

HALFWAY THRU FSF

You have just completed half of your 40 days journey, the time has come to once again test out your new skills!

TOPIC:

Create a simple 3-pages Grocery Management module

CONCEPTS TESTED:

1. Basics of MySQL
 - Query and Manipulating Database
2. Angular
 - Component
 - Directive
 - Services
 - Router
 - HttpClient
 - PWA
 - Observable/Promise
 - Angular Material
3. Express
 - Routes
 - RESTful API: Handling HTTP requests
 - Connecting to a Database

RESOURCES:

You are provided with a sample groceries schema here: <https://goo.gl/14f0Lg>

How to generate barcodes:

<http://www.barcodes4.me/barcode/c128a/{upc12}.png?IsTextDrawn=1>

Example:

<http://www.barcodes4.me/barcode/c128a/035200264013.png?IsTextDrawn=1>

CORE REQUIREMENTS:

Page 1

1. Search form (mandatory fields)
 - a. Brand
 - b. Product Name
2. List of Groceries
 - a. Display a list of 20 products by default, with the Product Name arranged in alphabetical order
 - b. Options to sort grocery list by Brand or Product Name from A-Z and Z-A
 - c. Each product in the grocery list should display the following:
 - i. UPC12 Barcode
 - ii. Brand

- iii. Product Name
 - iv. Edit button (that will lead to Page 2)
- d. Search result should include all possibilities based on keywords found in Brand or Product Name fields
- e. You may use GET or POST methods for the search
- f. Sort by all the column headers
- 3. Add New Product
 - a. An Add New Product button is to be placed at a prominent location above the List (you may determine the exact position)
 - b. Clicking on this button will lead to Page 3

Page 2

- 1. Edit product form (mandatory fields)
 - a. Brand (varchar)
 - b. Product Name (varchar)
 - c. Save button (saves changes made before returning to the Grocery List)
 - d. Cancel button (brings you back to the Grocery List without saving)
 - e. Prevent saving of empty Brand and Product Name fields
- 2. Editing a product
 - a. Pre-populate the form with the product's info (i.e. if you entered this page by clicking on the Edit button of Product A, the form should display the Brand, Product Name, and UPC12 Number of Product A)
 - b. Saving should be done by the PUT method only
 - c. Grocery List should also reflect the changes made to the product

Page 3

- 1. Add product form (same mandatory fields as Edit)
 - a. UPC12 Number (BIGINT) [Additional input field]
- 2. Adding a product
 - a. Validate that UPC12 is unique before allowing the new product to be added
 - b. Warn if invalid characters are entered into the fields (e.g. UPC12 field should accept only integers)
 - c. Saving should be done by the POST method only
 - d. Grocery List should show the newly added product

BONUS (Do not attempt if you have no time):

Page 1

- 1. Search form (validation)
 - a. Prevent attempts to search when both Brand and Product Name fields are blank
 - b. Warn if invalid characters are entered into the fields
 - c. Do you want to allow a search by BOTH Brand and Product Name?
- 2. List of Groceries
 - a. Product Image (replace with Placeholder Image if Product Image is unavailable)
 - b. Pagination with ability to change the number of products listed per page

Page 2

1. Edit product form (validation)
 - a. Product Image (file upload field)
 - b. Delete button (brings you back to the Grocery List after removing product)
2. Editing a product
 - a. Demonstrate the ability to upload an image file (JPG) upon saving, and rename its filename to the product's corresponding UPC12 Number before putting the image file to a folder on your local drive or to a cloud storage.
 - b. Can your edit form accept non-English Brand and Product Name?

DURATION:

You have **6 hours** (not inclusive of 1 hour lunch break) to complete this Assessment. Your final commit must be pushed to your BitBucket or GitHub repo **before 5.30pm**. Please indicate in Slack, if you are using a different username for your BitBucket/GitHub account.