**Team Budget Planner.**

index.html

<!DOCTYPEhtml>

<htmllang="en">

<head>

<metacharset="UTF-8" />

<metaname="viewport"content="width=device-width, initial-scale=1.0" />

<title>Budget App</title>

<linkrel="stylesheet"href="style.css" />

</head>

<body>

<divclass="container">

<divclass="header">

<h1>Budget App</h1>

</div>

<divclass="budget-section">

<divclass="budget col col-md col-sm">

<h2>budget</h2>

<imgsrc="image/money-bag.svg"width="40"alt="" />

<pclass="amount">$ <spanid="budgetAmount">0</span></p>

</div>

<divclass="expenses col col-md col-sm">

<h2>expenses</h2>

<imgsrc="image/accounting.svg"width="40"alt="" />

<pclass="exp-amount">$ <spanid="expensesAmount">0</span></p>

</div>

<divclass="balance col col-md col-sm">

<h2>balance</h2>

<imgsrc="image/law.svg"width="40"alt="" />

<pclass="amount bala">$ <spanid="balanceAmount">0</span></p>

</div>

</div>

<divid="displayExpenses">

<divclass="budget-list">

<divclass="col">

<h4>expense title</h4>

</div>

<divclass="col">

<h4>expense value</h4>

</div>

<divclass="col"></div>

</div>

<divid="expValue"></div>

</div>

</div>

<divclass="toggle">

<buttonid="myBtn"type="button">+</button>

</div>

<!-- The Modal -->

<divid="myModal"class="modal">

<!-- Modal content -->

<divclass="modal-content">

<divclass="modal-header">

<spanclass="close">&times;</span>

<h2>Budget Form</h2>

</div>

<divclass="modal-body">

<divclass="budget-form"id="budgetform">

<formid="addForm">

<labelfor=""> Make a budget</label><br />

<inputtype="number"id="number" /><br />

<buttontype="submit">Add Budget</button>

</form>

</div>

<divclass="expense-form"id="expense-form">

<formaction=""id="expForm">

<divclass="">

<labelfor="">please enter your expense</label><br />

<inputtype="text"id="expName" />

</div>

<divclass="">

<labelfor="">please enter expense amount</label><br />

<inputtype="number"id="expNumber" />

</div>

<buttontype="submit"id="submitExpen">Add expense</button>

</form>

<buttononclick="callBudget()"id="bug">

Back to add budget ->

</button>

</div>

<divclass="edit-form"id="editForm">

<formaction=""id="saveEdit">

<divclass="">

<labelfor="">Edit your expense</label><br />

<inputtype="text"id="editExpName" />

</div>

<divclass="">

<labelfor="">Edit expense amount</label><br />

<inputtype="number"id="editExpNumber" />

</div>

<buttontype="submit">Save changes</button>

</form>

</div>

</div>

</div>

</div>

<scriptsrc="main.js"></script>

</body>

</html>

main.js

constmodal=document.getElementById("myModal");

constbtn=document.getElementById("myBtn");

constspan=document.getElementsByClassName("close")[0];

btn.onclick=function () {

expName.value = "";

expNumber.value = "";

expenseForm.style.display = "block";

editForm.style.display = "none";

modal.style.display = "block";

};

span.onclick=function () {

modal.style.display = "none";

};

window.onclick=function (event) {

if (event.target == modal) {

modal.style.display = "none";

}

};

constamountInput=document.getElementById("number");

constaddForm=document.getElementById("addForm");

constbudgetAmount=document.getElementById("budgetAmount");

constbalanceAmount=document.getElementById("balanceAmount");

consteditForm=document.getElementById("editForm");

constsaveEdit=document.getElementById("saveEdit");

consteditExpValue=document.getElementById("editExpValue");

consteditExpNumber=document.getElementById("editExpNumber");

constexpForm=document.getElementById("expForm");

constexpensesAmount=document.getElementById("expensesAmount");

constexpValue=document.getElementById("expValue");

constdisplayExpenses=document.getElementById("displayExpenses");

constexpenseForm=document.getElementById("expense-form");

constbudgetform=document.getElementById("budgetform");

letexpName=document.getElementById("expName");

letexpNumber=document.getElementById("expNumber");

letid=0;

letdetails= [];

functiongetBudgetAmount(amount) {

if (!amount) {

amountInput.style.border = "1px solid #b80c09";

amountInput.placeholder = "input can not be empty";

amountInput.style.color = "#b80c09";

setTimeout(() => {

amountInput.style.color = "#495057";

amountInput.style.border = "1px solid gray";

}, 3000);

} else {

budgetAmount.innerText = amount;

balanceAmount.innerText = amount;

expenseForm.style.display = "block";

budgetform.style.display = "none";

editForm.style.display = "none";

amountInput.value = "";

}

}

functionaddExpenses(name, number) {

if (!name.length || !number.length) {

expName.style.border = "1px solid #b80c09";

expName.placeholder = "input can not be empty";

expName.style.color = "#b80c09";

expNumber.style.border = "1px solid #b80c09";

expNumber.placeholder = "input can not be empty";

expNumber.style.color = "#b80c09";

setTimeout(() => {

expName.style.color = "#495057";

expName.style.border = "1px solid gray";

expName.placeholder = "input can not be empty";

expNumber.placeholder = "input can not be empty";

expNumber.style.border = "1px solid gray";

expNumber.style.color = "#495057";

}, 3000);

} else {

constuserExp = {

id: id,

name: name,

number: parseInt(number),

};

details.push(userExp);

displayExp(details);

id++;

expName.value = "";

expNumber.value = "";

}

}

functiondisplayExp(details) {

expValue.innerHTML = null;

for (i = 0; i<details.length; i++) {

expValue.innerHTML += `

<div class="expValue" id="${details[i].id}">

<div id="expTitleName" class="exp"><p>${details[i].name}</p></div>

<div id="expValueAmount" class="exp"><p><span>$ </span>${details[i].number}</p></div>

<div id="edite\_delete">

<p>

<button id="${details[i].id}" onclick="editExpDetails(${details[i].id})"><imgsrc="image/edit.svg" width="15" alt="" /></button>

<button id="${details[i].id}" onclick="delExpenseDetails(${details[i].id})"><imgsrc="image/trash.svg" width="15" alt="" /></button>

</p>

</div>

</div>

`;

}

calcExpenses();

displayExpenses.style.display = "block";

}

functioncalcExpenses() {

lettotalExp = 0;

for (i = 0; i<details.length; i++) {

totalExp = details[i].number + totalExp;

}

expensesAmount.innerText = totalExp;

updateBalance();

}

functionupdateBalance() {

balanceAmount.innerText =

parseInt(budgetAmount.innerText) - parseInt(expensesAmount.innerText);

}

functiondelExpenseDetails(id) {

letindex = details.findIndex((item) =>item.id === id);

details.splice(index, 1);

displayExp(details);

}

functioneditExpDetails(id) {

expenseForm.style.display = "none";

budgetform.style.display = "none";

editForm.style.display = "block";

details.findIndex((item) => {

if (item.id === id) {

editExpName.value = item.name;

editExpNumber.value = item.number;

saveEdit.children[2].id = item.id;

modal.style.display = "block";

}

});

}

functiongetExpValue(editExpName, editExpNumber, id) {

edited = details.findIndex((obj) =>obj.id == id);

details[edited].name = editExpName;

details[edited].number = parseInt(editExpNumber);

displayExp(details);

}

functioncallBudget() {

budgetform.style.display = "block";

expenseForm.style.display = "none";

}

saveEdit.addEventListener("submit", (e) => {

e.preventDefault();

getExpValue(editExpName.value, editExpNumber.value, saveEdit.children[2].id);

});

expForm.addEventListener("submit", (e) => {

e.preventDefault();

addExpenses(expName.value, expNumber.value);

});

addForm.addEventListener("submit", (e) => {

e.preventDefault();

getBudgetAmount(amountInput.value);

});

style.css

@font-face {

font-family: "General";

src: url("./font/Nunito-ExtraLight.ttf");

}

@font-face {

font-family: "heading";

src: url("./font/Nunito-Bold.ttf");

}

body {

margin: 0;

padding: 0;

width: 100%;

font-family: "General";

height: 100vh;

background-color: #f5f5f5;

}

.container {

min-height: 80vh;

}

.header {

height: 60px;

width: 100%;

}

h1 {

margin: 0;

text-align: center;

font-family: "heading";

padding-top: 10px;

}

.budget-section {

display: flex;

justify-content: space-around;

align-items: flex-start;

width: 100%;

min-height: 160px;

font-weight: 900;

}

.budget-list {

display: flex;

justify-content: space-around;

align-items: flex-start;

width: 100%;

height: 70px;

font-size: 20px;

}

.exp-amount {

color: #b80c09;

}

.amount {

color: #317b22;

}

.col {

width: 40%;

text-align: center;

text-transform: capitalize;

}

.colp {

font-size: 40px;

padding-top: 5px;

margin: 0;

}

.toggle {

position: fixed;

right: 0;

bottom: 18px;

}

.togglebutton {

border: none;

border-radius: 600px;

background: #04458f;

padding: 14px20px;

color: #ffffff;

font-size: 30px;

margin: 10px50px00;

cursor: pointer;

}

/\* modal form \*/

.budget-form {

width: 100%;

min-height: 200px;

}

form {

margin: 10px020px10px;

min-height: 100px;

}

input {

width: 60%;

padding: 8px0;

border-radius: 6px;

border: 1pxsolidgray;

margin: 20px0;

padding-left: 5px;

color: #495057;

}

.budget-list.expp {

color: #b80c09;

font-size: 18px;

}

#displayExpenses {

width: 100%;

display: none;

}

.expValue {

display: flex;

justify-content: space-around;

height: 40px;

font-weight: 900;

text-align: center;

}

#expTitleName {

width: 40%;

}

#expValueAmount {

width: 40%;

}

#edite\_delete {

width: 40%;

}

#edite\_deleteimg {

cursor: pointer;

}

#edite\_deletebutton {

margin-left: 1px;

background: none;

border: none;

}

.budget-formbutton {

background: rgb(1, 83, 1);

border: none;

border-radius: 6px;

color: #ffffff;

padding: 10px40px;

cursor: pointer;

text-transform: capitalize;

font-size: 16px;

font-weight: 400;

}

#bug {

font-size: 14px;

background: none;

color: #04458f;

margin: 0px0px15px10px;

text-align: justify;

padding: 0;

}

#expense-form {

width: 100%;

display: none;

}

#editForm {

display: none;

width: 100%;

}

#editFormbutton {

background: #b80c09;

border: none;

border-radius: 6px;

color: #ffffff;

padding: 10px30px;

cursor: pointer;

text-transform: capitalize;

font-size: 16px;

font-weight: 400;

}

.expense-formbutton {

background: #b80c09;

border: none;

border-radius: 6px;

color: #ffffff;

padding: 10px30px;

cursor: pointer;

text-transform: capitalize;

font-size: 16px;

font-weight: 400;

}

/\* The Modal (background) \*/

.modal {

display: block;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

animation-name: fadeIn;

animation-duration: 0.4s;

text-transform: capitalize;

font-weight: 600;

}

/\* Modal Content \*/

.modal-content {

position: fixed;

top: 20%;

left: 25%;

background-color: #fefefe;

width: 50%;

min-height: 40px;

animation-name: slideIn;

animation-duration: 0.4s;

}

/\* The Close Button \*/

.close {

color: white;

float: right;

font-size: 28px;

font-weight: bold;

cursor: pointer;

color: #000;

}

.modal-header {

padding: 2px16px;

border-bottom: 2pxsolidgray;

}

.modal-body {

padding: 2px16px;

}

/\* Add Animation \*/

@keyframesslideIn {

from {

top: -300px;

opacity: 0;

}

to {

top: 0;

opacity: 1;

}

}

@keyframesfadeIn {

from {

opacity: 0;

}

to {

opacity: 1;

}

}