

# API User Management System (Digvijay Dhumal)

## ◆ BACKEND EXPLANATION

### (ASP.NET Core Web API)

---

#### 1 Backend Overview

The backend of this project is developed using **ASP.NET Core Web API**.

It is responsible for:

- Handling business logic
- Managing database operations
- Exposing RESTful APIs
- Providing data to the Angular frontend

The backend follows a **Controller–Model–DbContext** architecture.

---

#### 2 Project Structure (Backend)

APIUserManagement

```
|  
|   └── Controllers  
|       |   └── AuthController.cs  
|       |   └── MasterController.cs  
|       |   └── ProductsController.cs  
|       |   └── UsersController.cs  
|  
|   └── Data  
|       |   └── AppDbContext.cs  
|  
|   └── Models  
|       |   └── User.cs  
|       |   └── Product.cs  
|       |   └── Country.cs  
|       |   └── State.cs
```

```
|   |--- City.cs  
|  
|--- Program.cs  
|--- appsettings.json
```

---

### 3 Controllers Explanation

#### AuthController

Responsible for **user authentication**.

**Responsibilities:**

- Validates user email and password
- Returns user details on successful login

**Key API:**

- POST /api/auth/login
- 

#### UsersController

Handles **user CRUD operations**.

**Responsibilities:**

- Create new users
- Fetch all users
- Fetch user by ID
- Update user details
- Delete users

**APIs:**

- GET /api/users
  - POST /api/users
  - GET /api/users/{id}
  - PUT /api/users/{id}
  - DELETE /api/users/{id}
- 

#### MasterController

Manages **master data** like Country, State, and City.

### **Responsibilities:**

- Provide dropdown data for user forms
- Handle dependent dropdown logic

### **APIs:**

- GET /api/master/countries
  - GET /api/master/states/{countryId}
  - GET /api/master/cities/{stateId}
- 

## **ProductsController**

Manages **product operations** linked to users.

### **Responsibilities:**

- Add products for a specific user
- Fetch products user-wise
- Update product details
- Delete products

### **APIs:**

- POST /api/products
  - GET /api/products/user/{userId}
  - PUT /api/products/{id}
  - DELETE /api/products/{id}
- 

## **Models Explanation**

Each model represents a database table.

### **User Model**

Stores user-related data including personal and location details.

### **Product Model**

Stores products associated with a user.

### **Country, State, City Models**

Used for geographical master data and relational mapping.

---

## **AppDbContext**

`AppDbContext` acts as a bridge between the application and the database.

#### Responsibilities:

- Manages database connections
- Maps models to database tables using `DbSet<>`
- Enables LINQ queries

```
public DbSet<User> Users { get; set; }

public DbSet<Product> Products { get; set; }

public DbSet<Country> Countries { get; set; }

public DbSet<State> States { get; set; }

public DbSet<City> Cities { get; set; }
```

---

## 6 Configuration Files

### appsettings.json

- Stores database connection string
- Logging configuration

### Program.cs

- Registers `DbContext`
  - Enables CORS
  - Configures middleware
  - Maps controllers
- 

## 7 Backend Key Features

- RESTful API design
- Entity Framework Core
- LINQ-based queries
- Clean separation of concerns
- Scalable and maintainable structure

## ◆ BACKEND EXPLANATION

### (ASP.NET Core Web API)

---

## 1 Backend Overview

The backend of this project is developed using **ASP.NET Core Web API**.

It is responsible for:

- Handling business logic
- Managing database operations
- Exposing RESTful APIs
- Providing data to the Angular frontend

The backend follows a **Controller–Model–DbContext** architecture.

---

## 2 Project Structure (Backend)

APIUserManagement

```
|  
|   └── Controllers  
|       |   └── AuthController.cs  
|       |   └── MasterController.cs  
|       |   └── ProductsController.cs  
|       |   └── UsersController.cs  
|  
|   └── Data  
|       └── AppDbContext.cs  
|  
|   └── Models  
|       |   └── User.cs  
|       |   └── Product.cs  
|       |   └── Country.cs  
|       |   └── State.cs  
|       |   └── City.cs  
|  
|   └── Program.cs  
└── appsettings.json
```

---

## 3 Controllers Explanation

## AuthController

Responsible for **user authentication**.

### Responsibilities:

- Validates user email and password
- Returns user details on successful login

### Key API:

- POST /api/auth/login
- 

## UsersController

Handles **user CRUD operations**.

### Responsibilities:

- Create new users
- Fetch all users
- Fetch user by ID
- Update user details
- Delete users

### APIs:

- GET /api/users
  - POST /api/users
  - GET /api/users/{id}
  - PUT /api/users/{id}
  - DELETE /api/users/{id}
- 

## MasterController

Manages **master data** like Country, State, and City.

### Responsibilities:

- Provide dropdown data for user forms
- Handle dependent dropdown logic

### APIs:

- GET /api/master/countries
- GET /api/master/states/{countryId}

- GET /api/master/cities/{stateId}
- 

## ProductsController

Manages **product operations** linked to users.

### Responsibilities:

- Add products for a specific user
- Fetch products user-wise
- Update product details
- Delete products

### APIs:

- POST /api/products
  - GET /api/products/user/{userId}
  - PUT /api/products/{id}
  - DELETE /api/products/{id}
- 

## Models Explanation

Each model represents a database table.

### User Model

Stores user-related data including personal and location details.

### Product Model

Stores products associated with a user.

### Country, State, City Models

Used for geographical master data and relational mapping.

---

## AppDbContext

AppDbContext acts as a bridge between the application and the database.

### Responsibilities:

- Manages database connections
- Maps models to database tables using DbSet<>
- Enables LINQ queries

```
public DbSet<User> Users { get; set; }
```

```
public DbSet<Product> Products { get; set; }  
public DbSet<Country> Countries { get; set; }  
public DbSet<State> States { get; set; }  
public DbSet<City> Cities { get; set; }
```

---

## 6 Configuration Files

### appsettings.json

- Stores database connection string
- Logging configuration

### Program.cs

- Registers DbContext
  - Enables CORS
  - Configures middleware
  - Maps controllers
- 

## 7 Backend Key Features

- RESTful API design
- Entity Framework Core
- LINQ-based queries
- Clean separation of concerns
- Scalable and maintainable structure

## APIUserManagement 1.0.0 DASH

http://localhost:5118/swagger/v1/swagger.json

### Auth

`POST` /api/auth/login

### Master

`Get` /api/master/countries

`Get` /api/master/states/{countryId}

`Get` /api/master/cities/{stateId}

### Products

`POST` /api/products

`Get` /api/products/user/{userId}

`PUT` /api/products/{id}

`DELETE` /api/products/{id}

### Users

`POST` /api/users

`Get` /api/users

`Get` /api/users/{id}

`PUT` /api/users/{id}

`DELETE` /api/users/{id}

### WeatherForecast

`Get` /WeatherForecast

### Welcome Back

Please login to continue

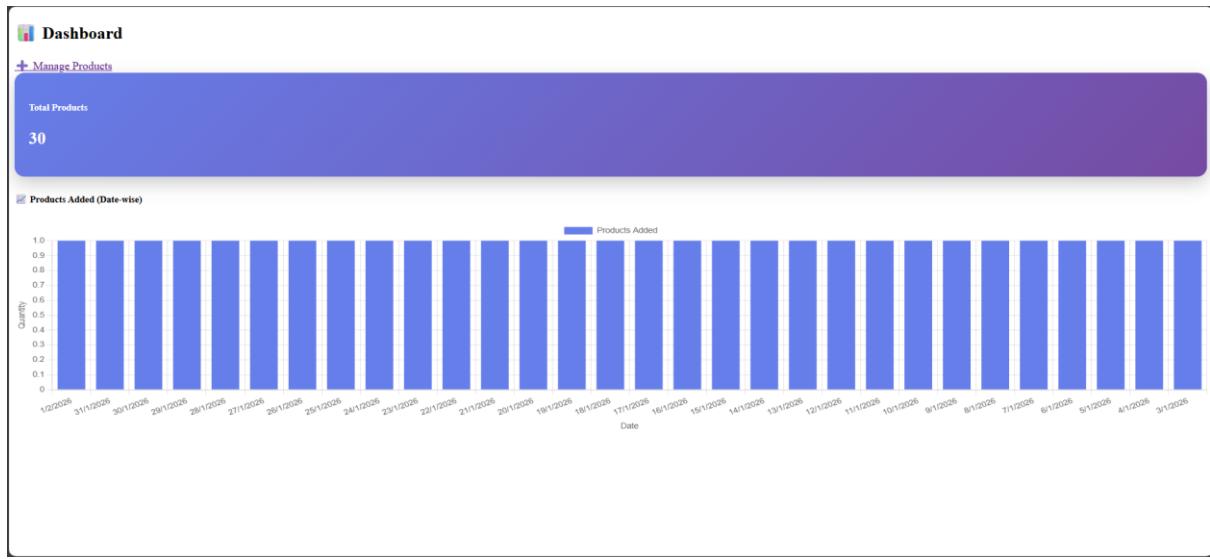
pradnya@mail.com

...

Login

OR

 Add New User



## My Products

[Add Product](#)

Name	Price (₹)	Qty	Action
Mouse	500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Keyboard	1500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Headphones	3000	1	<a href="#">Edit</a> <a href="#">Delete</a>
Pen Drive	800	1	<a href="#">Edit</a> <a href="#">Delete</a>
Laptop	55000	1	<a href="#">Edit</a> <a href="#">Delete</a>
Mouse	500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Keyboard	1500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Headphones	3000	1	<a href="#">Edit</a> <a href="#">Delete</a>
Pen Drive	800	1	<a href="#">Edit</a> <a href="#">Delete</a>
Laptop	55000	1	<a href="#">Edit</a> <a href="#">Delete</a>
Mouse	500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Keyboard	1500	1	<a href="#">Edit</a> <a href="#">Delete</a>
Headphones	3000	1	<a href="#">Edit</a> <a href="#">Delete</a>
Pen Drive	800	1	<a href="#">Edit</a> <a href="#">Delete</a>

## Add Product

Product Name

Price

Quantity

[Save](#)

[Cancel](#)

## Edit Product

Product Name

Pen Drive

Price

800

Quantity

1

Save

Cancel

### Add New User

Create a new user account

User Name Email

pradnya@mail.com Email address

Mobile Password

Mobile number .....

Country State City

Select Count Select State Select City

Save Cancel

Select Country

Australia

Brazil

Canada

China

France

Germany

India

Italy

Japan

Singapore

South Africa

Spain

UAE

UK

USA

### Add New User

Create a new user account

Email

Email address

Password

.....

State City

Select Count Select State Select City

Save Cancel

localhost:4200/users/add

localhost:4200 says

User Added Successfully

OK

Add New User  
Create a new user account

User Name	Email	
pradnya@mail.com	Email address	
Mobile	Password	
Mobile number	*****	
Country	State	City
UAE	Dubai Emirate	Dubai City

Save Cancel

**Users**

User Name	Email	Mobile
Pradnya Patil	pradnya@mail.com	9876543210
Amit Kulkarni	amit@mail.com	9876543211
Rahul Shah	rahul@mail.com	9876543212
John Smith	john@mail.com	900000001
Michael Lee	michael@mail.com	900000002
David Brown	david@mail.com	900000003
Emma Wilson	emma@mail.com	900000004
Olivia Martin	olivia@mail.com	900000005
Lukas Meyer	lukas@mail.com	900000006
Pierre Dubois	pierre@mail.com	900000007
pradnya@mail.com		

Add User