

Task – Fullstack Developer .NET (m/f)

Develop a web application using **.NET 8** for the backend and a TypeScript-based frontend framework, such as **Angular** or **React**. The application should implement a **CRUD** (Create, Read, Update, Delete) functionality for managing tickets, similar to the table shown in the provided image. The key details for each ticket are:

1. **Ticket ID:** A unique identifier for each ticket.
2. **Description:** A short summary of the ticket issue.
3. **Status:** Can be either "Open" or "Closed".
4. **Date:** The date the ticket was created.
5. **Actions:** Options to update or delete each ticket.

Ticket Id	Description	Status	Date	Actions
1002	Promotion code issued	Open	May-29-2022	Update Delete
1003	Additional user account	Open	May-27-2022	Update Delete
1004	Change payment method	Open	May-28-2022	Update Delete
1005	Activate account	Closed	May-28-2022	Update Delete
1007	Great job	Closed	May-29-2022	Update Delete
1008	Another Great Job	Closed	May-29-2022	Update Delete
1000	Help with Login	Closed	May-29-2022	Update Delete
1024	Happy Customer	Open	May-29-2022	Update Delete

[Add New](#)

Requirements

- **Backend (API):**
 - Use **.NET 8** to build a RESTful API.
 - The API should handle the basic CRUD operations (Create, Read, Update, Delete) for tickets.
 - Use a suitable database to store the ticket data (e.g., SQL Server, PostgreSQL).
 - Use EntityFramework
 - Implement pagination for the list of tickets.
- **Frontend (UI):**
 - Use **Angular** or **React** to create a responsive user interface that interacts with the .NET backend.
 - Implement a table view of tickets **that looks exactly as the provided image**.
 - The user should be able to:
 - Create a new ticket.
 - Edit/update an existing ticket.
 - Delete a ticket.
 - View the list of all tickets.
 - The UI should include an "Add New" button to allow users to add new tickets.

- **Additional Requirements:**
 - Implement client-side form validation (e.g., ensuring required fields are filled).
 - The status should be a dropdown allowing users to choose between "Open" or "Closed".
 - Use modern UI styling (Bootstrap, Material UI, or any preferred library) to make the application visually appealing.
- **Bonus Points:**
 - Implement sorting and filtering on the ticket table.
 - Add unit tests.

Submission

- Provide the source code in a GitHub repository.
- Include a **README** file with setup instructions for running the application.

YOUR HAHN SOFTWARE TEAM