

Practice Exercise

This document provides a list of exercises to be practiced by learners. Please raise feedback in Talent Next, should you have any queries.

Skill	Angular 15
Proficiency	S1
Document Type	Lab Practice Exercises
Author	L & D
Current Version	1.0
Current Version Date	13-April-2023
Status	Active

Document Control

Version	Change Date	Change Description	Changed By
1.0	13-Apr-2023	Scenario based Problem Statements	Shabarinath KP

Contents

Practice Exercise.....	1
Document Control	2
Introduction	4
Problem Statement 1: Creating Components	4
Problem Statement 2: Component to Component Communication	5
Problem Statement 3: Creating Custom Pipes	7
Problem Statement 4: Designing the Login Form	8
Problem Statement 5: Add / Edit / Delete Customer	9

Introduction

Contura is startup company who specialize in shipping of goods. You need to build a Single Page Application (SPA) using Angular to meet their requirements.

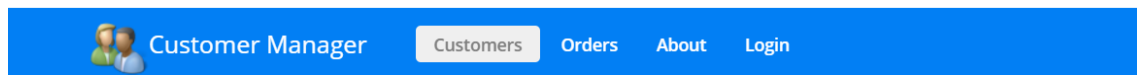
Note: Every Problem Statement starts in a new page

Problem Statement 1: Creating Components

Create an Angular application using CLI with following components

- Welcome Component – is the default component that is rendered when the application is run
- Customer Component – Display the list of customers
- Orders Component – Display the list of customers
- About component – Render details of the Contura
- Login Component – Display the login page that accepts credentials

The Above components must be loaded as per the below image in the welcome page



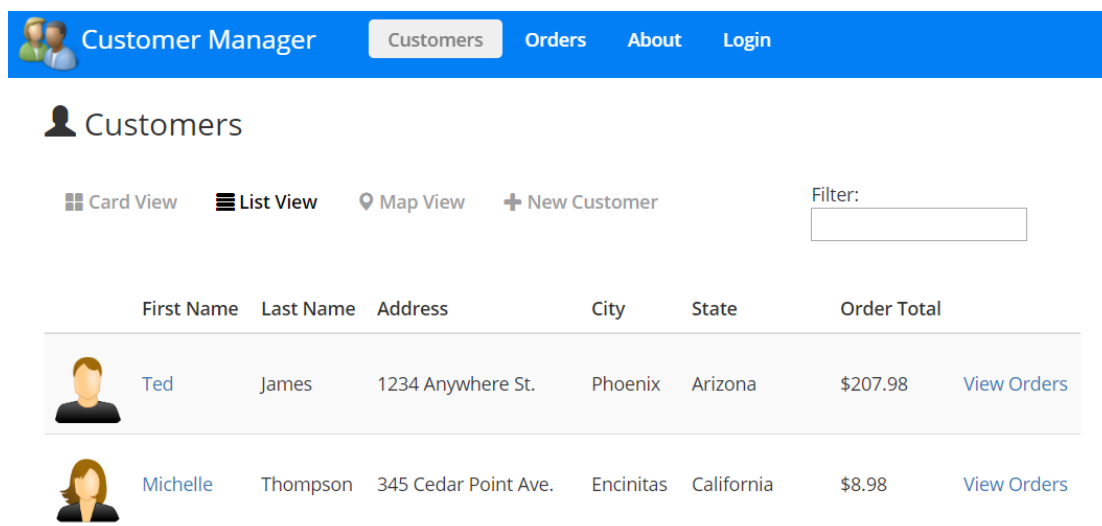
Problem Statement 2: Component to Component Communication

Create a list of customers and orders in a json object.



- Create Customers Component and Orders Component
- Pass data between Customers Component and Orders Component
- The customers and orders details should be hierarchical and have relationship.
- Enable pagination based on the count of users. Max # of customers to be displayed is 5 per page. Pagination can be implemented using Directives.

The Above components must be loaded as per the below image

Customers Component View



The screenshot shows a web application interface for a 'Customer Manager'. At the top is a blue navigation bar with a logo and links for 'Customers', 'Orders', 'About', and 'Login'. Below the navigation bar, the main heading is 'Customers'. There are four view toggles: 'Card View', 'List View' (which is selected), 'Map View', and '+ New Customer'. To the right of these is a 'Filter:' input field. Below the toggles is a table with columns: 'First Name', 'Last Name', 'Address', 'City', 'State', 'Order Total', and a 'View Orders' link. The table contains two rows of customer data.

	First Name	Last Name	Address	City	State	Order Total	
	Ted	James	1234 Anywhere St.	Phoenix	Arizona	\$207.98	View Orders
	Michelle	Thompson	345 Cedar Point Ave.	Encinitas	California	\$8.98	View Orders

Orders Component View is shown as below image

[Customer Manager](#)[Customers](#)[Orders](#)[About](#)[Login](#)

Orders

Ted James

Basketball	\$7.99
Shoes	\$199.99
<hr/>	
	\$207.98

Michelle Thompson


Frisbee	\$2.99
Hat	\$5.99
<hr/>	
	\$8.98

Angular 15 – S1 - Practice Exercise


Problem Statement 3: Creating Custom Pipes


Create a filter pipe that filters the customers as per the text entered in the Filter Textbox.


Custom Filter pipe View :


 Customer Manager


CustomersOrdersAboutLogin

 Customers


 Card View


 List View

 Map View

 New Customer

Filter:

Ted James

Phoenix,
Arizona
[View Orders](#)

«

1

2

3

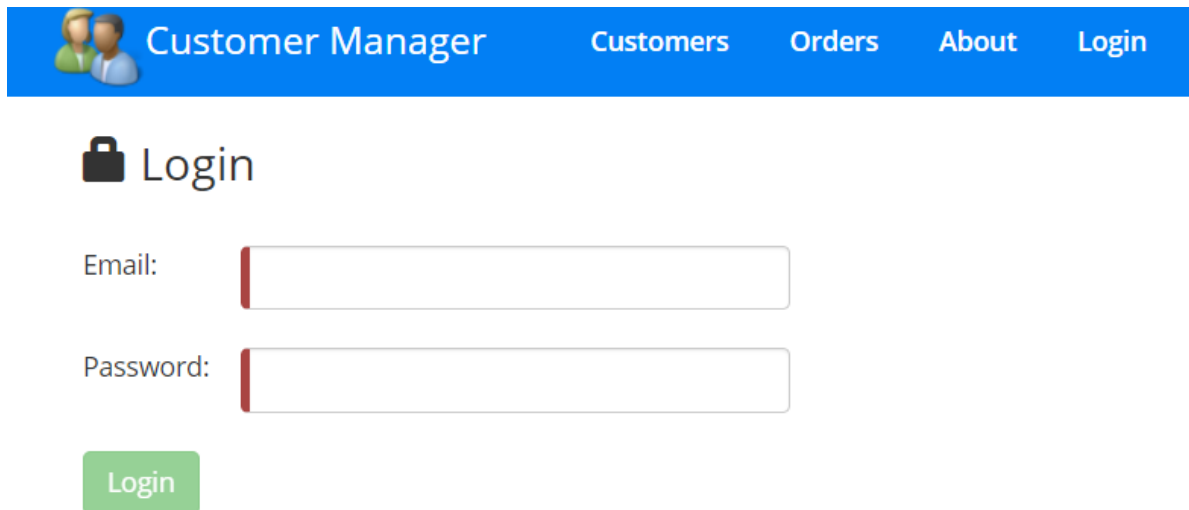
»

Problem Statement 4: Designing the Login Form

Create a login screen to accept the customer name and password.

- Implement Client-Side validations as below
 1. Textboxes cannot be empty
 2. Password min length – 6 Characters
 3. Password min length – 10 Characters
 4. Password should be a combination of letters ,number and 1 special character
 5. Submit button should be disabled till the validation is passed

Sample Login View looks like



The image shows a sample login view for a 'Customer Manager' application. At the top is a blue navigation bar with a logo of two people and the text 'Customer Manager'. To the right of the logo are four links: 'Customers', 'Orders', 'About', and 'Login'. Below the navigation bar is a section titled 'Login' with a briefcase icon. Under this title are two input fields: 'Email:' and 'Password:'. Each field has a red vertical bar on its left side, indicating a validation error. Below the password field is a green 'Login' button.

Problem Statement 5: Add / Edit / Delete Customer

Create a UI to add / edit / delete the customer.

- When an anonymous user tries to add / delete / edit customer – he should be redirected to login view
- After successful creation of the new customer / updating of the existing customer, update the JSON object created in the above assignment.

Add Customer View looks like below image



The form consists of five text input fields stacked vertically, each with a label to its left: 'First Name', 'Last Name', 'Address', 'City', and 'State'. Each input field has a red vertical bar on its left side. The 'State' field is a dropdown menu with a downward arrow on its right side. Below the input fields, there are two buttons: a 'Cancel' button with a light gray background and a green 'Insert' button with a green background.

Edit Customer View looks like below image

Customer Information

[Customer Details](#)[Customer Orders](#)[Edit Customer](#)

First Name

Last Name

Address

City

State

DeleteCancelUpdate