# Product Catalog API Documentation

## Project Overview

The Product Catalog API is a .NET Core Web API designed to manage a catalog of products.   
It provides CRUD (Create, Read, Update, Delete) functionalities for managing product information,   
such as name, category, description, price, and stock quantity. The API connects to a database to   
store product details using Entity Framework Core.

## Task Overview

Assigned to:  
- Rushkesh Katkar (Backend API Developer) - API Endpoints Implementation, Controller Logic,   
 Unit Testing, and API Documentation.  
- Abhishek Yadav (Database & Integration Developer) - Database Design, Data Validation,   
 Database Operations, and Testing.  
  
Deadline: 31st December 2024

## API Endpoints

### POST /api/products - Add a new product

Description: Adds a new product to the catalog.

#### Request Body:

{  
 "name": "Product Name",  
 "category": "Category",  
 "description": "Product Description",  
 "price": 100.00,  
 "stockQuantity": 50  
}

#### Response:

Status Code: 201 Created

{  
 "id": 1,  
 "name": "Product Name",  
 "category": "Category",  
 "description": "Product Description",  
 "price": 100.00,  
 "stockQuantity": 50  
}

### GET /api/products - Get all products

Description: Retrieves a list of all products in the catalog.

#### Response:

Status Code: 200 OK

[  
 {  
 "id": 1,  
 "name": "Product Name",  
 "category": "Category",  
 "description": "Product Description",  
 "price": 100.00,  
 "stockQuantity": 50  
 },  
 {  
 "id": 2,  
 "name": "Another Product",  
 "category": "Another Category",  
 "description": "Another Description",  
 "price": 50.00,  
 "stockQuantity": 30  
 }  
]

### GET /api/products/{id} - Get a product by ID

Description: Retrieves a product by its ID.

#### Path Parameter:

id (integer) - The ID of the product to retrieve.

#### Response:

Status Code: 200 OK (if the product is found)

{  
 "id": 1,  
 "name": "Product Name",  
 "category": "Category",  
 "description": "Product Description",  
 "price": 100.00,  
 "stockQuantity": 50  
}

Status Code: 404 Not Found (if the product is not found)

{  
 "message": "Product with ID 1 not found."  
}

### PUT /api/products/{id} - Update product details

Description: Updates an existing product in the catalog.

#### Path Parameter:

id (integer) - The ID of the product to update.

#### Request Body:

{  
 "name": "Updated Product Name",  
 "category": "Updated Category",  
 "description": "Updated Description",  
 "price": 150.00,  
 "stockQuantity": 60  
}

#### Response:

Status Code: 200 OK

{  
 "message": "Product updated successfully."  
}

### DELETE /api/products/{id} - Delete a product

Description: Deletes a product from the catalog.

#### Path Parameter:

id (integer) - The ID of the product to delete.

#### Response:

Status Code: 204 No Content (if the product is deleted)

Status Code: 404 Not Found (if the product is not found)

{  
 "message": "Product with ID 1 not found."  
}

## Models and DTOs

### Product Model

public class Product  
{  
 public int Id { get; set; }  
 public string Name { get; set; }  
 public string Category { get; set; }  
 public string Description { get; set; }  
 public decimal Price { get; set; }  
 public int StockQuantity { get; set; }  
}