

Team Budget Planner Source code

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <title>Team Budget Planner</title>
    <!-- Font Awesome Icons -->
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.1.1/css/all.min.css"
    />
    <!-- Google Fonts -->
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&di
splay=swap"
      rel="stylesheet"
    />
    <!-- Stylesheet -->
    <link rel="stylesheet" href="style.css" />
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  </head>
  <body>
    <nav class="navbar">
      <div class="navbar-container">
```

```
<h1 class="navbar-heading">Team Budget Planner</h1>
</div>
</nav>
```

```
<div class="wrapper">
  <div class="container">
    <div class="sub-container">
      <!-- Budget -->
      <div class="total-amount-container">
        <h3>Estimated Budget</h3>
        <p class="hide error" id="budget-error">
          Value cannot be empty or negative
        </p>
        <input
          type="number"
          id="total-amount"
          placeholder="Enter Estimeted Team Budget"
        />
        <button          class="submit"          id="total-amount-button">Set
Budget</button>
      </div>
      <!-- Expenditure -->
      <div class="user-amount-container">
        <h3>Expenses</h3>
        <p class="hide error" id="product-title-error">
          Values cannot be empty
        </p>
```

```
<input
  type="text"
  class="product-title"
  id="product-title"
  placeholder="Describe professional deal"
/>
<input
  type="number"
  id="user-amount"
  placeholder="Amount needed"
/>
<button class="submit" id="check-amount">Add Expense</button>
</div>
</div>
```

```
<!-- List -->
<div class="list">
  <h3>Expense List</h3>
  <div class="list-container" id="list"></div>
</div>
</div>
<!-- Output -->
<div class="output-container flex-space">
  <table class="output-table">
    <tr>
      <td>
        <p>Budget</p>
        <span id="amount">0</span>
```

```
</td>
<td>
  <p>Expenses</p>
  <span id="expenditure-value">0</span>
</td>
<td>
  <p>Balance</p>
  <span id="balance-amount">0</span>
</td>
</tr>
</table>
```

```
<div class="pie-chart-container">
  <canvas id="pie-chart"></canvas>
</div>
</div>
<!-- Script -->
<script src="script.js"></script>
</body>
</html>
```

Style.css

```
* {
padding: 0;
margin: 0;
box-sizing: border-box;
font-family: "Poppins", sans-serif;
```

```
}  
  
body {  
  background-color: #f7f9fd;  
}  
  
.wrapper {  
  width: 90%;  
  font-size: 16px;  
  max-width: 43.75em;  
  margin: 1em auto;  
}  
  
.output-heading {  
  margin-bottom: 15px;  
}  
  
.navbar {  
  background-color: #333;  
  color: #fff;  
  padding: 10px;  
}  
  
.navbar-container {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}  
  
.navbar-heading {  
  margin: 0;  
}
```

```
.container {  
  width: 100%;  
}  
.sub-container {  
  width: 100%;  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  gap: 3em;  
}  
.flex {  
  display: flex;  
  align-items: center;  
}  
.flex-space {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
}  
.wrapper h3 {  
  color: #363d55;  
  font-weight: 500;  
  margin-bottom: 0.6em;  
}  
.container input {  
  display: block;  
  width: 100%;  
  padding: 0.6em 0.3em;  
  border: 1px solid #d0d0d0;
```

```
border-radius: 0.3em;
color: #414a67;
outline: none;
font-weight: 400;
margin-bottom: 0.6em;
}
.container input:focus {
border-color: #587ef4;
}
.total-amount-container,
.user-amount-container {
background-color: #ffffff;
padding: 1.25em 0.9em;
border-radius: 0.3em;
box-shadow: 0 0.6em 1.2em rgba(28, 0, 80, 0.06);
}
.output-container {
background-color: #587ef4;
color: #ffffff;
border-radius: 0.3em;
box-shadow: 0 0.6em 1.2em rgba(28, 0, 80, 0.06);
margin: 2em 0;
padding: 1.2em;
display: flex;
flex-wrap: wrap;
}
```

```
.pie-chart-container {  
    flex: 1 1 100%;  
    margin-top: 1.5em;  
    text-align: center;  
}
```

```
.output-container p {  
    font-weight: 500;  
    margin-bottom: 0.6em;  
}
```

```
.output-container span {  
    display: block;  
    text-align: center;  
    font-weight: 400;  
    color: #e5e5e5;  
}
```

```
.submit {  
    font-size: 1em;  
    margin-top: 0.8em;  
    background-color: #587ef4;  
    border: none;  
    outline: none;  
    color: #ffffff;  
    padding: 0.6em 1.2em;  
    border-radius: 0.3em;  
    cursor: pointer;  
}
```

```
.list {
```



```
background-color: #ffffff;
padding: 1.8em 1.2em;
box-shadow: 0 0.6em 1.2em rgba(28, 0, 80, 0.06);
border-radius: 0.6em;
}

.sublist-content {
width: 100%;
border-left: 0.3em solid #587ef4;
margin-bottom: 0.6em;
padding: 0.5em 1em;
display: grid;
grid-template-columns: 3fr 2fr 1fr 1fr;
}

.product {
font-weight: 500;
color: #363d55;
}

.amount {
color: #414a67;
margin-left: 20%;
}

.icons-container {
width: 5em;
margin: 1.2em;
align-items: center;
}

.product-title {
margin-bottom: 1em;
```

```
}  
.hide {  
  display: none;  
}  
.error {  
  color: #ff465a;  
}  
.edit {  
  margin-left: auto;  
}  
.edit,  
.delete {  
  background: transparent;  
  cursor: pointer;  
  margin-right: 1.5em;  
  border: none;  
  color: #587ef4;  
}  
@media screen and (max-width: 600px) {  
  .wrapper {  
    font-size: 14px;  
  }  
  .sub-container {  
    grid-template-columns: 1fr;  
    gap: 1em;  
  }  
}  
@media screen and (min-width: 600px) {
```

```
.output-container {  
  flex-wrap: nowrap;  
}
```

```
.pie-chart-container {  
  flex: 1 1 auto;  
  margin-top: 0;  
  text-align: left;  
}  
}
```

Script.js

```
// Global variables for the pie chart
```

```
let pieChart = null;
```

```
let pieChartContainer = document.getElementById("pie-chart");
```

```
// Function to create the pie chart
```

```
const createPieChart = (data, labels, heading) => {
```

```
  if (pieChart) {
```

```
    pieChart.destroy(); // Destroy the previous chart if it exists
```

```
  }
```

```
  pieChart = new Chart(pieChartContainer, {
```

```
    type: "pie",
```

```
    data: {
```

```
      labels: labels,
```

```
      datasets: [
```

```
        {
```

```

    data: data,
    backgroundColor: ["#587ef4", "#ff465a", "#34eb86", "#ffc107",
"#8c54ff"],
    },
  ],
},
options: {
  responsive: true,
  legend: {
    position: "bottom",
  },
  title: {
    display: true,
    text: heading,
  },
},
});
};

```

// Function to update the pie chart based on the budget status

```

const updatePieChart = (budget, expenses) => {
  const balance = budget - expenses;
  const data = [expenses, balance > 0 ? balance : 0];
  const labels = ["Expenses", "Surplus Budget"];

  if (balance < 0) {
    labels[1] = "Deficit Budget";
  } else if (balance === 0) {

```

```
    labels[1] = "Balanced Budget";  
  }
```

```
    createPieChart(data, labels, "Budget Status");  
  };
```

```
let totalAmount = document.getElementById("total-amount");  
let userAmount = document.getElementById("user-amount");  
const checkAmountButton = document.getElementById("check-amount");  
const totalAmountButton = document.getElementById("total-amount-  
button");  
const productTitle = document.getElementById("product-title");  
const errorMessage = document.getElementById("budget-error");  
const productTitleError = document.getElementById("product-title-error");  
const productCostError = document.getElementById("product-cost-error");  
const amount = document.getElementById("amount");  
const expenditureValue = document.getElementById("expenditure-value");  
const balanceValue = document.getElementById("balance-amount");  
const list = document.getElementById("list");  
let tempAmount = 0;  
//Set Budget Part  
totalAmountButton.addEventListener("click", () => {  
  tempAmount = totalAmount.value;  
  //empty or negative input  
  if (tempAmount === "" || tempAmount < 0) {  
    errorMessage.classList.remove("hide");  
  } else {  
    errorMessage.classList.add("hide");
```

```
//Set Budget
amount.innerHTML = tempAmount;
//Set Balance
balanceValue.innerText = tempAmount - expenditureValue.innerText;
//Clear Input Box
totalAmount.value = "";
}

// Update the pie chart after setting the budget
updatePieChart(tempAmount, parseInt(expenditureValue.innerText));

});

//Function To Disable Edit and Delete Button
const disableButtons = (bool) => {
  let editButtons = document.getElementsByClassName("edit");
  Array.from(editButtons).forEach((element) => {
    element.disabled = bool;
  });
};

//Function To Modify List Elements
const modifyElement = (element, edit = false) => {
  let parentDiv = element.parentElement;
  let currentBalance = balanceValue.innerText;
  let currentExpense = expenditureValue.innerText;
  let parentAmount = parentDiv.querySelector(".amount").innerText;
  if (edit) {
    let parentText = parentDiv.querySelector(".product").innerText;
    productTitle.value = parentText;
```

```

    userAmount.value = parentAmount;
    disableButtons(true);
}

balanceValue.innerText = parseInt(currentBalance) +
parseInt(parentAmount);
expenditureValue.innerText =
    parseInt(currentExpense) - parseInt(parentAmount);
parentDiv.remove();
};

//Function To Create List
const listCreator = (expenseName, expenseValue) => {
    let sublistContent = document.createElement("div");
    sublistContent.classList.add("sublist-content", "flex-space");
    list.appendChild(sublistContent);
    sublistContent.innerHTML = `<p class="product">${expenseName}</p><p
class="amount">${expenseValue}</p>`;
    let editButton = document.createElement("button");
    editButton.classList.add("fa-solid", "fa-pen-to-square", "edit");
    editButton.style.fontSize = "1.2em";
    editButton.addEventListener("click", () => {
        modifyElement(editButton, true);
    });
    let deleteButton = document.createElement("button");
    deleteButton.classList.add("fa-solid", "fa-trash-can", "delete");
    deleteButton.style.fontSize = "1.2em";
    deleteButton.addEventListener("click", () => {
        modifyElement(deleteButton);
    });
};

```

```
sublistContent.appendChild(editButton);
sublistContent.appendChild(deleteButton);
document.getElementById("list").appendChild(sublistContent);
};

//Function To Add Expenses
checkAmountButton.addEventListener("click", () => {
  //empty checks
  if (!userAmount.value || !productTitle.value) {
    productTitleError.classList.remove("hide");
    return false;
  }
  //Enable buttons
  disableButtons(false);
  //Expense
  let expenditure = parseInt(userAmount.value);
  //Total expense (existing + new)
  let sum = parseInt(expenditureValue.innerText) + expenditure;
  expenditureValue.innerText = sum;
  //Total balance(budget - total expense)
  const totalBalance = tempAmount - sum;
  balanceValue.innerText = totalBalance;
  //Create list
  listCreator(productTitle.value, userAmount.value);
  //Empty inputs
  productTitle.value = "";
  userAmount.value = "";

  // Update the pie chart after adding expenses
```



```
updatePieChart(tempAmount, parseInt(expenditureValue.innerText));
```

```
});
```