Here is how I approached the problem:

1. Install .NET using the given link in the PDF.

2. Learn basics & syntax of C#:

I used the following sources to get myself familiar with C#:

https://www.w3schools.com/cs : This website has been one of my favorites since Uni, as it explains the very basics quickly and goes through simple examples as well. I also learned basics of SQL before using this website.

https://www.youtube.com/playlist?list=PLdo4fOcmZ0oULFjxrOagaERVAMbmG20Xe : this playlist was on the .NET install guide which I found quite useful. I went through the first few videos as well as the OOP videos as in the process I learned OOP is quite fundamental for backend development.

3. Learn Basics of .NET and how to use it for Back-end Web Development as it was required in the given task:

https://www.youtube.com/playlist?list=PLdo4fOcmZ0oWunQnm3WnZxJrseIw2zSAk: I watched the first 5 videos of this playlist to familiarize myself with .NET basics and how to build a simple API using this framework.

https://www.youtube.com/playlist?list=PL82C6-O4XrHfrGOCPmKmwTO7M0avXyQKc : I found this playlist useful as well but only had time to watch the first 2 videos of it. I will later spend more time going to the depth of this playlist. It explained the concepts in a quite comprehensible format. I installed the VSCode extensions advised in that video to make the coding part easier and more readable.

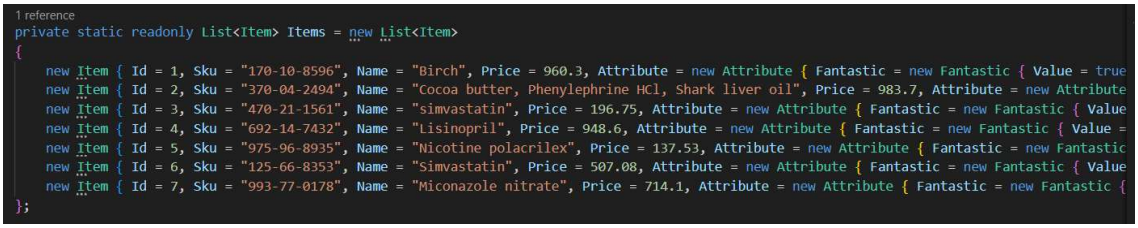
4. Learn how to build a simple C# .NET web API given a JSON database:
I was not sure how to learn how to build this so I used ChatGPT and I was given the following guideline which I later used as a general rule how to approach the problem :

a) Model: a model that represents the JSON data is necessary as simply our input and our outputs are dependent on it. After looking at the details of the JSON file I realized that in my model I should use both parent and child classes as the inheritance concept was seen in the database. I also set all my data types and variables exactly accordingly so that it completely reflects the mock data. For example, “value” in the “rating” class is a double whereas “type” in the “rating” class is a string.

b) Controller: as our main task is to provide the required filtering, a controller is necessary to handle the filtering logic. The controller also handles HTTP requests and returns appropriate responses. I will explain how I generated the code itself in the following section.

5. Code approach:

As I'm fairly new to this field I used a lot of online sources and VSCode extensions to make sure I don't miss anything and I'm on the right track, especially in the controller code. The only part I got so confused and could not understand despite of multiple sources available was how to import my MockData in my API folder, and how to scrape it in a way that I can have it accessible instead of copying and pasting the raw data:



```
1 reference
private static readonly List<Item> Items = new List<Item>
{
    new Item { Id = 1, Sku = "170-10-8596", Name = "Birch", Price = 960.3, Attribute = new Attribute { Fantastic = new Fantastic { Value = true
    new Item { Id = 2, Sku = "370-04-2494", Name = "Cocoa butter, Phenylephrine HCl, Shark liver oil", Price = 983.7, Attribute = new Attribute
    new Item { Id = 3, Sku = "470-21-1561", Name = "simvastatin", Price = 196.75, Attribute = new Attribute { Fantastic = new Fantastic { Value
    new Item { Id = 4, Sku = "692-14-7432", Name = "Lisinopril", Price = 948.6, Attribute = new Attribute { Fantastic = new Fantastic { Value =
    new Item { Id = 5, Sku = "975-96-8935", Name = "Nicotine polacrilex", Price = 137.53, Attribute = new Attribute { Fantastic = new Fantastic
    new Item { Id = 6, Sku = "125-66-8353", Name = "Simvastatin", Price = 507.08, Attribute = new Attribute { Fantastic = new Fantastic { Value
    new Item { Id = 7, Sku = "993-77-0178", Name = "Miconazole nitrate", Price = 714.1, Attribute = new Attribute { Fantastic = new Fantastic {
};
```

for the code to be readable for now, I formatted only the first 6 lines of the MockData until I learn an efficient way how to access my database. I am eager to hear your advice on the above and whether there are any libraries I can use that can assist me with the import part.

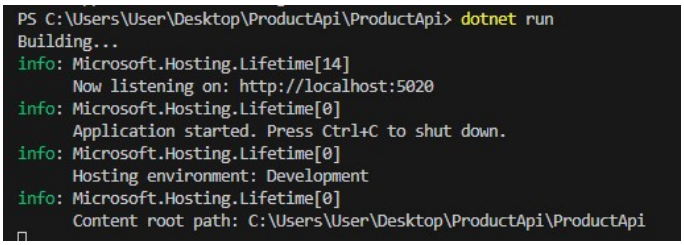
The Microsoft API documentation has a detailed tutorial and guidelines that I used to implement my code, as well as ChatGPT for cross checks, debugging, redundancy reductions, and learning new methods in C#.
https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0

I further used LINQ queries to simply apply the filters required in the test. I made sure the query is executed only if the filtered field has a value associated. (as for LINQ, I only learned the basics just for the test purpose, will spend more time learning the syntax and the details.)

Finally the filteredItems are returned in HTTP 200 OK response.

6. Testing the code:

Once I run the application:



I used the following URLs test my filters:

test #1:
http://localhost:5020/api/item?maxPrice=500&fantastic=true

result #1:

Pretty-print

```
[
  {
    "id": 3,
    "sku": "470-21-1561",
    "name": "simvastatin",
    "price": 196.75,
    "attribute": {
      "fantastic": {
        "value": true,
        "type": 1,
        "name": "fantastic"
      },
      "rating": {
        "name": "rating",
        "type": "2",
        "value": 4
      }
    }
  }
]
```


test #2:
http://localhost:5020/api/item?minPrice=100&maxPrice=500

result #2:

Pretty-print

```
[
  {
    "id": 3,
    "sku": "470-21-1561",
    "name": "simvastatin",
    "price": 196.75,
    "attribute": {
      "fantastic": {
        "value": true,
        "type": 1,
        "name": "fantastic"
      },
      "rating": {
        "name": "rating",
        "type": "2",
        "value": 4
      }
    }
  },
  {
    "id": 5,
    "sku": "975-96-8935",
    "name": "Nicotine polacrilex",
    "price": 137.53,
    "attribute": {
      "fantastic": {
        "value": false,
        "type": 1,
        "name": "fantastic"
      },
      "rating": {
        "name": "rating",
        "type": "2",
        "value": 2.1
      }
    }
  }
]
```

test #3:
http://localhost:5020/api/item?minRating=3&maxRating=4

result #3:

Pretty-print ☑


```
[
  {
    "id": 3,
    "sku": "470-21-1561",
    "name": "simvastatin",
    "price": 196.75,
    "attribute": {
      "fantastic": {
        "value": true,
        "type": 1,
        "name": "fantastic"
      },
      "rating": {
        "name": "rating",
        "type": "2",
        "value": 4
      }
    }
  }
]
```