**📄 Technical Requirements Document**

**🎯 Project: Simple Purchase Management System**

**Tech Stack:**

* **Backend: ASP.NET Core Web API**
* **Frontend: Angular + PrimeNG**
* **ORM: Entity Framework Core**
* **Database: SQL Server**
* **Authentication: JWT (Admin/User roles)**

**🧑‍💻 Purpose**

**Equip a new full stack developer with clear guidance, architecture, and DB schema to implement a simple purchase management system handling products, vendors, and purchase orders with CRUD capabilities.**

**🧱 System Features Scope**

**✅ Included**

* **User authentication (JWT)**
* **Role-based authorization**
* **Product CRUD**
* **Vendor CRUD**
* **Purchase Order CRUD**
* **Simple dashboard**

**❌ Excluded**

* **Inventory integration**
* **Assignment to employees/departments**
* **Purchase reporting (PDF, Excel)**
* **Workflow approvals**

**🗂 Pages & Descriptions**

| **Page** | **Description** |
| --- | --- |
| **Login Page** | **Authenticates users via JWT. Redirects based on role.** |
| **Dashboard (Home)** | **Simple summary (e.g., total products, total orders).** |
| **Product Management** | **CRUD operations for products (name, code, unit, price).** |
| **Vendor Management** | **CRUD operations for vendors (name, address, contact info).** |
| **Purchase Orders List** | **View all purchase orders, status, and basic filter/sort.** |
| **Purchase Order Create/Edit** | **Form to create/edit a purchase order with vendor, products, quantities, total.** |
| **User Management (Admin only)** | **Manage users with roles and status (active/inactive).** |
| **Error/Unauthorized Page** | **Fallback for failed access/authentication.** |

**🗃 Database Schema**

**1. Users**

**CREATE TABLE Users (**

**Id INT PRIMARY KEY IDENTITY,**

**Username NVARCHAR(50) NOT NULL,**

**PasswordHash NVARCHAR(MAX) NOT NULL,**

**Role NVARCHAR(20) NOT NULL, -- 'Admin' or 'User'**

**IsActive BIT NOT NULL DEFAULT 1**

**);**

**2. Vendors**

**CREATE TABLE Vendors (**

**Id INT PRIMARY KEY IDENTITY,**

**Name NVARCHAR(100) NOT NULL,**

**Address NVARCHAR(255),**

**ContactPerson NVARCHAR(100),**

**Phone NVARCHAR(50),**

**Email NVARCHAR(100)**

**);**

**3. Products**

**CREATE TABLE Products (**

**Id INT PRIMARY KEY IDENTITY,**

**Name NVARCHAR(100) NOT NULL,**

**Code NVARCHAR(50) UNIQUE NOT NULL,**

**Unit NVARCHAR(20),**

**UnitPrice DECIMAL(18, 2) NOT NULL**

**);**

**4. PurchaseOrders**

**CREATE TABLE PurchaseOrders (**

**Id INT PRIMARY KEY IDENTITY,**

**VendorId INT NOT NULL,**

**OrderDate DATE NOT NULL,**

**TotalAmount DECIMAL(18, 2) NOT NULL,**

**Status NVARCHAR(50) DEFAULT 'Pending',**

**FOREIGN KEY (VendorId) REFERENCES Vendors(Id)**

**);**

**5. PurchaseOrderItems**

**CREATE TABLE PurchaseOrderItems (**

**Id INT PRIMARY KEY IDENTITY,**

**PurchaseOrderId INT NOT NULL,**

**ProductId INT NOT NULL,**

**Quantity INT NOT NULL,**

**UnitPrice DECIMAL(18, 2) NOT NULL,**

**FOREIGN KEY (PurchaseOrderId) REFERENCES PurchaseOrders(Id),**

**FOREIGN KEY (ProductId) REFERENCES Products(Id)**

**);**

**🌐 API Endpoints (Sample)**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| **POST** | **/api/auth/login** | **Authenticate and return JWT** |
| **GET** | **/api/products** | **List all products** |
| **POST** | **/api/products** | **Create product** |
| **GET** | **/api/vendors** | **List vendors** |
| **POST** | **/api/purchaseorders** | **Create purchase order** |
| **GET** | **/api/purchaseorders/{id}** | **View order with line items** |
| **DELETE** | **/api/purchaseorders/{id}** | **Delete purchase order** |

**🎨 UI Components (Angular + PrimeNG)**

| **Component** | **Description** |
| --- | --- |
| **LoginComponent** | **User login** |
| **DashboardComponent** | **Overview of vendors, products, orders** |
| **ProductListComponent** | **Product DataTable** |
| **ProductFormComponent** | **Create/edit product** |
| **VendorComponent** | **Manage vendors** |
| **PurchaseOrderListComponent** | **List and search orders** |
| **PurchaseOrderFormComponent** | **Form to create/edit orders with product selector** |
| **UserComponent (Admin)** | **Manage users** |
| **NavbarComponent** | **Navigation and logout** |

**🧪 Testing Strategy**

**Backend**

* **Use xUnit to test:**
  + **Purchase service logic**
  + **CRUD controller actions**
  + **Validation for products and orders**

**Frontend**

* **Use Jasmine/Karma:**
  + **Form validation tests**
  + **Component rendering and service integration**

**📋 Developer Onboarding Checklist**

| **Task** | **Status** |
| --- | --- |
| **Set up backend & DB** | **☐** |
| **Run frontend with Angular CLI** | **☐** |
| **Test login and JWT roles** | **☐** |
| **Implement Product CRUD** | **☐** |
| **Implement Vendor CRUD** | **☐** |
| **Create simple purchase order flow** | **☐** |
| **Submit first pull request** | **☐** |

**Would you like this delivered in Markdown, Notion-style layout, or exported to a PDF? I can also prepare both this and the Fixed Assets System as code scaffolds if needed.**