Part 01: Application Setup

1. Using the terminal create an Angular 19 application using this command:

```
     ng new ng-app--skip-tests -skip-git
```

Choose the first option for CSS and N for Server-Side Rendering (just hit the enter key for CSS and N for SSR).

2. Open the ng-app application with VS Code. Open a terminal window and run the following command: `ng build`
If the app builds without errors, you are ready for the next step

3. Sign in to GitHub Copilot. There are a few ways to do this, we will discuss on the live session.

4. Here is the first prompt:
Use the app component to build a navigation bar with links to four components: home, products, contact-us and login. Do not show any other content on this template only the navigation bar. Also move all CSS to the styles.css file in the src folder.

5. At this point you should have a working application. If you are not able to follow along, use the ng-app-01 zipped file as you starting point for the next part.

Part 02: The Components

In this part we will build the four components so that each menu item can point to one component.

1. Here is my first prompt: **build four components one for each of the menu items in the navigation bar. use template urls and a separate css file for each component.**
This prompt will give you the content for each file, but it will not create the components for you. We will need to create these manually.
You can create each component like this: `ng generate component login -skip-tests`
Just replace login with home, contact-us and products.

2. Once you have your components, you can follow the Chat response for each file. This may take a while as you will have to carefully read the response and match it to the appropriate file.

3. After adding in all the suggested code from the Chat response, build and run the application. You should now have a working application. If you were not able to follow, you can always use the file named ng-app-02.zip.

Part 03 – Products

In this part we will change the products to bicycles. Obviously different participants will get different results after Part 02, but now we can attempt to sell a particular product.

1. In my case I got some kind of services as my sample product. In some cases you may not have gotten anything. Nevertheless try to prompt the tool to give you a product that you want to work with. Here is my first prompt: **`For the products component, I want to sell bicycles not services. Replace each service with a different bicycle. Use only royalty free images.`**

2. If you do not see images of bicycles, you can keep prompting. For example you can ask Copilot to use a different source for royalty-free images.

Part 04 – Contact-Us

The site looks good but we can improve the contact-us page

1. In my case, Copilot did not give me a field for telephone, so I am going to start there:

    `For the contact-us form, replace the subject field with a telephone field.`

2. The subject field is now replaced with a telephone field. I did not specifically ask for north American style telephone numbers but Copilot probably gathered that context from my location. Lets add validation: **`Add basic validation to the contact-us form, do not use the browser’s native validation.`**

3. Adjust the name field: **`The name field must not be less than three characters and must not contain any numeric values.`**

4. For the text area, lets try to create an array of invalid words. **`For the message field, I want to create an array of forbidden words. Lets start with the words “stupid” and “idiot”. If the user types these words, pop up a message that it is a forbidden word.`**

Part 05 – Profile

In this part lets try to build a database and have the user sign up and then login. We will keep everything local to the browser’s OS.

1. Add a server: **`I want the user to be able to complete a profile. We can start by adding the JSON server to this project.`**

In my case, I had to install the json server. Copilot gave me the commands. You will have to adjust the paths where the json-server picks up the data file. Something like this works on my machine:

```
"json-server": "json-server src/app/data/db.json --port 3000",
```

You can open a second terminal window to serve this file.

2. One of the suggested options is to create an API service in Angular to handle the interaction between the user and the json-server. For this manually create a service using a command like this:

```
ng g s api
```

3. You will also have to update the app.config.ts file to handle HTTP calls. The tool will guide you. So, if you have a service and the json-server installed, we can move on to the next step, posting a new user.

Part 06 – Signup Page

1. With the json-server installed and the db.json file, we can ask Copilot to help us build a sign-up page. Here is my prompt:

```
I want to create a sign-up page and collect a name, email and
password from the user.
```

After this prompt, I had to adjust the db.json file, then add a new sign-up component and of course update all three files in the component. I also had to update the routing and the api. I also had a problem with one of the paths, but Copilot help me resolve it.

2. Lets hook up the login page to the signup page:

```
     I want to test the new signup component. Add a link from the
     login page to the new signup page.
```

3. Try to post the data from the signup page to the db.json file we added in Part 05. Here is the prompt:

```
     i want to be able to post the data that the user enters in the
     signup page to the db.json file in the data folder using the
     json-server
```

4. Test the feature