1. Instructions Given to Copilot for the assignment :-  
     
   **1. Home Page**
2. **What You Need to Do**:
3. Create a welcoming home page for a grocery shopping website.
4. **Navigation Bar**: Include the logo "GroceryShop" with an icon. Add links for Home, Products, About, and Contact. Make sure this navbar looks good on all devices.
5. **Description Section**: Add a heading and a brief paragraph introducing GroceryShop. Place this next to a nice image of the store or products. Ensure this layout works well on mobile screens.
6. **Featured Products**: Show off some featured products in a card format. These should adjust properly across different screen sizes.
7. **2. About Page**
8. **What You Need to Do**:
9. Introduce the team behind GroceryShop.
10. **Header**: Add a heading like "About Us" and a short intro about the store’s mission.
11. **Founders Section**: Include images of the founders with their names and a brief description of their roles. Design this section to be both fancy and user-friendly on mobile devices.
12. **3. Products Page**
13. **What You Need to Do**:
14. Display your products in an organized grid.
15. **Header**: Add a heading like "Our Products".
16. **Product Grid**: Arrange products in a grid with 3 rows and 2 columns. Each product should have an image, name, and price. Ensure there’s enough space between items and that the grid looks great on all device sizes.
17. **4. Contact Page**
18. **What You Need to Do**:
19. Make it easy for customers to get in touch and find your store.
20. **Header**: Add a heading like "Contact Us".
21. **Contact Form**: Include fields for Name, Email, and Message. Ensure the form is responsive and easy to fill out on mobile devices.
22. **Map**: Embed a Google Map showing the store’s location. Make sure it’s properly sized and visible on all screen sizes.
23. **5. CSS and Responsive Design Instructions**
24. **Overall Design**:
25. Your site should look good and be easy to use on all devices—phones, tablets, and desktops.
26. Use media queries in your CSS to adjust layouts for different screen sizes.
27. **Specific Fixes**:
28. **Home Page Image**: Make sure the main image isn’t too big and fits nicely within the page.
29. **Footer**: Ensure the footer stays properly aligned and doesn’t get disrupted.
30. **Form and Map on Contact Page**: The contact form should be easy to use, and the map should display correctly across various devices.
31. **CSS Adjustments**:
32. **Navigation Bar**: Make sure it looks good and works well on both large and small screens.
33. **Founders Section**: Ensure that images and text are well-aligned and responsive.
34. **Product Grid**: The grid should adjust smoothly to different screen sizes.
35. **Contact Form and Map**: Both should be responsive and well-positioned.

How Website Looks on different devices :-  
  
**On desktop :-**  
The home screen looks pretty good with decent colours and images however the navigation bar is well aligned and the layout typically looks good .

The Products page have a good layout having an image inside a card with some description about the image which in our case all about grocery.

The about page looks decent with 3 cards which has Owners photo inside it pretty decent so to say .

The Contact us page has a contact form with embedded map it looking good enough for a contact page the footer is very basic and simple .

**On Mobile :-**

The navbar gets a little disrupted apart from that all other things a looking good the contact form and maps looks a little shrinked in size.

**What I like about the website :-**

The ease and speed of making layout of very basic and simple website using Copiloy just shocked me. You can easily create a layout for full fledged website using copilot within hours. The design was also very good and you can customize it according to your needs.

The colours used by copilot were pretty decent and the text size according to the screen size was good the images were resized according to the screen size which is pretty amazing .

**What I dislike about website :-**  
Althoght almost everything looks pretty good enough but still there are some places where changes can be made such the navbar can be aligned well in both the screen sizes and the

The images can be resized in a better way .

For the images the designer was not able to give me ehat I needed even after giving very specific details I pretty disappointed with copilot designer .

Overall, Copilot can be used for basic functionalities and layout in webdevlopment it can be really helpful and time saving sometimes but you can only play with it when you have a complete understanind of everything works cause you need to customize it according to your needs at the end afterall its an ai and it will make mistakes and its easier to rectify the mistake ourselves as ai is too bad at identifying and rectifying the mistakes it will give you the same code again and again and it be quite frustrating sometimes .

**Browser Compatibity:-**  
  
The Website is Compatible on Firefox, Edge and Chrome And Brave browser however copilot didn’t do a good job in making the website responsive . I can have done a better job then copilot.

**Mozzila:-**When comparing how a random website works on Mozilla Firefox and Google Chrome, several aspects can be considered:

1. **Rendering Engine**:
   * Firefox uses the Gecko rendering engine, while Chrome uses the Blink engine. Differences in these engines can lead to slight variations in how web pages are rendered.
2. **JavaScript Execution**:
   * Chrome uses the V8 JavaScript engine, whereas Firefox uses the SpiderMonkey engine. Although both are highly optimized, there may be differences in performance and behavior for complex scripts.
3. Chrome:-  
   **CSS Handling**:
   * While both browsers aim to adhere to CSS standards, there might be minor discrepancies in how CSS is interpreted and rendered, leading to subtle layout differences.
4. **Extensions and Add-ons**:
   * Both browsers support extensions, but they might interact differently with the website's content, potentially causing variations in behavior or performance.
5. **Developer Tools**:
   * Chrome DevTools and Firefox Developer Tools offer different features and user experiences. This can impact how developers debug and optimize websites for each browser.

To see these differences in action, one can open a website in both browsers and inspect the following aspects:

**Edge:-**

* **Layout and Styling**: Check for any visual differences in the layout, fonts, colors, and other styling elements.
* **JavaScript Behavior**: Look for differences in interactive elements, animations, and overall performance.
* **Console Errors/Warnings**: Open the developer console in both browsers and compare any errors or warnings that might appear.
* **Page Load Time**: Compare how quickly the page loads and responds in each browser.

Overall, modern web standards aim to minimize these differences, but subtle variations can still occur due to the unique implementations of each browser's engine.

4o

**Final questions:-**

1. **How you can do better than copilot?**

* The alignment was not good I could have done a better job.
* Css Colours were not appropriate
* Font Size was not good
* Logo and Hero Image placement was very hard with Copilot

1. **What things Copilot did Well?**
2. **Enhanced Context Understanding**:
   1. Better comprehend the overall design intent and style of the project to offer more relevant suggestions.
3. **Accessibility Awareness**:
   1. Improve suggestions to comply with web accessibility standards, ensuring inclusivity.
4. **Consistent Code Quality**:
   1. Maintain a consistent quality and style of code to avoid mixed coding practices.
5. **Better Handling of Complex Layouts**:
   1. Improve at generating code for complex layouts without causing layout issues or inconsistencies.
6. **Increased Responsiveness Awareness**:
   1. Provide suggestions that ensure the website is responsive across various devices and screen sizes.
7. **What are the Drawbacks of using copilot?**

 **Quality and Consistency**:

* **Risk of Inconsistent Code**: Copilot might generate code snippets that do not adhere to a consistent style or best practices, leading to inconsistent design and code quality.

 **Limited Understanding of Design Intent**:

* **Lack of Design Context**: Copilot can struggle to understand the specific design vision or user experience goals, resulting in suggestions that do not align with the desired outcome.

 **Potential for Over-Reliance**:

* **Skill Degradation**: Developers may become overly reliant on Copilot, leading to a decline in their own coding skills and problem-solving abilities over time.

 **Accessibility Concerns**:

* **Lack of Accessibility Considerations**: Copilot-generated code may not always prioritize web accessibility standards, potentially resulting in websites that are not fully accessible to all users.

 **Security Risks**:

* **Vulnerabilities in Code**: Copilot may suggest code that contains security vulnerabilities or outdated practices, which could compromise the security of the website.