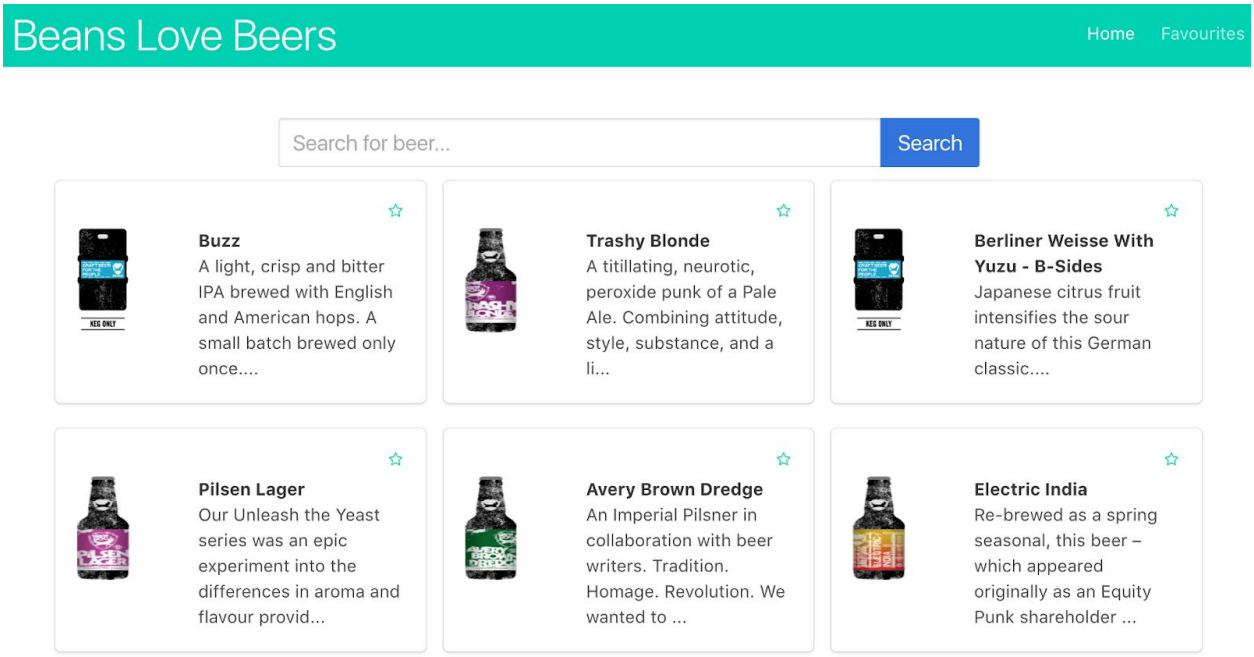
Coding Problem

Problem Statement:

Consider the following image carefully.



Please focus on building this web app/page piece by piece. Create the web-app in the following order:

- Develop your **responsive** frontend template in a grid layout of cards with the elements shown in the image.
- Each card contains **thumbnail image** of the product, **name**, **star button**, and **description**.
- You can use any thumbnail images of your choice.
- Add **favourite item** functionality which allows you to favourite a card of a product, using ‘star’ button.
- Create a separate route named as **/favourite** to display all the favorited cards
- Implement the **search** feature based on the name of the product, to search for a product. Search results should be displayed in a grid layout on the same page itself.

For REST APIs and Services:

- Create a **REST API** which would give the list of all the products to the client(front-end) via **GET** method.
- Endpoint name: **/products**
  Method: **GET**
- Use this API endpoint: to fetch all the products data.

- Create a **REST API** which would take product name as **query param** and would fetch the data of the searched product.
- Endpoint name: **/search?name=dummyname**
  Method: **GET**
- Use this API endpoint: to search any product's data.

- Create a **REST API** which would receive the product ID from the **request** body and would mark that product having the same product ID as favourite. Data.
- Endpoint name: **/setfavourite body: {'product_id': 'dummy ID'}**
- Method: **POST**
- Use this API endpoint: to mark any product as favourite.

- Create a **REST API** which would only fetch the list of **favourite products** which have been marked as favourite.
- Endpoint name: **/favourites**
  Method: **GET**
- Use this API endpoint: to fetch only the list of favourite products.

NOTE:

1. The assignment comprises of both frontend and backend APIs.
2. Completing both backend & frontend is mandatory and will give you **extra points**.
3. If in-case you're not able to finish up the backend part, you can use mock APIs to populate the front-end or JSON based model to emulate HTTP methods(GET, POST).

4. You can use **any tech stack or languages** to build the front-end and backend.

5. Recommended tech stack for Front-end: **ReactJS** or **Angular**, or **Simple JavaScript**, **HTML**, **CSS, Jquery.**
6. Recommended tech stack for Back-end: **Node.js** or **Express.js** or **Django** or **Django Rest Framework**, or **JAVA spring**.

Submission Guidelines:

Once you have completed your solution, please follow these guidelines to submit it.

- You are required to submit the assignment in a **.zip** file in a way, it is viewable from the folder itself.
- Send your solution via email to cv@navigus.in and attach the **.zip** file of the project as an attachment.
- NOTE: The name of the attached .zip file should follow this pattern: **firstname_rollnumber.zip**
  Example: **rahul_16CSE8987.zip**

- Submissions without a proper zipped attachment or name would get disqualified automatically.