



# E-Commerce Website

**Archit Sagar**

**Ashwin K**

## Overview:

Serve multiple users searching for products while also browsing and viewing products based on functionality like category and sorting. The Search/sort functionality is powered by the Unbxid API, while the Category functionality is powered by the API used to connect to the database.

## Description:

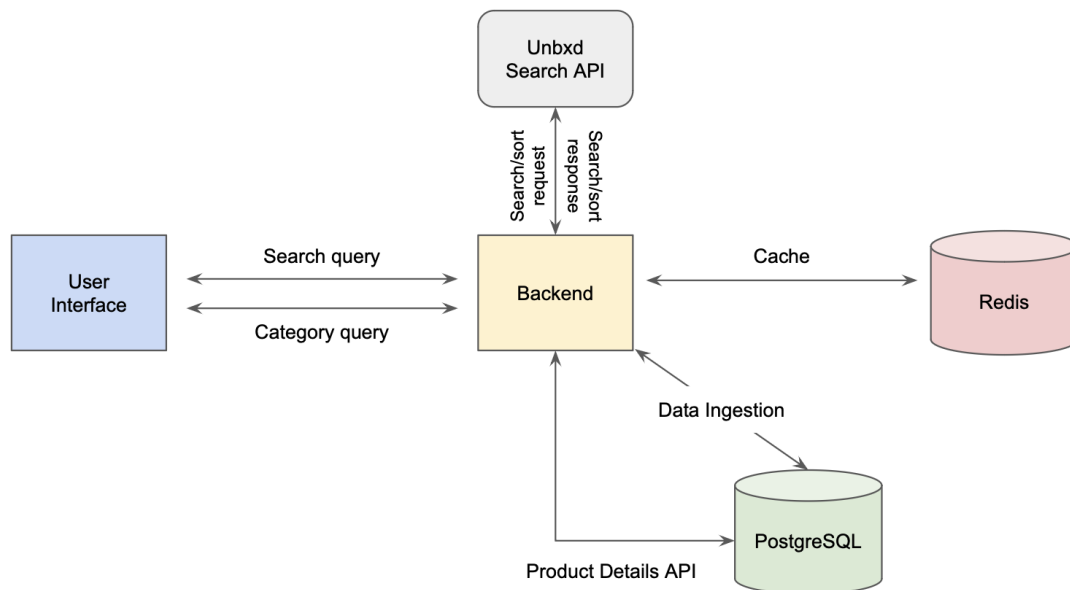
- Any user can view the products.
- Only one user role - Visitor.
  - Visitors can search for a product.
  - Visitors can view a particular product.
  - Optional - products can be recommended to users based on session actions.

## Technologies:

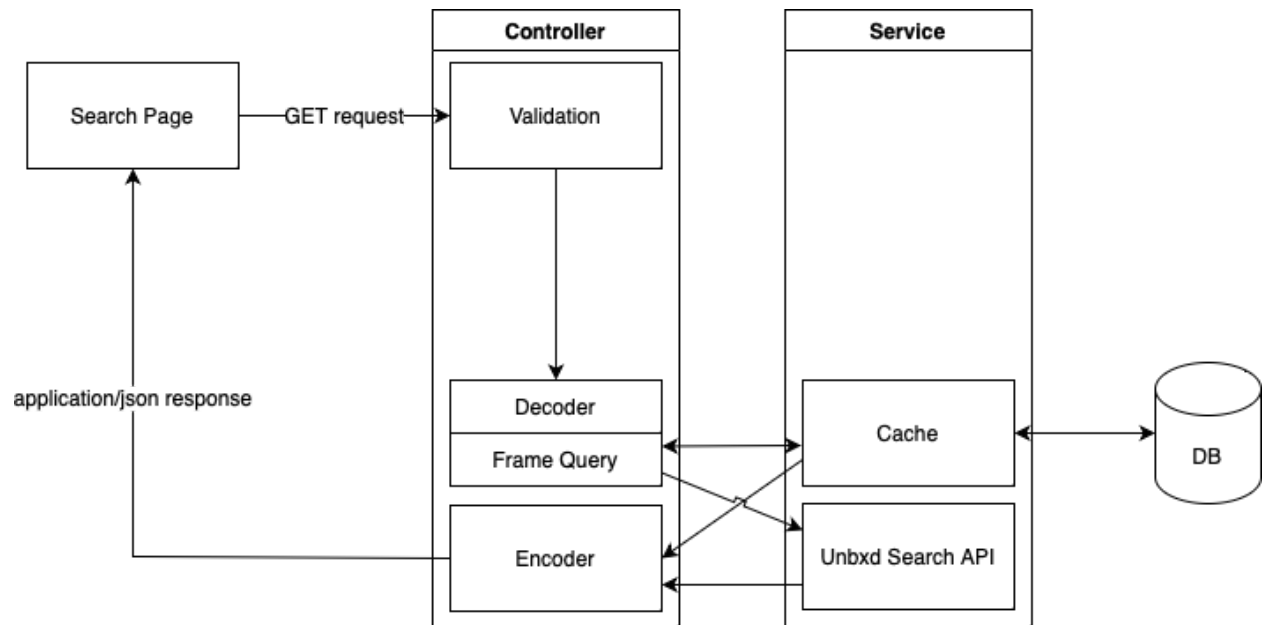
- Frontend - HTML, CSS, ReactJS
- Backend - Python - Flask
- Database -
  - Catalog information - PostgreSQL
  - Caches - Redis
  - Deployment - Kubernetes

## Project Design:

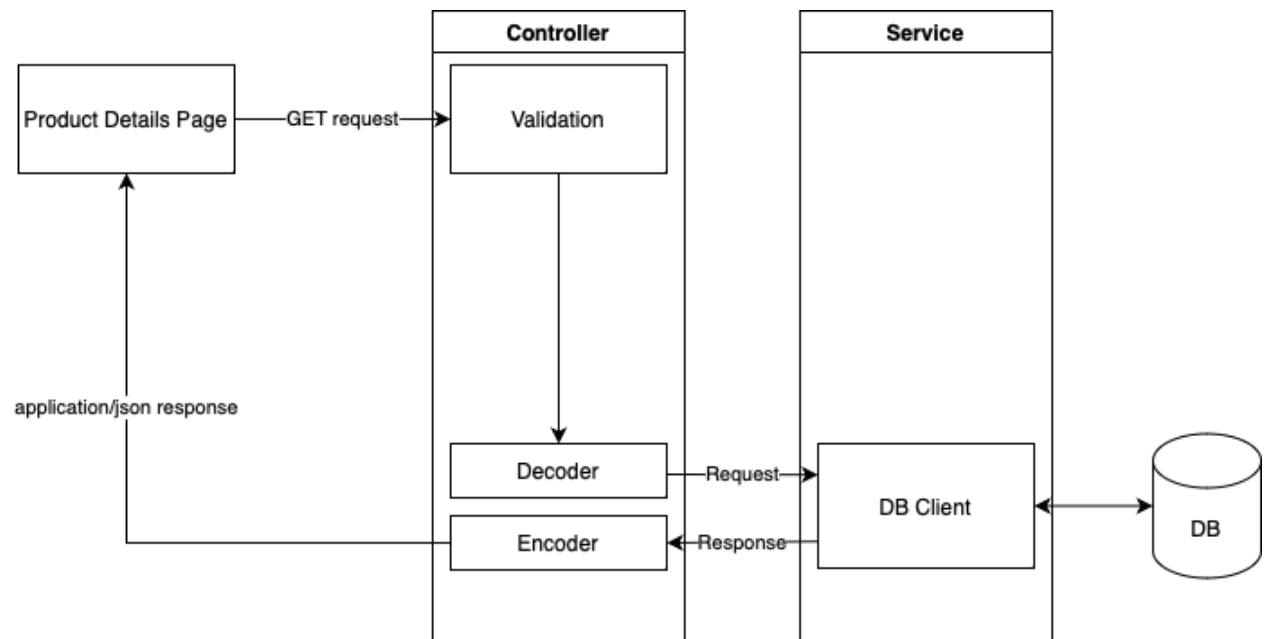
Application Diagram:



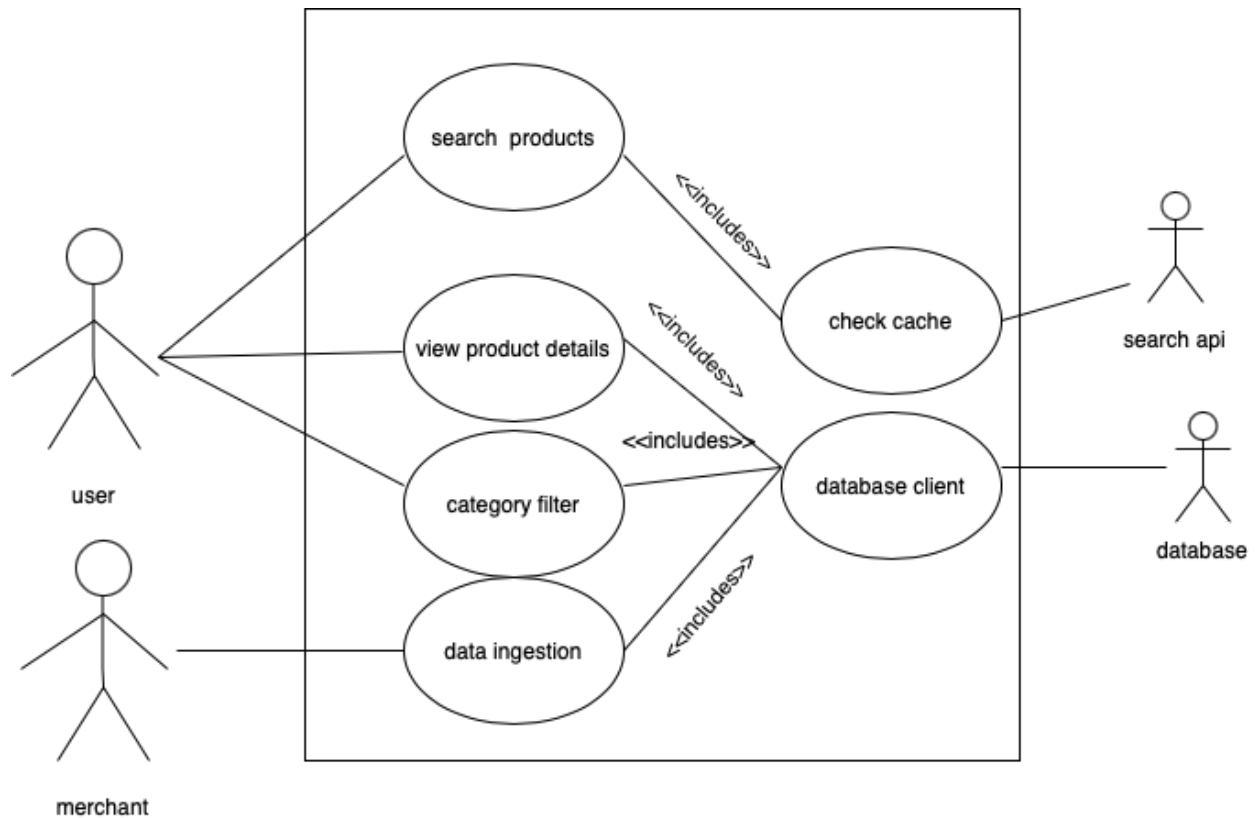
## Search Page:



## Product Details Page:



Use case diagram:



## E-Commerce Website Details:

- Search page:
  - Query typed in the search bar
  - Based on the query, get the relevant products' information using the Unbxid API.
  - Display the products on the search page.
- Category page:
  - 2-3 category levels are present.
  - Given the selected category values, fetch the relevant products from the database.
  - Sort the products based on the price in case the Sort functionality is used.
- Product details page:
  - Display product details like title, price and description.
  - Optional - retrieve relevant products based on user actions and display it at the bottom of the page.

## Data Management

- Data Description:

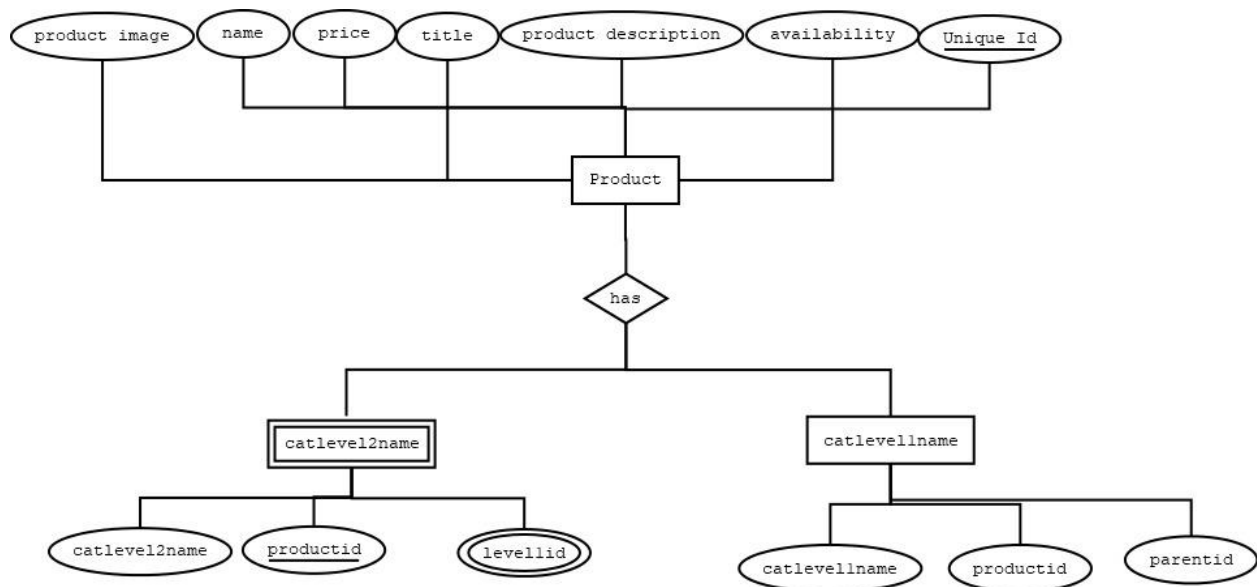
The database consists of:

- Products: Information about all the products is stored with each product having a unique ID.
- Category levels: Information about category level each product belongs to. Products are divided into 3 category levels - Category level 2, Category level 3 and Category level 4.

- Data Objects:

- Products: ID, name, price, title, product description, availability, product URL, product image, sku.
- Size: ID, size.
- Color: ID, color.
- Gender: ID, gender, catlevel1name.
- Category: ID, category.

- Database Relationships:



## API Specifications:

API for ingesting a json of catalog information:

PUT:

`PUT /ingestion HTTP/1.1`

`Host: 127.0.0.1`

`Content-Type: application/json`

```
{
    "Data": "insertion successful"
}
```

API to view product details:

GET:

`GET /category/02811264 HTTP/1.1`

`Host: 127.0.0.1`

`Content-Type: application/json`

```
{
    "price": "19.9",
    "title": "bar graphic crew neck t-shirt",
    "image-url":
"https://images.express.com/is/image/expressfashion/0023_02811264_0935
?cache=on&wid=361&fmt=jpeg&qlt=75,1&resmode=sharp2&op_usm=1,1,5,0&defa
ultImage=Photo-Coming-Soon"
}
```

**Github Repository:**

<https://github.com/unbxdinternship/ecommerce-website>