

1. System Overview

The Product Management System is a web application that allows users to:

- Manage Categories: Add, edit, and delete product categories.
- Manage Products: Add, edit, and delete products. Each product can have a name, description, category,
- Upload Images: Upload product images, which are stored in the wwwroot/images directory.
- Import/Export Products: Import products from an Excel file or export products to an Excel file.
- User Authentication: Log in and register users with JWT-based authentication.

2. Prerequisites

Before setting up the system, ensure the following software is installed on your machine:

Backend (.NET Web API)

- .NET 6 SDK
- Visual Studio 2022 (or any code editor like VS Code)
- SQL Server (or SQL Server Express)
- Entity Framework Core Tools (for database migrations)

Frontend (React)

- Node.js (v16 or higher)
- npm (comes with Node.js)
- Vite.js (for React development)

3. Setting Up the Backend

Step 1: Clone the Repository

Clone the repository to your local machine:

```
git clone https://github.com/your-repo/product-management-system.git
```

```
cd product-management-system/ProductManagementApi
```

Step 2: Set Up the Database

Open the appsettings.json file and update the connection string to point to your SQL Server instance:

```
"ConnectionStrings": {  
  "DefaultConnection": "Server=localhost;Database=ProductManagementDb;Trusted_Connection=True;TrustServerCertificate=True;"  
}
```

Run the following commands to apply database migrations:

```
dotnet ef database update
```

Step 3: Run the Backend

Start the backend server:

```
dotnet run
```

The backend will be available at <http://localhost:5190>.

4. Setting Up the Frontend

Step 1: Navigate to the Frontend Directory

```
cd ../product-management-app
```

Step 2: Install Dependencies

Install the required npm packages:

```
npm install
```

Step 3: Configure the API URL

Open the agent.js file and ensure the API_URL points to your backend:

```
const API_URL = 'http://localhost:5190/api';
```

Step 4: Run the Frontend

Start the development server:

```
npm run dev
```

The frontend will be available at <http://localhost:5173>.

5. Running the Application

- Backend: Ensure the backend is running at <http://localhost:5190>.
- Frontend: Open <http://localhost:5173> in your browser.
- Login/Register: Use the login or register page to create an account and log in.
- Manage Products/Categories: Use the dashboard to add, edit, or delete products and categories.
- Import/Export: Use the import/export functionality to manage products in bulk.

6. Using the Application

Dashboard

- Displays statistics about categories and products.
- Shows a list of recent products.

Categories

- Add, edit, or delete categories.
- Each category has a name, code, and status (active/inactive).

Products

- Add, edit, or delete products.
- Each product has a name, description, category, price, and image.
- Upload images for products.
- Import/export products using Excel files.

User Authentication

- Register a new account or log in with an existing one.
- JWT tokens are used for authentication.

7. Troubleshooting

Backend Issues

- Database Connection Errors: Ensure the connection string in appsettings.json is correct.
- Migrations Not Applying: Run `dotnet ef database update` to apply migrations manually.

Frontend Issues

- API Not Responding: Ensure the backend is running and the `API_URL` in `agent.js` is correct.
- Dependency Errors: Delete the `node_modules` folder and run `npm install` again.

Image Upload Issues

- Directory Not Found: Ensure the `wwwroot/images` directory exists. If not, create it manually or let the back