Back end part–ASP.net +MVC

1-task ➜ Create project skeleton

2-task ➜ Basic injections–

- 🌺 **2.1 – All required NuGet installations**
- 🌺**2.2 - app.setting–connection string**
- 🌺 **2.3 - program.cs**

3-task ➜ Models department

- 🌺 **3.1 — Create Your Models Folder & RegisterClass**
- 🌺**3.2— Paste the Register Model Code**
- 🌺**3.2-Education-career Class File**
- 🌺 **3.3-Family-information Class File**
- 🌺 **3.4-siblings Class File**
- 🌺 **3.5-LifestyleInfo Class File**
- 🌺**3.7-Astrology Class File**
- 🌺 **3.7-My-expectations Class File**
- 🌺**3.8-Upload photos File**
- 🌺**3.9-Verify-details Class File**

🌺 3.10-Payment Class File

🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸

1-task ➜ create Project skeleton

🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸

✅ STEP-BY-STEP GUIDE — Create MatrimonyAPI in Visual Studio 2022

🟢 Step 1 — Open Visual Studio 2022

- Launch Visual Studio 2022.
- On the start screen, click "Create a new project."

🟢 Step 2 — Select Project Template

1. In the search box, type:
   👉 "ASP.NET Core Web API"
2. Select ASP.NET Core Web API template (C#).
3. Click Next.

🟢 Step 3 — Configure Your Project

1. Project Name: MatrimonyAPI
2. Location: choose your backend folder (e.g., D:\API)

3. Solution Name: keep MatrimonyAPI or create your own.
4. Click Next.

🟢 Step 4 — Additional Information

1. Framework: Select .NET 8.0 (LTS)
2. Authentication Type: choose None (for now).
3. ✅ Use controllers (uncheck “Use minimal APIs”)
4. ✅ Configure for HTTPS
5. ✅ Enable OpenAPI support (this adds Swagger)
6. Click Create.

🟢 Step 5 — Verify the project

- Visual Studio creates the MatrimonyAPI folder and opens the project.

Solution Explorer should show:

```
 Controllers/
Program.cs
appsettings.json
...
```

- Press Ctrl + F5 (or click ▶️ “Start Without Debugging”).
- Swagger page will open automatically →
  👉 https://localhost:5001/swagger

✅ You’ve successfully created your API skeleton.

🔍 1. Folder Structure

Ensure your structure looks like this:

```
MatrimonyAPI/
 ├ Controllers/
 ├ Data/
 |  └ AppDbContext.cs
 ├ Models/
 |  ├ Register.cs
 |  ├ EducationCareer.cs
 |  ├ FamilyInfo.cs
 |  ├ Sibling.cs
 |  ├ LifestyleInfo.cs
 |  ├ AstrologyInfo.cs
 |  ├ Expectations.cs
 |  ├ UploadPhotos.cs
 |  ├ Verify.cs
 |  └ Payment.cs
 ├ appsettings.json
 ├ Program.cs
```

```
  └ MatrimonyAPI.csproj
```

🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸

2-task ➔Basic injections–

2.1–All required NuGet installations

2.2-app.setting–connection string

2.3-program.cs

🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸

2.1–All required NuGet installations

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖✅**1–STEP– Install NuGet packages**

Right-click the project ➜ Manage NuGet Packages ➜ Browse tab ➜ install:

`Microsoft.EntityFrameworkCore.SqlServer`

`Microsoft.EntityFrameworkCore.Tools`

(If you get errors later with migrations, you can also install `Microsoft.EntityFrameworkCore.Design`.)

2.2- 🧱 Appsettings.json–Add Your DB Connection String

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

```
"ConnectionStrings": {
    "DefaultConnection":
"Server=YOUR_SERVER_NAME;Database=MatrimonyDB;Trusted_Connection=True;TrustServerC
ertificate=True;"

}
```

2.3-program.cs

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖🌸🌸
🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸

3-task ➜ Models department

Model classes for all 10 sections

🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🌸🧱

3.1 — Create Your Models Folder & RegisterClass

🟢 **Step 1 — Create a Folder Named Models ➜ right-click your project name → Add → New Folder ➜ Name it `Models`**

🟢 **Step 2 — Add the Class File** → Now, right-click the **Models** folder → **Add** → **Class...**📝 Name it:`Register.cs`

🧱 3.2— Paste the Register Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    [Table("Register")]
    public class Register
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int UserID { get; set; }

        [Required, StringLength(100)]
        public string FirstName { get; set; } = string.Empty;
        [Required, StringLength(100)]
        public string LastName { get; set; } = string.Empty;

        [Required, StringLength(100)]
        [EmailAddress]
        public string Email { get; set; } = string.Empty;

        [Required, StringLength(15)]
        [Phone]
        public string PhoneNumber { get; set; } = string.Empty;

        [Required, StringLength(10)]
        public string Gender { get; set; } = string.Empty;

        [Required, StringLength(200)]
        public string Password { get; set; } = string.Empty;

        [StringLength(20)]
        public string LoginType { get; set; } = "Password";

        [StringLength(20)]
        public string AccountType { get; set; } = "Individual";

        [StringLength(20)]
        public string AccountStatus { get; set; } = "Active";

        public DateTime CreatedDate { get; set; } = DateTime.Now;

        public DateTime? LastLoggedIn { get; set; }

        [StringLength(50)]
        public string? Religion { get; set; }

        [StringLength(50)]
        public string? Caste { get; set; }

        [StringLength(50)]
        public string? SubCaste { get; set; }
    }
}
```

🧱 3.2-Education-career Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

EducationCareer.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;


namespace MatrimonyAPI.Models

{

    [Table("EducationCareer")]

    public class EducationCareer

    {

        [Key]

        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]

        public int EducationCareerID { get; set; }


        [ForeignKey("Register")]

        public int UserID { get; set; }
```

```
// 🎓 Education Information
[StringLength(100)]
public string? EducationLevel { get; set; }

[StringLength(200)]
public string? EducationDetails { get; set; }

// 💼 Employment Information
[StringLength(100)]
public string? Designation { get; set; }

[StringLength(150)]
public string? Organization { get; set; }

[StringLength(50)]
public string? TotalExperience { get; set; }

[StringLength(50)]
public string? RelevantExperience { get; set; }

[StringLength(50)]
public string? AnnualIncome { get; set; }

[StringLength(100)]
public string? WorkLocation { get; set; }

// 🏢 Business Information
```

```
        [StringLength(150)]
        public string? BusinessName { get; set; }

        [StringLength(100)]
        public string? BusinessType { get; set; }

        [StringLength(150)]
        public string? KeyDesignation { get; set; }

        // 🔗 Navigation Property
        public Register? Register { get; set; }
    }
}
```

🧱 3.3-Family-information Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

FamilyInfo.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    [Table("FamilyInfo")]
```

```
public class FamilyInfo
{
    [Key]
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int FamilyID { get; set; }

    [ForeignKey("Register")]
    public int UserID { get; set; }

    // 👨‍👩‍👦 Father Details
    [StringLength(100)]
    public string? FatherName { get; set; }

    [StringLength(100)]
    public string? FatherProfession { get; set; }

    [StringLength(100)]
    public string? FatherHometown { get; set; }

    // 👩‍👧 Mother Details
    [StringLength(100)]
    public string? MotherName { get; set; }

    [StringLength(100)]
    public string? MotherProfession { get; set; }
```

```
        [StringLength(100)]
        public string? MotherHometown { get; set; }

        // 🏠 Family Information
        [StringLength(200)]
        public string? LanguageKnown { get; set; }         // e.g., Tulu, Kannada, English

        [StringLength(50)]
        public string? FamilyValue { get; set; }           // e.g., Traditional, Moderate

        [StringLength(50)]
        public string? FamilyStatus { get; set; }          // e.g., Middle Class, Upper Middle Class

        public int? TotalSiblings { get; set; } = 0;       // Total number of siblings

        // 🔗 Navigation Property
        public Register? Register { get; set; }
    }
}
```

🧱 3.4-siblings Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

siblings.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    public class Sibling
    {
        [Key]
        public int SiblingID { get; set; }

        [Required]
        [ForeignKey("FamilyInfo")]
        public int FamilyID { get; set; }

        [MaxLength(100)]
        public string? SiblingName { get; set; }

        [MaxLength(10)]
        public string? Gender { get; set; }

        [MaxLength(50)]
        public string? MaritalStatus { get; set; }

        [MaxLength(100)]
        public string? Occupation { get; set; }

        // Navigation property for relationship
        public FamilyInfo? FamilyInfo { get; set; }
    }
}
```

🧱 3.5-LifestyleInfo Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

LifestyleInfo.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
```

```
    {
        public class LifestyleInfo
        {
            [Key]
            public int LifestyleID { get; set; }

            [Required]
            [ForeignKey("Register")]
            public int UserID { get; set; }

            // Physical Attributes
            [MaxLength(10)]
            public string? Height { get; set; }

            [MaxLength(10)]
            public string? Weight { get; set; }

            [MaxLength(50)]
            public string? Diet { get; set; }

            [MaxLength(50)]
            public string? Complexion { get; set; }

            [MaxLength(50)]
            public string? Physique { get; set; }
            [MaxLength(10)]
            public string? PhysicalDisability { get; set; }

            // Lifestyle Habits
            [MaxLength(50)]
            public string? SmokingHabit { get; set; }

            [MaxLength(50)]
            public string? DrinkingHabit { get; set; }

            // Navigation Property
            public Register? Register { get; set; }
        }
    }
```

🧱 3.7-Astrology Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

Astrology.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    public class AstrologyInfo
    {
        [Key]
        public int AstrologyID { get; set; }

        [Required]
        public int UserID { get; set; }

        [MaxLength(50)]
        public string? Raasi { get; set; }  // Zodiac sign

        [MaxLength(50)]
        public string? Nakshatra { get; set; }  // Birth star

        [Column(TypeName = "date")]
        public DateTime? DateOfBirth { get; set; }  // e.g., '1995-03-21'

        [MaxLength(20)]
        public string? TimeOfBirth { get; set; }  // e.g., '10:30 AM'

        [MaxLength(100)]
        public string? PlaceOfBirth { get; set; }  // City or town name

        // Navigation property (optional, if you have a Register model)
```

🧱 3.7-My-expectations Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 **Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:**

Myexpectations.cs

🟢 **Step 2 – Paste the Model Code**

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    public class Expectations
    {
        [Key]
        public int ExpectationID { get; set; }

        [Required]
        public int UserID { get; set; }

        // 1️⃣ Preferred Age
```

```
public int? PreferredAgeFrom { get; set; }

public int? PreferredAgeTo { get; set; }


// 2️⃣ Expected Salary

[MaxLength(50)]

public string? ExpectedSalaryFrom { get; set; }


[MaxLength(50)]

public string? ExpectedSalaryTo { get; set; }


// 3️⃣ Marital Status

[MaxLength(50)]

public string? PreferredMaritalStatus { get; set; }


// 4️⃣ Preferred Region

[MaxLength(100)]

public string? PreferredRegion { get; set; }


// 5️⃣ Preferred Language

[MaxLength(100)]

public string? PreferredLanguage { get; set; }


// 6️⃣ Education

[MaxLength(100)]

public string? PreferredEducation { get; set; }
```

```
        // 7️⃣ Your Opinion / Other Preferences
        public string? YourOpinion { get; set; }

        // 8️⃣ Height Expectation
        [MaxLength(10)]
        public string? PreferredHeightFrom { get; set; }

        [MaxLength(10)]
        public string? PreferredHeightTo { get; set; }

        // 9️⃣ Caste Preference
        [MaxLength(100)]
        public string? PreferredCaste { get; set; }

        // Navigation property (optional, links to Register table)
        [ForeignKey("UserID")]
        public Register? Register { get; set; }
    }
}
```

🧱 3.8-Upload photos File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

uploadphotos.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    public class UploadPhotos
    {
        [Key]
        public int PhotoID { get; set; }

        [Required]
        public int UserID { get; set; }

        [Required]
        [MaxLength(200)]
        public string PhotoPath { get; set; } = string.Empty;  // File path or URL

        public bool IsProfilePhoto { get; set; } = false;      // Default 0

        public DateTime UploadedDate { get; set; } = DateTime.Now;  // Default GETDATE()
        // Navigation property (optional, links to Register table)
        [ForeignKey("UserID")]
        public Register? Register { get; set; }
    }
}
```

🧱 3.9-Verify-details Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

verifydetails.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MatrimonyAPI.Models
{
    public class Payment
    {
        [Key]
        public int PaymentID { get; set; }
        [Required]
        public int UserID { get; set; }

        // 💳 Payment Details
        [Required]
        [MaxLength(100)]
        public string UPI_ID { get; set; } = string.Empty;  // User’s entered UPI ID

        [MaxLength(200)]
        public string? QRCodePath { get; set; }  // Path or URL for stored QR image

        [Required]
        [MaxLength(100)]
        public string TransactionID { get; set; } = string.Empty;  // Transaction reference

        // 🛒 Payment Product / Plan
        [MaxLength(100)]
        public string Product { get; set; } = "Profile Activation";

        [Column(TypeName = "decimal(10,2)")]
        public decimal Subtotal { get; set; } = 1000.00m;

        [Column(TypeName = "decimal(10,2)")]
        public decimal GST { get; set; } = 180.00m;
        // Computed Column — not updated directly
        [DatabaseGenerated(DatabaseGeneratedOption.Computed)]
        [Column(TypeName = "decimal(10,2)")]
        public decimal TotalAmount { get; private set; }

        // 📊 Payment Tracking
        [MaxLength(20)]
        public string PaymentStatus { get; set; } = "Pending";  // Pending / Success / Failed

        [MaxLength(50)]
        public string PaymentMethod { get; set; } = "UPI";  // UPI / Card / NetBanking

        public DateTime PaymentDate { get; set; } = DateTime.Now;

        // 🔗 Navigation property (link to Register)
        [ForeignKey("UserID")]
        public Register? Register { get; set; }
    }
}
```

🧱 3.10-Payment Class File + model class code

➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖➖

🟢 Step 1 – Right-click on the Models folder → Select Add → Class...📝 Name it exactly:

Payment.cs

🟢 Step 2 – Paste the Model Code

🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹🔹

```
using System;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;


namespace MatrimonyAPI.Models

{

    public class Payment

    {

        [Key]

        public int PaymentID { get; set; }


        [Required]

        public int UserID { get; set; }


        // 💳 Payment Details

        [Required]

        [MaxLength(100)]

        public string UPI_ID { get; set; } = string.Empty;  // User’s entered UPI ID


        [MaxLength(200)]
        public string? QRCodePath { get; set; }  // Path or URL for stored QR image

        [Required]
        [MaxLength(100)]
        public string TransactionID { get; set; } = string.Empty;  // Transaction reference

        // 🛒 Payment Product / Plan
        [MaxLength(100)]
        public string Product { get; set; } = "Profile Activation";

        [Column(TypeName = "decimal(10,2)")]
        public decimal Subtotal { get; set; } = 1000.00m;

        [Column(TypeName = "decimal(10,2)")]
        public decimal GST { get; set; } = 180.00m;

        // Computed Column — not updated directly
        [DatabaseGenerated(DatabaseGeneratedOption.Computed)]
        [Column(TypeName = "decimal(10,2)")]
        public decimal TotalAmount { get; private set; }

        // 📊 Payment Tracking
        [MaxLength(20)]
        public string PaymentStatus { get; set; } = "Pending";  // Pending / Success / Failed
        [MaxLength(50)]
        public string PaymentMethod { get; set; } = "UPI";  // UPI / Card / NetBanking

        public DateTime PaymentDate { get; set; } = DateTime.Now;

        // 🔗 Navigation property (link to Register)
        [ForeignKey("UserID")]
        public Register? Register { get; set; }
    }
}
```

4-task ➜ Create Your DbContext Class