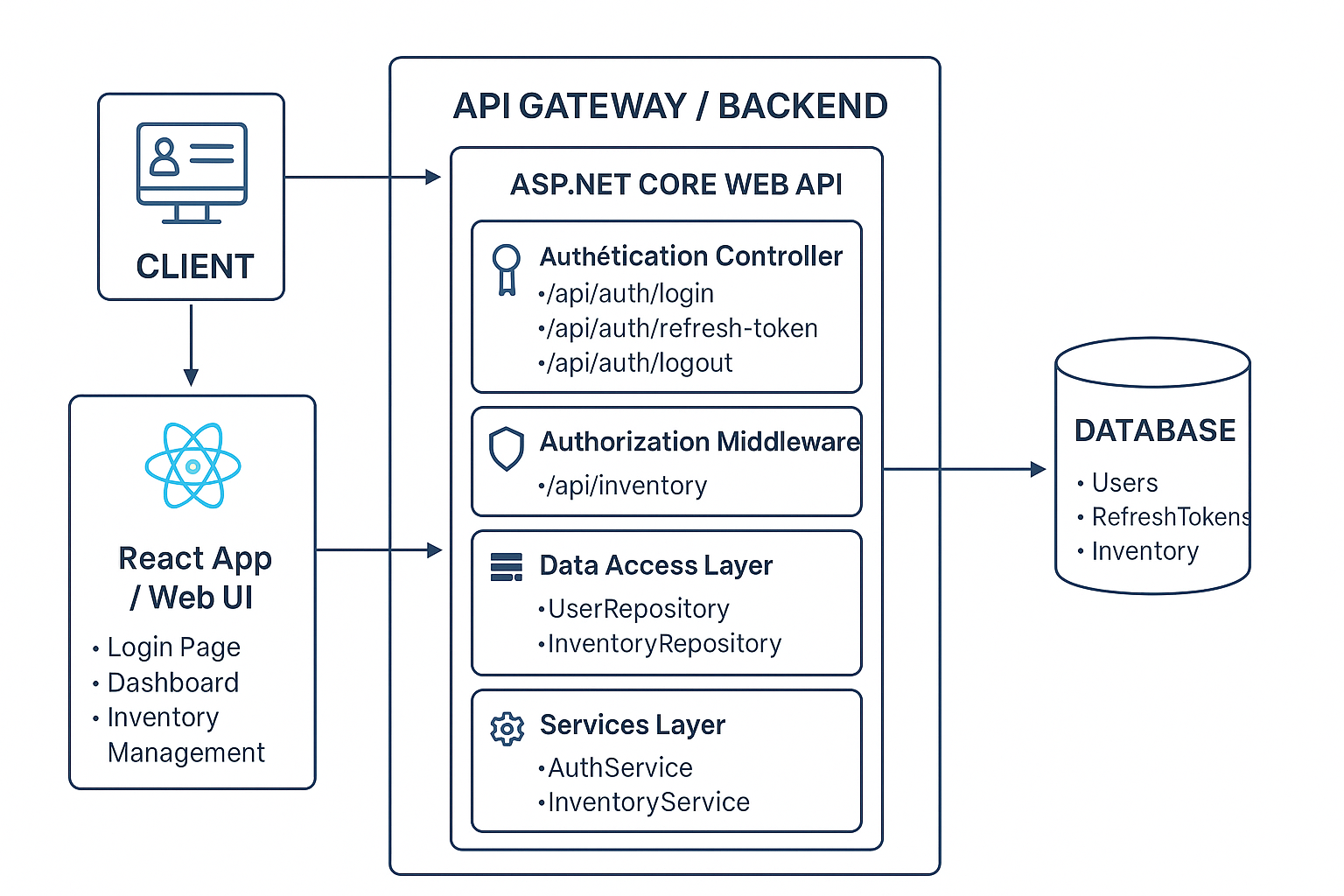
**ARCHITECTURE DIAGRAM**



## COMPONENT DESCRIPTIONS

## AUTHENTICATION & AUTHORIZATION FLOW

1. **User Login:**
   * User submits credentials → /api/auth/login.
   * Backend:
     + Validates password policy.
     + Checks lockout count.
     + Issues JWT (access token) + refresh token or secure cookie.
   * Frontend stores token securely (HttpOnly cookie or memory).
2. **Role Display:**
   * After login, the backend returns { username, role }.
   * React app shows: *“Welcome, Editor”* or *“Welcome, Viewer”*.
3. **Token Persistence:**
   * Refresh token stored in DB.
   * /api/auth/refresh-token can issue new JWT when access token expires.
4. **Failed Login Lockout:**
   * After 5 consecutive failures → IsLockedOut = true for X minutes.

**INVENTORY MANAGEMENT FLOW**

| **Role** | **Permissions** |
| --- | --- |
| Viewer | GET (view list/details only) |
| Editor | GET, POST, PUT, DELETE (full CRUD) |

**EXAMPLE FLOW**

1. **Viewer** calls /api/inventory → returns read-only list.
2. **Editor** calls:
   * POST /api/inventory → add new item.
   * PUT /api/inventory/{id} → update item.
   * DELETE /api/inventory/{id} → delete item.
3. Middleware checks Role claim in JWT to enforce permissions.

**DATA MODEL OVERVIEW**

**User Table**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Id | GUID | Primary Key |
| Username | string | Unique |
| PasswordHash | string | Hashed (e.g., PBKDF2 or bcrypt) |
| Role | enum(Viewer, Editor) | Used for RBAC |
| FailedAttempts | int | Track login failures |
| IsLockedOut | bool | Prevent login when true |

**Inventory Table**

| **Field** | **Type** |
| --- | --- |
| Id | GUID |
| Name | string |
| Description | string |
| Color | string |
| Type | string |
| Suppliers | string |
| Manufacturers | string |

DEPLOYMENTS

1. Run DB Scripts(file ***InventoryDB.sql***) to create DB and import data automatically(required SQL 2022 version, firstly need create a new database named “***InventoryDB***”
2. Run BE application from ***Inventory BE*** folder(required Visual Studio 2022 version), maybe need to modify ***ConnectionStrings*** in ***appsettings.json*** if difference SQL instance name
3. Run FE application from ***Inventory FE*** folder by using VS code or go to the terminal -> run command prompt as “***npm start***”, during use if there is any problem please refresh page or clear site data from application tab from Browser due to some of the problems described in section Technical Debt

Use Peter/dUY@NH123456 as Editor role

Use Mark/dUY@NH112233 as Viewer role

TECHNICAL DEBT

1. Some business constraints cannot be completed due to time constraints
2. Not applied yet Async Programming to achieve Non-Blocking operations
3. Review the entire source code thoroughly because of lack of time
4. Some performance issues may occur in this version(Paging mechanism not implemented yet, SQL Optimization,...)