Angular Assignment

Story:

Develop an **“e-commerce”** web app using Angular (latest), HTML5, CSS3. Application should display list of the products. User should be able to search product, view details/description of product, add product to cart and checkout.

**Following are the key functional aspects:**

1. **Login Screen** : Validation. Create dummy user. Only authenticated user can go into the application.
2. **Search Implementation :** Create few dummy products and list products as per search criteria.
3. **Grid Product** : Everyone can see a grid page. On Click of product image/title user should be able to view **product description page.**

**Hint:** (Use reusable components here)

1. **Product Detailed Page :** The product description page **(PDP)** will have the details of the product such as **product** **name**, **price**, **description**, **tags**, **category**, **image**, **quantity** etc. All these attributes can be hardcoded for simplicity. **(no need to go into category details).**

The PDP screen will have an **add to cart button.** Upon clicking on add to cart, the user will be taken into a new screen which will have all the products being added to cart.

1. **Cart Screen** : Only **logged in** user can perform this action. Redirect to login screen if user is not logged in. The cart screen can **update the quantity of the product**, **delete a product**. The cart screen will show the **total of all the items** in the cart.

On clicking on checkout button, user will be redirected to the checkout screen.

1. **Checkout Screen** : Only logged in user can see this screen. Enter delivery details, name, shipping address, phone number, email. Add necessary validation on this form. Upon clicking on submit, show a message to the user “Order placed successfully”
2. Write test cases for at least 1 component ( not just html components but verifying the functionality as well) and 1 service.

**Good to have/ Bonus points:**

1. **Category Tree**: Try to classify the products under categories (build your data or mock json accordingly). Based on this hierarchy display the category tree for products.
2. **Translation**: For Example: System should support two languages at the moment for all the labels displayed on UI. In the header part we can have a dropdown to change the system language**.**

**Deliverables :**

1. GitHub link where code is kept ( without node\_modules)
2. Link for deployed code.
3. README file where the above two links are kept.
4. Running application
5. Unit test cases of 1 component and 1 service.
6. If attempted the bonus points, do mention it in readme.

**Notes**:

1. Take any relevant assumptions in case requirements are not clear. Do mention it in the README file.
2. Do mention if you have attempted the bonus points section or not.
3. For **submission of the assignment**, build the application and deploy it on **surge.sh / netlify or any other platform.**
4. Its mandatory to upload the code on GitHub/ or any other platform and provide the link for the codebase. **Make sure you do not upload the node\_modules.**
5. You can use **in** **memory DB, RxJS, mock json** for data storage as per your choice.
6. **Do not copy code from GitHub code links provided for reference (in videos).**
7. Application should be modular.
8. Feel free to enhance the UI experience.
9. Use your creativity for the details which are ambiguous.
10. Use your creativity to make your app looks great.

**Evaluation Criteria:**

1. Application Deployed
2. Auth Guards
3. Routing Implementation
4. Form validation
5. Ease of using application
6. Library for CSS
7. Reusability of components
8. Documentation
9. Code Quality + Linting
10. Use of modules/ Lazy loading
11. Unit testing of 1 component (test functionality and not just HTML components) + 1 service
12. Bonus points for attempting good to have features

**Timeline:**

**2 weeks.**