**Requirement**: Develop a site to maintain different items. Item contains **Name**, **Description**, **Price** and **Photo**. Here we can Add, Update, and Remove the items at any time.

**Technologies**: -

**Front end** -> Angular, Visual Studio Code

**Back end** -> C#, WEB API, LINQ, Entity Framework

**Database** -> SQL .mdf file

**Unit testing tools** -> **MS unit** for WEB API, **Jasmine** for frontend

Please find the below screenshots for the developed site. And the site name as **Shop Bridge**.

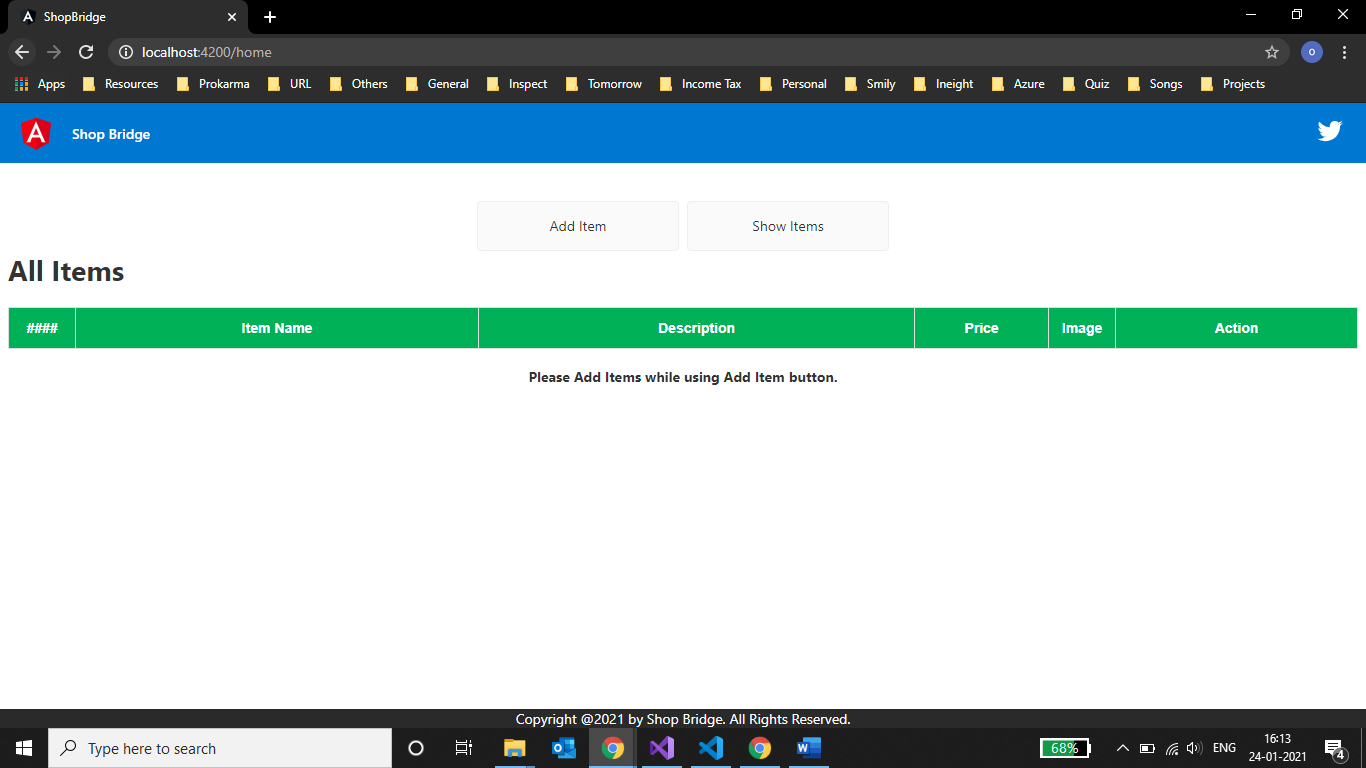
**Front-End**: -

* **Technology**: - Angular
* **IDE**: - Visual Studio Code

**Homepage**: while User navigate into site, he can see the home page as below. Here homepage contains header, body, and footer.

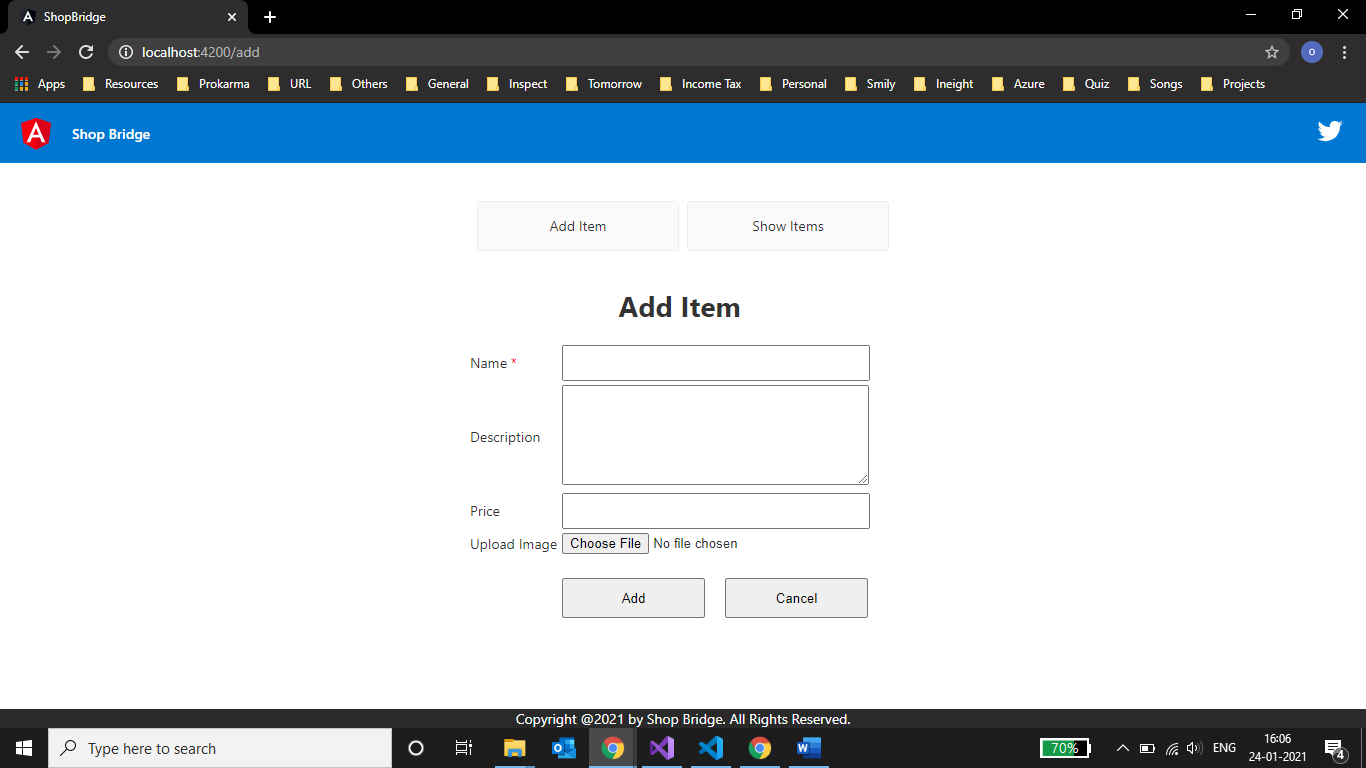
* **Header** section contains project/site title and twitter image to navigate to Think bridge site.
* **Body** section contains 2 buttons (i.e. Add Item & Show Items) and 1 Grid. Here we are using Grid to show the list of saved Items.
* **Add Item**: - while using this button we can add new items.
* **Show Items**: - while using this button we can navigate into home page to see all saved items in the grid.

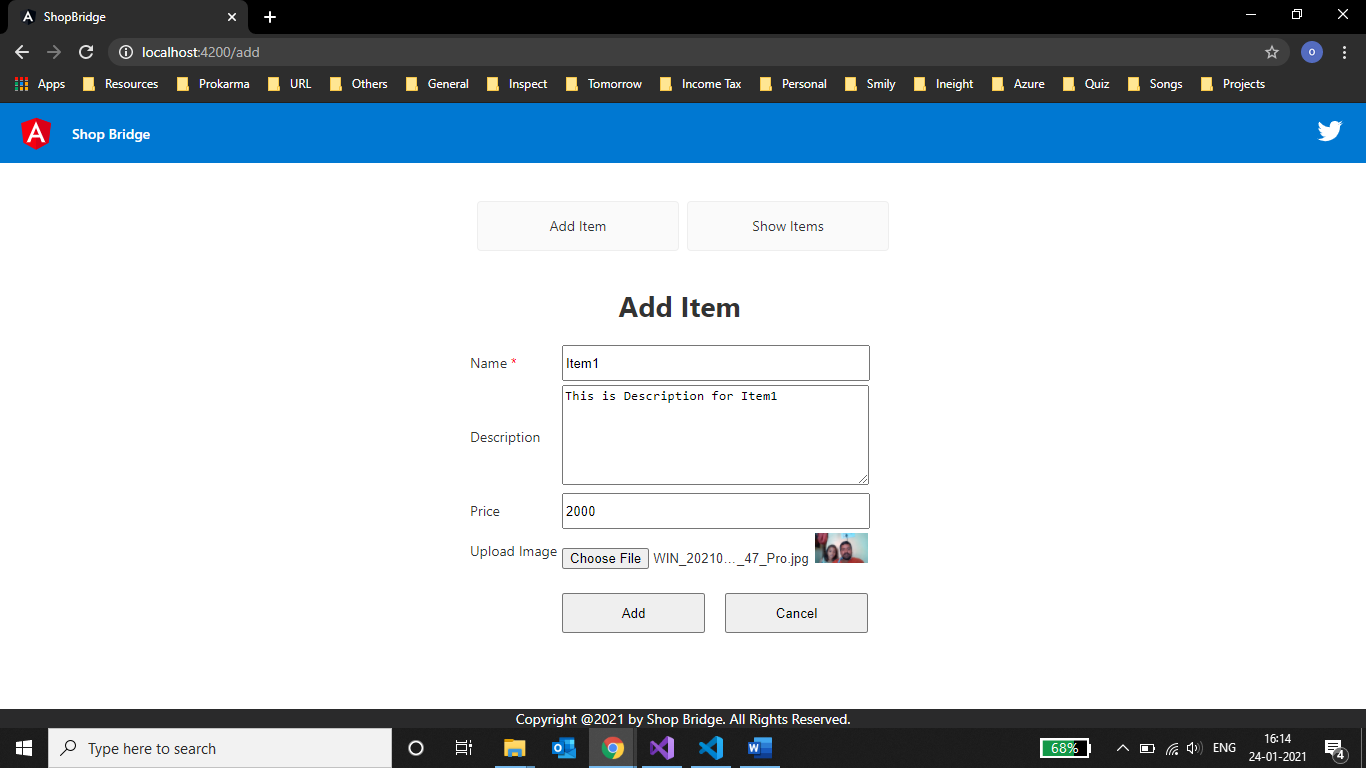
Note: - If grid is empty then we are showing a message on the Grid like *Add items by using Add button*.

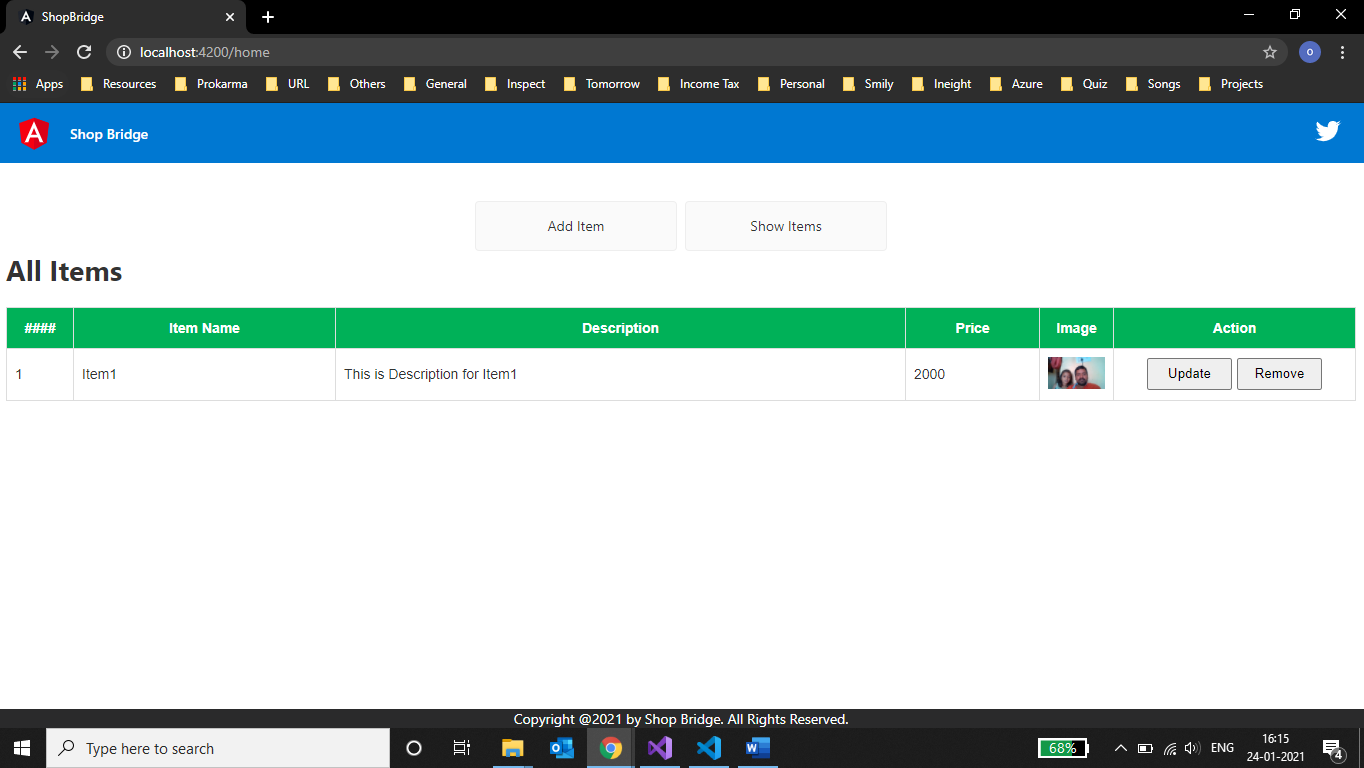


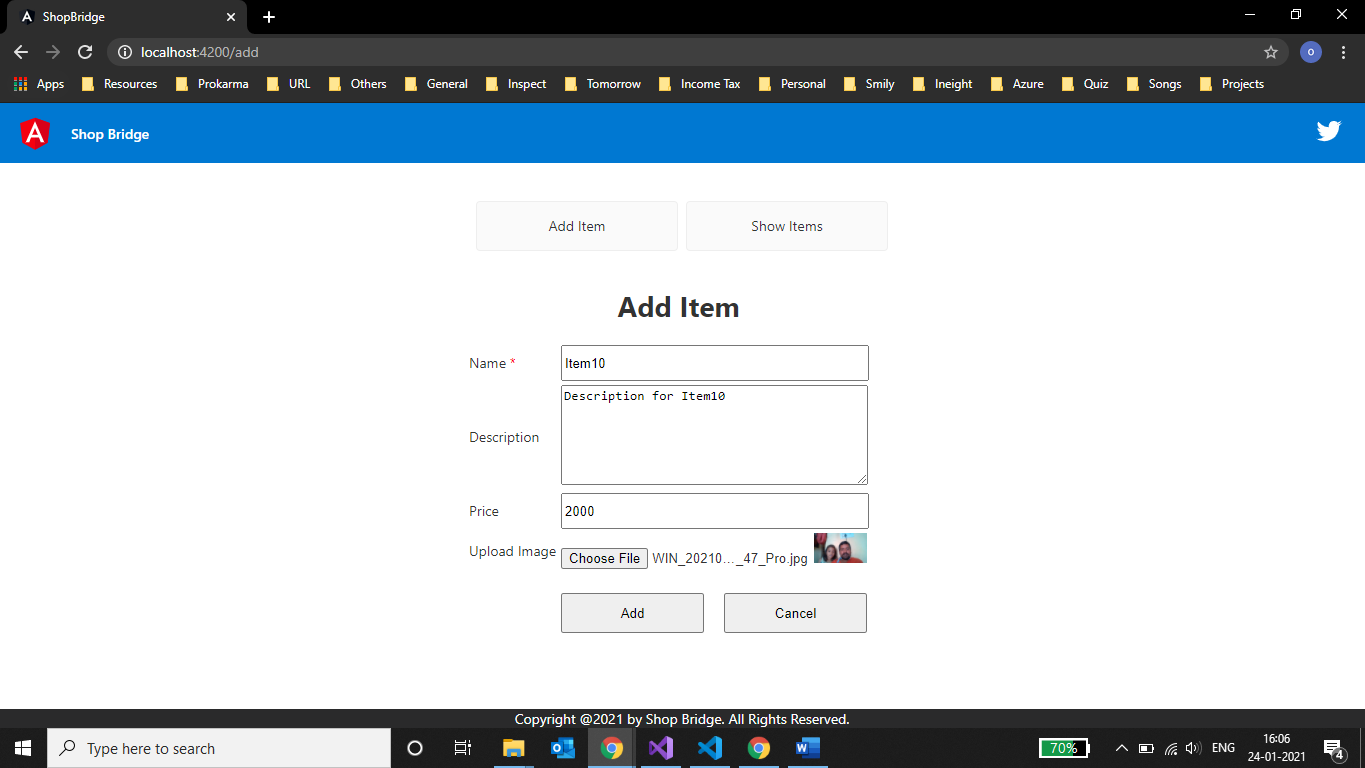
**Add Item**: - while selecting Add Item button, user can navigate into AddItem page. Here user can add new Item by providing Name, Description, Price, and Photo.

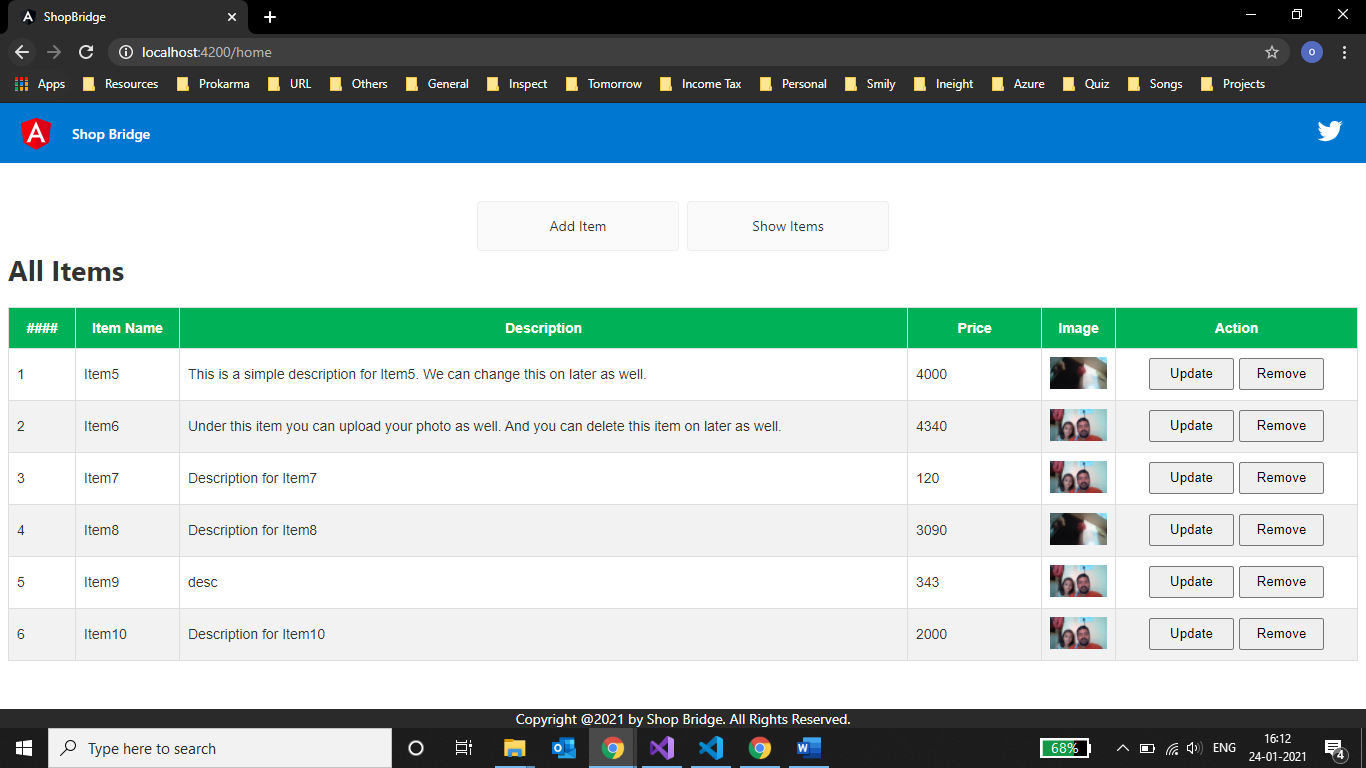
* **Name** is mandatory.
* **Price** will accept only numbers.
* While selecting a photo, preview will display beside choose button.
* This page has **Add** and **Cancel** buttons.
* If user click on **Add** button, then Item will save in database and display in grid in home page. Here user will navigate into home.
* If user click **Cancel** button, then page will dismiss, and user will navigate to home.
* By using **Add** button, user can add any number of items.



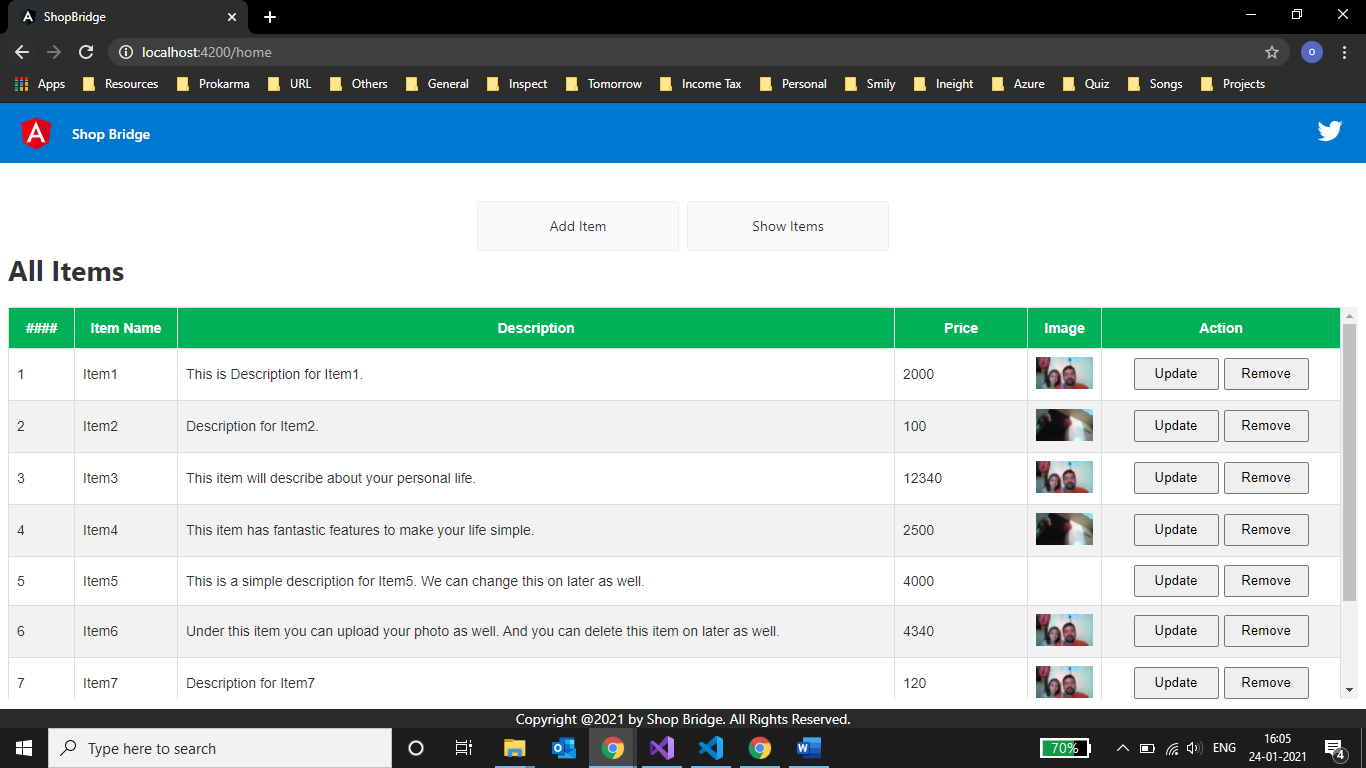






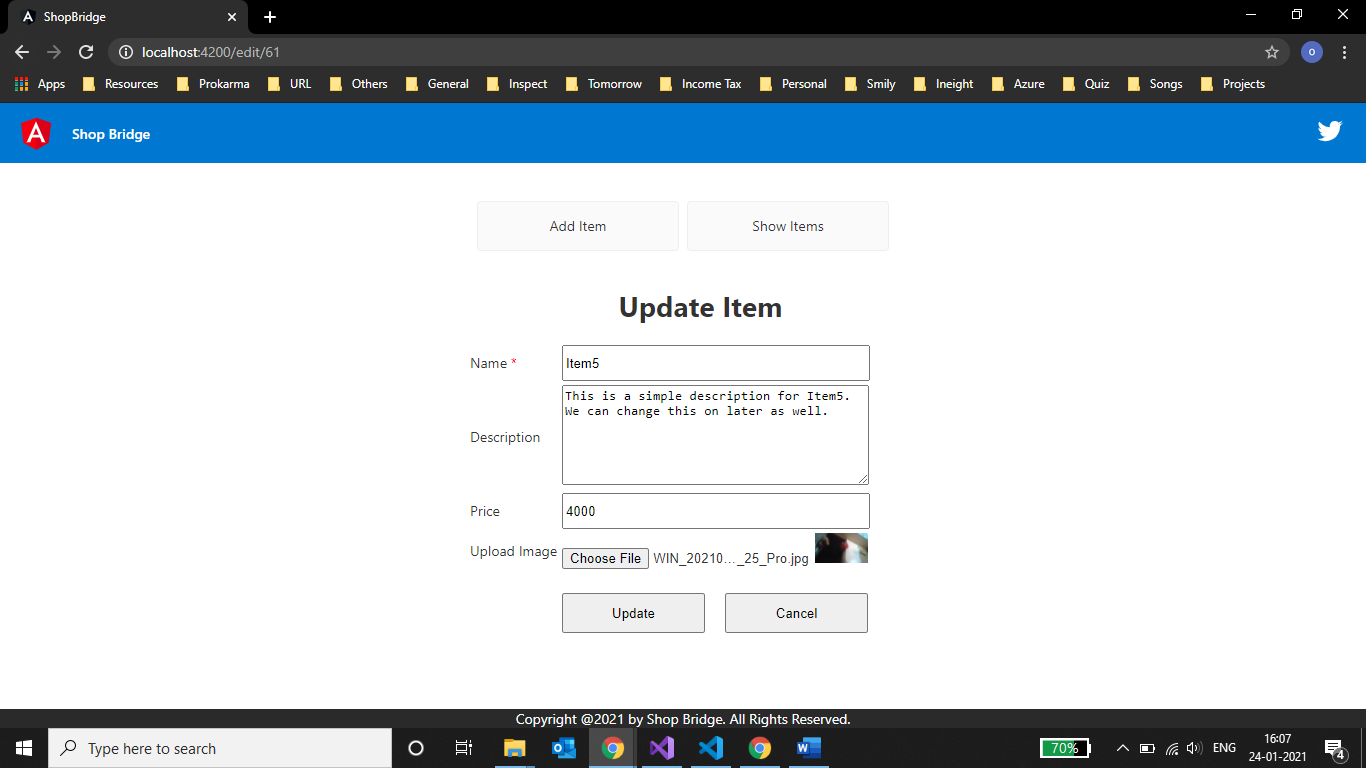


If user add more than 6 items, then vertical scroll will come in the grid as below,

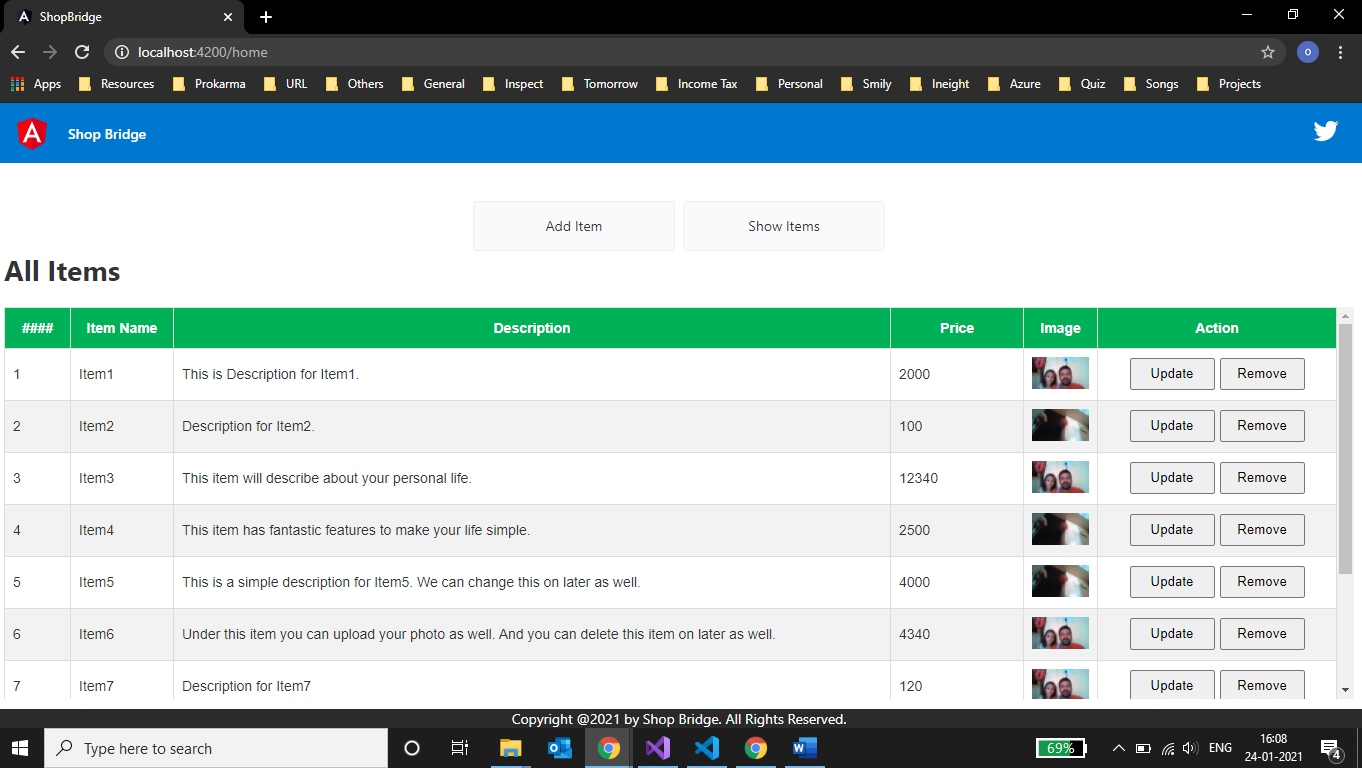


* User can Update or Remove the existing item.

**Update**: - If user wants to update any item, then he has to select **Update** button for that item in the grid. Here I am trying to update the photo for **Item5** as below,



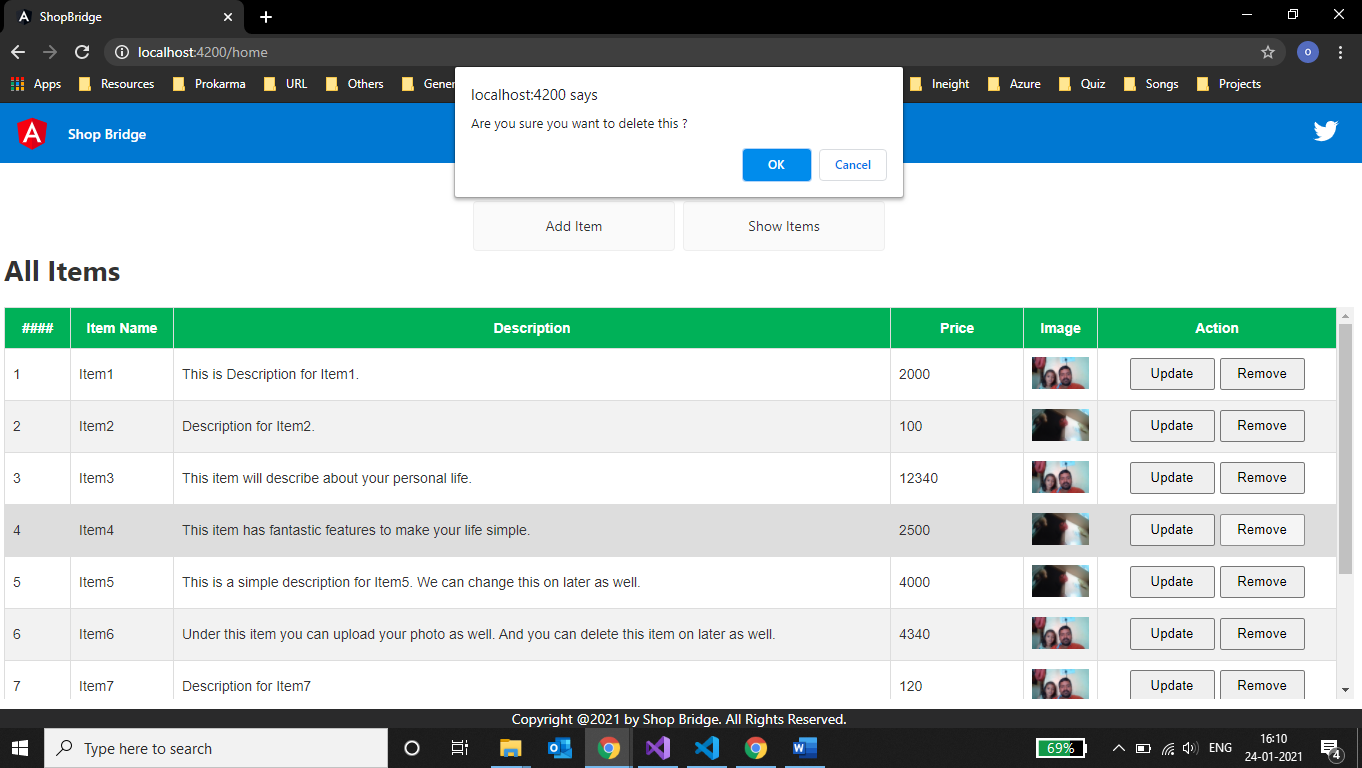
After click on Update button, changes will update in database and user will navigate into home page to see updates as below.



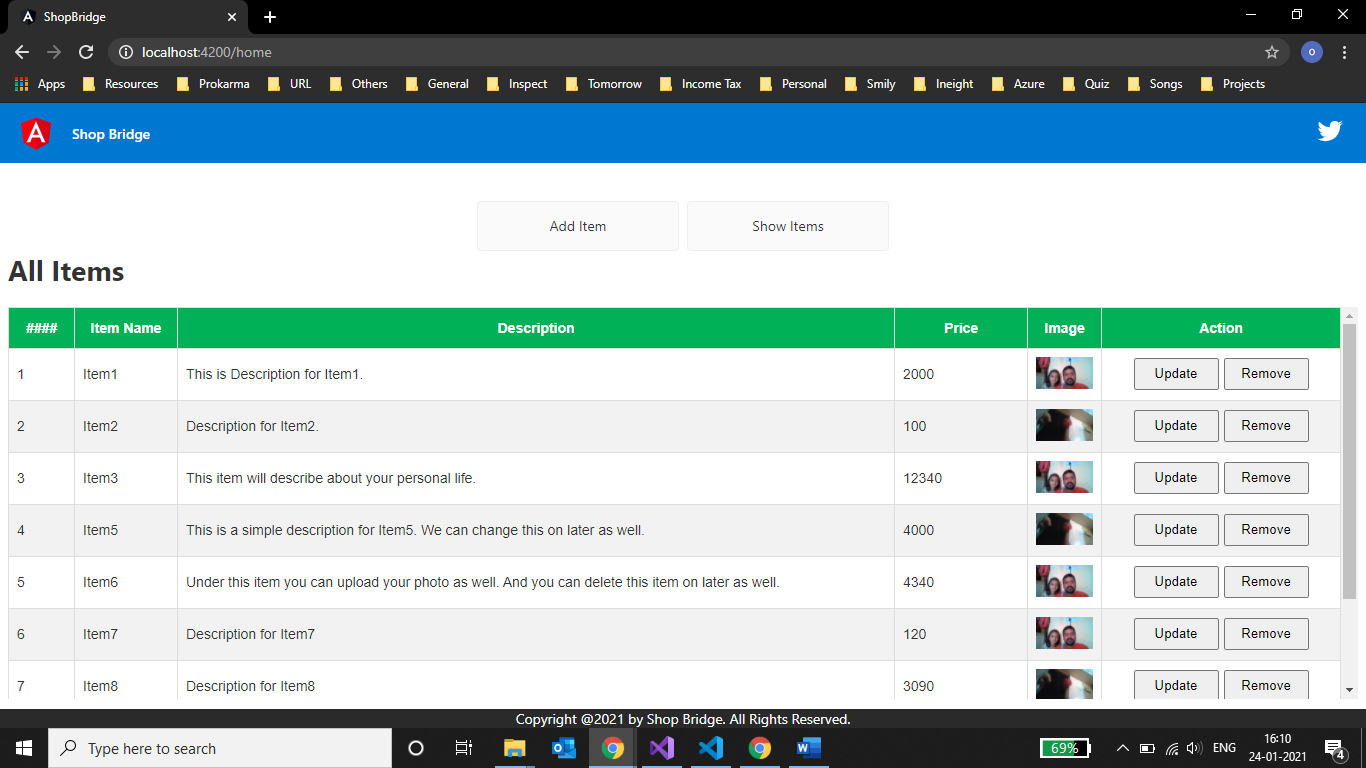
**Delete**: - If user wants to delete any item, then he has to select **Delete** button for that item in the grid. Here I am trying to delete the **Item4**.

Once click on delete button, then user will get the confirmation message to delete the item or not.

* If user will click **Yes**, then item will be deleting from database and grid will reload with updated data.
* If user will click **Cancel**, then item will not delete.



If delete option as **Yes**, then item will be deleted and showing grid as below, here **Item4** has deleted.



**Back-End**: -

* **Technology**: - C#, WEB API, LINQ, Entity Framework.
* **Database**: - SQL mdf file.

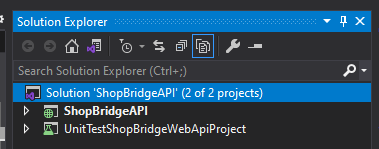
**ShopBridgeAPI** solution has 2 projects.

1. **WEB API** -> name as ShopBridgeAPI

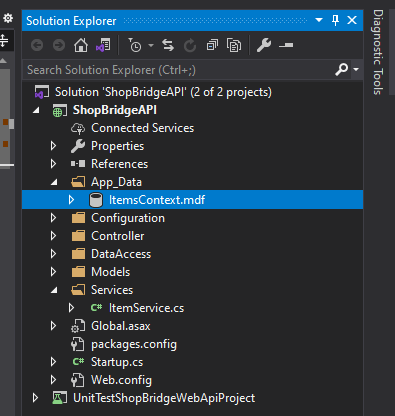
* C#, WEB API, LINQ, Entity Framework.

1. **Test project** -> name as UnitTestShopBridgeWebApiProject

* MS Unit



**Database design**: - For database, I used .mdf file (ItemsContex.mdf) and it is located under App\_Data folder in WEB API project as below,



CREATE TABLE [dbo].[Item] (

[ItemId] INT IDENTITY (1, 1) NOT NULL,

[Name] NVARCHAR (200) NOT NULL,

[Description] NVARCHAR (4000) NULL,

[Price] BIGINT NULL,

[IsActive] BIT DEFAULT ((1)) NOT NULL,

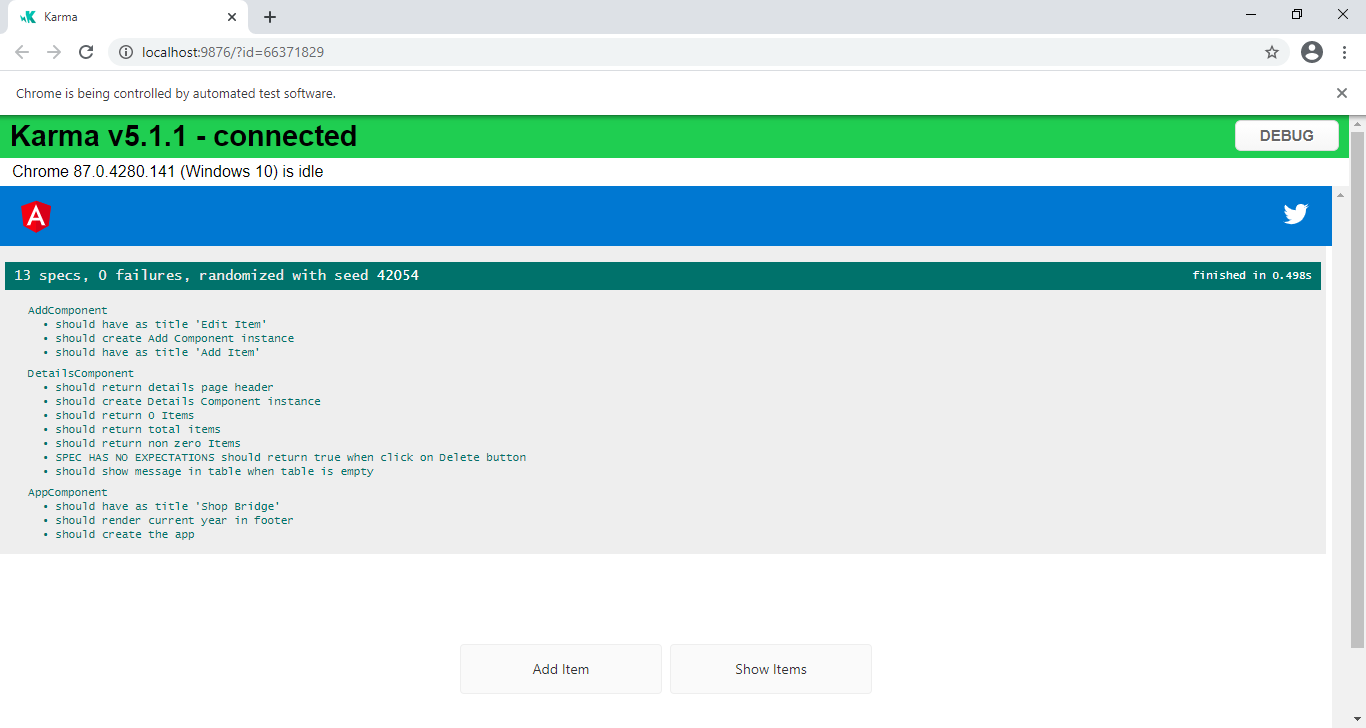
[ItemImageBase64] NVARCHAR (MAX) NULL,

PRIMARY KEY CLUSTERED ([ItemId] ASC)

);

**Unit test results**: -

* **Front end Unit test results**: -



* **WEB API Unit test results**: -

