# # ZomatoApp - A Case Study

**## Problem Statement**

Build a system to search for a Zomato Food with price, get Food and price statistics and add Food and prices to Bucket list.

# ## Requirements

find Food price, add Food price to Bucket list.

**-** Logout feature should be implemented.

* User can add a Foodprice to Bucket list and should be able view the Bucket l ist.
* JWT authentication (Optional)
* Sorting and pagination (optional)

# ## Modules

**### (User interface) - should be able to**

* + Register user, login and logout
  + Search a Food price by name
  + View Food price statistics
  + Add a Food price to Bucket list
  + should be able to see Food by prices

# ### UserService

* + Creating a server in Asp.net

Core to facilitate user registration and login using Sql Server

* + should be able to manage user accounts.

# ### BucketService

* + should be able to store all the Bucket Food prices for a user

# ## Tech Stack

* + - Asp.net Core API
    - Sql Server
    - Entiry Framework Core
    - React

# ## Flow of Modules ### Building

* + - * Register/Login
      * Search for a Food price
      * Food price list - populating from external API The application needs to work with Zomato states from

[https://rapidapi.com/axesso/api/axesso-Zomato-data-service](https://rapidapi.com/axesso/api/axesso-walmart-data-service) and price management - [https://developer.Zomato.com/api/us/mp/price](https://developer.walmart.com/api/us/mp/price)

# ### Building the UserService

1. Creating a server in Asp.net

Core to facilitate user registration and login with Sql Server as backend. Upon login.

1. Writing swagger documentation
2. Unit Testing of the entire code which involves the positive and negative scenarios

# ### Building the Bucket Service

1. Building a server in Asp.net

Core to facilitate CRUD operation over Bucket Food prices stored in Sql Server.

1. Writing Swagger Documentation
2. Write Unit Test Cases and get it executed.

# ### Demonstrate the entire application

* 1. Make sure all the functionalities are implemented
  2. server side (For all layers) codes are unit tested.