

Angular Project

For this project, you will work in assigned groups to create an app with a C# API back-end and an Angular Front End around one of these themes:

- Help Desk Ticket System
- Bootcamp Study Buddy
- Upmeet Event System

Build Specifications

Your application must include:

C# Backend

- A SQL Server Database
- A Data Access Layer (DbContext) including at least two model classes
- At least two API Controllers, with actions including a mixture of HTTP Get, Post, Put/Patch, and Delete, as required by the theme chosen.

Angular Front End

- At least 3 components
- At least 1 service for API access
- Routing between at least two of the components

User login is not required for either theme.

Help Desk Ticket System Required Functionality

1. User sees a list of tickets upon entering the site.
2. By clicking a ticket, user sees detail about the question or issue.
3. User can create a new ticket.
4. User can add a resolution and close a ticket.
5. User can bookmark a ticket.
6. User can see a list of bookmarked tickets.

Tickets should include fields identifying who opened the ticket and who resolved it, but you can take user-entered data (name and/or email) for each.

Tickets should go into one database table, favorites into another. User favorites should be stored in the database with some kind of UserID but login is not required.

Bootcamp Study Buddy Required Functionality

1. User sees a list of study questions upon entering the site.
2. User can click on a question to see the answer.
3. User can mark question & answer for list of favorites.

4. User can view their list of favorite questions and answers.
5. User can remove a question & answer from their list of favorites.
6. User can add a question and answer.

Q&A should go into one database table, favorites into another. User favorites should be stored in the database with some kind of UserID but login is not required.

Upmeet Event System Required Functionality

1. User sees a list of events upon entering the site.
2. User can click on an event to see more detail.
3. User can mark event for list of favorites.
4. User can view their list of favorite events.
5. User can remove an event from their list of favorites.
6. User can add an event.

Events should go into one database table, favorites into another. User favorites should be stored in the database with some kind of UserID but login is not required.