**TEAM BUDGET PLANNER SOURCE CODE**

**STYLE.CSS**

\* {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

.clearfix::after {

    content: "";

    display: table;

    clear: both;

}

body {

    color: #555;

    font-family: Open Sans;

    font-size: 16px;

    position: relative;

    height: 100vh;

    font-weight: 400;

}

.left-container {

    height: 100vh;

    background-image: linear-gradient(#46e992, #98a041);

    background-size: cover;

    background-position: center;

    position: relative;

}

.right-container {

    height: 100vh;

    width: 100%;

    background-image: linear-gradient(#6bcaf0, #e56f6f);

    background-size: cover;

    background-position: center;

    position: relative;

}

.header {

    font-weight: 700;

    font-size: 36px;

}

.sub-text {

    font-size: 22px;

    font-weight: 400;

}

.month-container {

    padding-top: 25%;

    padding-left: 5%;

    padding-right: 5%;

}

.calc-container {

    padding-top: 12%;

    padding-left: 5%;

    padding-right: 5%;

}

.fs-white {

    color: #c92c2c;

}

.fs-dark-grey {

    color: #2e088c;

}

.budget-container {

    display: inline-block;

    background: #f2e8e8;

    border-radius: 8px;

    box-shadow: 0 6px 4px #0e0e0d;

}

.month-amount {

    font-size: 36px;

    font-weight: 700;

}

.bottom-border {

    border-bottom: 1px solid #00446D;

}

.expense-row {

    padding: 10px;

}

.expense-date {

    color: #1c0305;

}

.expense-text {

    color: #891306;

}

.expense-list {

    overflow-y: scroll;

}

.fs-15 {

    font-size: 15px;

}

.expense-value {

    text-align: end;

}

.expense-saving {

    color: #039300;

}

.expense-cost {

     color: #E40000;

}

.expense-investment {

    color: #f48803;

}

#expense-chart {

    margin: 20% 0;

}

.btn-submit-expense {

    border-radius: 50%;

}

.currency-select {

    margin: 0 4%;

}

.selected-currency {

    color: #ffffff;

    font-size: 12px;

    font-weight: 700;

    margin-top: 1%;

}

**INDEX.HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Expense Manager</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

          integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

    <link rel="stylesheet" href="style.css">

    <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;700&display=swap" rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.css" rel="stylesheet">

    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"

            integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

            crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"

            integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"

            crossorigin="anonymous"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"

            integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

            crossorigin="anonymous"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.js" crossorigin="anonymous"></script>

</head>

<body>

<div class="row">

    <div class="col-4 left-container">

        <div class="month-container">

            <div class="header fs-white">Your Budget</div>

            <div id="current-month" class="sub-text fs-white"></div>

            <div class="budget-container p-2 mt-4">

                <span id="month-budget" class="month-amount">₹ 0</span>

            </div>

        </div>

        <div class="chart-container">

            <canvas id="expense-chart"></canvas>

        </div>

    </div>

    <div class="col-8 right-container">

        <div class="calc-container">

            <div class="header fs-dark-grey">BUDGET PLANNING</div>

            <div class="dropdown open">

                <button class="btn btn-info dropdown-toggle"

                        type="button" id="dropdownMenu3" data-toggle="dropdown"

                        aria-haspopup="true" aria-expanded="false">

                    Expense Type

                </button>

                <div class="dropdown-menu">

                    <a class="dropdown-item" id="type-savings">Savings</a>

                    <a class="dropdown-item" id="type-expense">Expense</a>

                    <a class="dropdown-item" id="type-investment">Investment</a>

                </div>

            </div>

            <div class="mt-3 tracking-text text-capitalize sub-text bottom-border">Tracking Savings 💰</div>

            <div class="row mt-4">

                <div class="col-7">

                    <input class="form-control input-expense-description" type="text" placeholder="Description">

                </div>

                <div class="col-4">

                    <input class="form-control input-expense-value" type="number" placeholder="Value">

                </div>

                <div class="col-1">

                    <button type="button" class="btn btn-success btn-submit-expense">&check;</button>

                </div>

            </div>

            <div class="expense-list mt-4">

            </div>

        </div>

    </div>

</div>

<script src="app.js"></script>

</body>

</html>

**APP.JS**

let ExpenseController = (() => {

    let total = 0, savings = 0, expenses = 0, investments = 0;

    return {

        inputEntry(userInput) {

            if (userInput['expenseType'] === 'savings') {

                savings += userInput['value'];

                total += userInput['value'];

            }

            if (userInput['expenseType'] === 'investment') {

                investments += userInput['value'];

                total -= userInput['value'];

            }

            if (userInput['expenseType'] === 'expense') {

                expenses += userInput['value'];

                total -= userInput['value'];

            }

        },

        getSavingsData() {

            return savings;

        },

        getExpensesData() {

            return expenses;

        },

        getInvestmentData() {

            return investments;

        },

        getTotalData() {

            return total;

        }

    }

})();

let UIController = (() => {

    let expenseType = 'savings';

    let HTMLStrings = {

        inExpenseDescription: '.input-expense-description',

        inExpenseValue: '.input-expense-value',

        btnSubmitExpense: '.btn-submit-expense',

        expenseList: '.expense-list',

        currentMonth: '#current-month',

        typeExpense: '#type-expense',

        typeSavings: '#type-savings',

        typeInvestment: '#type-investment',

        trackingText: '.tracking-text',

        expenseChart: '#expense-chart',

        monthBudget: '#month-budget'

    };

    return {

        numberFormat(number) {

            return Intl.NumberFormat('en-IN').format(number);

        },

        showCurrentMonth() {

            let now, month, year, months;

            now = new Date();

            month = now.getMonth();

            year = now.getFullYear();

            months = [

                'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October',

                'November', 'December'

            ];

            document.querySelector(HTMLStrings.currentMonth).textContent = months[month] + " " + year;

        },

        getHTMLStrings() {

            return HTMLStrings;

        },

        setExpenseType(type) {

            console.log('here', type);

            this.expenseType = type;

            let emoji ="💰";

            if (type === 'savings') {

                emoji ="💰";

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-warning')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-warning');

                }

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-danger')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-danger');

                }

                if (!document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-success')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.add('btn-success');

                }

            }

            if (type === 'expense') {

                emoji = "🧾";

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-warning')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-warning');

                }

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-success')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-success');

                }

                if (!document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-danger')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.add('btn-danger');

                }

            }

            if (type === 'investment') {

                emoji = "🏠";

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-danger')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-danger');

                }

                if (document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-success')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.remove('btn-success');

                }

                if (!document.querySelector(HTMLStrings.btnSubmitExpense).classList.contains('btn-warning')) {

                    document.querySelector(HTMLStrings.btnSubmitExpense).classList.add('btn-warning');

                }

            }

            document.querySelector(HTMLStrings.trackingText).textContent = "Tracking " + type + " " + emoji;

        },

        getUserExpenseInput() {

            return {

                description: document.querySelector(HTMLStrings.inExpenseDescription).value,

                value: parseInt(document.querySelector(HTMLStrings.inExpenseValue).value),

                date: new Date().toLocaleDateString(),

                expenseType: this.expenseType ? this.expenseType : 'savings'

            }

        },

        addListItem (inputObj) {

            let html, element;

            element = HTMLStrings.expenseList;

            if (inputObj['expenseType'] === 'savings') {

                html = '<div class="bottom-border"> <div class="row expense-row"><div class="col-2 expense-date fs-15">' + inputObj['date'] + ' </div><div class="col-8 expense-text fs-15"> ' + inputObj['description'] + ' </div><div class="col-2 expense-value expense-saving fs-15"> ₹ ' + this.numberFormat(inputObj['value']) + ' </div></div>'

            } else if (inputObj['expenseType'] === 'expense') {

                html = '<div class="bottom-border"> <div class="row expense-row"><div class="col-2 expense-date fs-15">' + inputObj['date'] + ' </div><div class="col-8 expense-text fs-15"> ' + inputObj['description'] + ' </div><div class="col-2 expense-value expense-cost fs-15"> ₹ ' + this.numberFormat(inputObj['value']) + ' </div></div>'

            } else if (inputObj['expenseType'] === 'investment') {

                html = '<div class="bottom-border"> <div class="row expense-row"><div class="col-2 expense-date fs-15">' + inputObj['date'] + ' </div><div class="col-8 expense-text fs-15"> ' + inputObj['description'] + ' </div><div class="col-2 expense-value expense-investment fs-15"> ₹ ' + this.numberFormat(inputObj['value']) + ' </div></div>'

            }

            // Add the new element

            document.querySelector(element).insertAdjacentHTML('beforeend', html);

            // Clear the input fields after adding element

            document.querySelector(HTMLStrings.inExpenseValue).value = "";

            document.querySelector(HTMLStrings.inExpenseDescription).value = "";

        },

        updateOverallTotal(totalValue) {

            document.querySelector(HTMLStrings.monthBudget).textContent  = "₹ " + this.numberFormat(totalValue);

            if (totalValue > 0) {

                if (document.querySelector(HTMLStrings.monthBudget).classList.contains('expense-cost')) {

                    document.querySelector(HTMLStrings.monthBudget).classList.remove('expense-cost');

                }

                document.querySelector(HTMLStrings.monthBudget).classList.add('expense-saving');

            } else {

                if (document.querySelector(HTMLStrings.monthBudget).classList.contains('expense-saving')) {

                    document.querySelector(HTMLStrings.monthBudget).classList.remove('expense-saving');

                }

                document.querySelector(HTMLStrings.monthBudget).classList.add('expense-cost');

            }

        },

        displayChart(savings = 0, expenses = 0, investments = 0) {

            let ctx = document.querySelector(HTMLStrings.expenseChart);

            let expenseChart = new Chart(ctx, {

                type: 'doughnut',

                data: {

                    labels: ['Savings', 'Expenses', 'Investments'],

                    datasets: [{

                        data: [savings, expenses, investments],

                        backgroundColor: [

                            'rgba(32, 137, 56, 1)',

                            'rgba(255, 84, 98, 1)',

                            'rgba(255, 206, 86, 1)'

                        ],

                        borderWidth: 0.5

                    }]

                },

                options: {

                    legend: {

                        labels: {

                            fontColor: 'Black'

                        }

                    }

                }

            });

        }

    }

})();

((UIController, ExpenseController) => {

    let HTMLStrings = UIController.getHTMLStrings();

    let setupEventListeners = () => {

        document.querySelector(HTMLStrings.btnSubmitExpense).addEventListener('click', addExpense);

        document.querySelector(HTMLStrings.typeExpense).addEventListener('click', () => {

            setExpenseType('expense')

        });

        document.querySelector(HTMLStrings.typeInvestment).addEventListener('click', () => {

            setExpenseType('investment')

        });

        document.querySelector(HTMLStrings.typeSavings).addEventListener('click', () => {

            setExpenseType('savings')

        });

    };

    let setExpenseType = (type) => {

        UIController.setExpenseType(type);

    }

    let addExpense = () => {

        let input = UIController.getUserExpenseInput();

        console.log(input);

        if (input.description !== "" && !isNaN(input.value) && input.value > 0) {

            console.log('Adding item');

            UIController.addListItem(input);

            ExpenseController.inputEntry(input);

            UIController.updateOverallTotal(ExpenseController.getTotalData());

            UIController.displayChart(ExpenseController.getSavingsData(), ExpenseController.getExpensesData(),

                ExpenseController.getInvestmentData());

        }

    }

    let init = () => {

        console.log('Initializing...');

        setupEventListeners();

        UIController.showCurrentMonth();

    }

    init();

})(UIController, ExpenseController);