EMP No: 2577108
Name: Pramod

Source Code: Team Budget Planner

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link
      rel="stylesheet"
      href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
      integrity="sha384-
UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="style.css" />
    <title>Budget App</title>
  </head>

  <body>
    <div class="wrapper">
      <h1>budget app</h1>

      <form class="form-budget">
        <label for="">Please Enter Your Budget</label> <br />
        <input type="number" id="budget-field" /><br />
        <div class="warning-budget d-none">
          value cannot be empty or negative.
        </div>
        <button class="btn" id="calculate-btn">Calculate</button>
      </form>

      <form class="form-expense">
        <label for="">Please Enter Your Expense Name</label> <br />
        <input type="text" id="expense-name-field" /> <br />
        <div class="warning-expense d-none">Expense name cannot be empty.</div>
        <label for="">Please Enter Expense Amount</label> <br />
        <input type="number" id="expense-value-field" /><br />
        <button class="btn" id="add-expense-btn">Add Expense</button>
      </form>
```

```
   <div class="amts">
    <ul class="amts-head">
     <li>Budget</li>
     <li>Expenses</li>
     <li>Balance</li>
</ul>

    <ul class="amts-icon">
     <li><i class="fas fa-money-bill-alt"></i></li>
     <li><i class="far fa-credit-card"></i></li>
     <li><i class="fas fa-money-bill-alt"></i></li>
    </ul>
    <ul class="amts-value">
     <li class="clr-green" id="budget"><span>RS</span>0</li>

     <li class="clr-red" id="expense"><span>RS</span>0</li>

     <li id="balance"><span>RS</span>0</li>
    </ul>
   </div>
   <div class="expenses">
    <ul class="expenses-head">
     <li>Expense title</li>
     <li>expense value</li>
     <li></li>
    </ul>
   </div>
  </div>
  <script src="main.js"></script>
 </body>
</html>
```

Main.js

```
window.onload = function() {
  let balance = document.querySelector("#balance");
  let expense = document.querySelector("#expense");
  let budget = document.querySelector("#budget");
  let warningBudget = document.querySelector(".warning-budget");
  let warningExpense = document.querySelector(".warning-expense");
  let budgetField = document.querySelector("#budget-field");
  let expenseNameField = document.querySelector("#expense-name-field");
  let expenseValueField = document.querySelector("#expense-value-field");
  let calculateBtn = document.querySelector("#calculate-btn");
  let addExpBtn = document.querySelector("#add-expense-btn");
  let expenseList = document.querySelector(".expenses");

  let state = {
    list: [],
    amount: {
      budgetAmt: 0,
      expenseAmt: 0,
      balanceAmt: 0
    }
  };

  function updateExpAmt() {
    state.amount.expenseAmt = state.list.reduce((total, obj) => {
      return total + parseInt(obj.amt);
    }, 0);
  }

  function updateExpenses(e) {
    e.preventDefault();
    if (!expenseNameField.value) {
      warningExpense.classList.remove("d-none");
      setTimeout(() => {
        warningExpense.classList.add("d-none");
      }, 1500);
    } else {
      //updating state
      state.list.push({
        id: randomNum(),
        name: expenseNameField.value,
        amt: parseInt(expenseValueField.value) || 0
      });

      updateExpAmt();

      //resetting entry fields
      expenseNameField.value = "";
      expenseValueField.value = "";
      //render updated amount and expenses
      renderAmounts();
      renderExpenses();
    }
  }

  function updateBudget(e) {
    e.preventDefault();
    if (!budgetField.value) {
      warningBudget.classList.remove("d-none");
      setTimeout(() => {
        warningBudget.classList.add("d-none");
      }, 1500);
    } else {
      state.amount.budgetAmt = parseInt(budgetField.value);
      renderAmounts();
      budgetField.value = "";
    }
  }

  addExpBtn.addEventListener("click", updateExpenses);
  calculateBtn.addEventListener("click", updateBudget);

  //edit and delete button functionality

  function deleteItem(id) {
    let newList = state.list.filter(obj => obj.id !== id);
    state.list = newList;
    updateExpAmt();
    renderAmounts();
    renderExpenses();
  }

  function editItem(id) {
    let newList = state.list.filter(obj => obj.id !== id);
    let item = state.list.filter(obj => obj.id == id)[0];

    expenseNameField.value = item.name;
    expenseValueField.value = item.amt;

    state.list = newList;
    updateExpAmt();
    renderAmounts();
    renderExpenses();
```

```
   }

   //function render
   function renderExpenses() {
     while (expenseList.childNodes.length > 2) {
       expenseList.removeChild(expenseList.lastChild);
     }

     state.list.forEach(obj => {
       let ul = document.createElement("ul");
       ul.innerHTML = `
         <li>${obj.name}</li>
         <li>${obj.amt}</li>
         <li>
            <button class="btn-edit">
            <i class="fas fa-edit"></i>
            </button>
            <button class="btn-trash" id=${obj.id} onclick="greetings()">
            <i class="fas fa-trash-alt"></i>
            </button>
         </li>
         `;
ul.querySelector(".btn-trash").onclick = () => deleteItem(obj.id);
    ul.querySelector(".btn-edit").onclick = () => editItem(obj.id);
    ul.setAttribute("class", "expenses-row");
    expenseList.appendChild(ul);
  });
}

function renderAmounts() {
  budget.childNodes[1].nodeValue = state.amount.budgetAmt;
  expense.childNodes[1].nodeValue = state.amount.expenseAmt;
  state.amount.balanceAmt =
    state.amount.budgetAmt - state.amount.expenseAmt;
  balance.childNodes[1].nodeValue = state.amount.balanceAmt;

  if (state.amount.balanceAmt < 0) {
    balance.classList.remove("clr-green");
    balance.classList.add("clr-red");
  } else {
    balance.classList.remove("clr-red");
    balance.classList.add("clr-green");
  }
}
```

```
renderExpenses();
renderAmounts();

//--------utility function------------------------------------------------
function randomNum() {
  return Math.random() * 1000;
}
};
```

style.css

```
@import url("https://fonts.googleapis.com/css?family=Roboto");
* {
    box-sizing: border-box;
}

.d-none {
    display: none;
}

body {
    padding: 0 2rem;
    font-family: roboto;
}

.wrapper {
    display: grid;
    grid-template-columns: 1fr 2fr;
    height: auto;
}

h1 {
    grid-column: 1/-1;
    text-transform: uppercase;
    font-weight: 100;
}

.warning-budget,
```

```css
.warning-expense {
  border: 2px solid #ef2929;
  border-radius: 3px;
  padding: 0.3rem;
  background: rgba(248, 166, 166, 0.3);
  margin-bottom: 0.5rem;
}

form {
  grid-column: 1/2;
  width: 100%;
  border: 2px solid;
  border-radius: 4px;
  padding: 0.7rem;
  margin-bottom: 2.5rem;
}

input {
  width: 100%;
  border-radius: 3px;
  border: 1px solid grey;
  margin: 0.5rem 0 0.7rem 0;
  font-size: 1.1rem;
  padding: 0.2rem;
  box-sizing: border-box;
  background: #fff;
}

.btn {
  all: unset;
  font-size: 0.8rem;
  text-align: center;
  border: 2px solid;
  border-radius: 3px;
  padding: 0.3rem 0.5rem;
  transition: box-shadow 0.3s;
  font-weight: 400;
}

.btn:hover {
  box-shadow: 0 0 10px blue;
}

.form-budget {
    background: rgb(193, 254, 211, 0.1);
    border-color: rgb(84, 165, 3);
}

.form-budget button {
    border-color: rgb(84, 165, 3);
    color: rgb(84, 165, 3);
}

.form-expense {
    background: rgba(254, 193, 193, 0.1);
    border-color: #ef2929;
    align-self: flex-start;
    height: auto;
}

.form-expense button {
    border-color: #ef2929;
    color: #ef2929;
}

li {
    list-style-type: none;
}

.amts {
    grid-column: 2/-1;
    grid-row: 2/3;
}

.amts ul {
    margin: 0;
    padding: 0;
    display: flex;
}

.amts li {
    flex: 1;
    text-align: center;
}

.amts-head li {
    font-size: 1.6rem;
    font-weight: 400;
    letter-spacing: 1px;
    text-transform: uppercase;
}

.amts-icon li {
    font-size: 3rem;
    color: #3465a4;
    padding: 0.5rem;
}

.amts-value span {
    align-self: center;
    font-size: 2.5rem;
}

.amts-value li {
    font-size: 2.5rem;
}

.clr-red {
    color: #e01616;
}
.clr-green {
    color: rgb(75, 148, 2);
}

.expenses {
    height: auto;
    max-height: 20rem;
    overflow-y: auto;
}

.expenses ul {
    display: flex;
    padding: 0;
    align-items: center;
}

.expenses-head {
    font-size: 1.1rem;
    text-transform: capitalize;
}

.expenses-row li {
    font-size: 1.05rem;
    color: #e01616;
}

.expenses li {
    flex: 1;
    text-align: center;
}

.expenses-row li:first-child::before {
    content: "- ";
    margin-left: -20px;
}
.expenses-row li:first-child {
    margin-left: 20px;
}

.btn-edit {
    background: none;
    border: none;
    font-size: 1.1rem;
    color: #3465a4;
}
.btn-trash {
    background: none;
    border: none;
    color: #e01616;
    font-size: 1.1rem;
}
```