

# Project Development Steps

1. Form your groups for minimum 1 maximum 4 friends.
2. After your project's phases are completed, you will need to upload your Solution to any friend's GitHub account and send the GitHub Repository link to the instructor including the project team member names.
3. Choose a domain other than Users and Locations for your project from:  
[https://need4code.com/DotNet?path=.NET%5C00\\_Files%5CProjects%5CDiagrams.jpg](https://need4code.com/DotNet?path=.NET%5C00_Files%5CProjects%5CDiagrams.jpg)

If you want to develop your project from another domain of yours, create the E-R or Class Diagram and send it to the instructor via e-mail for approval.

For another domain of yours, you must have different data types (string, decimal, DateTime, bool, int, etc.) in entities, and one 1 to Many and one Many to Many relationship between your domain entities.

If you need another project idea, you can contact the instructor via e-mail.

4. Create your entity classes in your solution's Application (APP) Project Domain.
5. Create DbSet in your Database Context class for each entity in your APP Project Domain.
6. Define the connection string in your API Projects' appsettings.json file.
7. Manage the dependency of your Database Context class in the Inversion of Control (IoC) Container section in the Program.cs file of your API Projects.
8. Create your databases.
9. Create request, response and handler classes for CRUD (Create, Read, Update and Delete) operations for each entity other than the many to many relational entities of your APP Project Domain. Implement MediatR interfaces.
10. Manage Inversion of Control for mediator in the Inversion of Control (IoC) Container section in the Program.cs file of your API Project.
11. Copy the Templates folder from any API Project of any previous Solution under your Solution's API Projects' root folders.
12. Scaffold the controllers for each entity other than the many to many relational entities of your domain.
13. Test your API endpoints.