Table Management Web Application

Approach

The Table Management web application is designed to provide users with functionality to dynamically create and manage database tables, input column information, and view data through a user-friendly interface. The application is split into frontend and backend components, utilizing Docker for containerization to simplify deployment.

Assumptions Made

1. **Frontend Technology**: Angular is chosen for the frontend.
2. **Database**: PostgreSQL is used for storing tables and data.
3. **Authentication**: JWT-based authentication is implemented for user login and registration.
4. **Data Display**: A data grid with pagination, sorting, and filtering functionalities is implemented to display table data.

Setup and Running the Project Locally Using Docker

Prerequisites:

- Docker installed on your machine.

Steps:

1. **Clone the Repository:**
   bash
   Copy code
   git clone https://github.com/your-username/your-repo.git
   cd your-repo
2. **Build and Start Docker Containers:** Use Docker Compose to build and start the frontend, backend, and PostgreSQL database services:
   bash
   Copy code
   docker-compose up -d --build

   This command builds the Docker images and starts the containers in detached mode (-d).
3. **Access the Application:**
   - **Frontend (Angular):**
     - URL: http://localhost:4201
   - **Backend (ASP.NET Core):**
     - Table API: https://localhost:8081/swagger/index.html
4. **Stopping the Services:** To stop and remove the containers, networks, and volumes defined in docker-compose.yml, run:
   bash
   Copy code
   docker-compose down

Project Structure

- **Frontend (Angular):**
  - Contains components for forms, data grid, authentication pages, and services to interact with backend APIs.

- **Backend (ASP.NET Core):**
  - Implements JWT-based authentication with endpoints for user management.
  - Provides endpoints for creating, reading, updating, and deleting tables (Table Management).
  - Supports CRUD operations for table data (Data Management).
- **Database (PostgreSQL):**
  - Stores dynamically created tables and associated data.

GitHub Repository

- Contains:
  - Frontend and backend code.
  - PostgreSQL database setup instructions.
  - Dockerfile and Docker Compose configuration for containerization.
  - Instructions on setting up and running the application using Docker.

Summary

The Table Management web application integrates frontend, backend, and database components using Docker for easy deployment and scalability. It offers dynamic table creation, CRUD operations, and secure authentication, catering to database management needs with a user-friendly interface.

This documentation provides an overview of how to set up, run, and interact with the application locally using Docker, ensuring a seamless development and deployment experience.