

# CRUD - Code Affection

✓ get nugets

## # model

- props
- nullable warning (supress or initialise)
- creat physical table by migration
  - before that those props should be decorated as [column].  
Bcs it will correspond column in DB.
- [column(TypeName = "nvarchar(50)")] ]

✓ DbContext.cs

- ctor (options): base(options)
- to connect DbContext to MainModel we need DbSet<Model> in the DB Context.cs

### ✓ Dependency Injection

- builder.Services.AddDbContext<>(options  
   ⇒ options.UseSqlServer(builder.conf....  
                                     ..connectionstring);

### ✓ ConnectionString to app settings json: to check all services on your pc ⇒

- [commandprompt = services.msc]
- service requires running
  - ... Database = Give a name for migration  
   or initialcatalog = give anyname ...
  - authentication = true

- Package Manager Console
  - add-migration "give a name"
  - update-database → executes columns in Model

## # controller

- add MVC cont with entityframework.
- choose model and context we built.
- code generation. design package (check installed)
- · · scaffolding mechanism connects context and model to controller and creates methods!

✓ Sqlserver instance is created under program / builder ... config ... connection string. This is dependency injection. Because of (DI), now controller gets same context and sqlserver instance

- as a parameter!
- methods built as views

# Program.cs

- DI
- Route Control

# Layout view

- parent view
- html's
- renderbody - carrier on all views
- head reference : bootstrap
- ✓ bootstrap reference - cdnjs.com

# ~~CREATE~~ - MIN26 -

# VALIDATION - MIN32 -

# FORM SUBMISSION - MIN37 -