

Exit Test

Problem Statement

The problem statement is a product community website where a customer can visit, register themselves, browse products, raise for review, post reviews.

Please make reasonable assumptions wherever required and state them clearly in a separate text file (if any).

2 applications are to be developed.

First application (REST API – No UI, only backend):

User authentication API

Build a Rest API to support successful/unsuccessful user authentication validation.

User registration API

Build a Rest API to register new users.

Search product API

Build a Rest API to search a product by one or many of the following parameters:

1. Product name
2. Brand
3. Product code

Post reviews API

Build a Rest API to post a new review.

[BONUS] Approve review API

Build a Rest API to approve a review.

Request review API

Build a Rest API to request a review for a product.

Stats API

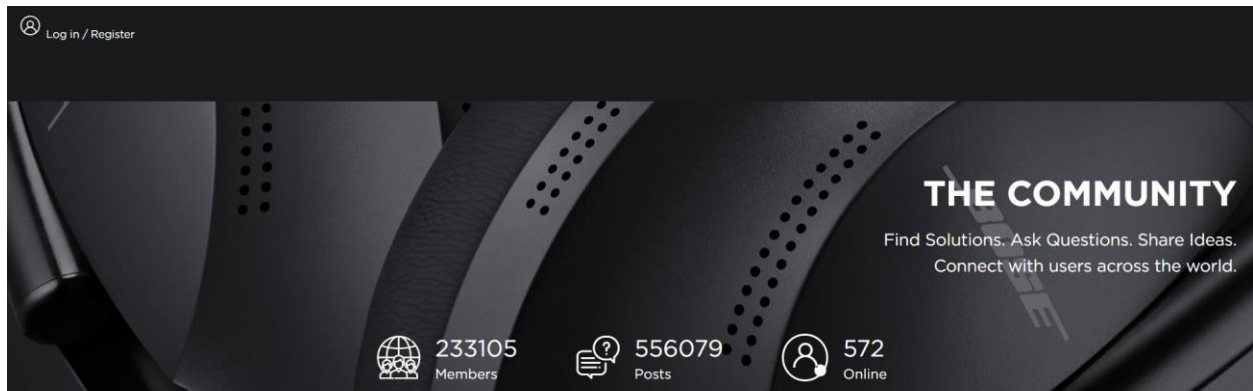
Build a Rest API to get count of stats to be shown on homepage.

Second Application (using Angular):

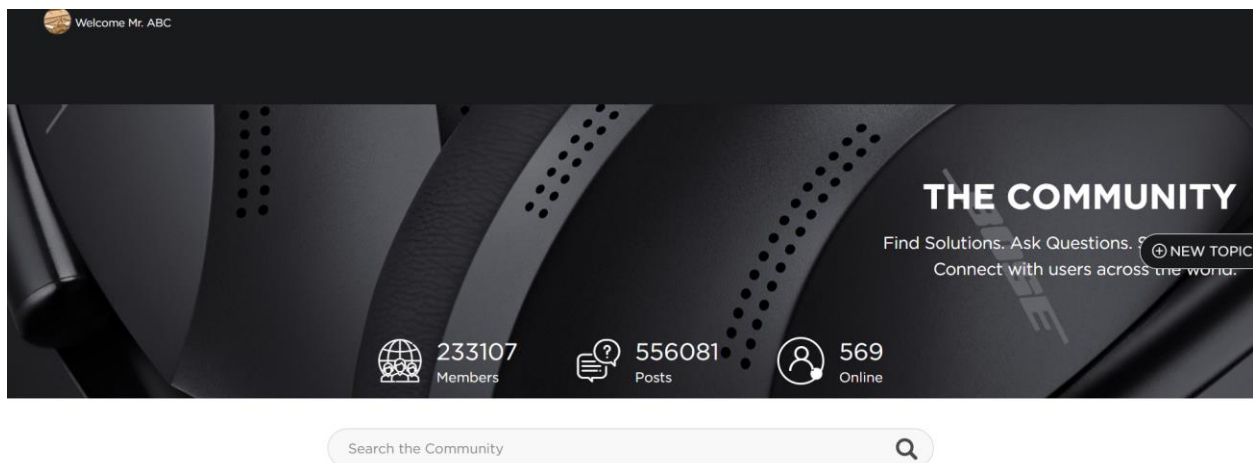
Technology stack: Latest version of Angular 11, Responsive design, SASS/LESS (Bonus), Angular CLI, npm

Homepage

Landing page should have links for Registration and Login. Homepage should show few stats such as number of registered users, number of products, number of reviews etc.



For a logged in user, Homepage should show a search box with which a customer can search for a product.

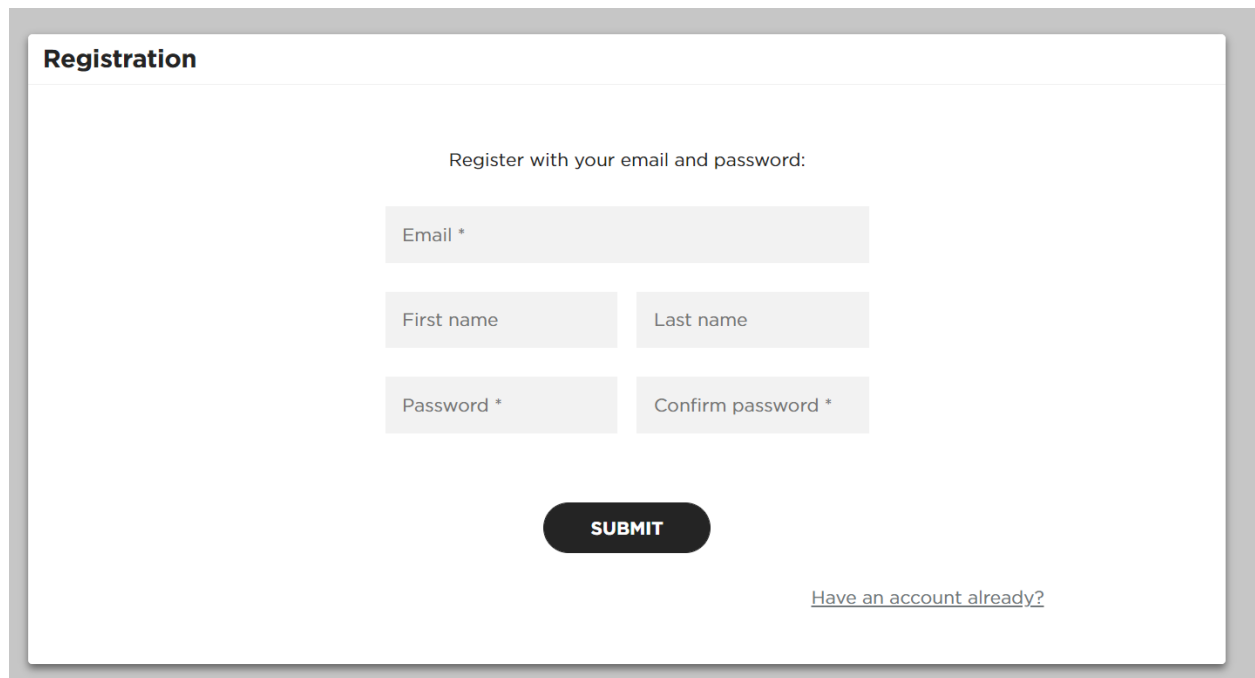


Login Page

The application should have a login page. User should be able to login. Proper error messages should be handled in case of invalid authentication.

Registration Page

The application should have a registration page. User should be able to register. Validation for email address, password policy, mandatory fields such as name etc. should be performed.

A registration form titled "Registration" with a subtitle "Register with your email and password:". The form contains five input fields: "Email *" (single line), "First name" and "Last name" (two columns), and "Password *" and "Confirm password *" (two columns). Below the fields is a dark "SUBMIT" button. At the bottom right is a link "Have an account already?".

Registration

Register with your email and password:

Email *

First name Last name

Password * Confirm password *

SUBMIT

[Have an account already?](#)

Product search page

Search screen should search the input text against all input parameters specified in as name, code, brand

Results Page

Once Search is triggered after specifying the search parameters, It will display the search results using the API from 1st Application.

The user should be able to see the products and filter them further by name, code, brand. User should be able to see all products with average ratings, number of reviews etc.

If no results are found, error messages should be displayed.

Ask for reviews.

User can raise for reviews on a product by submitting a simple form with fields such as: Product name, code, brand. If a product code already exists in a system, it should provide an error message that the product already exists and should redirect to the product review page after 30 seconds.

Product details

User can select a product and view all reviews provided for that product.

Post a review.

User can post a review to a product by submitting a simple form with fields such as ratings (1-5), heading and review (min. 20 characters and max 400 characters)

[BONUS] Admin

Logged in administrator can view all the new reviews and either approve or reject them. Only approved reviews are then visible on storefront and average ratings are calculated from approved reviews only.

Logout

User should be provided with an option to logout.

Some clarifications:

- can design pages as per your creativity.
- Proper coding conventions should be followed.
- Concepts such as interceptors, guards etc. should be implemented.
- Both applications should be fully functional

Timelines:

7 PDs

Evaluation Criteria

1. Rest API implementation completeness.
2. Authentication
3. Adherence to Restful APIs best practices.
4. Integration between Angular and Rest API for searching and displaying the results.
5. Implementation of Logout functionality
6. Adherence to best coding and design practices for both the applications.
7. Bonus points will be awarded for bonus parts.

All the Best!!!