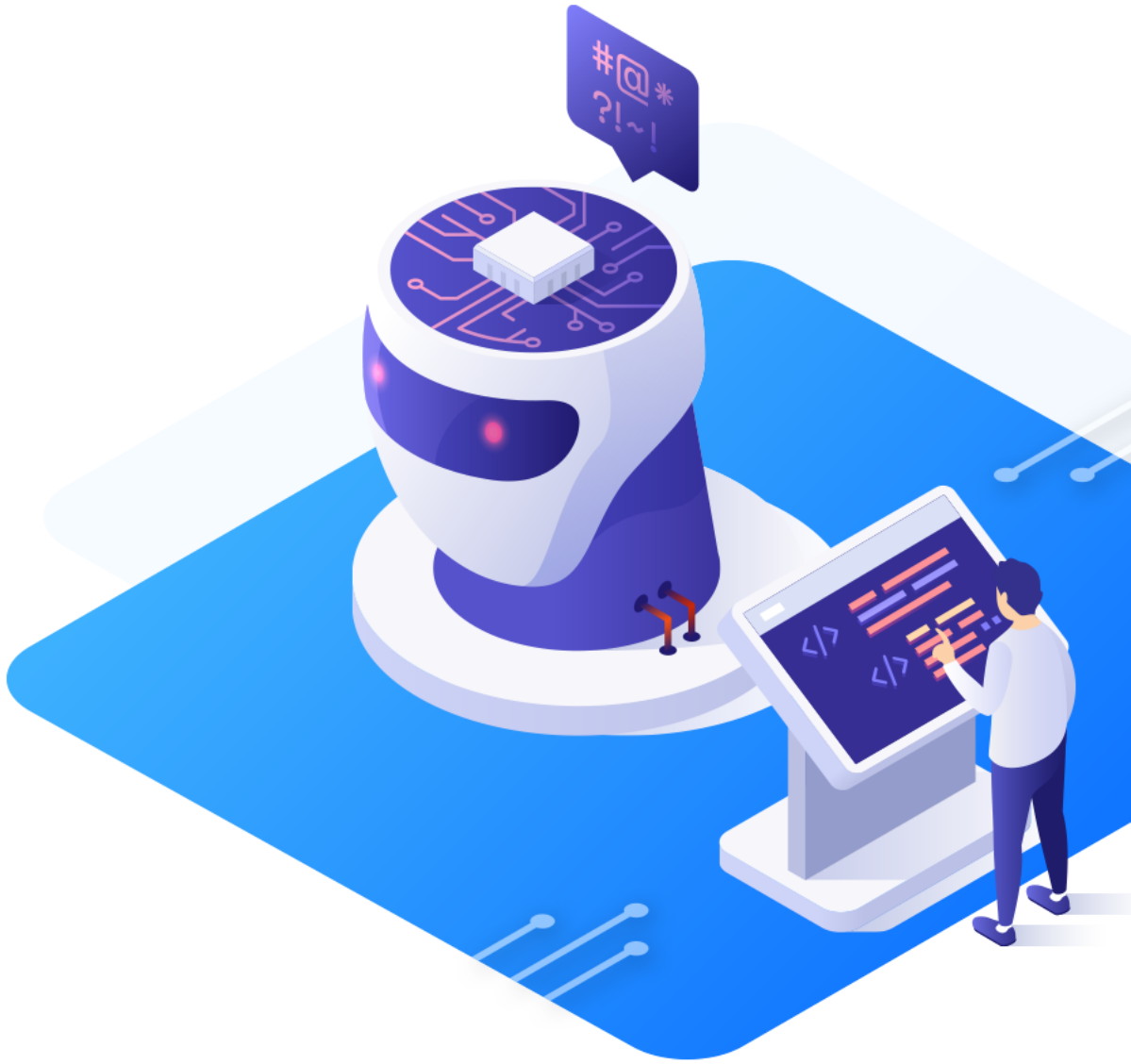
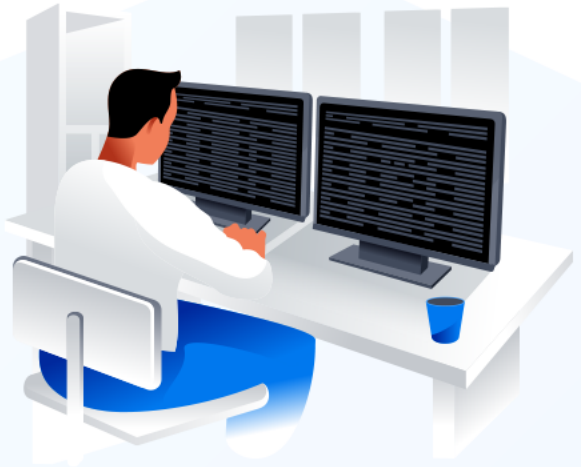
GitHub Copilot

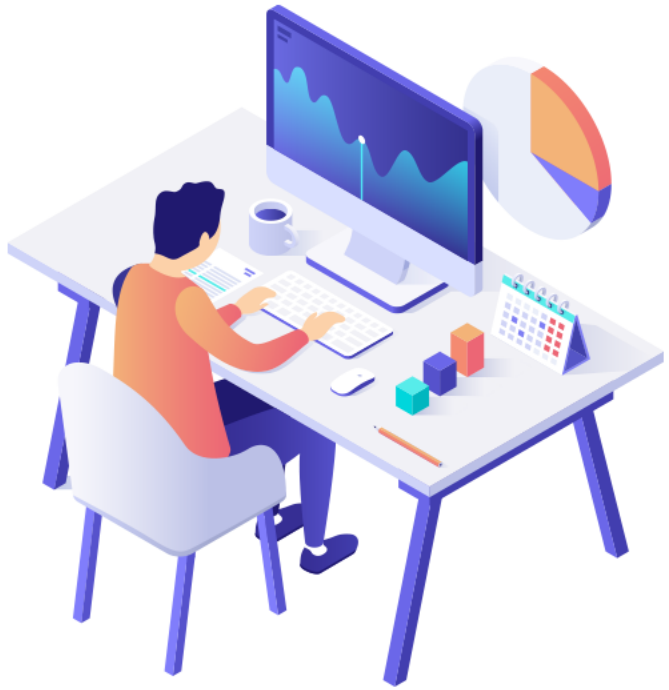


PROJECT



Personal Expense Tracker

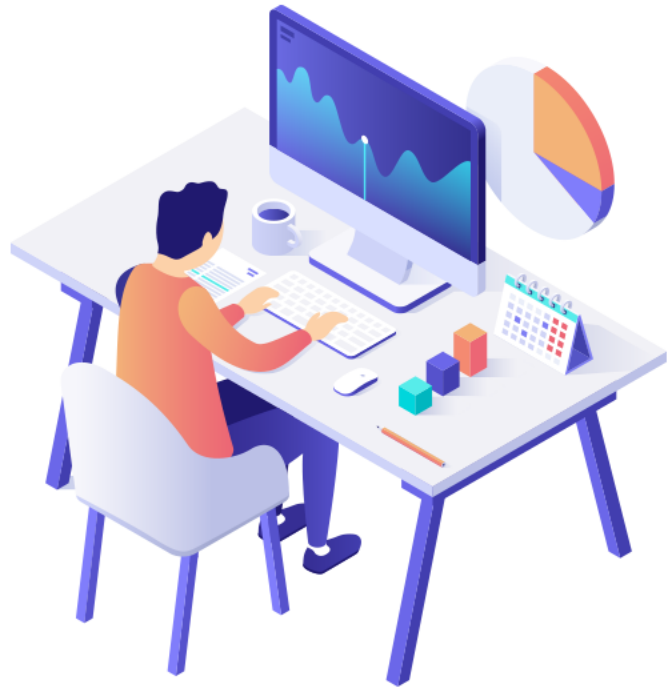
Creating a **High-Level Project**: a **Personal Expense Tracker** using JavaScript. This project will help users track their income and expenses, categorize them, and visualize the data through charts. We'll leverage GitHub Copilot for code suggestions and guidance throughout the development process.



Project Overview

Personal Expense Tracker

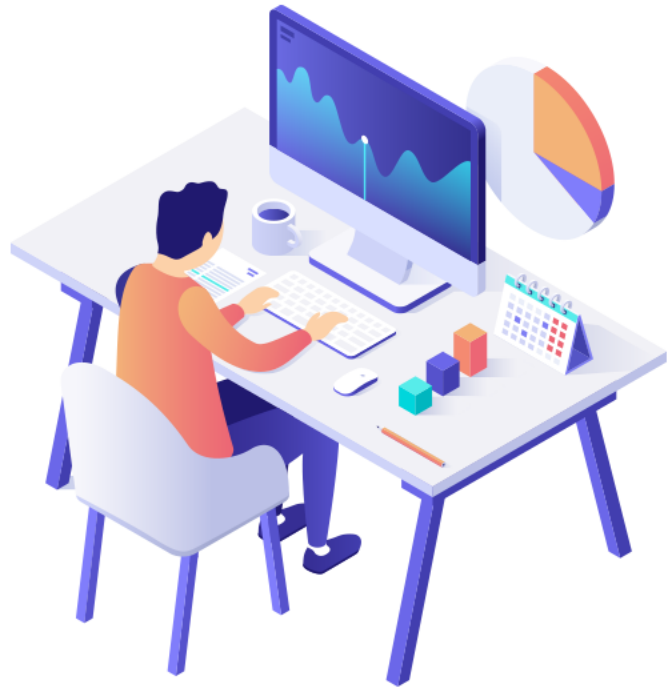
- **Features:**
  - Add income and expenses
  - Categorize transactions (e.g., food, utilities, entertainment)
  - Display transaction history
  - Visualize data with charts
  - Filter by category
  - Save data in `localStorage`



Set Up the Project

- **Create a Project Directory**

```
bash
Copy code
mkdir expense-tracker
cd expense-tracker
```

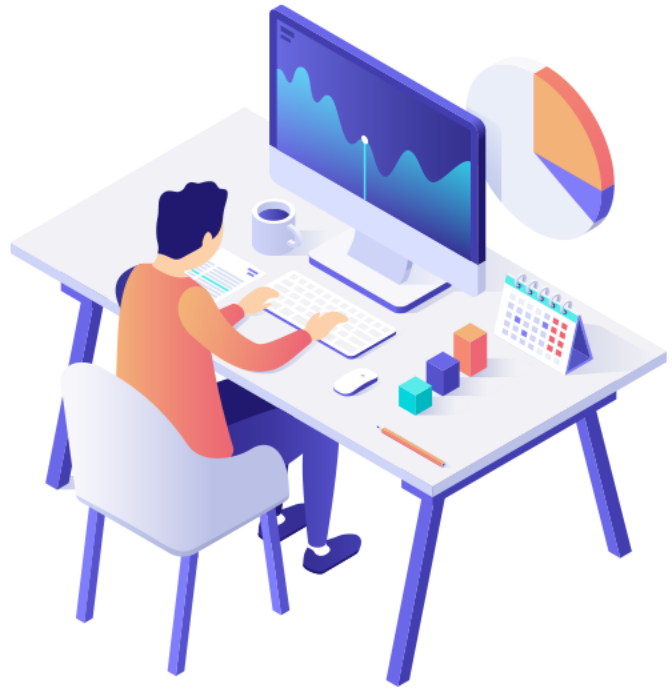


Initialize a Git Repository

```
git init
```

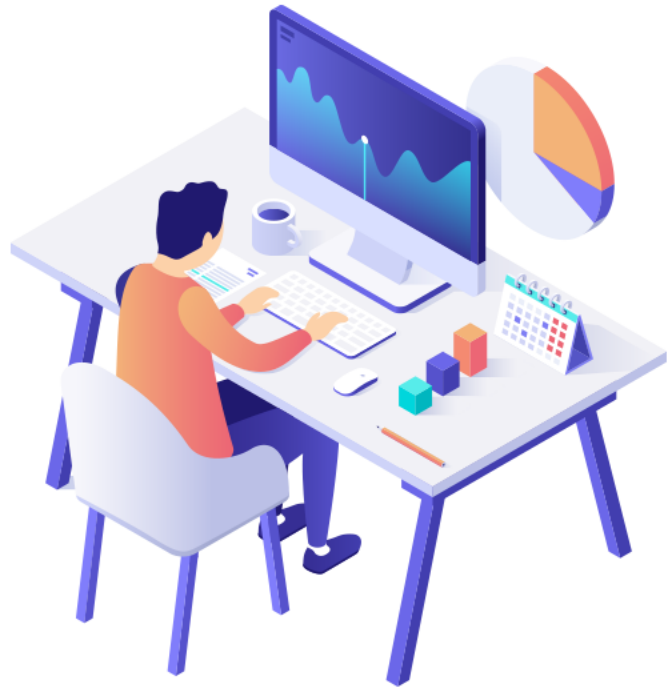
Create HTML, CSS, and JavaScript Files

```
touch index.html styles.css script.js
```

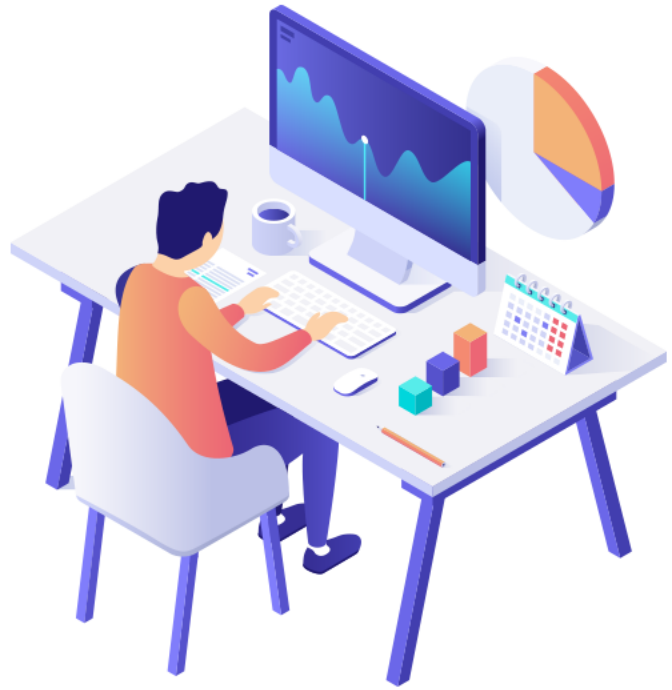


Set Up Basic HTML Structure Open `index.html` and add the following code:

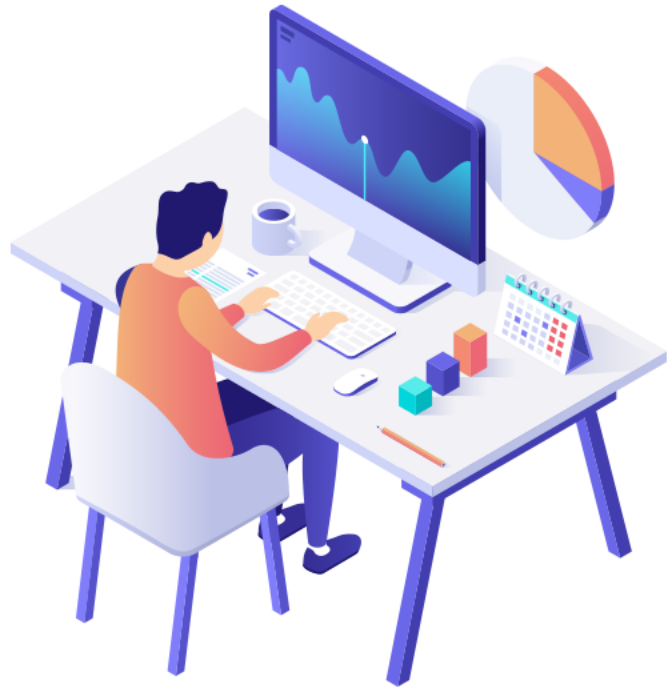
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <title>Expense Tracker</title>
</head>
```



```
<body>
    <div class="container">
        <h1>Expense Tracker</h1>
        <form id="transactionForm">
            <input type="text" id="description" placeholder="Description"
required>
            <input type="number" id="amount" placeholder="Amount"
required>
            <select id="category">
                <option value="income">Income</option>
                <option value="expense">Expense</option>
            </select>
            <button type="submit">Add Transaction</button>
        </form>
```



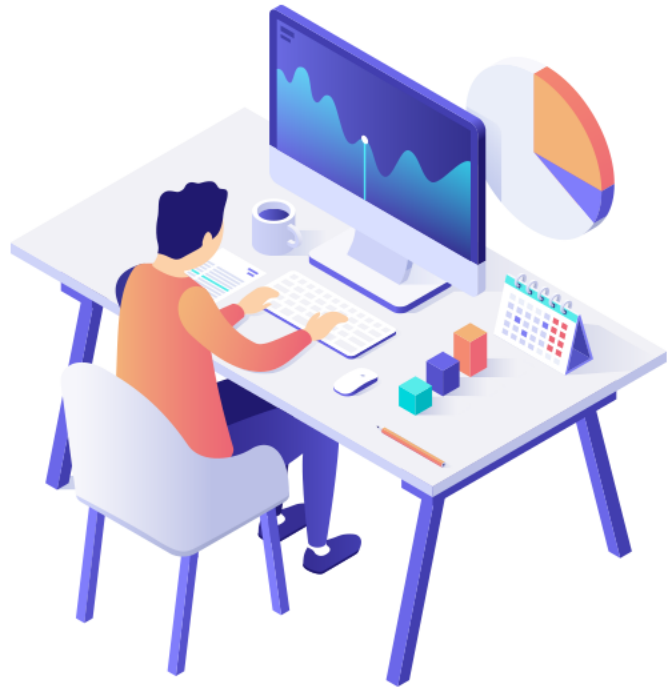
```
<h2>Transaction History</h2>
        <ul id="transactionList"></ul>
        <h2>Statistics</h2>
        <div id="stats"></div>
        <canvas id="chart"></canvas>
    </div>
    <script src="script.js"></script>
</body>
</html>
```



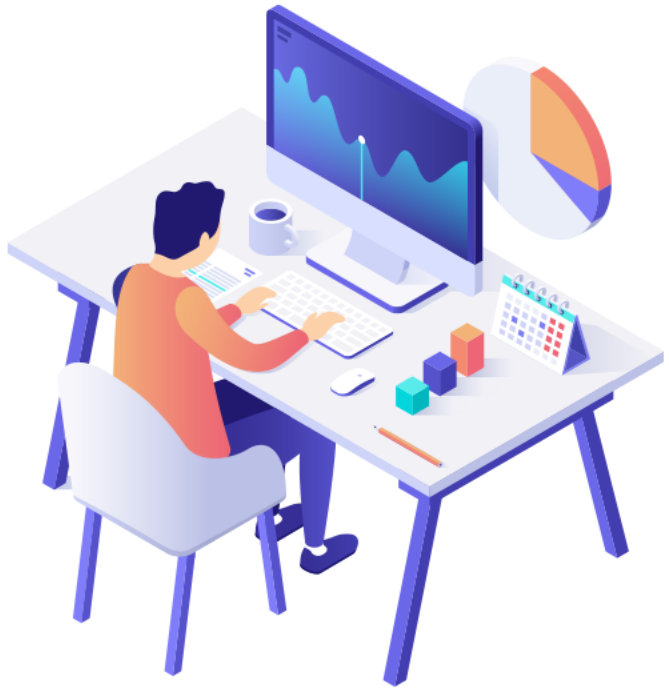
Style the Application

- **Add Basic Styles** Open `styles.css` and add some basic styling:

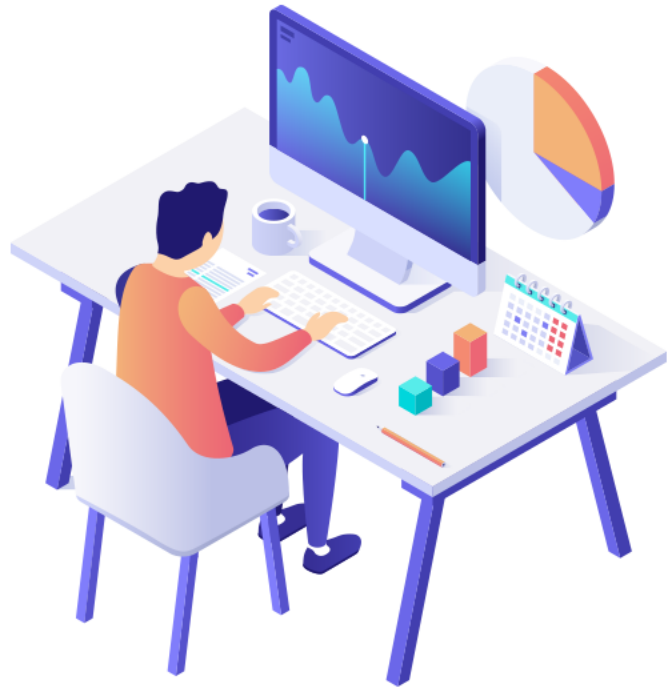
```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
```



```
.container {
    background: white;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    width: 400px;
}
input, select {
    padding: 10px;
    margin: 10px 0;
    width: 100%;
}
```



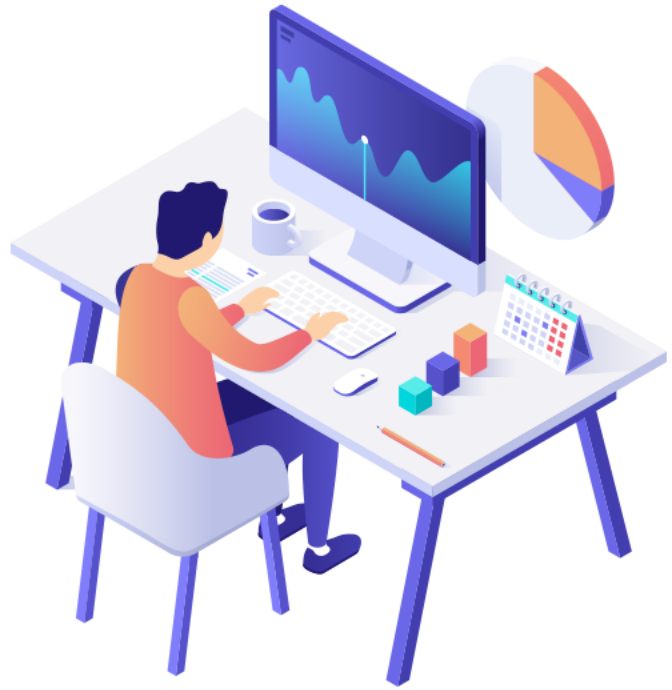
```
button {
    padding: 10px;
    width: 100%;
}
ul {
    list-style: none;
    padding: 0;
}
li {
    display: flex;
    justify-content: space-between;
    padding: 10px;
    border-bottom: 1px solid #ccc;
}
```



Implement JavaScript Functionality

- **Add Event Listeners** Open `script.js` and set up event listeners for form submission.

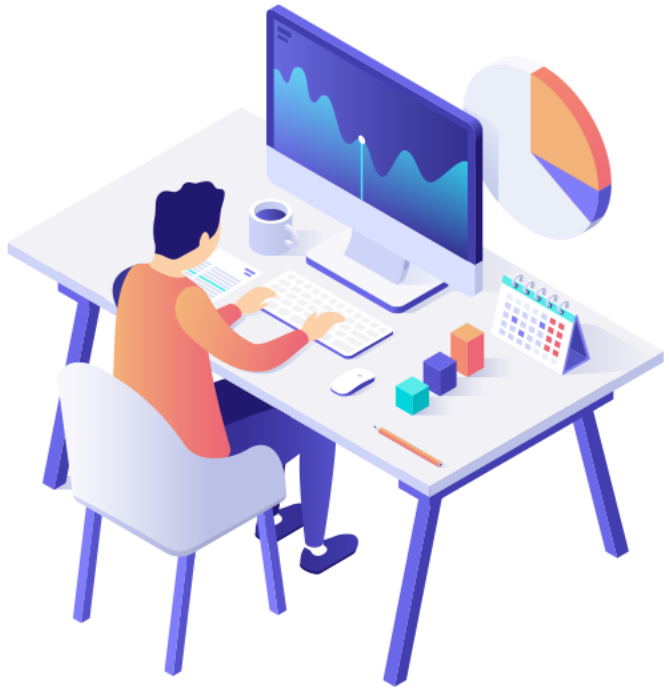
```
document.getElementById('transactionForm').addEventLis
tener('submit', addTransaction);
```



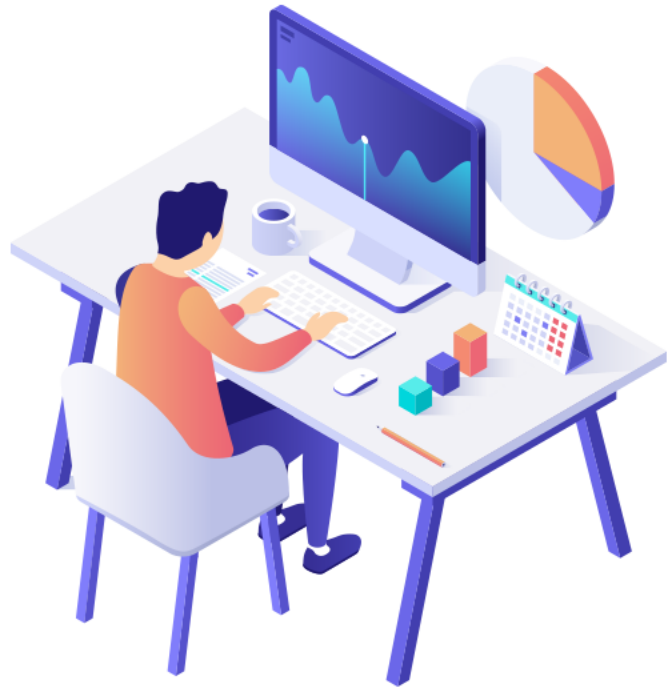
Implement the addTransaction Function

- Use a comment to guide Copilot.

```
// Function to add a new transaction
function addTransaction(e) {
    e.preventDefault(); // Prevent form submission
    const description =
document.getElementById('description').value;
    const amount =
parseFloat(document.getElementById('amount').value)
;
    const category =
document.getElementById('category').value;
```



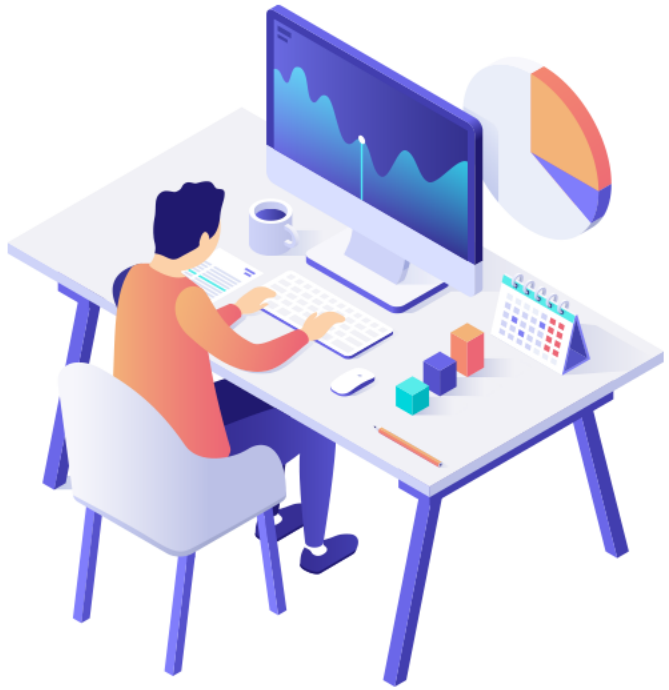
```
// Create a transaction object
    const transaction = { description, amount, category, id: Date.now() };
    addToLocalStorage(transaction); // Save to localStorage
    displayTransaction(transaction); // Display the transaction
    updateStats(); // Update statistics
    clearForm(); // Clear input fields
}
```



Display Transactions

```
function displayTransaction(transaction) {
    const li = document.createElement('li');
    li.innerHTML = `${transaction.description} -
$${transaction.amount} (${transaction.category})`;

document.getElementById('transactionList').appendChild
(li);
}
```

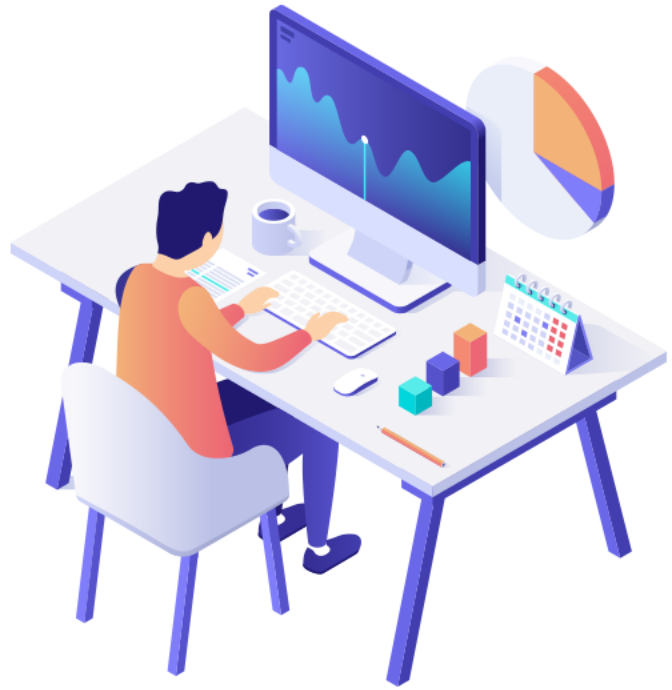


Implement Local Storage Functions

- Save and load transactions from `localStorage`.

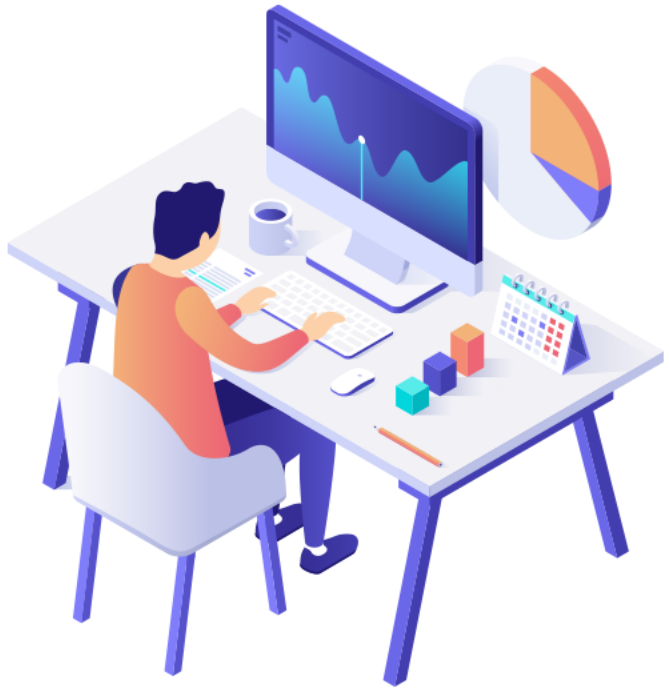
```
function addToLocalStorage(transaction) {
    const transactions =
JSON.parse(localStorage.getItem('transactions')) || [];
    transactions.push(transaction);
    localStorage.setItem('transactions',
JSON.stringify(transactions));
}

function loadTransactions() {
    const transactions =
JSON.parse(localStorage.getItem('transactions')) || [];
    transactions.forEach(displayTransaction);
}
```



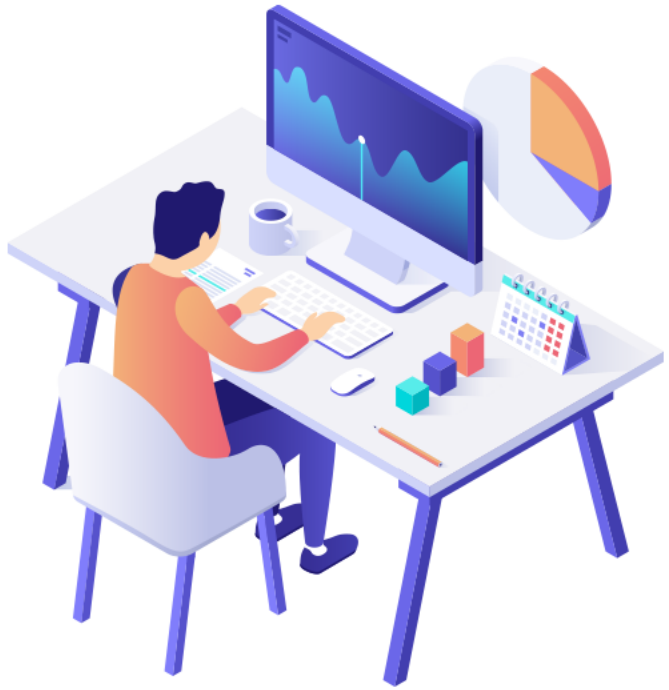
Update Statistics

```
function updateStats() {
    const transactions = JSON.parse(localStorage.getItem('transactions')) || [];
    const income = transactions.filter(t => t.category === 'income').reduce((acc, t) => acc + t.amount, 0);
    const expenses = transactions.filter(t => t.category === 'expense').reduce((acc, t) => acc + t.amount, 0);
    document.getElementById('stats').innerHTML = `Income: $${income} | Expenses: $${expenses}`;
}
```



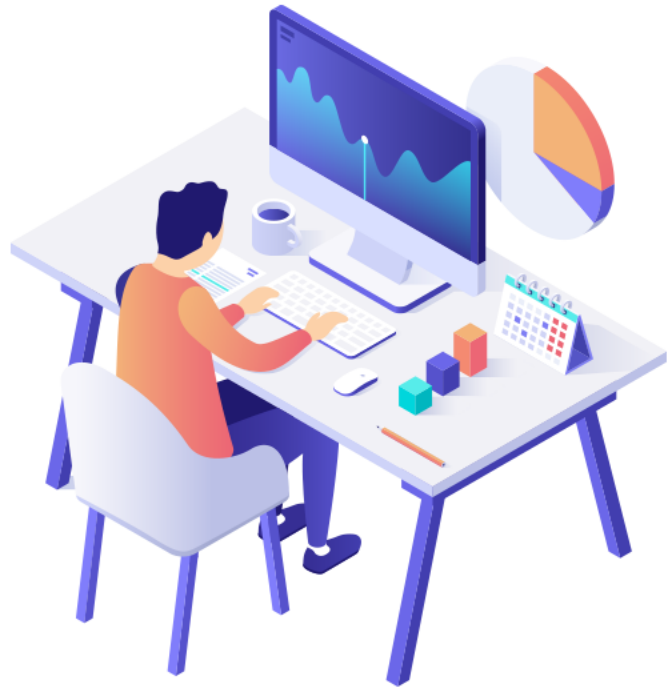
Clear the Form

```
function clearForm() {
    document.getElementById('description').value = '';
    document.getElementById('amount').value = '';
    document.getElementById('category').value =
'income';
}
```



Load Transactions on Page Load

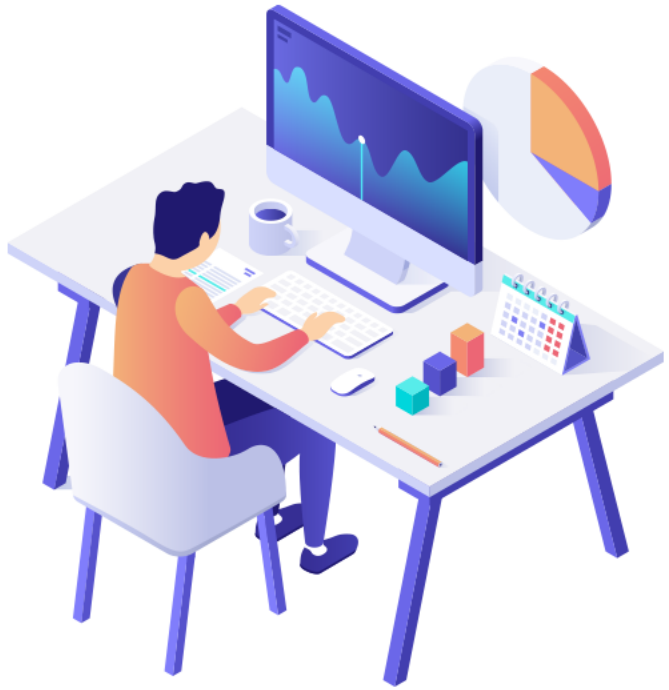
```
window.onload = () => {
    loadTransactions();
    updateStats();
};
```



Visualizing Data

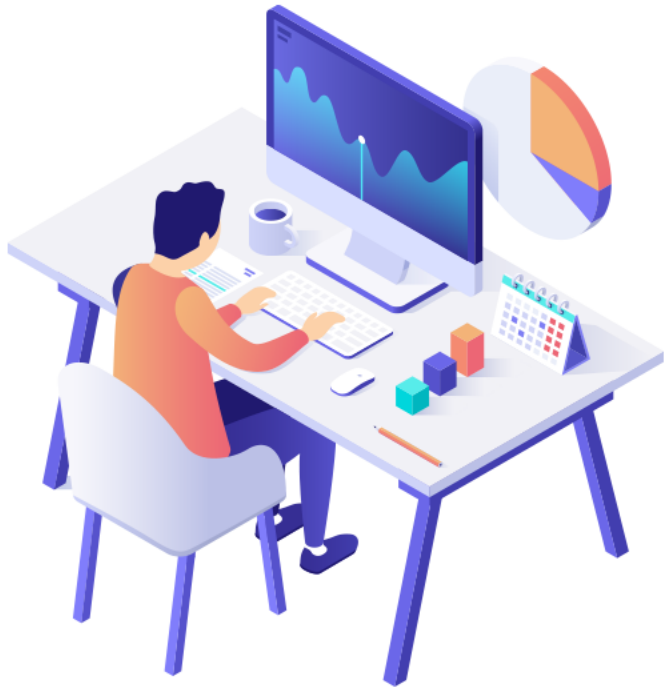
- **Install Chart.js**
  - Add the Chart.js library for visualizing transactions.

```
<script
  src="https://cdn.jsdelivr.net/npm/chart.js"><
  /script>
```

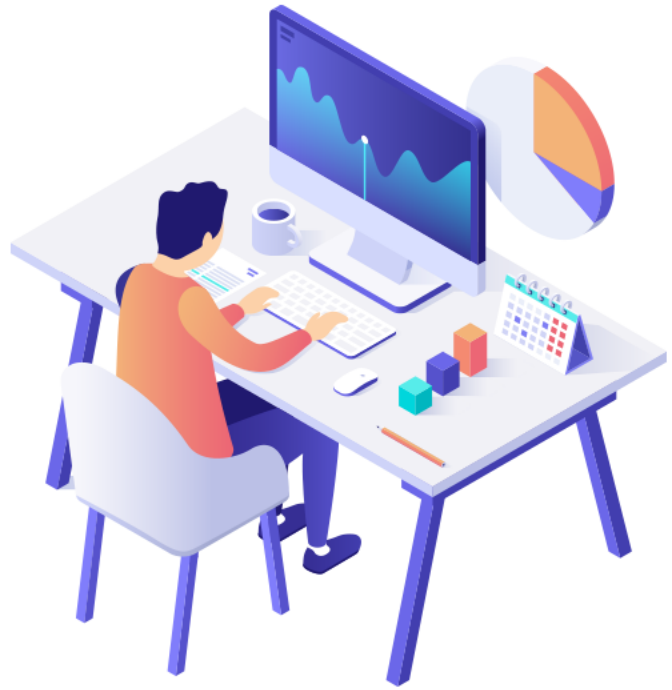


Create a Function to Generate Charts

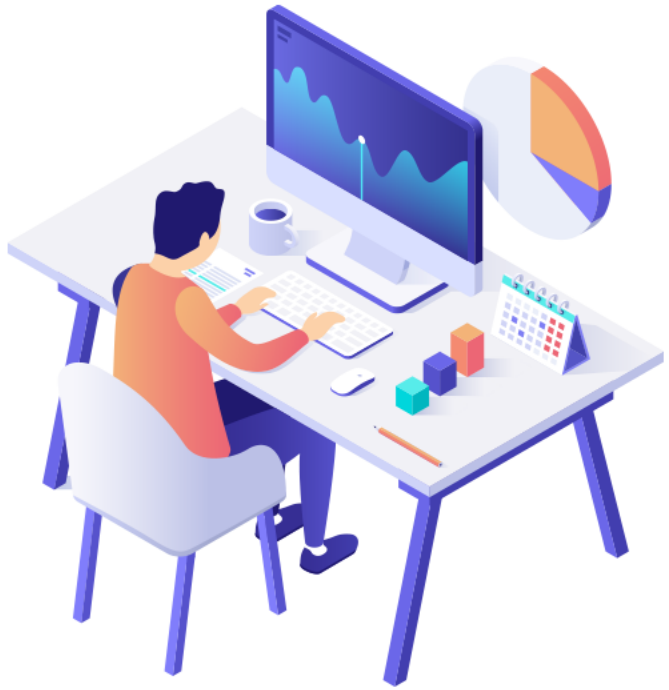
```
function generateChart() {
    const transactions = JSON.parse(localStorage.getItem('transactions')) || [];
    const categories = [...new Set(transactions.map(t => t.category))];
    const data = categories.map(cat => {
        return transactions.filter(t => t.category === cat).reduce((acc, t) => acc + t.amount, 0);
    });
```



```
const ctx = document.getElementById('chart').getContext('2d');
    new Chart(ctx, {
        type: 'bar',
        data: {
            labels: categories,
            datasets: [{
                label: 'Expenses by Category',
                data: data,
                backgroundColor: ['#36a2eb', '#ff6384', '#ffce56'],
            }]
        },
```

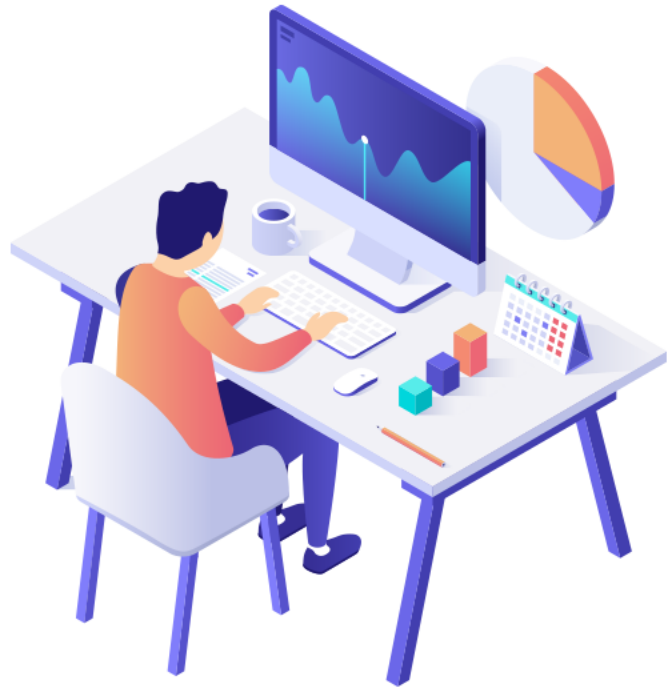


```
options: {
        responsive: true,
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});
}
```



Call generateChart after Loading Transactions

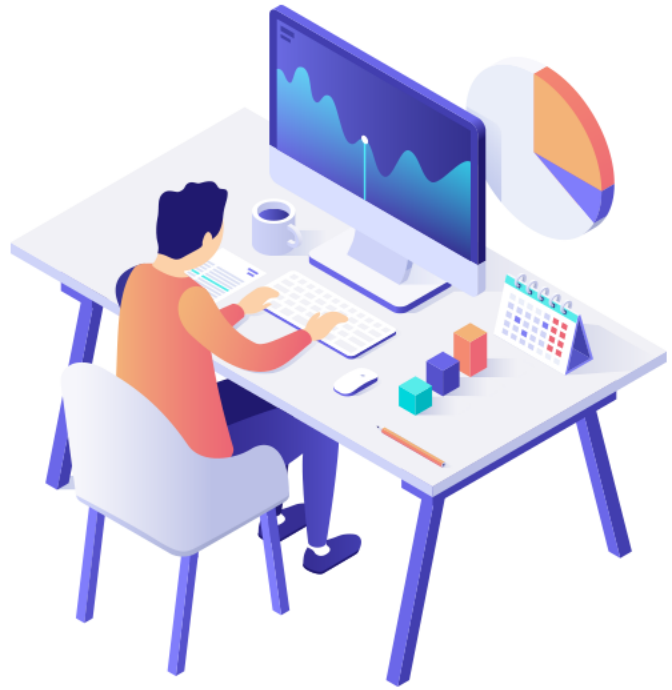
```
function loadTransactions() {
    const transactions =
JSON.parse(localStorage.getItem('transactions')
) || [];
    transactions.forEach(displayTransaction);
    generateChart(); // Generate chart after
loading transactions
}
```



Testing and Debugging

- **Test Functionality**
  - Open `index.html` in a browser and test adding, categorizing, and visualizing transactions.

- **Debugging with GitHub Copilot**
  - If you encounter issues, write comments to guide Copilot in generating debugging code or logging errors.

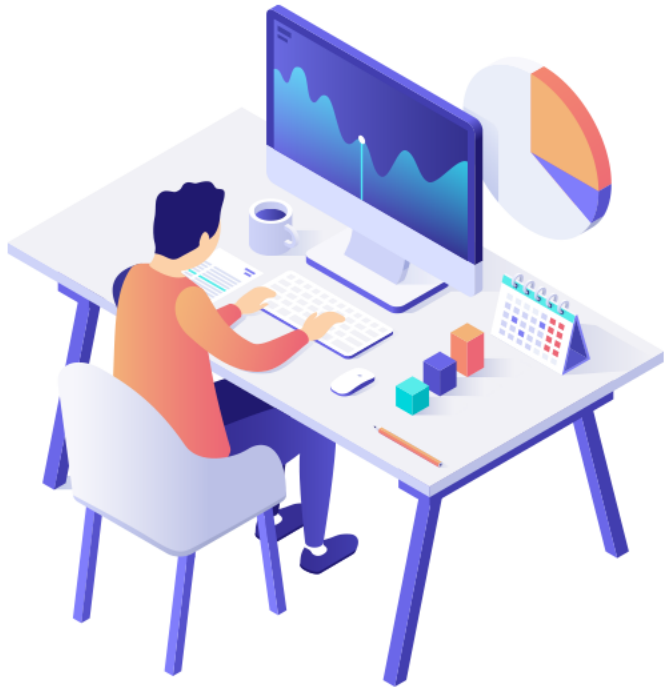


Version Control

- **Commit Your Changes**

```
git add .

git commit -m "Initial commit: Personal Expense
Tracker with basic functionality"
```

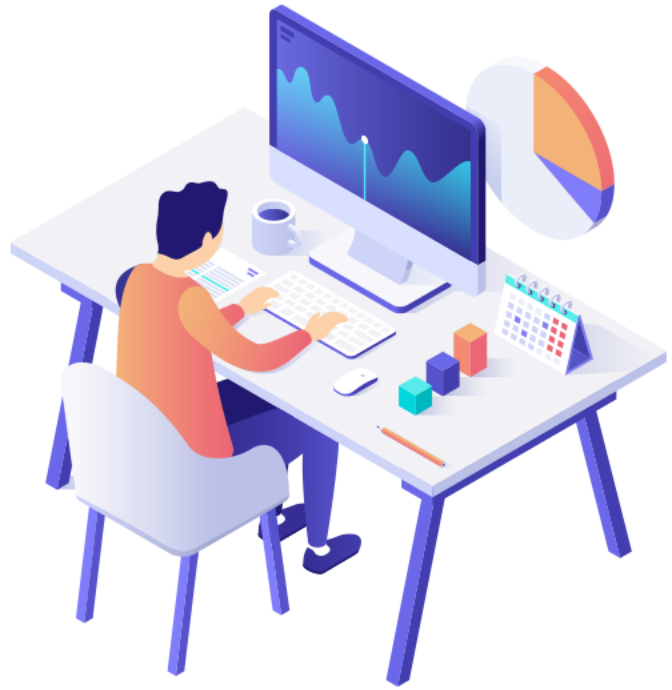


- **Push to GitHub (Optional)**
  - Create a repository on GitHub and follow the instructions to push your local repository.

Future Enhancements

- Consider adding more features:
- **User Authentication**: Allow users to create accounts.

- **Export Data**: Enable users to export transaction history as a
- CSV.
- **Mobile Responsiveness**: Improve styles for mobile viewing.



NOTE :

This Personal Expense Tracker project is a comprehensive way to practice using GitHub Copilot alongside JavaScript. By following these steps, you can build a functional application that incorporates user input, local storage, data visualization, and more. As you continue to develop and enhance the project, you'll gain a deeper understanding of JavaScript and the capabilities of GitHub Copilot.