

Login App

A full-stack **Login App** built with:

- **Backend:** ASP.NET Core 9.0 with JWT Authentication and Identity Core.
- **Frontend:** React + Vite with Material-UI.

This app provides basic login and registration features with secure authentication using JWT tokens.

Features

- **Authentication:** Secure login and registration using JWT tokens.
- **Authorization:** Protected routes for authenticated users.
- **Frontend:**
 - Clean UI with Material-UI components.
 - Responsive design for user-friendly experience.
- **Backend:**
 - ASP.NET Core 9.0 with Entity Framework Core.
 - SQL Server database integration.
- **CORS Enabled:** Seamless interaction between frontend and backend during development.

Technologies Used

Backend:

ASP.NET Core 9.0
Entity Framework Core
SQL Server
JWT Authentication
Identity Core

Frontend:

React
Vite
Material-UI
Axios
React Router

Requirements

Backend:

.NET 9.0 SDK
SQL Server

Frontend:

Node.js (v16+ recommended)

Setup Instructions

1. Clone the Repository:

```
git clone https://github.com/your-repo-url/login-app.git  
cd login-app
```

2. Backend Setup

a. Navigate to the backend folder:

```
cd Backend
```

b. Configure appsettings.json:

Update the ConnectionStrings and JwtSettings in appsettings.json:

```
"ConnectionStrings": {
  "conn": "Data Source=DESKTOP-
U0BU3J3\\SQLEXPRESS;Database=TaskManagementApp;User
Id=sa;Password=VD@49812;MultipleActiveResultSets=true;TrustServerCerti
ficate=true"
},
"JwtSettings": {
  "Issuer": "LoginApp",
  "Audience": "LoginAppAPI",
  "SecretKey": "AppSuperSecretKey4981201234578789"
}
```

c. Migrate the database:

Run the following commands in the terminal to create the database and apply migrations:

```
dotnet ef migrations add InitialCreate
dotnet ef database update
```

d. Run the backend:

```
dotnet run
```

The backend will be available at <http://localhost:5226>.

3. Frontend Setup

a. Navigate to the frontend folder:

```
cd Frontend
```

b. Install dependencies:

```
npm install
```

c. Run the frontend:

`npm run dev`

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

Usage

Access the App:

Open the frontend in your browser at <http://localhost:5173>.

Register a new user or log in with an existing account.

Protected Routes:

The `/dashboard` route is accessible only after logging in.

Once logged in, the JWT token is stored in local storage for secure access to protected routes.

API Endpoints

Authentication Routes:

Register: POST `/api/Auth/register`

```
Body: {  
  "username": "testuser",  
  "email": "testuser@example.com",  
  "password": "Password123!"  
}
```

Login: POST `/api/Auth/login`

```
Body: {  
  "username": "testuser",  
  "password": "Password123!"  
}
```

Future Enhancements

- Add email verification during registration.
- Implement password recovery and reset functionality.
- Add role-based authorization.
- Deploy to production (Azure or AWS).

Contributions

Contributions are welcome! Feel free to fork this repository and create a pull request with improvements or bug fixes.