

1. Project Overview

This is an ASP.NET Core API designed for user and role management using Unit of Work and Onion Architecture. The application allows administrators to perform CRUD operations for users and roles and ensures separation of concerns using a layered architecture. The API supports authentication, authorization, and role-based access.

2. Architecture Overview

Onion Architecture is a pattern that helps separate concerns in an application. It consists of concentric layers where each layer has a specific responsibility, and dependencies flow inward, not outward.

Layers in the Onion Architecture:

Core Layer: Contains your core entities and business logic.

Service Layer: Contains your interfaces for business operations (use cases) and data transfer objects (DTOs).

Infrastructure Layer: Handles implementation details, including data access, external services, and other dependencies.

API Layer: The API layer (controllers) that interacts with the client.

Unit of Work (UOW)

The Unit of Work pattern manages database transactions by allowing multiple operations to be treated as a single unit, ensuring all operations are completed successfully or none at all.

3. Security and Authentication

Description: Logs in a user and generates a JWT token.

Request Body

```
{
  "username": "string",
  "password": "string"
}
```

Response body

[illegible]

Download

3. CRUD Operations

User:

User	
1	GET /api/User/all
1	GET /api/User/{id}
2	PUT /api/User/{id}
	DELETE /api/User/{id}
3	POST /api/User/create
4	POST /api/User/AddRolesToUser
5	POST /api/User/login

Json Body:

```
{
  "user_Id": "string",
  "full_Name": "string",
  "user_Name": "string",
  "email": "string",
  "roles": [
    "string"
  ]
}
```

1 2

```
{
  "userId": "string",
  "roles": [
    "string"
  ]
}
```

4

```
{
  "full_Name": "string",
  "user_Name": "string",
  "passWord": "string",
  "email": "string",
  "roles": [
    "string"
  ]
}
```

3

```
{
  "username": "string",
  "password": "string"
}
```

5

Role:

Role

1

POST /API/Role/AddRole

2

GET /API/Role/GetAllRoles

DELETE /API/Role/DeleteRole

3

PUT /API/Role/UpdateRole

Json Body:

Request body

Example Value | Schema

```
{
  "oldRole": "Admin",
  "newRole": "SuperAdmin"
}
```

3

Request body

Example Value | Schema

"string"

1

Response body

```
[
  {
    "roleId": "9862e641-b27b-4010-9695-0647730fcf66",
    "roleName": "Admin"
  },
  {
    "roleId": "a888b049-55a3-430a-aa5e-a60bd5876b11",
    "roleName": "Doctor"
  },
]
```

2