1. Create an angular 8 application called mydist. The application needs to include two additional components as well as a service. The first component is called my-image and the second component is called my-button. Both components should render in the app component or host view. When a user clicks the “like” button next to the image it should call a member function in the my-button component which passes a boolean member property to a service called my-service. The service method likeImage should accept a boolean parameter and return a banged version of the param. So every time the user clicks the button is should update the member property. Add a span element to the my-button component that renders “Image is liked == True”, replacing true with a template string literal bound to the member property. You can use the following url to an image of Homer Simpson or find your own image url.
   https://images.app.goo.gl/4bCeemd32fvrhYEj6

2. Add a css class to the my-button.component.css file called is-liked and set the “background-color” property to green. Add the ngClass property to the button element. When the button is clicked, component member property evaluates to true, the color should update to have a green background color.
3. Add a folder under src/app/models. Design the class or model architecture to support a chat application. The model must include a message class, messages class, and a user class. The message class should include a timestamp and text attribute, the messages class should contain a member property that is an array that holds messages.
4. Create an additional service called messages.service.ts. Import the classes from the models folder into the services. Add a member function that instantiates several message objects and append them to a messages object which should contain a member property that is an array of messages.
5. Extra Credit: Add a constructor method to app.component.ts and import the messages service to the app component. Inside the constructor invoke the member function which should return an array of messages. Assign the messages array to a member property in the app component. In the app component template below the image add a div and use the *ngFor directive to iterate over the messages rendering the text message and the timestamp for each message
6. SUBMISSION GUIDELINES: add a readme file with any instructions at the root of the project, commit your changes and push a version to your public github account. Include the full github url in the readme file along with any additional instructions if needed. create a dist file using ng-build from the angular cli. Zip the file and submit on D2L.

   HINT: SLACK IS YOUR FRIEND! POST QUESTIONS TO THE HOMEWORK CHANNEL !!!