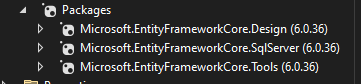


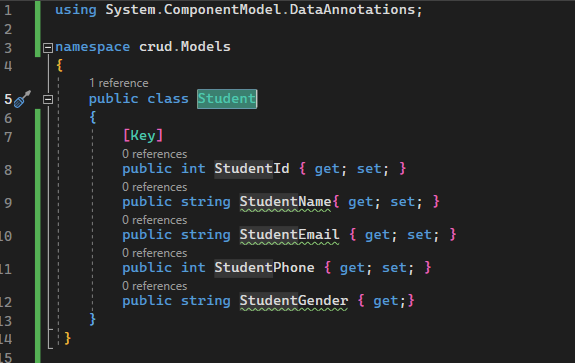
**CODE FIRST APPROACH:**

**3 STEPS PROCESS:**

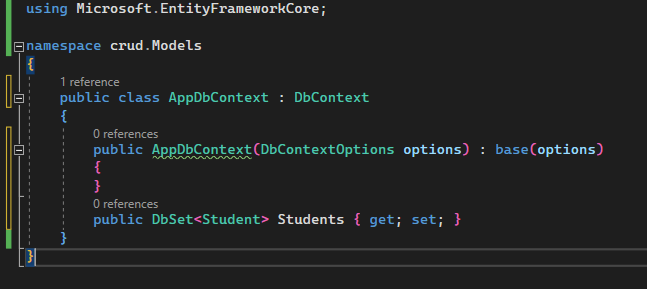
**1ST step: install 3 packages in your asp.net core 6 MVC application.**



**2nd step: create model class, create DbContext class.**

Model class:  


DbContextClass:



**3rd step: create a connection string in appsettings.json file.**

"ConnectionStrings": {

"DefaultConnection": "Server=YOURSERVERNAME; Database=YOURDATABASENAME; Trusted\_Connection=True; MultipleActiveResultSets=true"

}

Or

"ConnectionStrings": {

"DefaultConnection": "Data Source=Faculty-9; Initial Catalog=CodeFirstDB; User Id = sa; Password = aptech;"

},

**4th step: Registering connection string in Program.cs file.**

var provider = builder.Services.BuildServiceProvider();

var config = provider.GetRequiredService<IConfiguration>();

builder.Services.AddDbContext<ApplicationDbContext>(item => item.UseSqlServer(config.GetConnectionString("DefaultConnection")));

var app = builder.Build();

**5th step: Add a migration and run the migration.**

**Tools>nuget>colsole**

PM> add-migration CodeFistDB

PM> update-database

**In controller file:**

public class HomeController : Controller

{

//private readonly ILogger<HomeController> \_logger;

//public HomeController(ILogger<HomeController> logger)

//{

// \_logger = logger;

//}

**Comment it, then add below code**

public class HomeController : Controller

{

private readonly AppDbContext \_db;

public HomeController(AppDbContext db)

{

this.\_db = db;

}

//private readonly ILogger<HomeController> \_logger;

//public HomeController(ILogger<HomeController> logger)

//{

// \_logger = logger;

//}

Crud: create

[HttpPost]

public IActionResult AddStudent(Student student){

Student admin1 = new Student ()

{

Username = admin.Username,

Email = admin.Email,

Password = admin.Password

};

\_applicationdb.Students.Add(student);

// Save changes to the database…jo kam kr rhy hn wo save b krana h

\_db.SaveChanges();

TempData["successMessage"] = "Student inserted successfully!";

// Redirect to another view, e.g., Index, after successful creation

return RedirectToAction("Index");

}

// If the model is invalid, return the same view with validation messages. Same page fields ya form wala return hoga

<div class="form-group">

<**label** **asp-for**="Name" class="control-label"></**label**>

<**input** **asp-for**="Name" class="form-control" />

<**span** **asp-validation-for**="Name" class="text-danger"></**span**>

</div>

//ab add krengy alert k lye data. To wo temp data use hoga.

<div class="alert alert-success alert-dismissible fade show" role="alert">

@TempData["successMessage"]

</div>

//cross end me add

<div class="alert alert-success alert-dismissible fade show" role="alert">

@TempData["successMessage"]

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>

</div>

//condition b add krdengy ab. Ta k insert hony k bad he ho show alert home page py

@{

if (TempData["successMessage"] != null)

{

<div class="alert alert-success alert-dismissible fade show" role="alert">

@TempData["successMessage"]

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>

</div>

}

}

Then, open pg 40

**Update the HomeController**

Modify the Index action in the HomeController to retrieve and pass the data to the view:

public IActionResult Index()

{

// Retrieve the list of students from the database

var students = \_applicationdb.Students.ToList();

// Pass the students to the view

return View(students);

}

**Delete:**

public IActionResult Delete(int id)

{

// Retrieve the list of students from the database

var students = \_applicationdb.Students.Find(id);

// Pass the students to the view

return View(students);

}

Delete.cshtml update kro yh last me

**POST:**Delete**(for actual deletion)**

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult DeleteConfirmed(int id)

{

var student = \_applicationdb.Students.Find(id);

if (ModelState.IsValid)

{

// Add the student to the database context

\_applicationdb.Students.Remove(student);

// Save changes to the database

\_applicationdb.SaveChanges();

TempData["successMessage"] = "Student deleted successfully!";

// Redirect to another view, e.g., Index, after successful creation

return RedirectToAction("Index");

}

// If the model is invalid, return the same view with validation messages

return View(student);

}

Or

//[HttpPost]

//[ValidateAntiForgeryToken]

public IActionResult DeleteConfirmed(int id)

{

var admins = \_db.Admin.Find(id);

if (ModelState.IsValid)

{

\_db.Admin.Remove(admins);

\_db.SaveChanges();

}

return RedirectToAction("Index");

}

**Index.cshtml:**

<**a** **asp-action**="Delete" **asp-route-id**="@item.Id" class="btn btn-danger">Delete</**a**>

Delete.cshtml:

<**a** **asp-action**="DeleteConfirmed" class="btn btn-danger" **asp-route-id**="@Model.AdminId">Delete</**a**>

or

<**form** **asp-action**="DeleteConfirmed" **asp-controller**="Home" method="post">

<**input** **type**="hidden" **asp-for**="Id" />

<input type="submit" value="Delete" class="btn btn-danger" /> |

<**a** **asp-action**="Index">Back to List</**a**>

</**form**>

**Add image:**

**Updated Student model:**

[Column(TypeName = "varchar(200)")]

public string MainImage { get; set; }

[NotMapped]

public IFormFile ProImg { get; set; }

**Migration add karne ka command:**

Tools>nuget package manager> console

Add-Migration AddMainImageColumn

Isse ek nayi migration file generate hogi Migrations folder me.

**Migration ko database par apply karne ke liye command run karein:**

Update-Database

**Folder Path Define Karna:**

string folderUpload = Path.Combine(\_webhostEnv.WebRootPath, "images");

Path.Combine: Folder ka path banata hai. Yahan server ke root folder (wwwroot) ke andar ek images folder ka path banaya gaya hai.

\_webhostEnv.WebRootPath: ASP.NET Core ka predefined property hai jo wwwroot ka physical path return karta hai. Static files (CSS, JS, images) yahin store hoti hain.

**Unique File Name Generate Karna(img ka name):**

filename = Guid.NewGuid().ToString() + "\_" +student.Proimg.FileName;

**Explanation**:

* Guid.NewGuid(): Ek unique identifier generate karta hai (e.g., a3f0b6cd-1234-4abc-9d6d-567890def012).
* student.Proimg.FileName: Uploaded file ka original naam return karta hai (e.g., profile.jpg).
* Combined Result: Unique naam generate hota hai, e.g., a3f0b6cd-1234-4abc-9d6d-567890def012\_profile.jpg.
* **Why Unique Name?**:
  + Har file ka unique naam hone se overwrite hone ka risk nahi hota.

**Full File Path Create Karna:**

string filepath = Path.Combine(folderUpload, filename);

**Explanation**:

* Path.Combine: folderUpload (destination folder ka path) aur filename (file ka naam) ko combine karta hai.
* Example:
  + Folder Path: C:\MyProject\wwwroot\images
  + Filename: a3f0b6cd-1234-4abc-9d6d-567890def012\_profile.jpg
  + Final Path: C:\MyProject\wwwroot\images\a3f0b6cd-1234-4abc-9d6d-567890def012\_profile.jpg

**Missing Part (Save File to Path)**

student.Proimg.CopyTo(new FileStream(filepath, FileMode.Create));

**Example:**

* **student.Proimg**: Suppose user ne ek profile.jpg file upload ki.
* Humne filepath banaya: C:\MyProject\wwwroot\images\a123\_profile.jpg.
* CopyTo ye ensure karega ki profile.jpg file ko is folder mein a123\_profile.jpg ke naam se save kar diya jaye.

**Create Student Object**

Student student2 = new Student()

{

Name = student.Name,

Age = student.Age,

Address = student.Address,

Phone = student.Phone,

Email = student.Email,

MainImage = filename

};

Ek naya Student object create hota hai jisme form ki details (jaise Name, Age, Address) aur uploaded image ka filename assign hota hai.

**Database Operations**

\_applicationdb.Students.Add(student2);

\_applicationdb.SaveChanges();

 **Add**: Naye student record ko database context ke Students table mein add karta hai.

 **SaveChanges**: Database ke changes ko persist karta hai.

TempData["successMessage"] = "Student inserted successfully!";

return RedirectToAction("Index");

}

**Return View**

return View(student);

}

Agar sab kuch sahi se ho jaye ya koi error ho, toh view return hota hai.