

Financial Planner Web Application

Overview

The Financial Planner web application empowers users to manage their finances effectively. Features include:

- **Budget Planning:** Allocate budgets and track expenses with visual feedback.
- **Goal Tracking:** Set and monitor financial goals with progress indicators.
- **Investment Management:** Track portfolio growth over time with dynamic visualizations.
- **Resources:** Access curated financial literacy tools and materials.
- **Secure Authentication:** Login and registration functionalities to protect user data.

Features

Budgeting

- Enter monthly income and allocate budgets.
- Track expenses visually with a dynamic pie chart.
- Real-time warnings for budget overages.

Goal Tracking

- Define financial goals with target amounts and deadlines.
- View progress through visual indicators and update goals as needed.

Investment Tracking

- Add and track investments.
- Visualize portfolio growth with interactive line charts.

Resources

- Educational tools and links to improve financial literacy.

Authentication

- Register and login securely to manage personal finance.

Getting Started

Prerequisites

- Modern web browser (e.g., Chrome, Firefox)
- Internet connection for loading external resources (e.g., Chart.js)

Installation

<https://github.com/marianogarciamelo/312Project.git>

Dependencies

- [Chart.js](#): For data visualizations (pie and line charts).

Testing Procedure

Budgeting

1. Validate income input and remaining income display.
2. Add budget categories and ensure updates in the list and dropdown.
3. Track expenses and verify dynamic updates in the pie chart.

Goal Tracking

1. Add goals with target amounts and deadlines.
2. Confirm progress visualization and functionality to update goals.

Investment Tracking

1. Add investments and verify chart updates with new data points.

Authentication

1. Test registration and login with valid and invalid credentials.

Navigation

1. Verify navigation links redirect to correct pages.
2. Confirm modal display for login and registration.

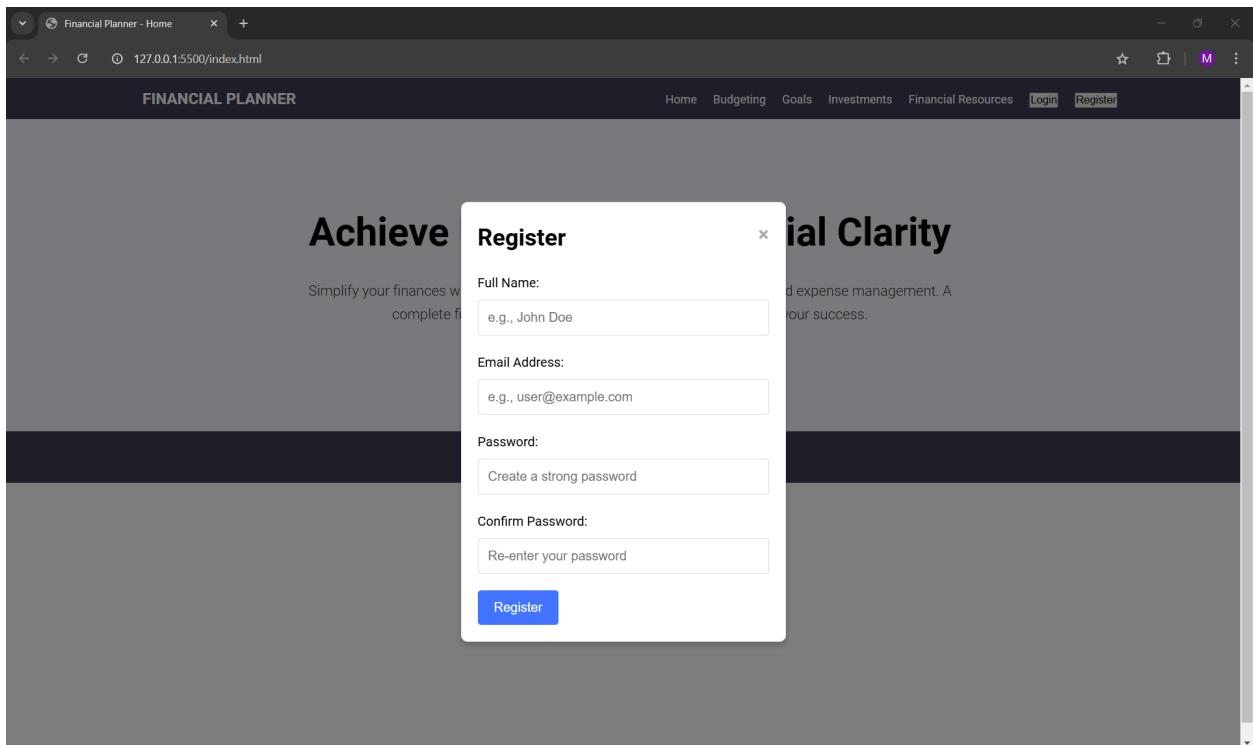
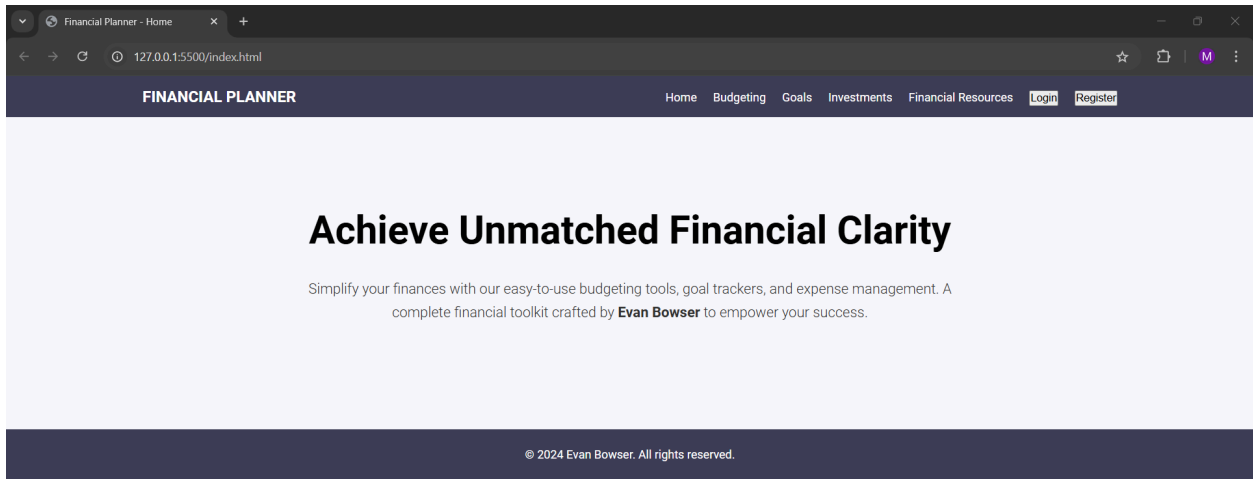
File Structure

```
|— index.html          # Home page
|— budget.html        # Budgeting page
|— goals.html         # Goal tracking page
|— investments.html   # Investments page
|— resources.html     # Resources page
|— login.html         # Login page
|— register.html      # Registration page
|— scripts/
|   |— app.js         # JavaScript logic
|— styles/
|   |— style.css      # CSS for styling
```

Contributors

- **[Evan Bowser]**: Budgeting module development.
- **[Chris Muller]**: Goal tracking functionality.
- **[Mariano Garcia Melo]**: Investment tracking and visualizations.
- **[Jaedon Satchell]**: Authentication and resources pages.
- **[Evan Bowser]**: UI design and responsive layout.

Screenshots



```
File Edit Selection View Go Run Terminal Help
Financial Planner_

EXPLORER
  FINANCIAL PLANNER_
    scripts
    styles
    # style.css
    budget.html
    goals.html
    index.html
    investments.html
    login.html
    register.html
    resources.html

  app.js
  index.html
  style.css

scripts > app.js
1 document.addEventListener('DOMContentLoaded', () => {
2   const incomeForm = document.getElementById('incomeForm');
3   const categoryForm = document.getElementById('categoryForm');
4   const expenseForm = document.getElementById('expenseForm');
5   const incomeInput = document.getElementById('income');
6   const categoryNameInput = document.getElementById('categoryName');
7   const categoryBudgetInput = document.getElementById('categoryBudget');
8   const expenseCategorySelect = document.getElementById('expenseCategory');
9   const expenseAmountInput = document.getElementById('expenseAmount');
10  const remainingIncomeDisplay = document.getElementById('remainingIncome');
11  const budgetList = document.getElementById('budgetList');
12  const warningMessage = document.getElementById('warning');
13  const expenseChartCanvas = document.getElementById('expenseChart');
14
15  let totalIncome = 0;
16  let remainingIncome = 0;
17  const categories = {};
18
19  // Initialize the Pie Chart
20  const expenseChart = new Chart(expenseChartCanvas, {
21    type: 'pie',
22    data: {
23      labels: [],
24      datasets: [{
25        data: [],
26        backgroundColor: ['#ff6384', '#36a2eb', '#ffce56', '#4bc0c0', '#9966ff', '#ff9f40'],
27        hoverOffset: 4,
28      }]
29    },
30    options: {
31      responsive: true,
32      plugins: {
33        legend: {
34          position: 'top',
35        },
36      },
37    },
38  });
39
40  incomeForm.addEventListener('submit', (e) => {
41    e.preventDefault();
42    totalIncome = parseFloat(incomeInput.value) || 0;
```

