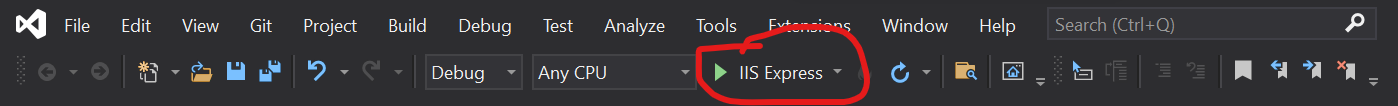
**User Guide and Time Track**

**Prerequisites: DOTNET Core 2.1, MS SQL Server**

1. **Steps to Run the API**
2. Copy the Solution folder named “ShopBridge” to your machine.
3. Open the File ShopBridge.sln in Visual Studio.
4. Click on Run Button which says IIS Express on the top middle of your screen in Visual Studio Editor.



1. Now you are ready to consume the API.
2. **Steps to consume the API or test it through postman.**
3. Download Postman from the Internet if you don’t have it already
4. After Installing it search postman on start menu and open it

Graphical user interface, text, application

Description automatically generated

1. Click on New

Graphical user interface, text, application

Description automatically generated

1. Select the Option HTTP Request

Graphical user interface, application

Description automatically generated

1. In the selected section you must paste the URL for testing the web API and you can also select the type of HTTP request you want to send from that panel

Graphical user interface, application

Description automatically generated

1. Important URLs for Consuming API are Listed here

Note: The port number in your URL may vary as you run the API on your system please change it accordingly

* To get the list of Products
* Type Of Request: GET
* URL: <http://localhost:29884/api/product>
* To get a single Product
* Type of Request: GET
* URL: <http://localhost:29884/api/product/id>
* Example: <http://localhost:29884/api/product/1>
* To add a Product
* Type of Request: POST
* URL: <http://localhost:29884/api/product>
* In addition to it you must provide the body to the request in the form of JSON object an example of which is in the screenshot below.

Graphical user interface, text, application, email

Description automatically generated

* To Update/ Edit a Product
* Type of Request: PATCH
* URL: <http://localhost:29884/api/product/id>
* In addition to it you must provide a body to the request in the format below

Graphical user interface, application

Description automatically generated

* Note: You can add more objects depending on the fields you want to update
* To Delete a Product
* Type of Request: DELETE
* URL: <http://localhost:29884/api/product/id>
* Example: <http://localhost:29884/api/product/1>

1. You can use either postman or you can configure the API calls from your frontend application to test the API.
2. **Time Taken for development of different modules**
3. Data Store Design

* Deciding Properties and creating Models – 20-25 Minutes
* Creating Entities – 5-10 Minutes
* Implementing Entity Framework Through code first approach – 15-20 Minutes
* Implementing Migration for Database Versioning – 5-10 Minutes
* Overall time taken – 1 Hour approx.

1. Creating and configuring Services

* Creation of Interface for Product Repository – 10-15 Minutes
* Implementing the Product Repository Interface – 20-25 Minutes
* Configuring Services Like Automapper and custom Services along with setting up Dependency Injection in startup class – 10-15 Minutes
* Overall time taken – 50 Minutes

1. Creating Controllers to handle HTTP requests and creating profile for Automapper – 1-1.5 Hours approx.
2. Testing the API throughout for different test cases and providing relevant comments to methods and classes – 1 Hour approx.
3. Total time taken to Develop the full API Solution – 4-5 Hours approx.