**Blazor Todo App Documentation**

**Overview**

The Blazor Todo App is a task management application built with .NET 9, featuring a .NET Core Web API as the backend and a Blazor WebAssembly standalone app as the frontend. The application follows best practices for Blazor architecture and provides a seamless task management experience.

**Features**

1. Task Management

* Create, edit, delete tasks.
* Mark tasks as completed, storing the completion date.

2. Filtering & Sorting

* Filter tasks by status (Completed, Not Completed).
* Dynamically fetch and display tasks using API calls.

3. User Interface

* Responsive Bootstrap-powered UI.
* Status indicators with color-coded badges.

4. API Integration

* Fetch, filter, and modify tasks via REST API calls.

**Setup Instructions**

***Prerequisites***

* + .NET 9 SDK installed.
  + A SQL Server instance.
  + A CORS-enabled API

***Backend (Web API)***

1. Clone and Set Up

* Clone the repository

[*https://github.com/PubuduAbeywickrama/TodoApp\_Vario.git*](https://github.com/PubuduAbeywickrama/TodoApp_Vario.git)

2. Configure Database

* Update `appsettings.json` with your SQL Server connection string:

*"ConnectionStrings": {*

*"DefaultConnection": "Server=your-server;Database=TodoDb;User Id=your- user;Password=your-password;"*

*}*

3. Apply Migrations & Run

*dotnet ef database update*

*API will start at* [*https://localhost:6605*](https://localhost:6605)*.*

**Frontend (Blazor WebAssembly)**

1. Navigate to Frontend

*cd ../* [*BlazorTodoApp*](https://github.com/PubuduAbeywickrama/TodoApp_Vario/tree/master/BlazorTodoApp)

2. Update API Endpoint

* In `Program.cs`, update the base address for the API:

builder.Services.AddScoped(sp => new HttpClient { BaseAddress = new Uri("https://localhost:6605/") });

3. Run the Application

* Blazor app will start at https://localhost:7295.

**API Endpoints**

1. Fetch All Tasks

GET /api/TodoTask

2. Fetch Filtered Tasks

*GET /api/TodoTask?status=Completed*

3. Mark Task as Completed

PATCH /api/TodoTask/{id}/complete

4. Delete a Task

*DELETE /api/TodoTask/{id}*

**Assumptions**

* + The app assumes that all API calls are successful (error handling is minimal).
  + The frontend and backend can run on separate hosts (CORS settings may be required).

**Next Steps**

* + Implement user authentication for task ownership.
  + Improve error handling and API responses.
  + Add due date reminders and notifications.

Github Link : <https://github.com/PubuduAbeywickrama/TodoApp_Vario>

For further details, refer to the project repository or contact the developer Pubudu Abeywickrama.