**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>BudgetAnalyzer</title>

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

</head>

<body>

<div class="budget-container">

<div class="app-title">

<a href="">Budget<b>Analyzer</b></a>

</div>

<div class="budget-header">

<div class="balance">

<div class="title">

Balance

</div>

<div class="value">

<small>$</small>0

</div>

</div>

<div class="account">

<div class="income">

<div class="title">

TotalIncome

</div>

<div class="income-total">

<small>$</small>0

</div>

</div>

<div class="chart"></div>

<div class="outcome">

<div class="title">

TotalExpenses

</div>

<div class="outcome-total">

<small>$</small>0

</div>

</div>

</div>

</div>

<div class="budget-dashboard">

<div class="dash-title">Dashboard</div>

<div class="toggle">

<div class="tab1">Expenses</div>

<div class="tab2">Income</div>

<div class="tab3 active">All</div>

</div>

<div class="hide" id="income">

<ul class="list"></ul>

<div class="input">

<input type="text" id="income-title-input" name="title" placeholder="Title">

<input type="number" id="income-amount-input" name="amount" placeholder="$0">

<div class="add-income"><img src="icon/plus.png" alt=""></div>

</div>

</div>

<div class="hide" id="expense">

<ul class="list"></ul>

<div class="input">

<input type="text" id="expense-title-input" name="title" placeholder="Title">

<input type="number" id="expense-amount-input" name="amount" placeholder="$0">

<div class="add-expense"><img src="icon/plus.png" alt=""></div>

</div>

</div>

<div id="all">

<ul class="list"></ul>

</div>

</div>

</div>

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

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

</body>

</html>

**Chart.js**

const chart = document.querySelector(".chart");

const canvas = document.createElement("canvas");

canvas.width = 50;

canvas.height = 50;

chart.appendChild(canvas);

const ctx = canvas.getContext("2d");

ctx.lineWidth = 8;

const R = 20;

function drawCircle(color, ratio, anticlockwise){

ctx.strokeStyle = color;

ctx.beginPath();

ctx.arc( canvas.width/2, canvas.height/2, R, 0, ratio \* 2 \* Math.PI, anticlockwise);

ctx.stroke();

}

function updateChart( income, outcome){

ctx.clearRect(0, 0, canvas.width, canvas.height);

let ratio = income / (income+outcome);

drawCircle("#FFFFFF", - ratio, true);

drawCircle("#F0624D", 1 - ratio, false);

}

**Budget.js**

// SELECT ELEMENTS

const balanceEl = document.querySelector(".balance .value");

const incomeTotalEl = document.querySelector(".income-total");

const outcomeTotalEl = document.querySelector(".outcome-total");

const incomeEl = document.querySelector("#income");

const expenseEl = document.querySelector("#expense");

const allEl = document.querySelector("#all");

const incomeList = document.querySelector("#income .list");

const expenseList = document.querySelector("#expense .list");

const allList = document.querySelector("#all .list");

// SELECT BTNS

const expenseBtn = document.querySelector(".tab1");

const incomeBtn = document.querySelector(".tab2");

const allBtn = document.querySelector(".tab3");

// INPUT BTS

const addExpense = document.querySelector(".add-expense");

const expenseTitle = document.getElementById("expense-title-input");

const expenseAmount = document.getElementById("expense-amount-input");

const addIncome = document.querySelector(".add-income");

const incomeTitle = document.getElementById("income-title-input");

const incomeAmount = document.getElementById("income-amount-input");

// VARIABLES

let ENTRY\_LIST;

let balance = 0, income = 0, outcome = 0;

const DELETE = "delete", EDIT = "edit";

// LOOK IF THERE IS SAVED DATA IN LOCALSTORAGE

ENTRY\_LIST = JSON.parse(localStorage.getItem("entry\_list")) || [];

updateUI();

// EVENT LISTENERS

expenseBtn.addEventListener("click", function(){

show(expenseEl);

hide( [incomeEl, allEl] );

active( expenseBtn );

inactive( [incomeBtn, allBtn] );

})

incomeBtn.addEventListener("click", function(){

show(incomeEl);

hide( [expenseEl, allEl] );

active( incomeBtn );

inactive( [expenseBtn, allBtn] );

})

allBtn.addEventListener("click", function(){

show(allEl);

hide( [incomeEl, expenseEl] );

active( allBtn );

inactive( [incomeBtn, expenseBtn] );

})

addExpense.addEventListener("click", function(){

// IF ONE OF THE INPUTS IS EMPTY => EXIT

if(!expenseTitle.value || !expenseAmount.value ) return;

// SAVE THE ENTRY TO ENTRY\_LIST

let expense = {

type : "expense",

title : expenseTitle.value,

amount : parseInt(expenseAmount.value)

}

ENTRY\_LIST.push(expense);

updateUI();

clearInput( [expenseTitle, expenseAmount] )

})

addIncome.addEventListener("click", function(){

// IF ONE OF THE INPUTS IS EMPTY => EXIT

if(!incomeTitle.value || !incomeAmount.value ) return;

// SAVE THE ENTRY TO ENTRY\_LIST

let income = {

type : "income",

title : incomeTitle.value,

amount : parseInt(incomeAmount.value)

}

ENTRY\_LIST.push(income);

updateUI();

clearInput( [incomeTitle, incomeAmount] )

})

incomeList.addEventListener("click", deleteOrEdit);

expenseList.addEventListener("click", deleteOrEdit);

allList.addEventListener("click", deleteOrEdit);

function deleteOrEdit(event){

const targetBtn = event.target;

const entry = targetBtn.parentNode;

if( targetBtn.id == DELETE ){

deleteEntry(entry);

}else if(targetBtn.id == EDIT ){

editEntry(entry);

}

}

function deleteEntry(entry){

ENTRY\_LIST.splice( entry.id, 1);

updateUI();

}

function editEntry(entry){

console.log(entry)

let ENTRY = ENTRY\_LIST[entry.id];

if(ENTRY.type == "income"){

incomeAmount.value = ENTRY.amount;

incomeTitle.value = ENTRY.title;

}else if(ENTRY.type == "expense"){

expenseAmount.value = ENTRY.amount;

expenseTitle.value = ENTRY.title;

}

deleteEntry(entry);

}

function updateUI(){

income = calculateTotal("income", ENTRY\_LIST);

outcome = calculateTotal("expense", ENTRY\_LIST);

balance = Math.abs(calculateBalance(income, outcome));

// DETERMINE SIGN OF BALANCE

let sign = (income >= outcome) ? "$" : "-$";

// UPDATE UI

balanceEl.innerHTML = `<small>${sign}</small>${balance}`;

outcomeTotalEl.innerHTML = `<small>$</small>${outcome}`;

incomeTotalEl.innerHTML = `<small>$</small>${income}`;

clearElement( [expenseList, incomeList, allList] );

ENTRY\_LIST.forEach( (entry, index) => {

if( entry.type == "expense" ){

showEntry(expenseList, entry.type, entry.title, entry.amount, index)

}else if( entry.type == "income" ){

showEntry(incomeList, entry.type, entry.title, entry.amount, index)

}

showEntry(allList, entry.type, entry.title, entry.amount, index)

});

updateChart(income, outcome);

localStorage.setItem("entry\_list", JSON.stringify(ENTRY\_LIST));

}

function showEntry(list, type, title, amount, id){

const entry = ` <li id = "${id}" class="${type}">

<div class="entry">${title}: $${amount}</div>

<div id="edit"></div>

<div id="delete"></div>

</li>`;

const position = "afterbegin";

list.insertAdjacentHTML(position, entry);

}

function clearElement(elements){

elements.forEach( element => {

element.innerHTML = "";

})

}

function calculateTotal(type, list){

let sum = 0;

list.forEach( entry => {

if( entry.type == type ){

sum += entry.amount;

}

})

return sum;

}

function calculateBalance(income, outcome){

return income - outcome;

}

function clearInput(inputs){

inputs.forEach( input => {

input.value = "";

})

}

function show(element){

element.classList.remove("hide");

}

function hide( elements ){

elements.forEach( element => {

element.classList.add("hide");

})

}

function active(element){

element.classList.add("active");

}

function inactive( elements ){

elements.forEach( element => {

element.classList.remove("active");

})

}