Part 2

1. ~~Create a new Angular project with routing via the command line interface (CLI).  The name of the project is up to you.~~
2. ~~Add the Angular Material library to your project via the CLI.  The choice of theming is up to you.~~
3. ~~Add two new components via the CLI, giving appropriate names for each component as they will hold the two pages shown in the video:~~ ~~(1) Grocery List and (2) Purchase History.~~
4. ~~Set up the Routes so that a route exists for each page component in Step 3.  Give appropriate paths for each component~~.  ~~Also set up the default route (where the path is an empty string) to redirect to the Grocery List page.~~
5. Edit the application component's HTML and CSS, adding the necessary app modules along the way, so that the application contains:
   * A Material Toolbar at the top displaying the title of the current page component displayed (see #8) and a button with the "menu" icon that when clicked, toggles the visibility of the Material Drawer (Sidenav)
   * A Material drawer container, that should take up the rest of the space under the Material Toolbar
6. The Material drawer container of the application should contain:
   * A Material Drawer (Sidenav) with a Material Navigation List containing two items: links that direct the router to the two different page components of your application
   * The Angular router outlet, to display the active page component in the container
7. In the two page components, edit the HTML, CSS and TypeScript, adding the necessary app modules along the way, such that:
   * In conjunction with #8, the appropriate titles in the application's Toolbar will be displayed when the component is the active page in the Angular router outlet
   * They contain the controls as shown in the video for each page: a Material Select, Material Input and button for the Grocery List page and a button for the Purchase History page.
   * The form controls in the Grocery List page should be wrapped in Material Form Field elements and all be descendants of a single <form> element, in preparation for Assignment #3 where user input will be processed.
8. Attach an event handler to the Angular router outlet's **activate** event so that the title displayed in the Material Toolbar shows the title of the current page component displayed.

|  |  |  |  |
| --- | --- | --- | --- |
| Graphical user interface, application, Teams  Description automatically generated | Graphical user interface, text, application  Description automatically generated | Graphical user interface, application, Teams  Description automatically generated | A picture containing application  Description automatically generated |

Part 3

Applying the knowledge and experience you have up until this point, polish the application you have so far from Part 2 and add the following:

* A third page component containing information about you, the developer.  This page doesn't have to look like the one in the video but it should:
  + Include one image asset (.png, .jpg or .svg)
    - The image asset can be placed under the src/assets folder of your project
    - In the <img> element of your component's template HTML, you can precede the filename with "assets/" in the **src** attribute value, for example <img src="assets/mylogo.png"> if you placed mylogo.png under src/assets
  + Style the page so it has a good look and is user-friendly
  + Give it a title like your other components, so that if it is the active page component in the router outlet, its title will show in the application's Toolbar
  + Link to it in your application's Navigation List in your Drawer
* Icons for each Navigation List item
  + If you need to see an example, see: <https://material.angular.io/components/list/overview#lists-with-avatars>
  + If you need to include a module for the icons, see: <https://material.angular.io/components/icon/api>
  + For a list of icons included with the Material library, visit <https://fonts.google.com/icons>
  + You do not need to use the same icons shown in the video — choose your own
* The divider in the Drawer between Purchase History and About is optional, but if you accept the challenge, finding the details in the documentation to know which module to import, what tag name to use and where to place the divider element is an exercise left to you!