



Building a Customer REST API with C# and CI/CD Deployment via Azure DevOps

Web/QA Team

Our Team



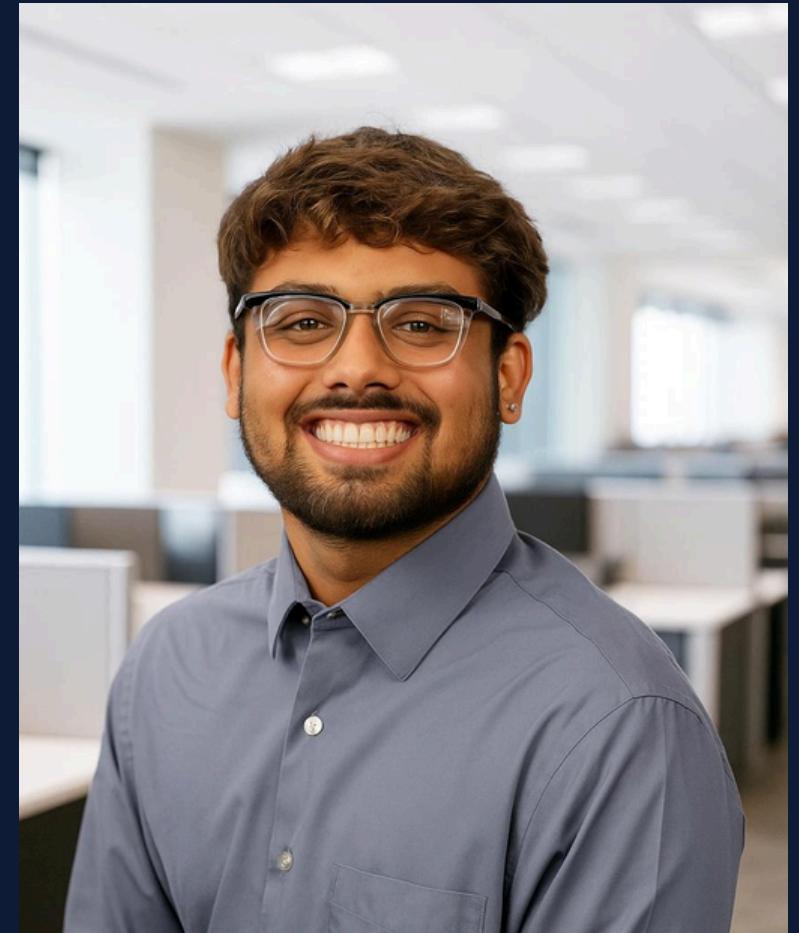
Tatyana



Tetiana



Nazar



Jeet

Technology Stack

Language: C# and .NET 8.0



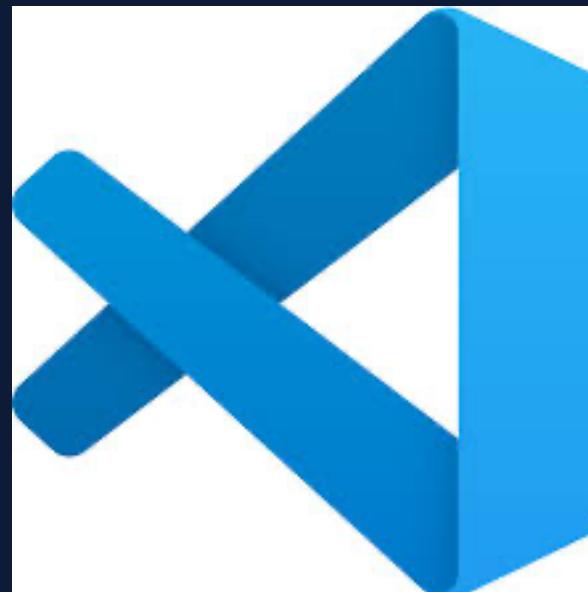
Framework: ASP.NET Core Web API



Database: Azure SQL Database

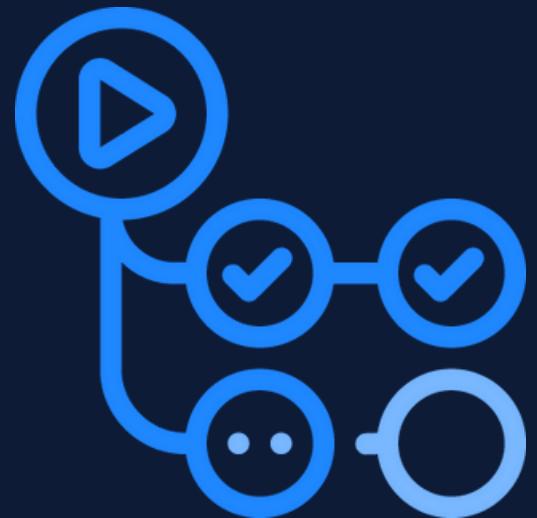


IDE: Visual Studio Code



Technology Stack

CI/CD: GitHub Actions



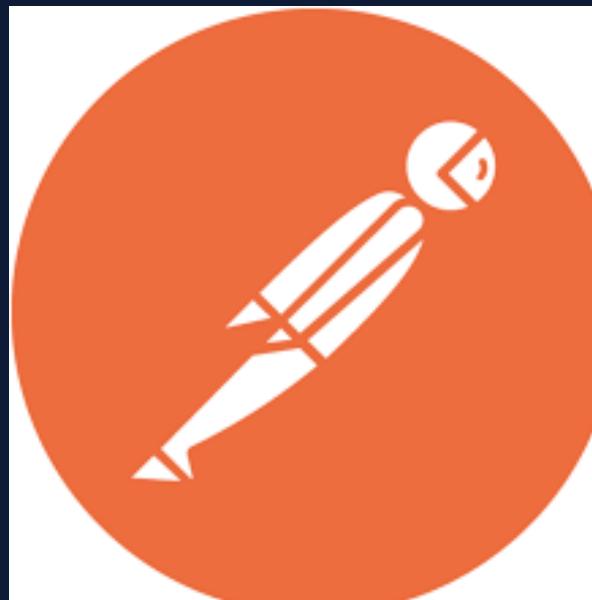
Version Control: GitHub



Hosting: Azure Apps Services



API Testing: Postman



API Design & Database Schema

- Id is the primary key
- Utilized by Entity Framework Core to store in Azure SQL Database

```
CustomerApi > Models > C# Customer.cs > Customer > Id
1  namespace CustomerApi.Models
2  {
3      public class Customer
4      {
5          public int Id { get; set; } // e.g. 1
6          public string Name { get; set; } = ""; // e.g. "Jeet Patel"
7          public string Email { get; set; } = ""; // e.g. "jeet@example.com"
8          public string Phone { get; set; } = ""; // e.g. "jeet@example.com"
9      }
10 }
11
```

API Design & Database Schema

- Implements full CRUD operations (Create, Read, Update, Delete)
- Built using ASP.NET Core Minimal APIs in Program.cs
- Routes defined with app.MapGet, MapPost, MapPut, and MapDelete
- Uses asynchronous database access via AppDbContext
- Interacts directly with Azure SQL Database

```
CustomerApi > C# Program.cs > Program > <top-level-statements-entry-point>
6   var builder = WebApplication.CreateBuilder(args);
1   using CustomerApi.Models;
62
63   // CRUD endpoints for Customers using EF Core
64
65   app.MapGet("/customers", async (AppDbContext db) =>
66     await db.Customers.ToListAsync());
67
68   app.MapGet("/customers/{id}", async (int id, AppDbContext db) =>
69   {
70     var customer = await db.Customers.FindAsync(id);
71     return customer is not null ? Results.Ok(customer) : Results.NotFound();
72   });
73
74   app.MapPost("/customers", async (Customer newCustomer, AppDbContext db) =>
75   {
76     db.Customers.Add(newCustomer);
77     await db.SaveChangesAsync();
78     return Results.Created($"/customers/{newCustomer.Id}", newCustomer);
79   });
80
81   app.MapPut("/customers/{id}", async (int id, Customer updatedCustomer, AppDbContext db) =>
82   {
83     var existing = await db.Customers.FindAsync(id);
84     if (existing is null) return Results.NotFound();
85
86     existing.Name = updatedCustomer.Name;
87     existing.Email = updatedCustomer.Email;
88     existing.Phone = updatedCustomer.Phone;
```

Development Workflow

Local Development Setup

- Installed .NET 8 SDK, VS Code, Azure CLI, and Git
- Created ASP.NET Core Web API project
 - dotnet new webapi -n CustomerApi
- Ran API locally
 - dotnet run

```
Last login: Thu Jul 31 12:30:06 on ttys000
[jeetpatel@Jeets-MacBook-Air ~ % dotnet new webapi -n CustomerApi
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring /Users/jeetpatel/CustomerApi/CustomerApi.csproj:
Restore succeeded.

jeetpatel@Jeets-MacBook-Air ~ %
```



Development Workflow

Git Source Control

- Initialized local Git repo, committed code

- **git commit -m "Initial commit"**
 - **git push -u origin main**

- Connected remote repo on GitHub

- **git remote add origin**
<https://github.com/Jeet52/CustomerApi.git>

- Created personal access token for pushing code securely

```
Last login: Thu Jul 31 20:25:53 on ttys004
[jeetpatel@Jeets-MacBook-Air ~ % cd ~/CustomerApi
[jeetpatel@Jeets-MacBook-Air CustomerApi % git init
Reinitialized existing Git repository in /Users/jeetpatel/CustomerApi/.git/
[jeetpatel@Jeets-MacBook-Air CustomerApi % git add .
[jeetpatel@Jeets-MacBook-Air CustomerApi % git commit -m "Presentation"
[main 9aea0fd] Presentation
 593 files changed, 19982 insertions(+), 4 deletions(-)
  create mode 100644 CustomerApi.csproj
  create mode 100644 CustomerApi.http
  create mode 100644 CustomerApi/.DS_Store
  create mode 100644 CustomerApi/.github/workflows/dotnet.yml
  create mode 100644 CustomerApi/CustomerApi/.github/workflows/deploy.yml
  create mode 100644 CustomerApi/CustomerApi/CustomerApi.csproj
```



CI/CL Pipeline

Azure App Service

- Created App Service Plan and Web App with .NET 8 runtime
- Set up resource group and service plan using Azure CLI

GitHub Actions

- Automates build, publish, and deploy steps on git push
- Workflow file located in .github/workflows/deploy.yml

Stored Azure Publish Profile in GitHub Secrets

- Downloaded .PublishSettings file from Azure
- Added it as AZURE_WEBAPP_PUBLISH_PROFILE secret in GitHub

Note: Deployment is Triggered Automatically via git push
GitHub Actions builds the app and deploys to Azure App Service

Live Demo

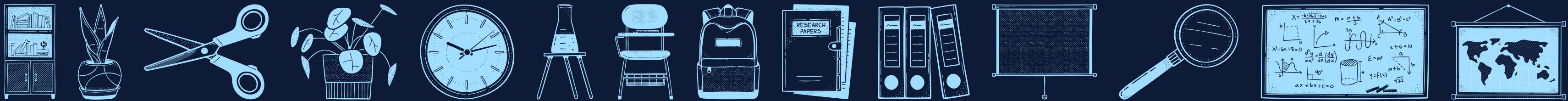
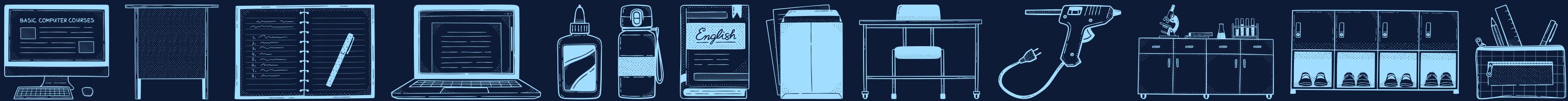
<https://customer-api-app-jeet123.azurewebsites.net/swagger/index.html>

The screenshot shows the Swagger UI interface for the Customer API. At the top, there's a navigation bar with the Swagger logo, the text "Supported by SMARTBEAR", a dropdown for "Select a definition" set to "Customer API V1", and a search bar. Below the header, the title "Customer API" is displayed with "v1 OAS 3.0" and a link to "/swagger/v1/swagger.json". The main content area is titled "CustomerApi" and lists the following endpoints:

- GET /
- GET /customers
- POST /customers
- GET /customers/{id}
- PUT /customers/{id}
- DELETE /customers/{id}

The "POST /customers" endpoint is highlighted with a green background, indicating it is the current selection.

Thank You!



Any Questions?