**Day 6 Lab Assignments**

1. Create new link in the navbar (Admin) that opens new component (new Product).
   1. Add new route: ‘/admin/insertproduct’ which redirects to the new product component.
   2. Create a Product object in the class of the component and initialize it with static data in the constructor.
   3. Create a button on the new component, that when pressed uses addProduct(product) method to insert the new product and on success redirects to products (order) component.
2. Change the previous (add product) component and add new form for inserting the product.
   1. Make a dropdown to select the product category (Fill categories from the new service), and inputs for the other fields of the product.
   2. addProduct(product) that adds new product (Use POST API: <http://localhost:3000/products>), knowing that this API return the inserted product object ton success, so you need this function to return an observable of Product.
   3. Use Template Driven Form
   4. Use 2-way binding to bind from inputs to the Product object in the class.
   5. Handle the submit button to use the service to add the new product and redirects to the products (Order) component on success.
3. Handle Edit and Delete products.
   1. In Edit, you should show form with existing values, and user can change and then press update.
      1. Use the same Insert product component, but take URL param: product ID, and add new product in case if there’s no Prd ID, and update in case of the Prd ID is sent.
   2. In Delete, you should display confirm message before delete.
      1. You need to delete the deleted product from the products collection after delete success response, if the delete is implemented on the product list component.
4. Add new model for User (Using below fields) and add new collection in the JSON server for users.
5. Create User registration component and use reactive form on it:
   1. Add the following fields:
      1. Full Name, use: required, min length 5 validators.
      2. Email, use required, Email validators.
      3. Mobile number, use required, pattern validators.
         1. Show button (add another mobile), that will show new input with the same pattern validation, and other button (remove) to remove only additional numbers (Use Dynamic form building).
      4. Address (City, Postal Code, street): make the city, postal code and street nested from Address field, and make all of them required.
      5. Password : required, min length 6.
      6. Confirm password