**Project Title: Personal Finance Tracker**

**Problem Statement**

This project aims to create a simple, user-friendly web application using **HTML**, **CSS**, and **JavaScript** that allows users to:

* Record their daily expenses.
* View their financial activity in a clear and visually appealing way.
* Manage their budget effectively and build good financial habits.

**Detailed Project Pages and Features**

**1. Header (Appears on all pages)**

**What to Include:**

* A **Logo** (can just be the app name styled with CSS).
* A **Navigation Menu** with links to:
  + Home
  + Dashboard
  + Add Expense
  + Reports
  + Settings
  + Login/Logout (optional).
* A **Responsive Design** that works on all devices (CSS Media Queries for smaller screens).

**Beginner Tips:**

* Use a <nav> element for the navigation bar.
* Add hover effects using CSS (e.g., change color or underline links on hover).
* Use Flexbox or Grid for layout alignment.

**2. Landing Page**

This is the first page users see and should showcase the app's purpose and features.

**Elements:**

1. **Hero Section:**
   * A welcoming title (e.g., *"Track Your Finances Easily!"*).
   * A tagline (e.g., *"Stay on top of your expenses with just a few clicks."*).
   * A big "Get Started" button that links to the Sign-Up page.
2. **Features Section:**
   * List the app's features with simple icons and text. For example:
     + *"Add your daily expenses effortlessly."*
     + *"View your spending trends."*
     + *"Set and track your savings goals."*
3. **Sign-Up Section:**
   * A simple form where users can enter their email and password (you can simulate this without actual backend integration).

**Beginner Tips:**

* Use a <section> for each part of the page.
* Use basic animations with CSS (e.g., fade-in or hover effects).
* For buttons, use a <button> or <a> tag styled with CSS.

**3. Dashboard Page**

The dashboard is where users can see a summary of their financial data.

**Elements:**

1. **Summary Section:**
   * Display information such as:
     + Total Expenses (e.g., "You spent $500 this month.")
     + Remaining Budget (e.g., "You have $200 left to spend.")
   * Use boxes or cards styled with CSS to organize this information.
2. **Recent Transactions:**
   * Show a list of recent expenses in a table format:
     + Date | Expense Name | Amount | Category
3. **Charts (Optional for Beginners):**
   * Use a simple bar chart or pie chart to visualize spending (e.g., categories like food, travel).
   * Use **Chart.js** for an easy and beginner-friendly chart library.

**Beginner Tips:**

* Use <div> for boxes and tables for organizing data.
* Start simple: Focus on text data first, then add charts later.

**4. Add Expense Page**

This page allows users to log a new expense.

**Elements:**

* A simple form with:
  + **Date** (use an input type="date").
  + **Expense Name** (e.g., "Lunch").
  + **Amount** (use input type="number").
  + **Category** (e.g., dropdown list with options like Food, Travel, Bills).
  + **Notes** (optional).
* A Submit button that stores the data (use localStorage for beginners).

**Beginner Tips:**

* Validate the form using JavaScript (e.g., check if fields are filled out).
* Use CSS to style the form and make it look clean and user-friendly.

**5. Reports Page**

The reports page provides an overview of spending habits.

**Elements:**

1. **Expense Breakdown:**
   * Display a pie chart or bar graph showing how much money is spent on each category (use **Chart.js**).
2. **Trends Over Time:**
   * Show a basic line graph for spending over time (optional).
3. **Insights Section:**
   * Provide simple tips like: *"You spent 30% of your budget on food. Consider reducing eating out."*

**Beginner Tips:**

* Start with a basic table or text report, then add visual charts once comfortable with JavaScript.
* Use a <canvas> element for charts if using libraries like Chart.js.

**6. Settings Page**

This page allows users to personalize their experience.

**Elements:**

* **Change Budget:** A simple input field where users can set or update their monthly budget.
* **Theme Toggle:** A button to switch between light and dark mode (use JavaScript to change CSS).
* **Reset Data:** A button to clear all stored data (use localStorage clear method).

**Beginner Tips:**

* Use basic buttons styled with CSS.
* Practice DOM manipulation in JavaScript to update user settings.

**7. Footer**

The footer provides additional navigation and information.

**Elements:**

* Links: Privacy Policy, Terms of Service, Contact Us.
* Social Media Icons (e.g., Facebook, Twitter).
* Copyright Notice: *"© 2024 Personal Finance Tracker. All rights reserved."*

**Beginner Tips:**

* Use a <footer> element and style it with a consistent background and font color.

**General Tips for Beginners**

1. **Start Simple:** Focus on creating static pages first with HTML and CSS. Add interactivity later with JavaScript.
2. **Responsive Design:** Use CSS media queries to make the app look good on phones and desktops.
3. **Break Tasks Into Steps:** Work on one feature or page at a time.
4. **Learn LocalStorage:** Use JavaScript's localStorage API to save and retrieve data without needing a backend.
5. **Practice Reusing Code:** Use external CSS files and reusable components (e.g., header and footer).