РАЗРАБОТКА НА СОФТУЕР

**Тема:**

Управление на екскурзиите в туристическа агенция "Dream Journey"

**Изготвили:** Йоана Гаджакова,

Михаела Балийска,

Шейда Неби

**Ръководител:** Красимира Юрукова

Съдържание

**Увод 2**

**Анализ на изискванията 2**

**Процес на разработка2**

Създаване на проекта 2

Структура на проекта3

Разработка на проекта5

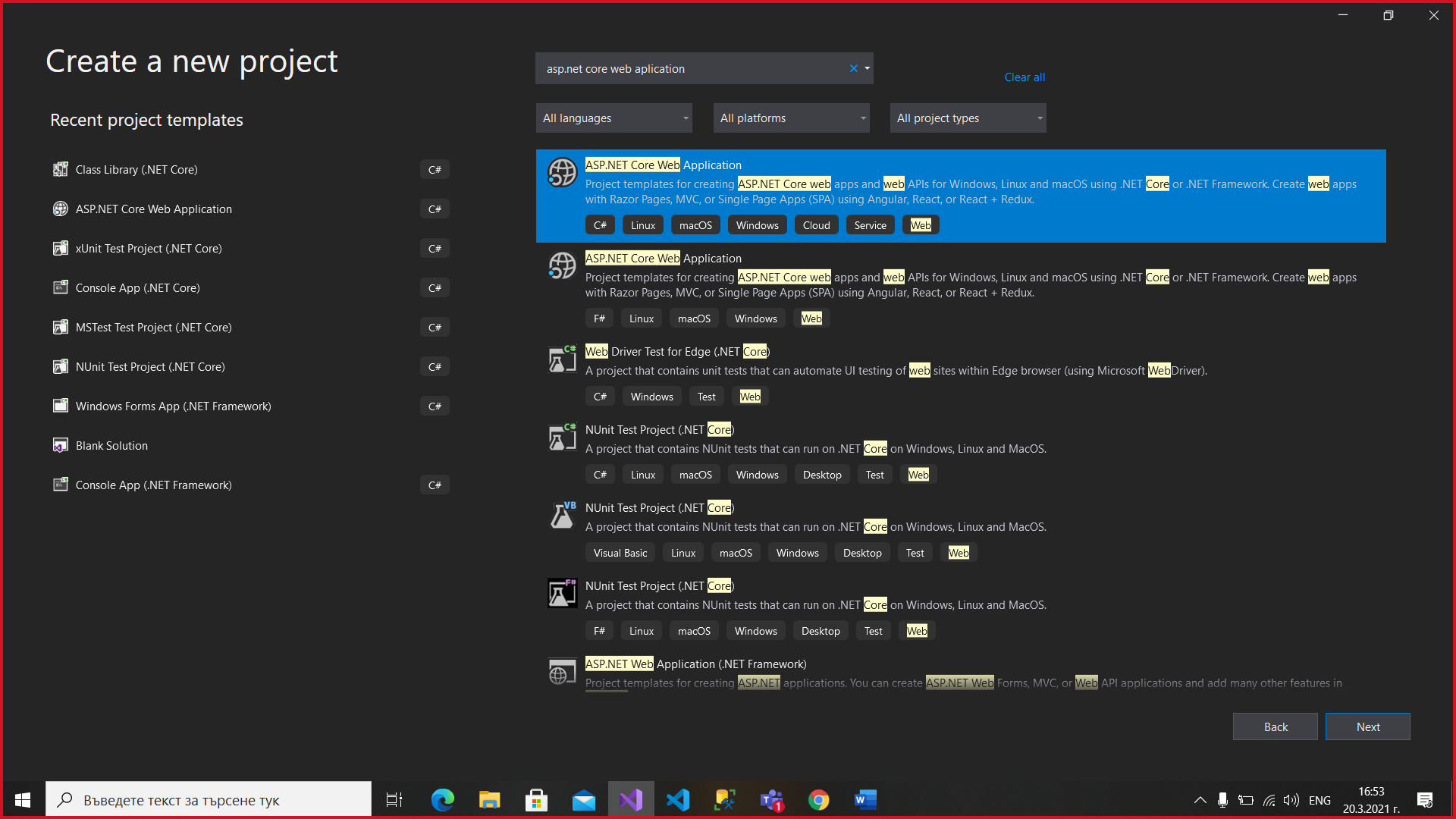
**Работа с GitHub7**

1. **Увод**

Създаването на уеб приложение в интегрирана среда за програмиране Visual Studio се осъществява чрез платформата ASP.NET, използвайки езика C#. За поддържане на базата се използва SQL Management Studio Server, като двете програми са свързани посредством миграция. Екипната работа се извършва през GitHub.

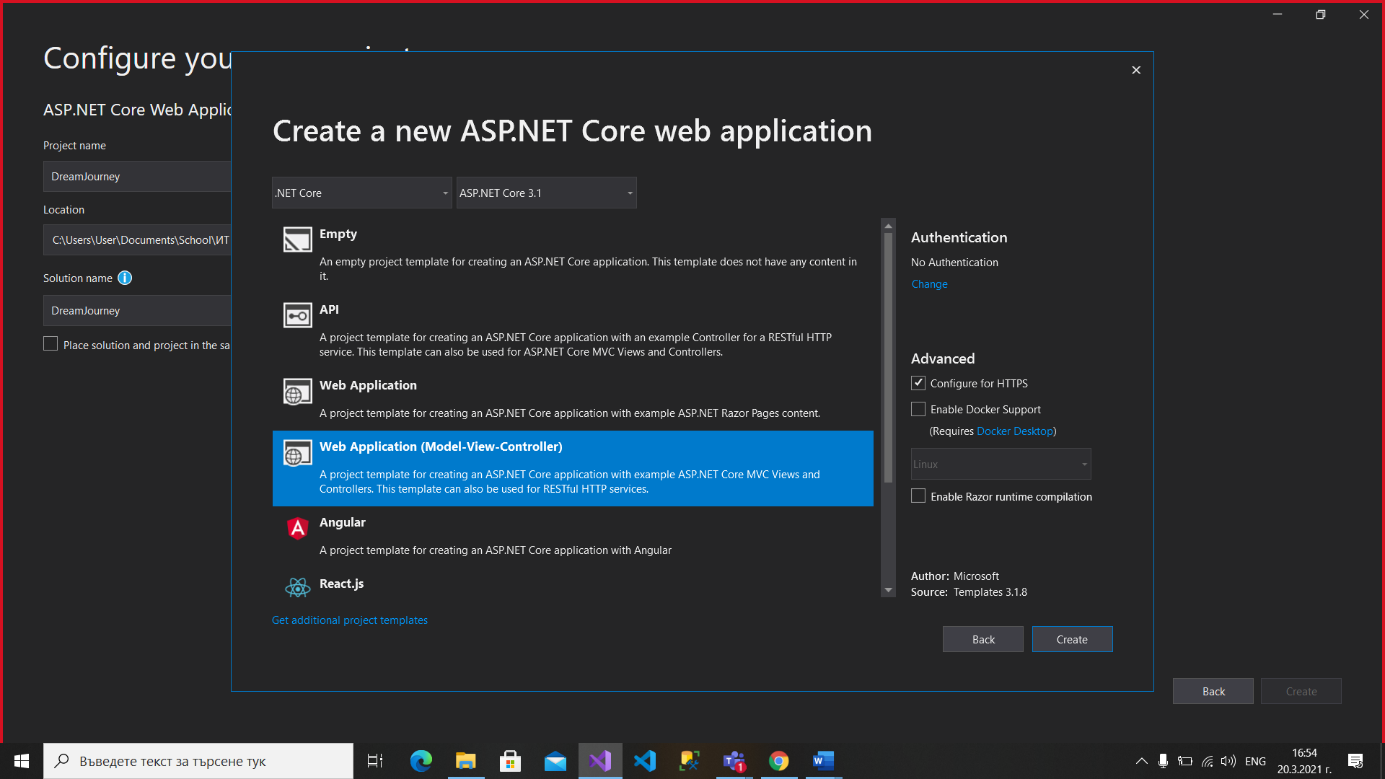
1. **Анализ на изискванията**

* Наличие и правилно структуриране на слой за данни, слой за услуги и потребителски интерфейси
* Наличие на подходящ интерфейс/и
* Използване на база данни
* Използване на Entity framework
* Покритие с компонентни тестове
* Спазване на добрите практики за писане на програмен код

1. **Процес на разработка**
   1. Създаване на проекта

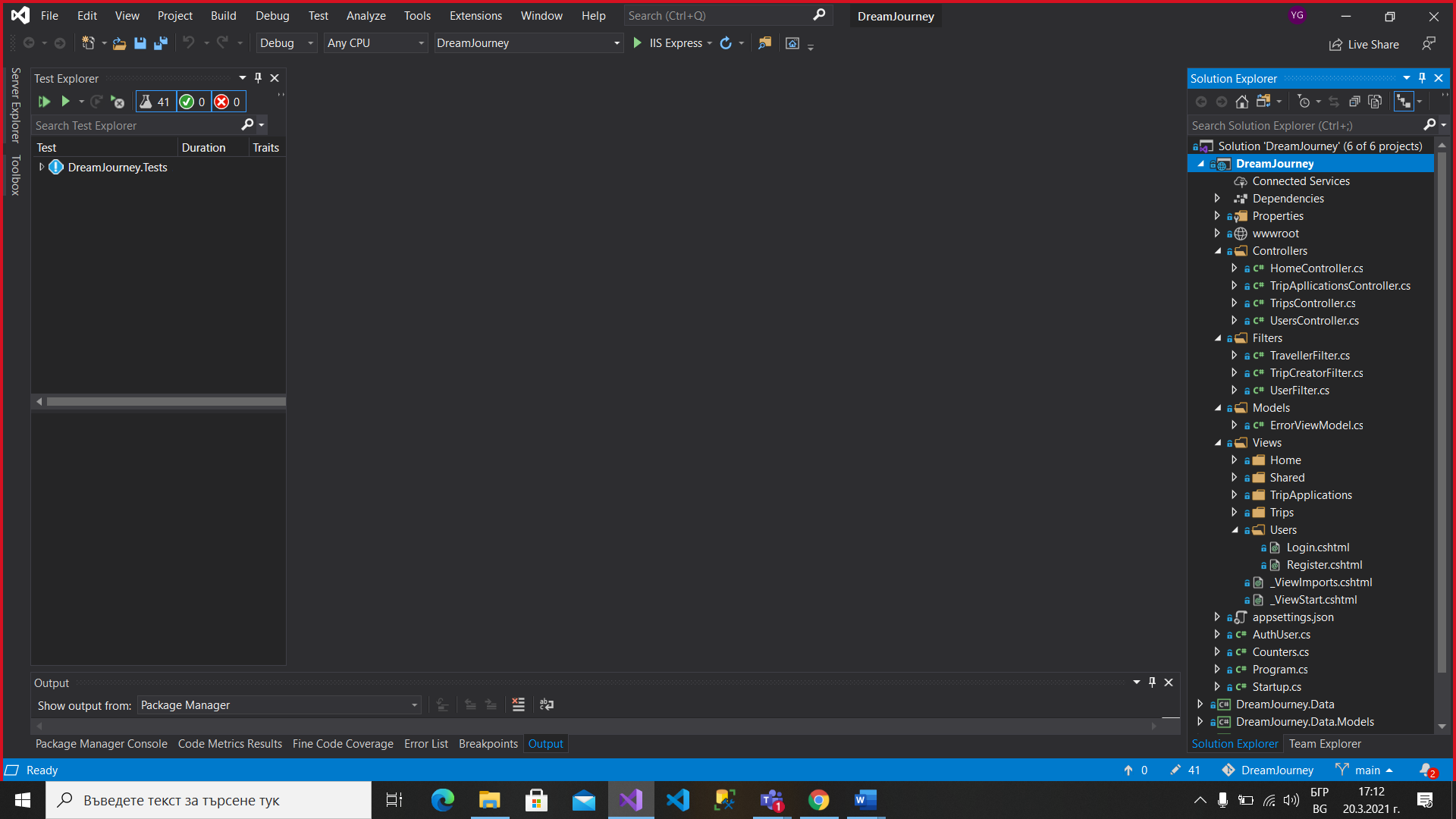
Създава се ASP.NET проект

Фигура 1



Избира се вида на проекта – MVC Model

* 1. Структура на проекта
* DreamJourney

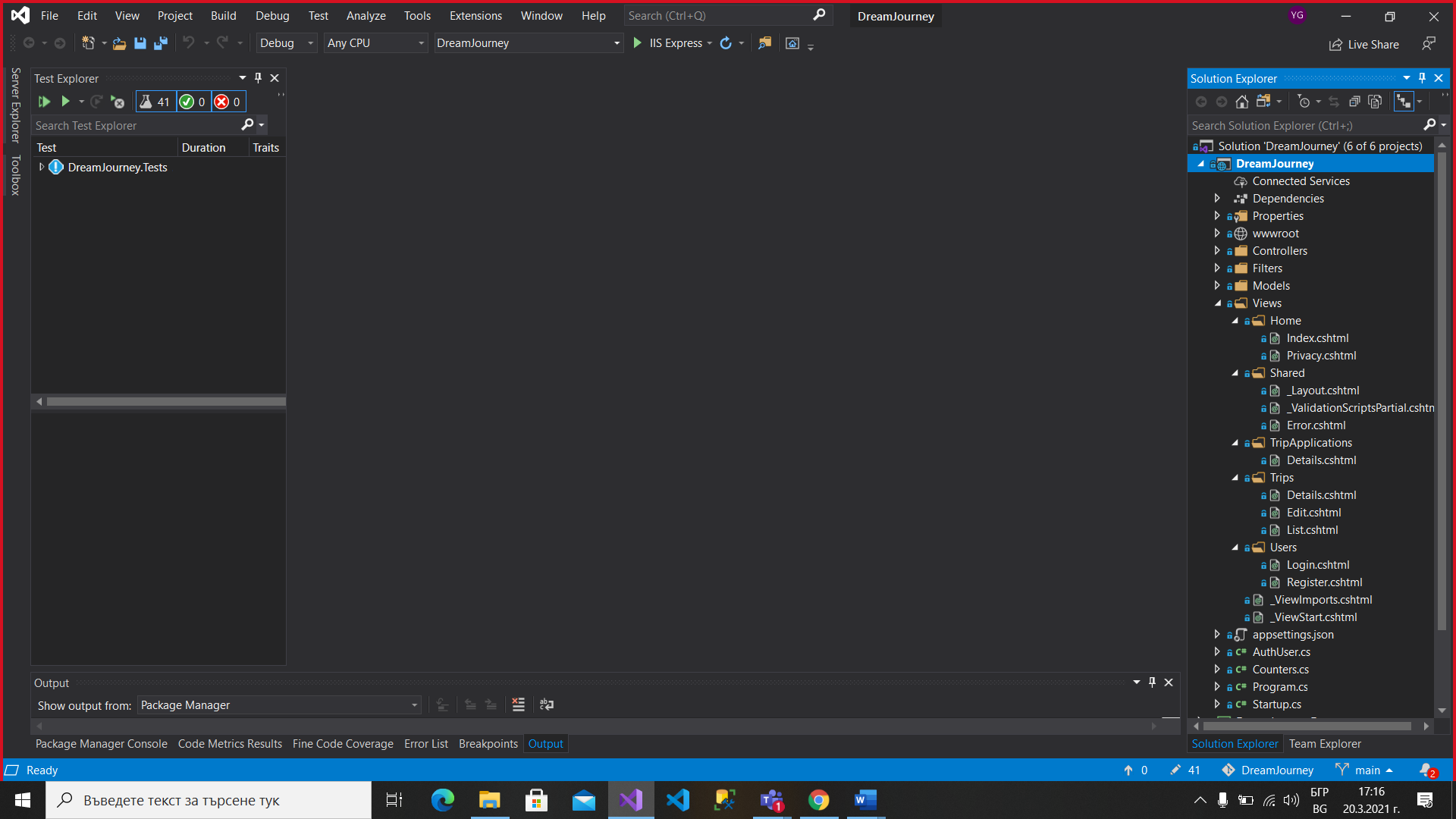


В папка Controllers се намират контролерите, които управляват логиката на приложението.

Класовете в папка Filters определят изгледа, в зависимост от ролята на потребителя, който използва приложението

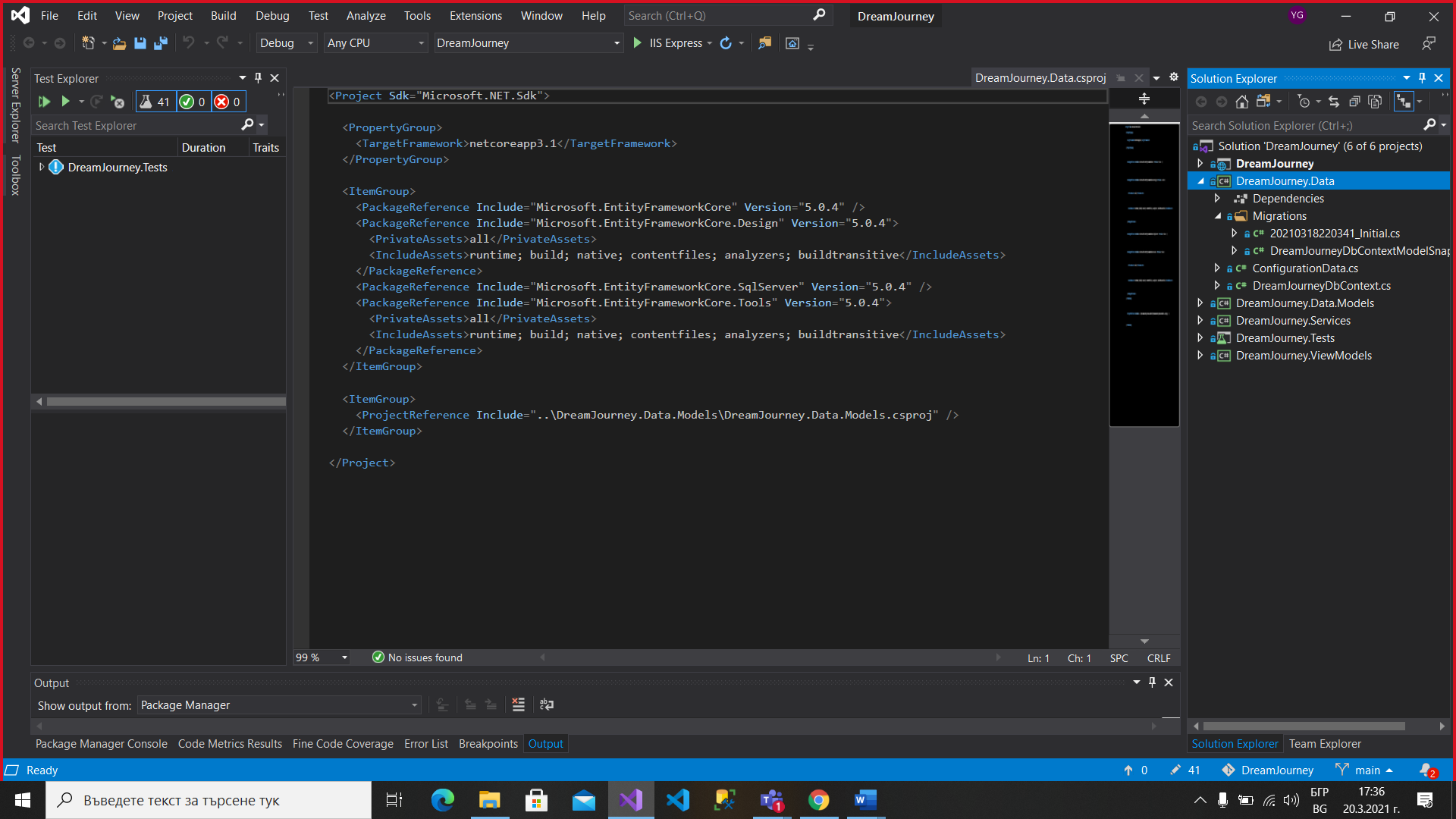
Index.cshtml – изгледа на началната страница

\_Layout.cshtml – горната и долната част, които се виждат непрекъснато

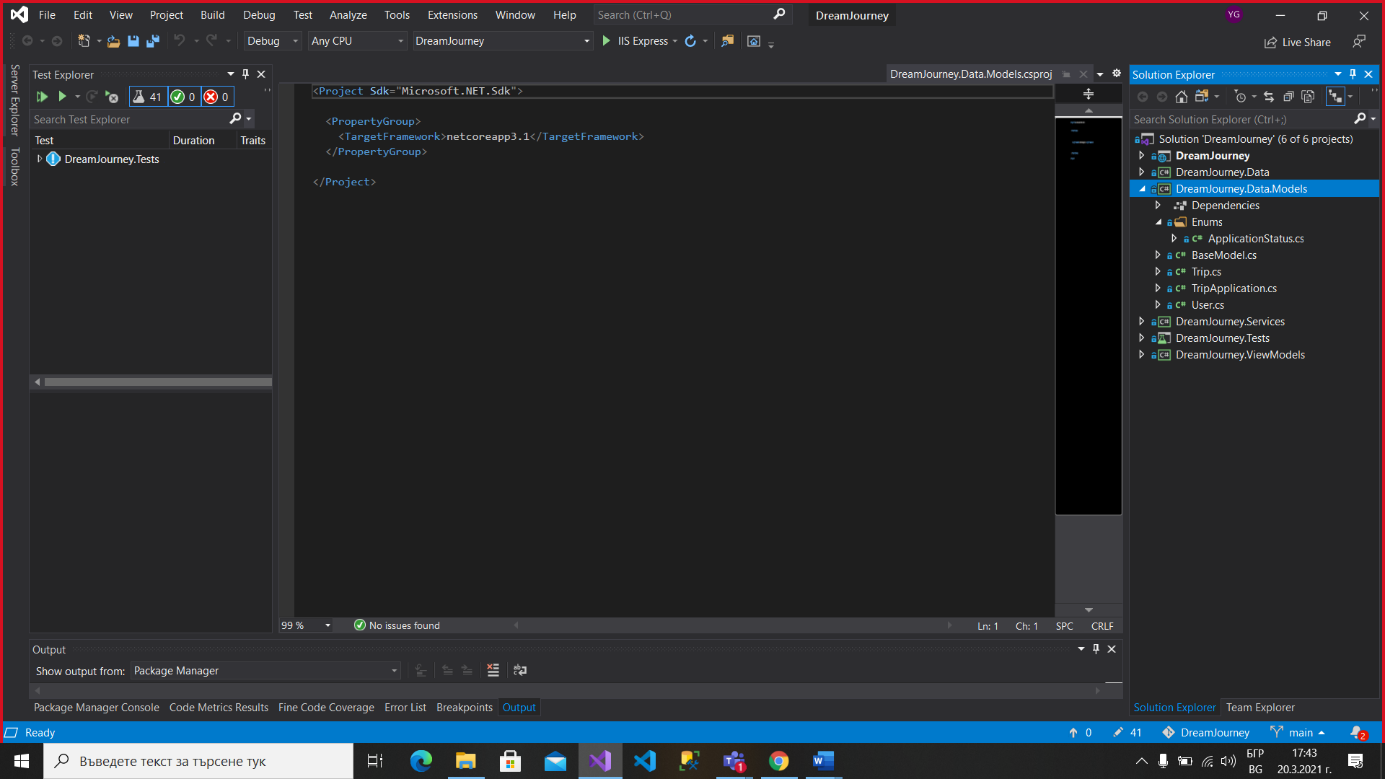


В папка Views се пазят .cshtml файловете, управляващи изгледа на приложението. Има различни папки с имената на моделите, които съхраняват различните изгледи, необходими за функционалностите на приложението.

* DreamJourney.Data

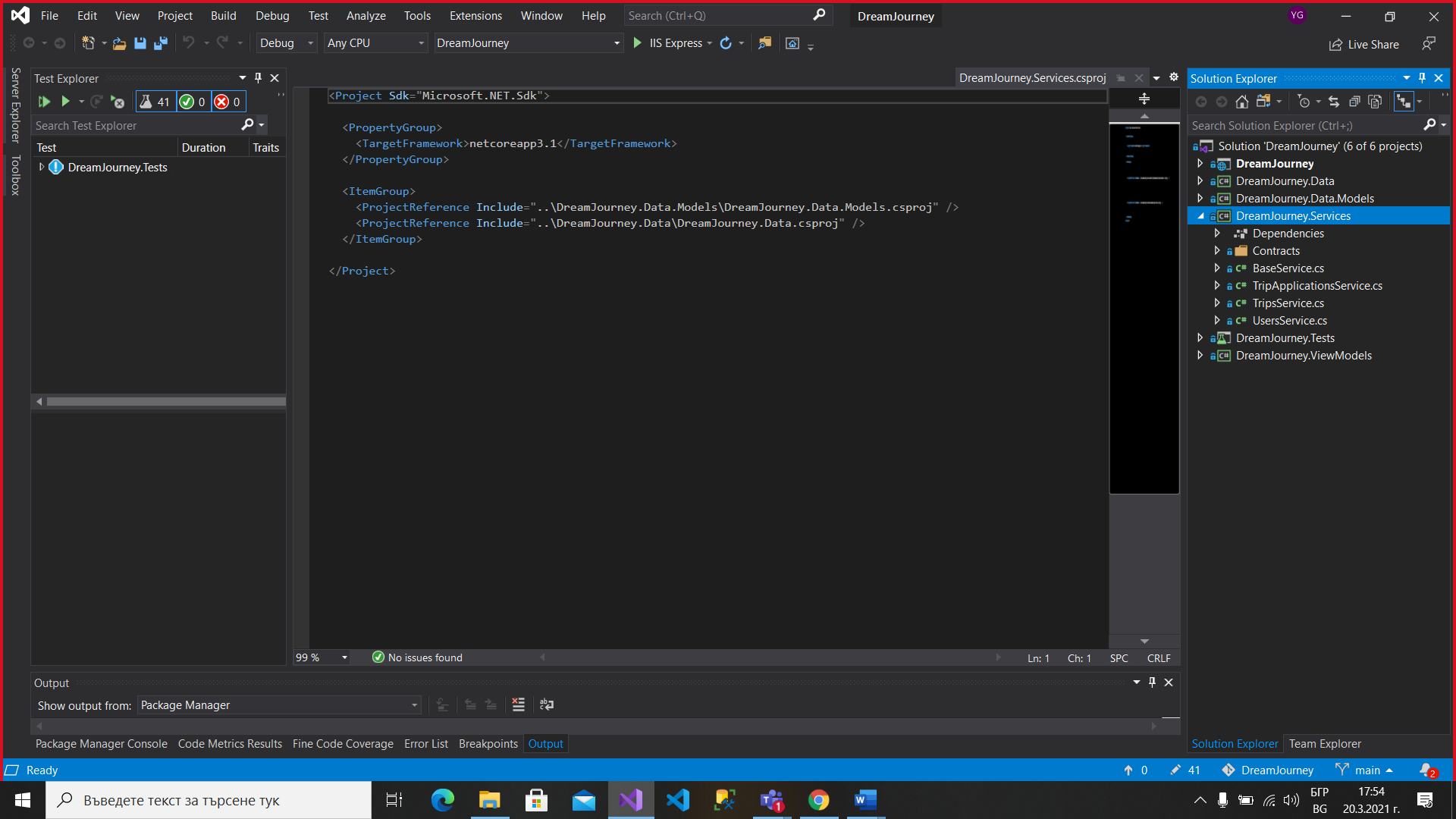
В този проект се намира класът ConfigurationData.cs, който съдържа connection string, миграцията, която осъществява връзката с базата данни, както и DbContex-a, който съдържа информацията за таблиците.

* DreamJourney.Data.Models

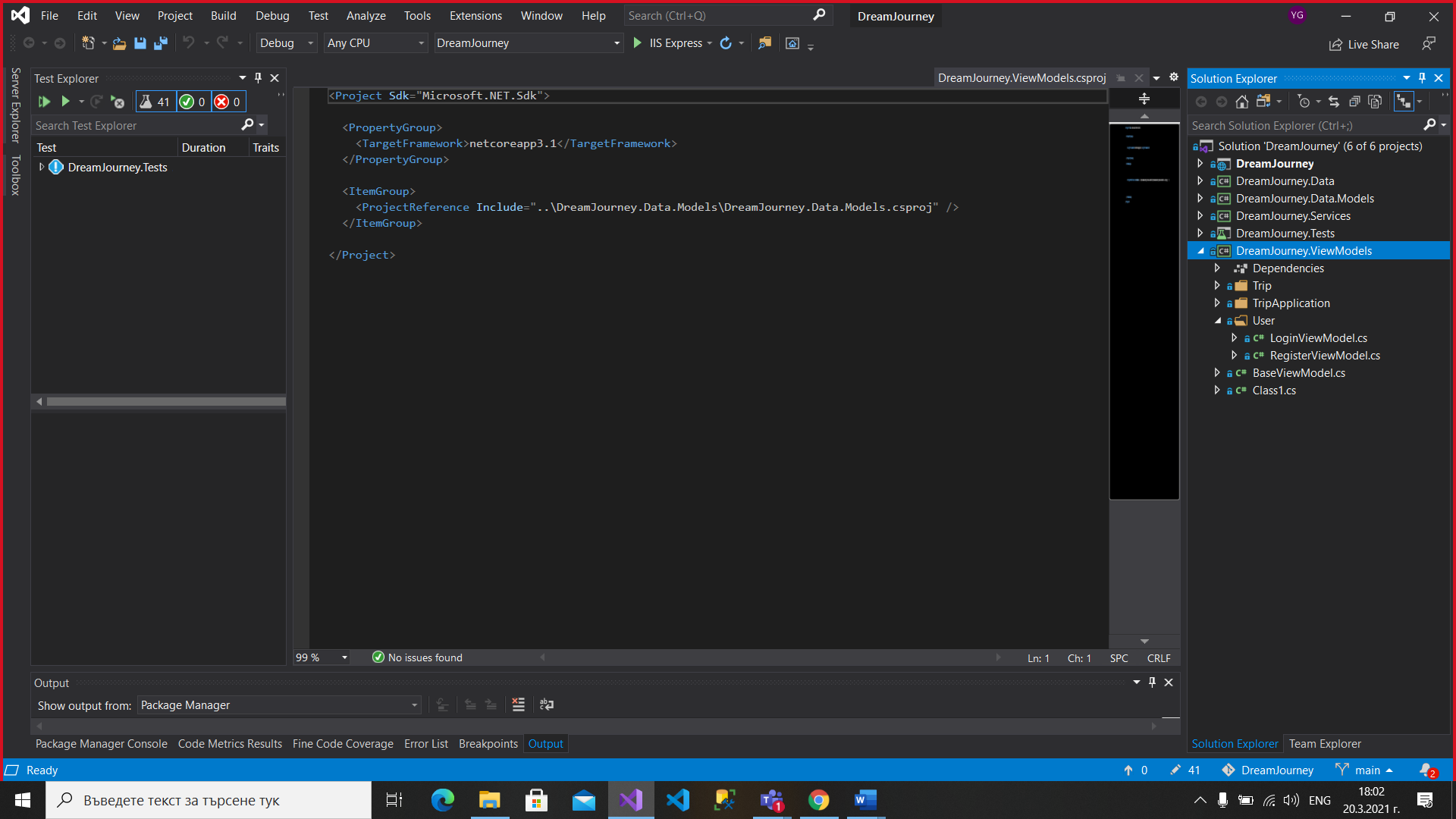


Проектът съхранява класовете, които описват полетата на съответните таблици от базата данни.

* DreamJourney.Services

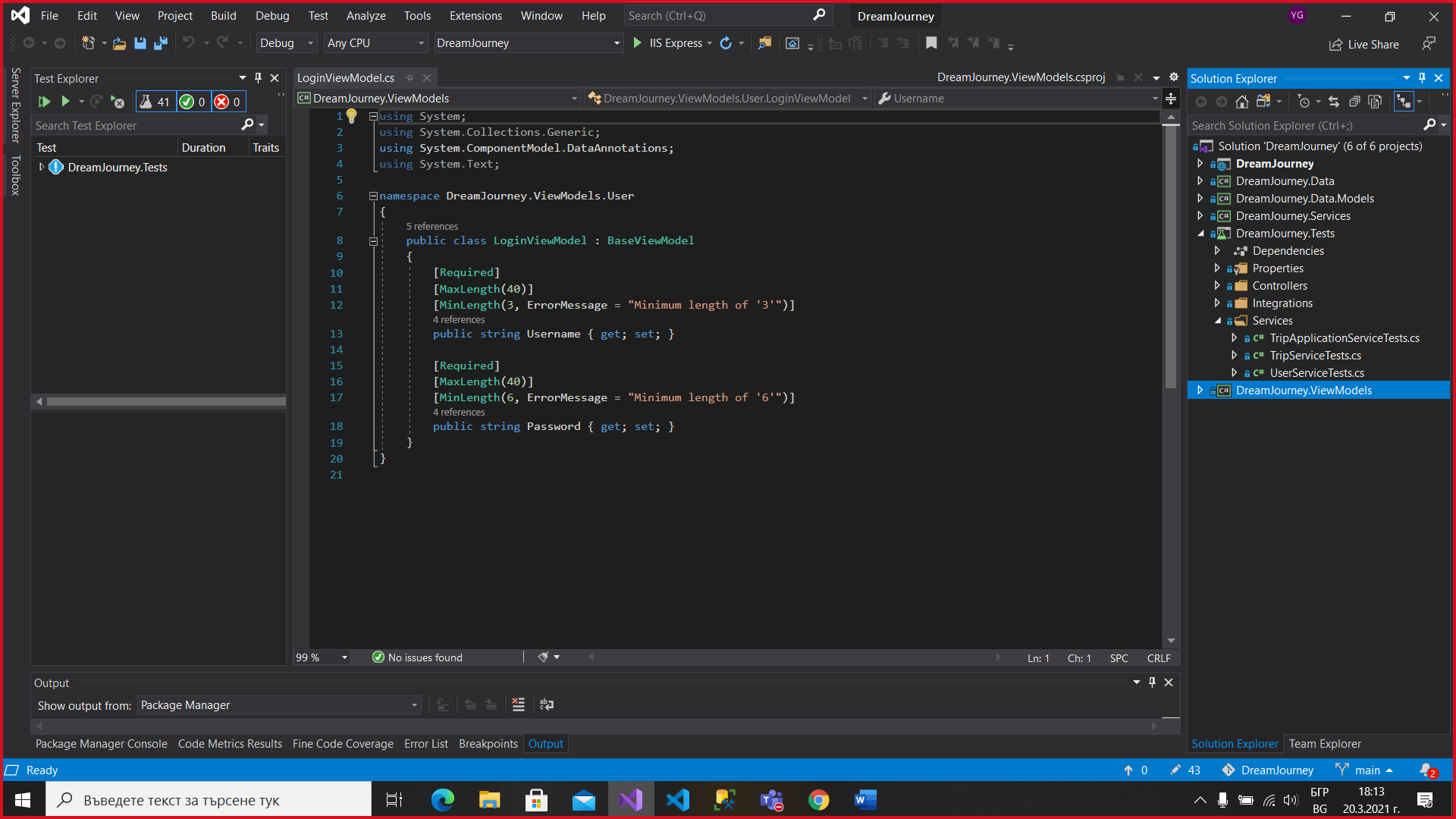


В този проект се съдържат моделите контролерите под формата на Services, а в папката Contracts - техните интерфейси.

* DreamJourney.ViewModels

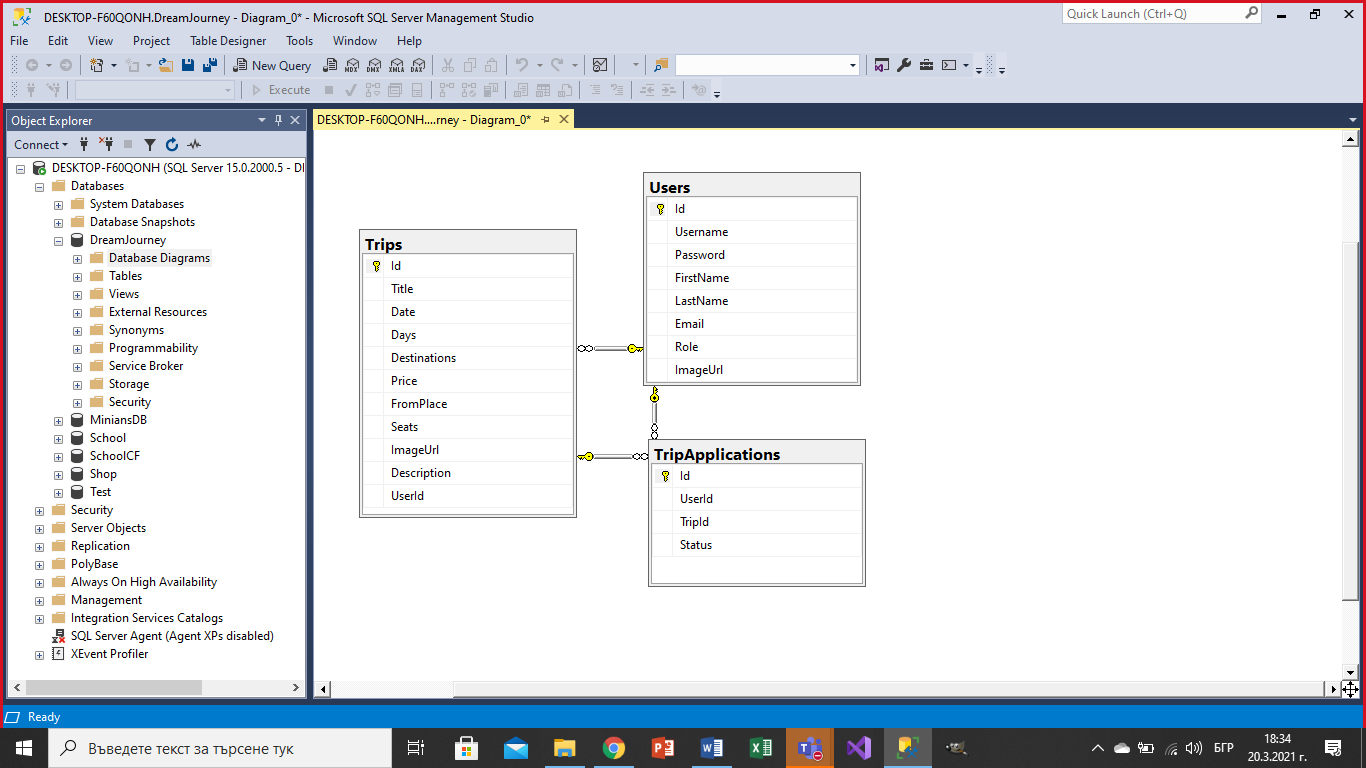
В проекта има папки с имената на моделите, съдържащи класове, в които под формата на полета се съхраняват нужните компоненти за всички изгледи, които може да извика контролера на съответния модел.

* DreamJourney.Tests



В този проект са написани тестовете на приложението, разделени по папки за сървиси, контролери и изгледи.

* Диаграма на базата данни



* 1. Разработка на проекта

1. Добавяне на 4 проекта .NET Core Class Library
2. Създаване на клас static Configuration в DreamJourney.Data
3. Добавяне на публичен клас DreamJourneyDbContext и наследяваме DbContext, пишем конструктор и пренаписваме два метода
4. Създаваме моделите в DreamJourney.Data.Models
5. Папка Enums за TripApplication в DreamJourney.Data.Models
6. Добавяме DbSet-ове в контекста и ги правим virtual (за unit tests)
7. В StartUp пишем services.AddDbContext
8. Правим миграция в DreamJourney.Data
9. В миграцията за създаване на TripApplication променяме OnDelete(Cascade => NoAction)
10. Създава се папка Contracts, в която има интерфейси на сървисите:

* IBaseService - типизиран
* IUsersService : IBaseService<User>
* ITripsService : IBaseService<Trip>
* ITripApplicationsService : IBaseService<TripApplocation>

1. Създават се сървисите:

* BaseService : IBaseService
* UsersService : IUsersService
* TripsService : ITripsService
* TripApplicationsService : I TripApplicationsService

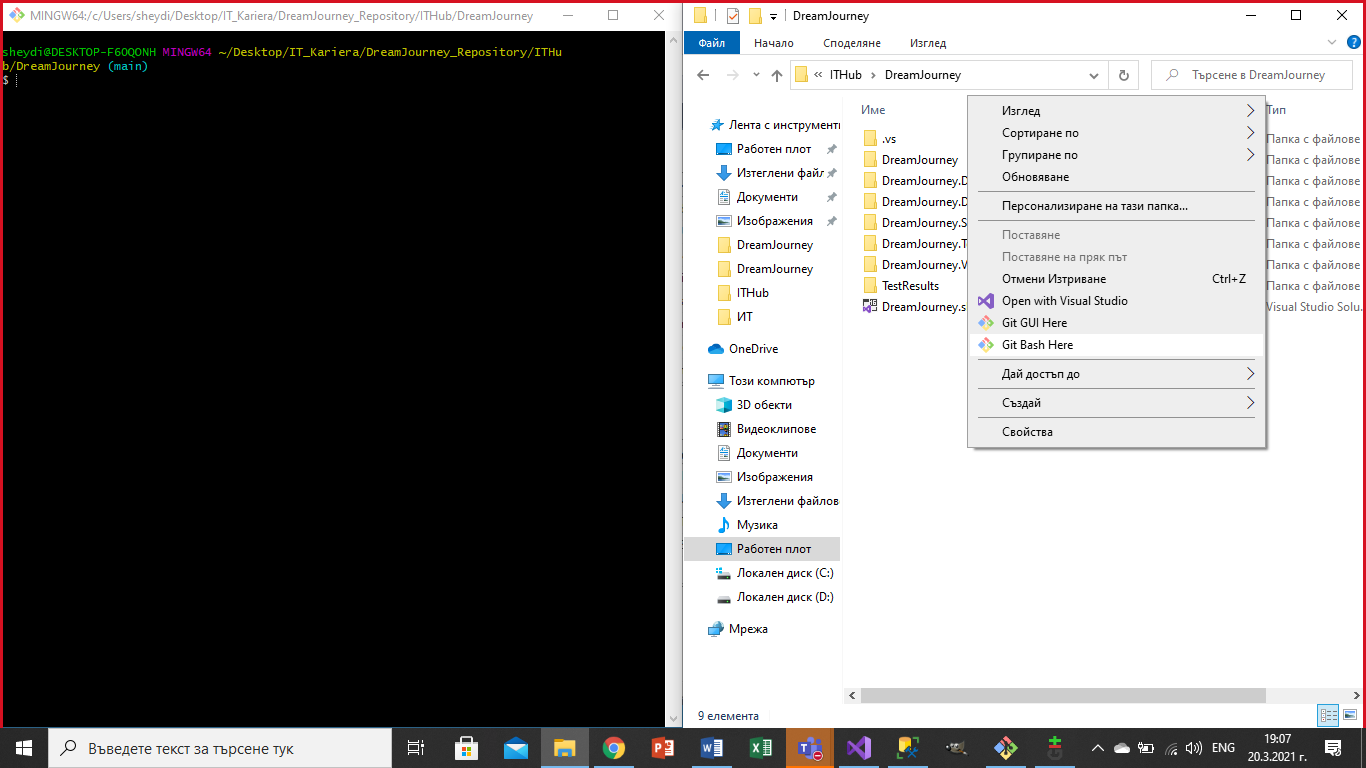
1. В StartUp добавяме services.AddScoped<Interface, Class> за всички сървиси
2. Добавяме папка Filters в DreamJourney, като в нея има класове – TripCreatorFilter, TravellerFilter, UserFilter
3. Създаваме клас AuthUser в DreamJourney
4. В StartUp добавяме:

* Services. AddSession();
* Services.AddScoper<TripCreatorFilter>

1. Създаваме ViewModels на User, Trip и TripApplications – пазят полетата нужни на Views
2. Създаване на контролерите в DreamJourney като се наследява Controller
3. Прави се клас Counter в DreamJourney, който има статично поле, пазещо броя на свободните места
4. Създаваме в папка Views от DreamJourney:

* Users – Login, Register (от ViewModels)
* Trips – Details, Edit, List (от ViewModels)
* TripApplications – Details (от ViewModels)

1. Променяме \_Layout.cshtml от Views/Shared
2. Променяме Index.cshtml от Views/Home и добавяме нужната снимка в DreamJourney/wwwroot
3. Създаваме проект DreamJourneyTests, където пишем тестовете, разпределени по папки
4. **Работа с GitHub**



Препращането на проекта се осъществява чрез Git Bash Here, като се използват команди.