

Task Management System

A full-stack task management application built with React (frontend) and ASP.NET Core (backend). Features user authentication, CRUD operations for tasks, searching, pagination, and a responsive UI.

Features

Authentication

- User login with JWT token-based authentication
- Token stored in localStorage for session persistence
- Protected routes requiring authentication
- Logout functionality to clear session

Task Management

- Create new tasks with title, description, and status
- View all tasks in a paginated grid (2 tasks per page)
- Update task status (Pending ↔ Completed)
- Delete tasks with confirmation dialog
- Filter tasks by status (All, Pending, Completed)
- Search tasks by title or description
- Display creation and update timestamps

UI/UX

- Responsive design with modern styling
- Success/error message banners for user feedback
- Confirmation dialogs for destructive actions
- Pagination with Next/Previous buttons
- Real-time updates after operations

Tech Stack

Frontend

- **React** - UI library
- **Axios** - HTTP client for API calls
- **CSS** - Custom styling
- **LocalStorage** - Token persistence

Backend (Assumed)

- **ASP.NET Core** - Web API
- **Entity Framework** - ORM
- **JWT** - Authentication
- **SQL Server** - Database

Getting Started

Prerequisites

- Node.js (v14 or higher)
- npm or yarn
- Backend API running on <http://localhost:7167>

Installation

1. Clone the repository
2. Navigate to the frontend directory:

```
cd frontend
```

3. Install dependencies:

```
npm install
```

4. Start the development server:

```
npm start
```

5. Open <http://localhost:3000> in your browser

Login Credentials

The application uses static credentials for authentication (for demonstration purposes). Upon successful login, a JWT token is generated and stored in localStorage.

- **Username:** admin
- **Password:** admin123

API Endpoints

Authentication

- [POST /api/auth/login](#) - User login

Tasks

- [GET /api/tasks](#) - Get all tasks (optional ?status=0|1)
- [GET /api/tasks/{id}](#) - Get task by ID
- [POST /api/tasks](#) - Create new task
- [PUT /api/tasks/{id}](#) - Update task

- **DELETE** `/api/tasks/{id}` - Delete task

Project Structure

```
frontend/
├── public/
│   └── index.html
├── src/
│   ├── components/
│   │   ├── Login.js           # Login component
│   │   ├── TaskManager.js     # Main task management component
│   │   └── TaskManager.css    # Styles for components
│   ├── services/
│   │   └── api.js             # API service functions
│   ├── App.js                 # Main app component
│   ├── App.css                # Global styles
│   ├── index.js               # App entry point
│   └── setupTests.js
├── package.json
└── README.md
```

Available Scripts

- **npm start** - Runs the app in development mode
- **npm test** - Launches the test runner
- **npm run build** - Builds the app for production
- **npm run eject** - Ejects from Create React App (not recommended)

Usage

1. **Login:** Enter credentials to authenticate
2. **Create Task:** Fill the form and click "Add Task"
3. **View Tasks:** Browse tasks in the paginated list
4. **Search:** Use the search bar to filter by title/description
5. **Filter:** Use dropdown to filter by status
6. **Update:** Click "Mark Completed/Pending" to toggle status
7. **Delete:** Click "Delete" and confirm in the dialog
8. **Logout:** Click the "Logout" button in the top-right to end session

Contributing

1. Fork the repository
2. Create a feature branch
3. Make your changes
4. Test thoroughly
5. Submit a pull request

License

This project is licensed under the MIT License.