SimpliLearn

Assesment project 1: budget application

Source code

Index.html

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Budget App</title>
    <link rel="stylesheet" href="index.css" />
  </head>
  <body>
    <div class="container">
      <div class="header">
        <h1>Budget App</h1>
      </div>
      <div class="budget-section">
        <div class="budget col col-md col-sm">
          <h2>budget</h2>
          <img src="image/money-bag.svg" width="40" alt="" />
          <p class="amount">$ <span id="budgetAmount">0</span></p>
        </div>
        <div class="expenses col col-md col-sm">
          <h2>expenses</h2>
          <img src="image/accounting.svg" width="40" alt="" />
          <p class="exp-amount">$ <span id="expensesAmount">0</span></p>
        </div>
        <div class="balance col col-md col-sm">
          <h2>balance</h2>
          <img src="image/law.svg" width="40" alt="" />
          <p class="amount bala">$ <span id="balanceAmount">0</span></p>
        </div>
      </div>
    </div>
```

```html
<div id="displayExpenses">
  <div class="budget-list">
    <div class="col">
      <h4>expense title</h4>
    </div>
    <div class="col">
      <h4>expense value</h4>
    </div>
    <div class="col"></div>
  </div>
  <div id="expValue"></div>
</div>
</div>
<div class="toggle">
  <button id="myBtn" type="button">+</button>
</div>

<!-- The Modal -->
<div id="myModal" class="modal">
  <!-- Modal content -->
  <div class="modal-content">
    <div class="modal-header">
      <span class="close">&times;</span>
      <h2>Budget Form</h2>
    </div>
    <div class="modal-body">
      <div class="budget-form" id="budgetform">
        <form id="addForm">
          <label for=""> Make a budget</label> <br />
          <input type="number" id="number" /> <br />
          <button type="submit">Add Budget</button>
        </form>
      </div>

      <div class="expense-form" id="expense-form">
        <form action="" id="expForm">
          <div class="">
            <label for="">please enter your expense</label> <br />
            <input type="text" id="expName" />
          </div>
```

```html
        <div class="">
          <label for="">please enter expense amount</label> <br />
          <input type="number" id="expNumber" />
        </div>
        <button type="submit" id="submitExpen">Add expense</button>
      </form>
      <button onclick="callBudget()" id="bug">
        Back to add budget ->
      </button>
    </div>

    <div class="edit-form" id="editForm">
      <form action="" id="saveEdit">
        <div class="">
          <label for="">Edit your expense</label> <br />
          <input type="text" id="editExpName" />
        </div>
        <div class="">
          <label for="">Edit expense amount</label> <br />
          <input type="number" id="editExpNumber" />
        </div>
        <button type="submit">Save changes</button>
      </form>
    </div>
      </div>
     </div>
    </div>
    <script src="index.js"></script>
  </body>
</html>
```

Index.css

```css
@font-face {
font-family: "General";
   src: url("./font/Nunito-ExtraLight.ttf");
```

```css
    }
    @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;
```

```css
  font-size: 20px;
}
.exp-amount {
color: #b80c09;
}
.amount {
 color: #317b22;
}
.col {
 width: 40%;
 text-align: center;
 text-transform: capitalize;
}
.col p {
 font-size: 40px;
 padding-top: 5px;
 margin: 0;
}
.toggle {
 position: fixed;
 right: 0;
 bottom: 18px;
}
.toggle button {
border: none;
 border-radius: 600px;
 background: #04458f;
 padding: 14px 20px;
 color: #ffffff;
 font-size: 30px;
 margin: 10px 50px 0 0;
 cursor: pointer;
}
/* modal form */
.budget-form {
 width: 100%;
 min-height: 200px;
}
form {
 margin: 10px 0 20px 10px;
```

```css
  min-height: 100px;
  }
  input {
  width: 60%;
   padding: 8px 0;
   border-radius: 6px;
   border: 1px solid gray;
   margin: 20px 0;
   padding-left: 5px;
   color: #495057;
  }
  .budget-list .exp p {
   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_delete img {
   cursor: pointer;
  }
  #edite_delete button {
  margin-left: 1px;
```

```css
  background: none;
  border: none;
  }
.budget-form button {
  background: rgb(1, 83, 1);
  border: none;
  border-radius: 6px;
  color: #ffffff;
  padding: 10px 40px;
  cursor: pointer;
  text-transform: capitalize;
  font-size: 16px;
  font-weight: 400;
  }
#bug {
  font-size: 14px;
  background: none;
  color: #04458f;
  margin: 0px 0px 15px 10px;
  text-align: justify;
padding: 0;
  }
#expense-form {
  width: 100%;
  display: none;
  }
#editForm {
  display: none;
  width: 100%;
  }
#editForm button {
  background: #b80c09;
  border: none;
  border-radius: 6px;
  color: #ffffff;
  padding: 10px 30px;
  cursor: pointer;
  text-transform: capitalize;
  font-size: 16px;
  font-weight: 400;
```

```css
    }
    .expense-form button {
    background: #b80c09;
     border: none;
     border-radius: 6px;
     color: #ffffff;
     padding: 10px 30px;
     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;
    }
```

```css
/* The Close Button */
.close {
 color: white;
 float: right;
 font-size: 28px;
 font-weight: bold;
cursor: pointer;
 color: #000;
}
.modal-header {
 padding: 2px 16px;
border-bottom: 2px solid gray;
}
.modal-body {
padding: 2px 16px;
}

/* Add Animation */
@keyframes slideIn {
from {
 top: -300px;
opacity: 0;
}
 to {
 top: 0;
opacity: 1;
 }
}

@keyframes fadeIn {
from {
opacity: 0;
}
 to {
opacity: 1;
 }
}
```

Index.js

```javascript
const modal = document.getElementById("myModal");
const btn = document.getElementById("myBtn");
const span = 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";
  }
};

const amountInput = document.getElementById("number");
const addForm = document.getElementById("addForm");
const budgetAmount = document.getElementById("budgetAmount");
const balanceAmount = document.getElementById("balanceAmount");

const editForm = document.getElementById("editForm");
const saveEdit = document.getElementById("saveEdit");
const editExpValue = document.getElementById("editExpValue");
const editExpNumber = document.getElementById("editExpNumber");

const expForm = document.getElementById("expForm");
const expensesAmount = document.getElementById("expensesAmount");
const expValue = document.getElementById("expValue");
```

```javascript
const displayExpenses = document.getElementById("displayExpenses");
const expenseForm = document.getElementById("expense-form");
const budgetform = document.getElementById("budgetform");

let expName = document.getElementById("expName");
let expNumber = document.getElementById("expNumber");
let id = 0;
let details = [];

function getBudgetAmount(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 = "";
  }
}

function addExpenses(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";
```

```javascript
      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 {
    const userExp = {
      id: id,
      name: name,
      number: parseInt(number),
    };
    details.push(userExp);
    displayExp(details);
    id++;
    expName.value = "";
    expNumber.value = "";
  }
}

function displayExp(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(${details[i].id})"> <img
src="image/edit.svg" width="15" alt=""  /></button>
 <button id="${details[i].id}" onclick="delExpenseDetails(${details[i].id})"><img
src="image/trash.svg" width="15" alt="" /></button>
 </p>
 </div>
 </div>
  `;
  }
```

```javascript
  calcExpenses();
  displayExpenses.style.display = "block";
}

function calcExpenses() {
  let totalExp = 0;
  for (i = 0; i < details.length; i++) {
    totalExp = details[i].number + totalExp;
  }
  expensesAmount.innerText = totalExp;
  updateBalance();
}

function updateBalance() {
  balanceAmount.innerText =
    parseInt(budgetAmount.innerText) - parseInt(expensesAmount.innerText);
}

function delExpenseDetails(id) {
  let index = details.findIndex((item) => item.id === id);
  details.splice(index, 1);
  displayExp(details);
}

function editExpDetails(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";
    }
  });
}

function getExpValue(editExpName, editExpNumber, id) {
  edited = details.findIndex((obj) => obj.id == id);
```

```javascript
  details[edited].name = editExpName;
  details[edited].number = parseInt(editExpNumber);
  displayExp(details);
}

function callBudget() {
  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);
});
```