**Executing the project:**

1. Visual Studio: open the solution and run it.
2. Visual Studio Code or CLI: in the command line execute **dotnet watch run**

**1. Shortcuts you felt necessary to take.**

Since it is a proof-of-concepts and the technical requirements are simple (SQL database and C#) I took multiples shortcuts:

* I decided to use EF with code first workflow. Meaning that when the project is executed for the first time, the database is generated in LocalDB.
* Meeting table is not normalized.
* I used Razor pages with CRUD scaffolding.
* I took advantage of Data Notations (Meetings Model) to use default validations.
* A simple pink background is used to indicate that a Follow-Up has not been done.

**2. What you would do differently for a production ready system.**

* Obviously create multiples tables and their relationships, such as: Company, Contacts, Meetings, etc.
* For the back-end I would create a Web API project, and evaluate the possibility to use EF or Dapper (to retrieve information faster).
* For the front-end I would use Angular to create a SPA (and some nice UI components like PrimeNG). And I would use Modal Dialog windows for the CRUD operations.
* Authentication and Authorization, and once the user is log in, display the username in the right corner with a bell icon to display the number of meetings without follow-up (non-intrusive alerts). On click, this icon should display a clickable list of alerts to find/edit items easier.