify each segment.
- The "Update Pie (Static)" button is non-functional (returns false) because updates happen live on input.

---

5. Advantages

- Real-time feedback without page reloads or button clicks.
- No external dependencies or libraries needed.
- Clean, modern, and accessible design.
- Demonstrates effective use of CSS gradients and JavaScript integration.

---

6. Limitations and Future Enhancements

- Does not save data between sessions.
- Limited to four predefined account types; can be expanded.
- Could add percentage labels or tooltips on the pie chart.
- Export, printing, or detailed report generation could be added.
- More robust validation and accessibility improvements can be integrated.

---

7. Conclusion

This webpage project effectively visualizes financial data with minimal code and no dependencies. It’s perfect for learning or simple personal budgeting, showing how modern CSS features like conic-gradient combined with JavaScript can create interactive, informative visualizations.
