## **Problem Statement: JWT-Based RBAC in MERN Paginated Product App**

This MERN stack application demonstrates a product management system with:

- JWT-based authentication
- Role-Based Access Control (RBAC)
- Paginated product listings
- Search, edit, and delete capabilities based on user role

There are two roles supported:

- 1. Admin: Can view, edit, delete, and manage all products.
- 2. Consumer: Can only view paginated product listings.

Authentication is handled using JWT tokens stored in `localStorage`. Based on the user's role, the UI dynamically enables or disables edit/delete actions.

## **MongoDB Collection Schemas**

```
User Schema:
{
    username: String,
    password: String (hashed with bcrypt),
    role: { type: String, enum: ['admin', 'consumer'] }
}

Product Schema:
{
    name: String,
    price: Number,
    category: String,
    inStock: Boolean
}
```

## **Response Messages**

#### Register:

- Success: 'Registration successful. You can now login.'
- Failure: 'Registration failed: <error message>'

#### Login:

- Success: Stores JWT and role in localStorage
- Failure: 'Login failed. Please try again.'

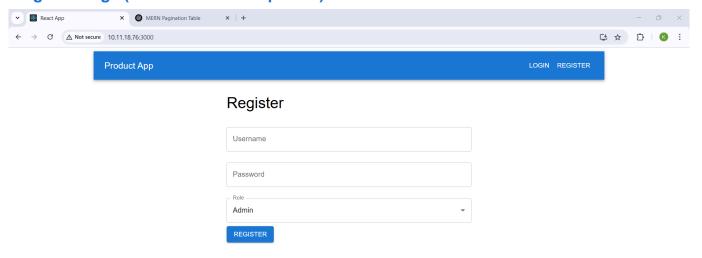
## **LocalStorage Structure (auth)**

```
Structure of the `auth` item in localStorage:

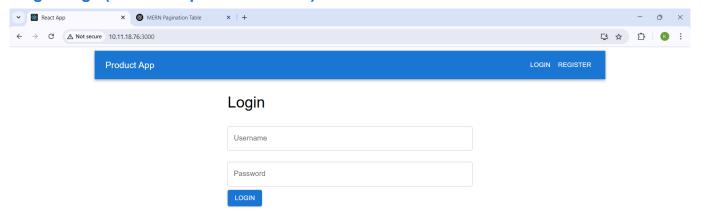
{
    "token": "<JWT_TOKEN>",
    "role": "admin" or "consumer"
}
```

# **Application Screenshots**

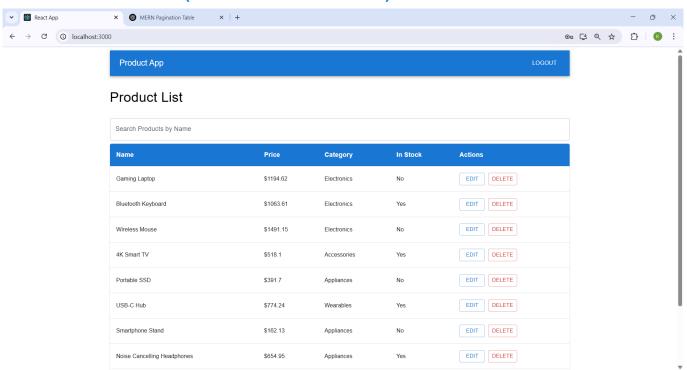
### Register Page (Role selection dropdown)



### Login Page (username/password fields)



#### **Product List for Admin (with Edit & Delete actions)**



**Product List for Consumer (read-only view)** 

## **KMIT Group - MERN Pagination App with JWT & RBAC**

