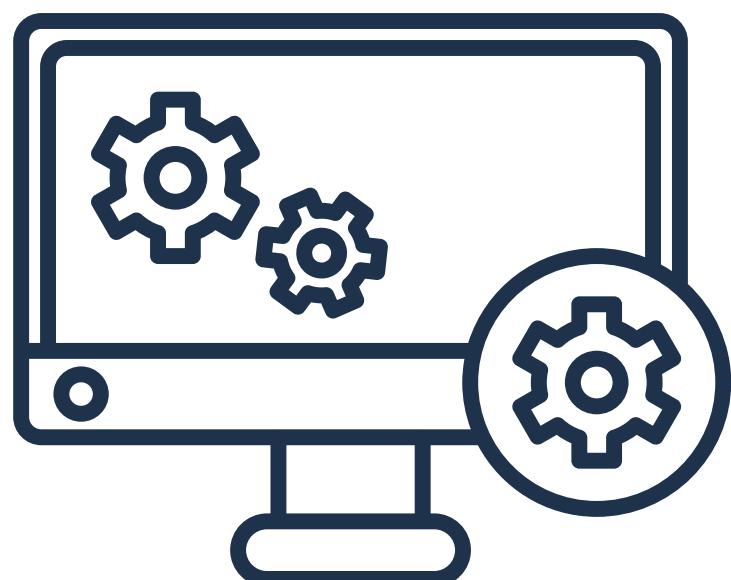


```
return <  
  <React.Fragment>  
    <div className="py-5">  
      <div className="container">  
        <Title name="our" title="product">  
        <div className="row">  
          <ProductConsumer>  
            {(value) => {  
              console.log(value)  
            }}  
          </ProductConsumer>  
        </div>  
      </div>  
    </div>  
  </React.Fragment>  
</div>
```

# ASP.NET, ANGULAR, AND CHATGPT APP SOFTWARE DOCUMENTATION



# PART 1: ASP.NET API

## 1.1. CHATGPT API TOKEN CONFIGURATION

- OPEN THE OPENAISERVICE.CS FILE WITHIN THE OPENAI DIRECTORY.
- LOCATE THE SECRETKEY VARIABLE WITHIN THE GETDATARESPONSE METHOD.
- REPLACE THE EXISTING VALUE WITH YOUR CHATGPT API TOKEN:

csharp

```
var secretKey = "your-api-token-here";
```

## 1.2. ENTITY FRAMEWORK CORE SETUP

- CONNECTION STRINGS: UPDATE THE CONNECTION STRINGS IN THE STARTUP.CS FILE OR APPSETTINGS.JSON TO MATCH YOUR DATABASE.
- MIGRATIONS: USE THE FOLLOWING COMMAND TO CREATE AND APPLY MIGRATIONS:

bash

```
dotnet ef migrations add InitialCreate  
dotnet ef database update
```

## 1.3. RUNNING THE .NET CORE API

- NAVIGATE TO THE PROJECT DIRECTORY AND EXECUTE:

bash

```
dotnet build  
dotnet run
```

- THE API WILL BE ACCESSIBLE AT [HTTPS://LOCALHOST:7055](https://localhost:7055) OR THE CONFIGURED PORT.

# PART 2: ANGULAR APPLICATION

## 2.1. USER SERVICE CONFIGURATION

- OPEN THE USER.SERVICE.TS FILE.
- MODIFY THE API\_URL VARIABLE TO POINT TO THE CORRECT AUTHENTICATION ENDPOINT (E.G., 'HTTPS://LOCALHOST:7055/API/AUTHENTICATION').

## 2.2. RUNNING THE ANGULAR APPLICATION

- NAVIGATE TO THE ANGULAR PROJECT DIRECTORY.
- INSTALL DEPENDENCIES:

```
bash
```

```
npm install
```

- RUN THE APPLICATION:

```
bash
```

```
ng serve
```

- THE APPLICATION WILL BE ACCESSIBLE AT **HTTP://LOCALHOST:4200** OR THE CONFIGURED PORT.

# PART 3: INTEGRATION AND USAGE

## 3.1. AUTHENTICATION FLOW

- REGISTRATION: USERS CAN REGISTER THROUGH THE REGISTERCOMPONENT.
- LOGIN: USERS CAN LOG IN THROUGH THE LOGINCOMPONENT.
- AUTHENTICATION GUARD: THE AUTHGUARD ENSURES THAT CERTAIN ROUTES ARE PROTECTED.

## 3.2. CONVERSATIONS AND OPENAI INTEGRATION

- OPENAI CALLS: THE OPENAICONTROLLER IN .NET CORE HANDLES OPENAI-RELATED CALLS.
- CONVERSATIONS: THE CONVERSATIONCONTROLLER MANAGES CONVERSATION-RELATED OPERATIONS.

## 3.3. ANGULAR COMPONENTS AND ROUTING

- COMPONENTS: EXPLORE COMPONENTS FOR LOGIN, REGISTRATION, SEARCH, AND DETAILS.
- ROUTING: DEFINED IN APP-ROUTING.MODULE.TS, THE ROUTING CONFIGURATION CONTROLS THE NAVIGATION FLOW.

# PART 4: TROUBLESHOOTING AND BEST PRACTICES

- SECURING API KEYS: USE ENVIRONMENT VARIABLES OR SECURE STORAGE FOR SENSITIVE INFORMATION.
- ERROR HANDLING: IMPLEMENT PROPER ERROR HANDLING IN BOTH ANGULAR AND .NET CORE TO ENSURE A SMOOTH USER EXPERIENCE.
- TESTING: WRITE UNIT AND INTEGRATION TESTS TO VALIDATE FUNCTIONALITY.

# PART 5: VIDEO LINK ON HOW TO USE

**HTTPS://YOUTU.BE/  
W-9UEVPUSME**