

HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Budget App - JavaScript</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="budget-container">
    <div class="app-title">
      <a href="">Code<b>Explained</b></a>
    </div>
    <div class="budget-header">
      <div class="balance">
        <div class="title">
          Balance
        </div>
        <div class="value">
          <small>$</small>0
        </div>
      </div>
      <div class="account">
        <div class="income">
          <div class="title">
            Income
          </div>
          <div class="income-total">
            <small>$</small>0
          </div>
        </div>
        <div class="chart"></div>
        <div class="outcome">
          <div class="title">
            Income
          </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"></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"></div>
        </div>
    </div>
    <div id="all">
        <ul class="list"></ul>
    </div>
</div>

<script src="chart.js"></script>
<script src="budget.js"></script>
</body>
</html>

```

CSS CODE-----

```

@font-face {
    font-family: 'Gilroy';
    src: url('font/Gilroy-Regular.woff') format('woff');
    font-weight: bold;
    font-style: normal;
}

```

```
@font-face {
  font-family: 'Gilroy Bold';
  src: url('font/Gilroy-Bold.woff') format('woff');
  font-weight: bold;
  font-style: normal;
}

body{
  background-color: #222;
  font-family: 'Gilroy';
}

*{
  font-family: 'Gilroy';
  list-style: none;
  text-decoration: none;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
}

/* REMOVE SPINNERS FROM NUMBER INPUT */
/* Chrome, Safari, Edge, Opera */
input::-webkit-outer-spin-button, input::-webkit-inner-spin-button {
  -webkit-appearance: none;
  margin: 0;
}

/* Firefox */
input[type=number] {
  -moz-appearance: textfield;
}

.budget-container{
  position: absolute;
  width: 360px;
  height: 780px;
  top: 50%;
  left: 50%;
  transform: translateX(-50%) translateY(-50%);
  background-color: #34006a;
}

.budget-container .app-title{
  margin-top: 45px;
  margin-left: 20px;
}
```

```
.budget-container .app-title a{
  color: #FFF;
}
.budget-container .app-title a b{
  font-family: 'Gilroy Bold';
}

.budget-header{

}

.budget-header .balance{
  margin-top: 45px;
  margin-left: 20px;
}

.budget-header .balance .title{
  color: #FFF;
  font-size: 1em;
  font-family: 'Gilroy Bold';
}

.budget-header .balance .value{
  color: #FFF;
  font-size: 2em;
  margin-top: 10px;
  font-family: 'Gilroy Bold';
}

.budget-header .account{
  display: block;
  width: 310px;
  margin: 35px auto 0;
  line-height: 1.7em;
  text-align: center;
}

.budget-header .account .income, .chart, .outcome{
  height: 50px;
  display: inline-block;
  width: 100px;
  vertical-align: middle;
}

.budget-header .account .income .title{
  color: #FFF;
```

```

        opacity: 0.5;
    }
    .budget-header .account .income .income-total{
        color: #FFF;
        font-family: 'Gilroy Bold';
    }

    .budget-header .account .outcome .title{
        color: #FFF;
        opacity: 0.5;
    }
    .budget-header .account .outcome .outcome-total{
        color: #f0624d;
        font-family: 'Gilroy Bold';
    }

    /*///////////////// DASHBOARD //////////////////*/
    .budget-dashboard{
        display: block;
        width: 360px;
        height: 470px;
        background-color: #FFF;
        margin-top: 35px;
        border-radius: 30px 30px 0 0;
    }

    .budget-dashboard .dash-title{
        display: inline-block;
        color: #1a0034;
        font-size: 1.7em;
        margin-left: 20px;
        margin-top: 30px;
        font-family: 'Gilroy Bold';
    }

    .toggle{
        display: block;
        margin-top: 30px;
        margin-left: 20px;
    }

    .toggle .tab1, .tab2, .tab3{
        display: inline-block;
        cursor: pointer;
        font-size: 1.1em;
        color: #1a0034;
        font-family: 'Gilroy Bold';
        opacity: 0.4;
    }

```

```

}

.toggle .tab2, .tab3{
    margin-left: 10px;
}

.active{
    opacity : 1 !important;
}

.hide{
    display: none;
}

.budget-dashboard input{
    /* MAKE INPUT SELECTABLE */
    -webkit-user-select: auto;
    -moz-user-select: auto;
    -ms-user-select: auto;
    user-select: auto;
}

/* ////////// ALL ////////// */
.budget-dashboard #all{
    width: 360px;
    height: 339px;
    position: relative;
}

.budget-dashboard #all ul{
    width: 360px;
    height: 285px;
    list-style: none;
    margin: 0;
    padding: 0;
    margin-top: 20px;
    overflow-x: auto;
}

.budget-dashboard #all ul.list li{
    position: relative;
    font-size: 1em;
}

.budget-dashboard #all ul.list li *{
    display: inline-block;
    height: 30px;
    vertical-align: middle;
}

```

```
.budget-dashboard #all ul.list li .entry{
    width:250px;
    height: auto;
    padding-left: 20px;
    margin-bottom: 10px;;
    word-wrap: break-word;
    font-family: 'Gilroy Bold';
}

.budget-dashboard #all ul.list li #edit{
    visibility: hidden;
}

.budget-dashboard #all ul.list li #delete{
    position: absolute;
    width:20px;
    height: 20px;
    right: 20px;
    cursor: pointer;
    background-image: url('icon/trash.png');
    background-size: 20px 20px;
    background-repeat: no-repeat;
}

.budget-dashboard #all ul.list li.income{
}

.budget-dashboard #all ul.list li.expense{
    color: #f0624d;
}

/* ////////// EXPENSE ////////// */
.budget-dashboard #expense{
    width: 360px;
    height: 339px;
    position: relative;
}

.budget-dashboard #expense .input{
    position: absolute;
    display: block;
    width: 360px;
    bottom: 0;
    border-top: 1px solid #ccc;
}

.budget-dashboard #expense .input *{
```

```

display: inline-block;
vertical-align: middle;
}

.budget-dashboard #expense .input input#expense-title-input{
  height: 40px;
  width : 150px;
  padding-left: 10px;
  font-size: 1.2em;
  font-family: 'Gilroy';
  margin: 5px;
  margin-left: 10px;
  border-radius: 10px 0 0 10px;
  border: none;
}

.budget-dashboard #expense .input input#expense-amount-input{
  height: 40px;
  width : 110px;
  padding-left: 10px;
  font-size: 1.2em;
  font-family: 'Gilroy';
  margin: 5px;
  margin-left: -10px;
  border-radius: 0 10px 10px 0;
  border: none;
}

.budget-dashboard #expense .input .add-expense img{
  width: 40px;
  height: 40px;
  cursor: pointer;
}

.budget-dashboard #expense ul{
  width: 360px;
  height: 285px;
  list-style: none;
  margin: 0;
  padding: 0;
  margin-top: 20px;
  overflow-x: auto;
}

.budget-dashboard #expense ul.list li{
  position: relative;
  font-size: 1em;
}

```



```

.budget-dashboard #expense ul.list li *{
    display: inline-block;
    height: 30px;
    vertical-align: middle;
}

.budget-dashboard #expense ul.list li .entry{
    width:250px;
    height: auto;
    padding-left: 20px;
    margin-bottom: 10px;;
    word-wrap: break-word;
    font-family: 'Gilroy Bold';
    color: #f0624d;
}

.budget-dashboard #expense ul.list li #edit{
    position: absolute;
    width:20px;
    height: 20px;
    right: 50px;
    cursor: pointer;
    background-image: url("icon/edit.png");
    background-size: 20px 20px;
    background-repeat: no-repeat;
}

.budget-dashboard #expense ul.list li #delete{
    position: absolute;
    width:20px;
    height: 20px;
    right: 20px;
    cursor: pointer;
    background-image: url('icon/trash.png');
    background-size: 20px 20px;
    background-repeat: no-repeat;
}

/* ////////// INCOME ////////// */
.budget-dashboard #income{
    width: 360px;
    height: 339px;
    position: relative;
}

.budget-dashboard #income .input{
    position: absolute;

```

```
display: block;
width: 360px;
bottom: 0;
border-top: 1px solid #ccc;
}

.budget-dashboard #income .input *{
display: inline-block;
vertical-align: middle;
}

.budget-dashboard #income .input input#income-title-input{
height: 40px;
width : 150px;
padding-left: 10px;
font-size: 1.2em;
font-family: 'Gilroy';
margin: 5px;
margin-left: 10px;
border-radius: 10px 0 0 10px;
border: none;
}

.budget-dashboard #income .input input#income-amount-input{
height: 40px;
width : 110px;
padding-left: 10px;
font-size: 1.2em;
font-family: 'Gilroy';
margin: 5px;
margin-left: -10px;
border-radius: 0 10px 10px 0;
border: none;
}

.budget-dashboard #income .input .add-income img{
width: 40px;
height: 40px;
cursor: pointer;
}

.budget-dashboard #income ul{
width: 360px;
height: 285px;
list-style: none;
margin: 0;
padding: 0;
margin-top: 20px;
```

```

        overflow-x: auto;
    }

    .budget-dashboard #income ul.list li{
        position: relative;
        font-size: 1em;
    }

    .budget-dashboard #income ul.list li *{
        display: inline-block;
        height: 30px;
        vertical-align: middle;
    }

    .budget-dashboard #income ul.list li .entry{
        width:250px;
        height: auto;
        padding-left: 20px;
        margin-bottom: 10px;;
        word-wrap: break-word;
        font-family: 'Gilroy Bold';
    }

    .budget-dashboard #income ul.list li #edit{
        position: absolute;
        width:20px;
        height: 20px;
        right: 50px;
        cursor: pointer;
        background-image: url("icon/edit.png");
        background-size: 20px 20px;
        background-repeat: no-repeat;
    }

    .budget-dashboard #income ul.list li #delete{
        position: absolute;
        width:20px;
        height: 20px;
        right: 20px;
        cursor: pointer;
        background-image: url('icon/trash.png');
        background-size: 20px 20px;
        background-repeat: no-repeat;
    }

```

BUDGET.JS FILE CODE-----

```

// 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);

// HELPERS

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>${sign}</small>${outcome}`;
    incomeTotalEl.innerHTML = `<small>${sign}</small>${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");
    })
}

```

CHART.JS FILE-----

```

// SELECT CHART ELEMENT
const chart = document.querySelector(".chart");

// CREATE CANVAS ELEMMENT
const canvas = document.createElement("canvas");
canvas.width = 50;
canvas.height = 50;

// APPEND CANVAS TO CHART ELEMENT
chart.appendChild(canvas);

// TO DRAW ON CANVAS, WE NEED TO GET CONTEXT OF CANVAS
const ctx = canvas.getContext("2d");

// CHANGE THE LINE WIDTH
ctx.lineWidth = 8;

```



```
// CIRCLE RADIUS
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);
}
```