E-SHOPPING

Contents

[1 Problem Statement 2](#_Toc108597655)

[2 WIREFRAMES 3](#_Toc108597656)

[3 Business Requirements: 4](#_Toc108597657)

[4 Proposed Rest Endpoints to be exposed 5](#_Toc108597658)

[4.1 Rest APIs: 5](#_Toc108597659)

[5 Key Rubrics/Expected Deliverables 5](#_Toc108597660)

[6 Platform 5](#_Toc108597661)

# Problem Statement

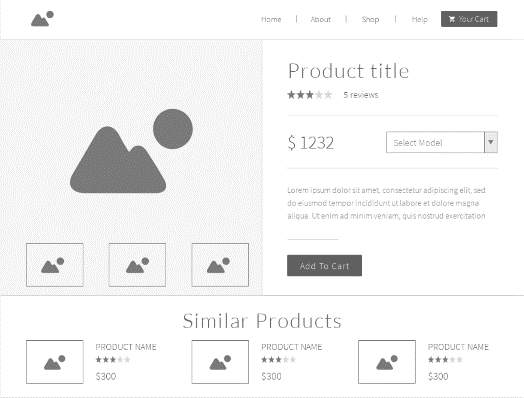
Build a software system which lets user search for a product and purchase it & includes Admin related activities. User can also cancel the order.

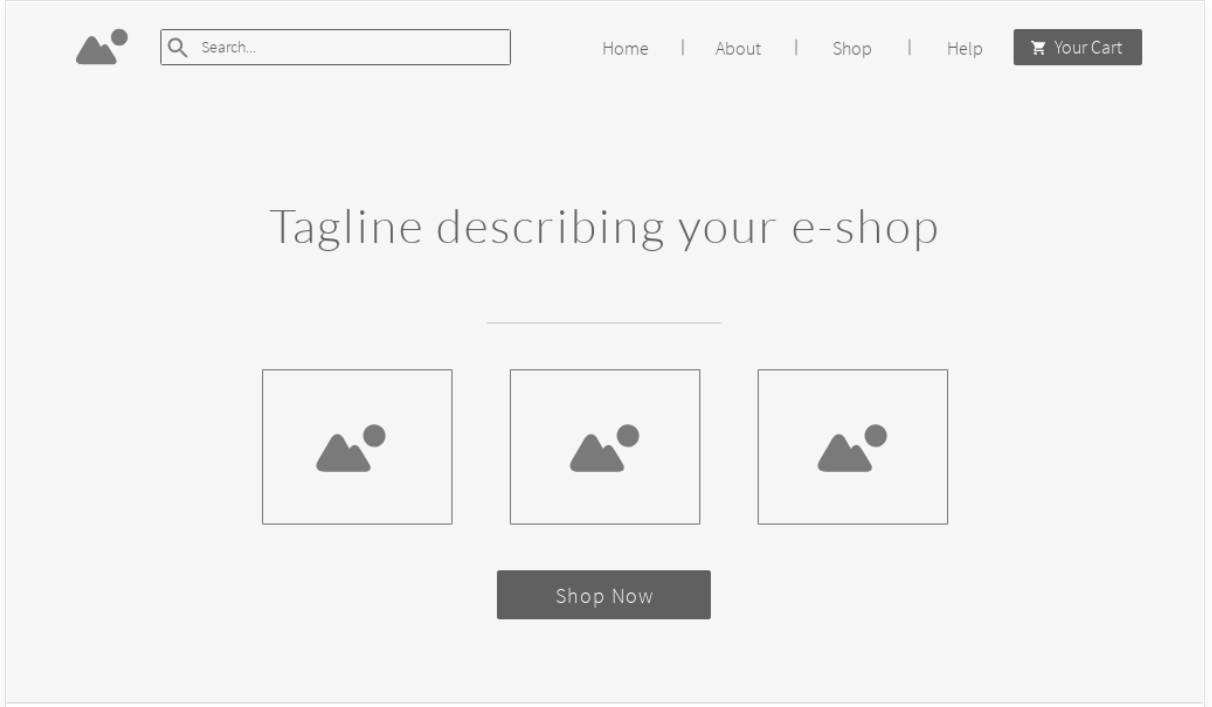
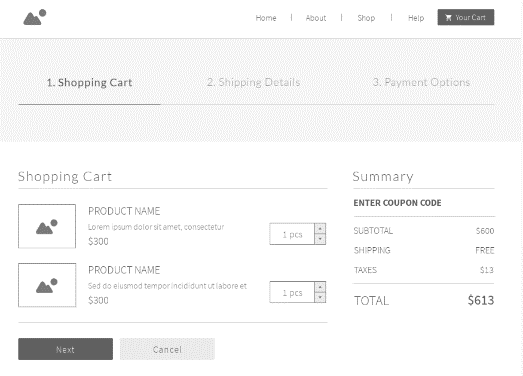
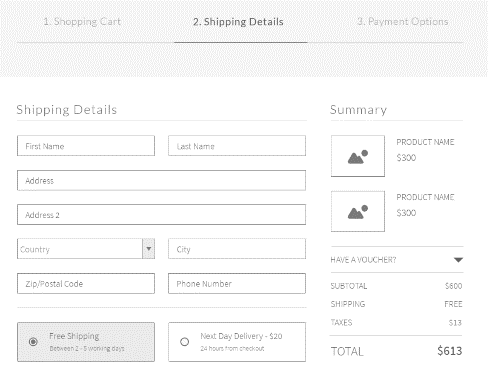
Below are the different roles, which need to be supported by above Software System.

1. User
2. Admin

The scope includes developing the application using dot net core as the backend

# WIREFRAMES

UI needs improvisation and modification as per given use case.



# Business Requirements:

As an application developer, develop microservices with below guidelines:

|  |  |  |
| --- | --- | --- |
| User  Story # | User Story Name | User Story |
| US\_01 | User Mode | 1. User can search for a Product based on any text 2. Each Search result need to display Product details, product Name/image, Price (to & round trip – TBD) 3. From Search results, User should be able to select a product and go ahead and complete order Booking by providing below details  * Name and Email ID * Quantity. * Address.  1. On successful Order Placement, OrderID need to be generated 2. With Order ID user should be able to  * view Orders placed, * Cancel an Order.  1. With OrderID view the order details |
| US\_02 | Admin Mode | 1. Admin shall be able to login/logout. 2. There can be pre-defined username/password for Admin. 3. Admin shall be able to add/block a product. When product is Blocked, product will not be shown in Search results. 4. Admin shall be able to add Inventory of an existing product by specifying below details:  * Product ID  1. Admin shall be able to add product with below details: -  * Product name * Product Description * Category * Product discount * Product MRP * Product selling price |

# Proposed Rest Endpoints to be exposed

## Rest APIs:

|  |  |  |
| --- | --- | --- |
| **POST** | **/api/v1.0/eshopping/user/register** | **New user register** |
| **POST** | **/api/v1.0/ eshopping /admin/login** | **Admin login** |
| **POST** | **/api/v1.0/ eshopping /catalog/inventory/add** | **Add Inventory** |
| **POST** | **/api/v1.0/ eshopping /search** | **Searches for product** |
| **POST** | **/api/v1.0/ eshopping /order/add** | **Place order** |
| **GET** | **/api/v1.0/ eshopping / order /{OrderID}** | **Get order details based on orderID** |
| **GET** | **/api/v1.0/ eshopping /order/history/** | **Get orders history** |
| **DELETE** | **/api/v1.0/ eshopping / order /cancel/{ OrderID }** | **Cancel an order** |

# Key Rubrics/Expected Deliverables

As an application developer:

* 1. Ensure layer project Structure with proper naming conventions and model classes.
  2. Use application.json file to maintain all configuration like connection string
  3. Implemented the layered structure - Controller, Interface, Service, DAO, Testing, Validation, Security etc
  4. Secure all Rest End Points by configuring SSL Certificate for Cloud

# Platform

Use above Business Requirements to implement the below.

* + - 1. Use Azure to deploy application on cloud.
      2. Use Azure SQL Server as a database for the Application.
      3. Use Azure Functions and DB to build a backend process for handling requests for E-Shopping App.
      4. Use Azure Service to send email after order placement.

Note: Minimum 2APIs (UI+Backend) to be hosted in cloud