**SHOP-BRIDGE**

**(BACKEND SOLUTION)**

**Submitted By: - Abhay Kardekar**

**(**[**auk1113029@gmail.com**](mailto:auk1113029@gmail.com)**)**

**(8329756639)**

**Index**

1. Technology and Tools Used
2. Configurations
3. Assumptions
4. SQL Database Schema
5. Action methods/API
6. Other Operation Methods
7. SOP To Run And Use APIs
8. **Technology and Tools Used:**
   1. WebApi dotnet core - 5.0 framework
   2. C#
   3. MS SQL database
   4. Visual Studio 2019
   5. Postman
9. **Configurations**
   1. DB Connection String – In appSettings.json, Under “ConnectionString”:”Conn”
   2. Used required Nuget packages
   3. Localhost URL configured in launchSettings.json in project
10. **Assumptions**
    1. The designed DB schema contains the required master data for category and sub-category of products.
    2. Authorized user will only call the developed methods/APIs. User Authentication handling will be at webserver. In this scenario – Product Admin.
    3. Text logs generates at server level for hosted application/service.
    4. Category and Sub-Category of Product can’t be modified in Product Master.
    5. Category and Sub-Category IDs will be proper from front end dropdown lists while calling api’s. (As front end will get the list of categories and sub-categories as per point (a)).
11. **SQL Database Schema**
    1. **Tables:**
       1. tbl\_Product\_Master – Contains all product data
       2. tbl\_category\_master – Contains all categories data
       3. tbl\_sub\_category\_master – Contains all sub categories data
       4. tbl\_ShopBridge\_APILogs – Contains API wise request/response DB logs
    2. **Stored Procedures**
       1. usp\_add\_productdetails – adds product and image path details
       2. usp\_update\_productdetails – update product details
       3. usp\_update\_productImagedetails – update product new image path
       4. usp\_delete\_productdetails – delete product
       5. usp\_get\_productdetails – get list of all products
       6. usp\_get\_productImagePath – get existing product image path
       7. usp\_get\_subcategorydetails – get list of all subcategories
       8. usp\_InsertAPILogs – insert DB logs
       9. usp\_get\_categorydetails – get list of all categories
12. **Action methods/API**
    1. **GetProductList** – Default, Get all products list

**URL:** …/api/product

**Type:** HttpGet

**Response:** Json – List of product with details

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Success | Json List of products |
| E101 | No Product in Inventory |  |
| E102 | Exception |  |

* 1. **GetProductImage** – Get product image against product ID

**URL:** …api/product/GetProductImage

**Type:** HttpPost

**Request:**

{

“product\_id”:1

}

**Response:** File of type “image/jpeg”

* 1. **AddProduct** – Add new product to inventory

**URL:** …api/product/AddProduct

**Type:** HttpPost

**Request:**

{

  "product\_id": 1,

  "category\_id": 1,

  "sub\_category\_id": 1,

  "product\_name": "DINNING TABLE 2X4",

  "product\_desc": "2X4 CHAIRS WITH DINING",

  "unit\_price": 10500.00,

  "sell\_price": 10000.00,

  "discount": 0.00,

  "discount\_price": 0.00,

  "product\_img": null,

  "stock\_qty": 20,

  "created\_by": "U202",

  "modified\_by": "U202"

}

**Response:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Product added successfully! |  |
| 1 | Product already added! |  |
| E101 | Input Field Validation. Mandatory fields/Invalid values - |  |
| E102 | Exception |  |
| E103 | Failed to get product added status |  |

* 1. **UpdateProduct** – Update product data fields

**URL:** …api/product/UpdateProduct

**Type:** HttpPost

**Request:**

{

  "product\_id": 1,

  "product\_name": "DINNING TABLE 2X4",

  "product\_desc": "2X4 CHAIRS WITH DINING",

  "unit\_price": 10500.00,

  "sell\_price": 10000.00,

  "discount": 0.00,

  "stock\_qty": 20,

  "modified\_by": "U202"

}

**Response:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Product updated successfully! |  |
| 1 | Product not present in inventory to update! |  |
| E101 | Input Field Validation. Mandatory fields/Invalid values - |  |
| E102 | Exception |  |
| E103 | Failed to get product update status |  |

* 1. **UpdateProductImage** – Update product image

**URL:** .../api/product/UpdateProductImage

**Type:** HttpPost

**Request:**

{

“product\_id”:1,

“modified\_by”:”U203”,

“product\_img”:”<base64 of image>”

}

**Response:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Product image updated successully! |  |
| 1 | Product not present in inventory to update! |  |
| E101 | Input Field Validation. Invalid - |  |
| E102 | Exception |  |
| E103 | Failed to get product image update status |  |
| E104 | No image exists! |  |

* 1. **DeleteProduct** – delete the product from inventory

**URL:** …/api/product/DeleteProduct

**Type:** HttpPost

**Request:**

{

“product\_id”:1

}

**Response:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Product deleted successfully! |  |
| 1 | Product not present in inventory to remove! |  |
| E102 | Exception |  |
| E103 | Failed to get product delete status |  |

* 1. **GetProductCategories**

**URL:** .../api/product/GetProductCategories

**Type:** HttpPost

**Reponse:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Success | Json List of Product Categories |
| E102 | Exception |  |
| E103 | No Product Categories exists! |  |

* 1. **GetProductSubCategories**

**URL:** …/api/product/GetProductSubCategories

**Type:** HttpPost

**Request:**

{

“category\_id”:1

}

**Response:**

|  |  |  |
| --- | --- | --- |
| **ErrorCode** | **ErrorDesc** | **Data** |
| 0 | Success | Json List of Product Sub-Categories |
| E102 | Exception |  |
| E103 | No Product Sub-Categories exists! |  |

1. **Other Operation Methods**
   1. SaveImage – Save image to defined path
   2. GetImage – Get image for specific product from the respective path and filename
   3. WriteDBLog – Writes all developed API response logs with some unique identifiers.
2. **SOP To Run And Use APIs**