Details Steps

1. Created a database” Shop bridge” and created Inventory table

USE [ShopBridge]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Inventory] Script Date: 17-04-2021 16:13:15 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Inventory](

[ID] [int] IDENTITY(1,1) NOT NULL,

[ItemName] [varchar](50) NULL,

[ItemDescription] [varchar](50) NULL,

[ItemPrice] [varchar](30) NULL,

[ItemExpiryDate] [datetime] NULL,

[ItemCode] [varchar](30) NULL,

CONSTRAINT [PK\_Inventory] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

2.Created ShopBridgeApi Web API for all CRUD functionality for Inventory

--Used Entity Framework as ORM tool added through Nuget

Invoked below methods of api on call

GetInventories ()

http://localhost:64593/api/Inventories

GetInventory()

http://localhost:64593/api/Inventories/5

PutInventory ()

http://localhost:64593/api/Inventories/5

PostInventory ()

http://localhost:64593/api/Inventories

DeleteInventory ()

http://localhost:64593/api/Inventories/5

3.Created backend solution for Product functionality

1. Add a new item to the inventory (The item should require a name, description, and price as basic fileds, think of additional information that would be useful)

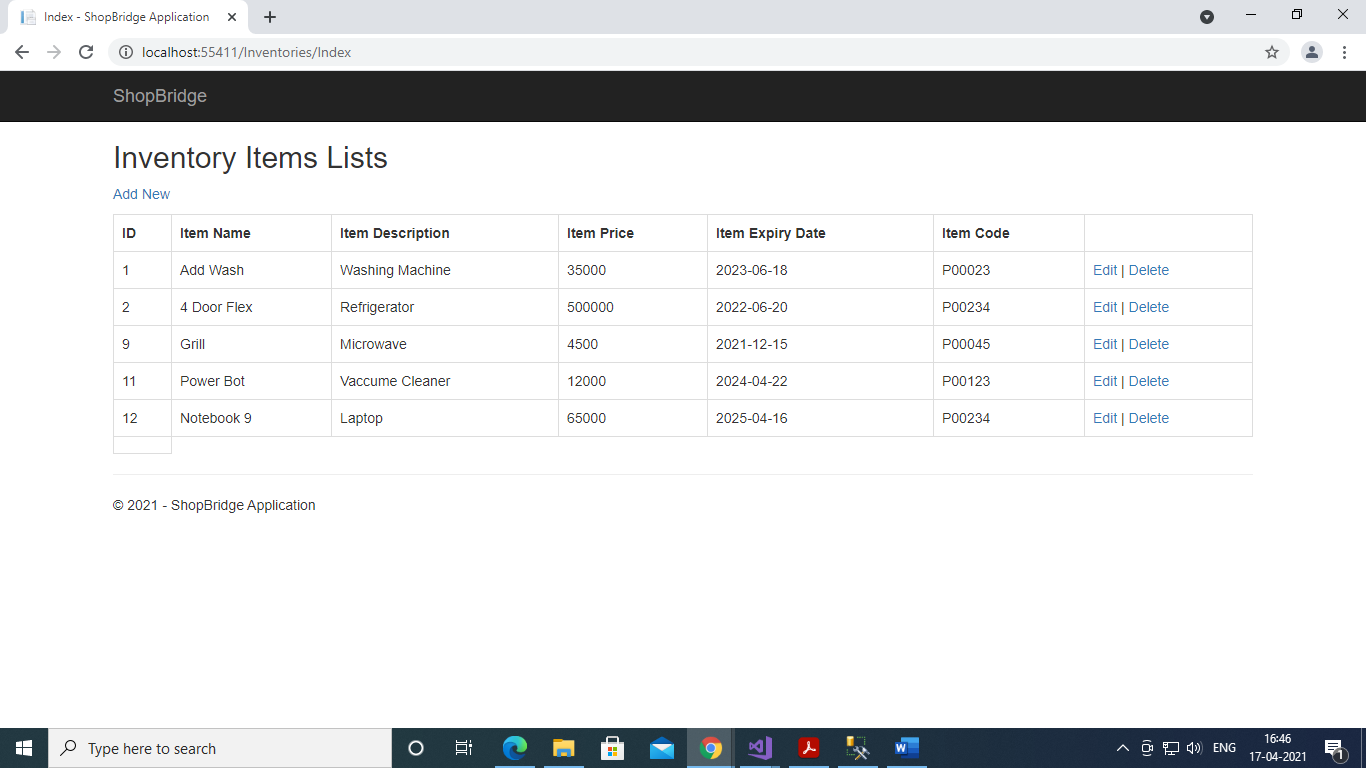
2. Modify an item in the inventory.

3. Delete an item from the inventory.

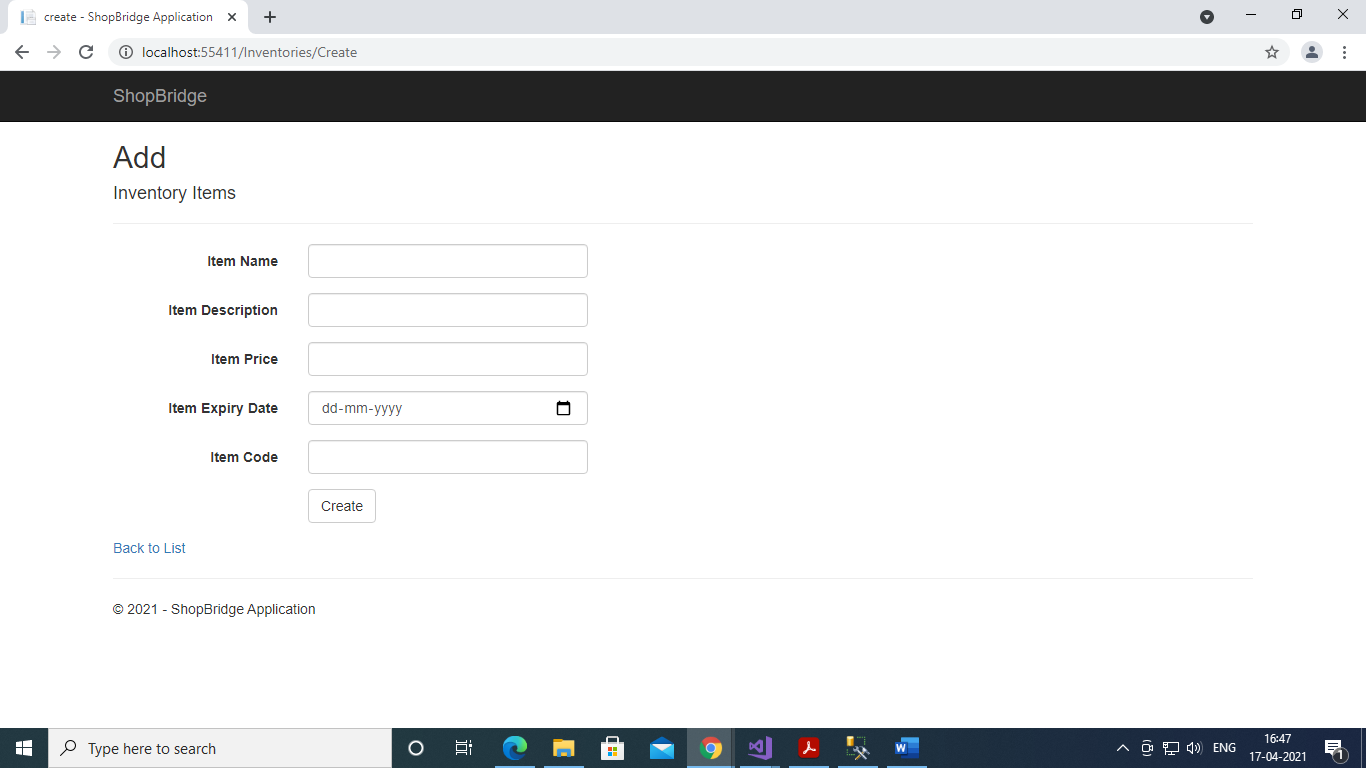
4. Lists the items in the inventory.

4.Consumed web API in front end application for performing all CRUD functionality.

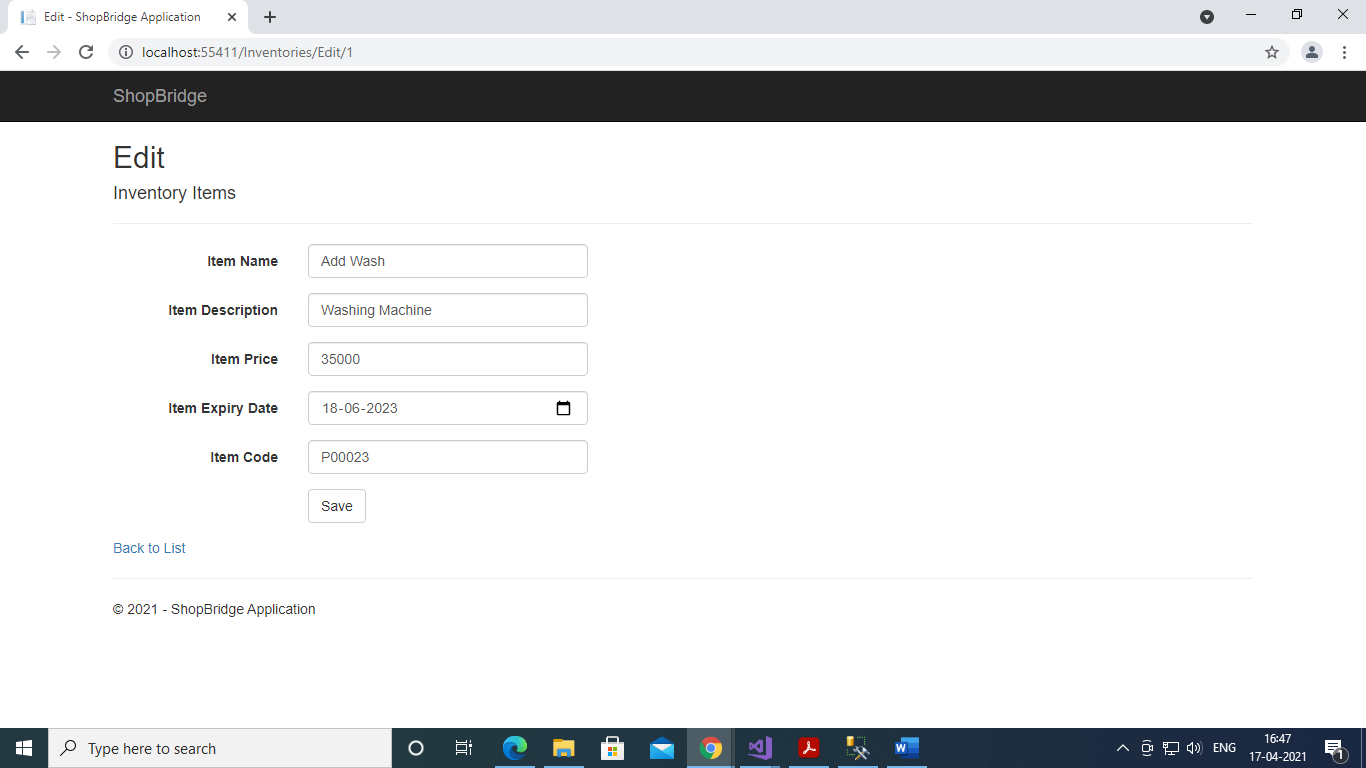
Screen Shot



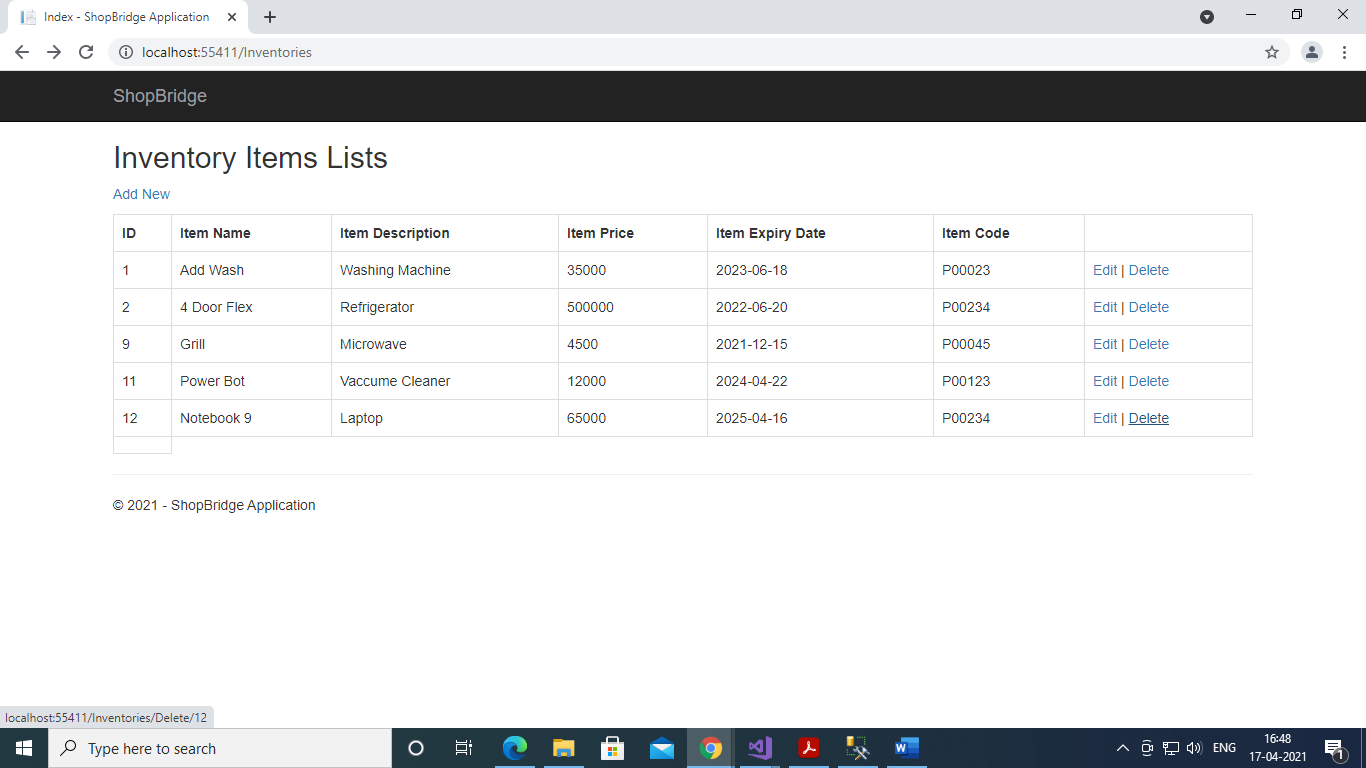
Add Inventory



Edit Inventory



Delete Inventory



.

1.Requiment Understanding 2 hour

2.Data store design 30 min

3.API and service logic 4 hour

4.Front end Design 3 hour

5. Unit Test Coverage 0