Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

Raport

la Lucrarea de laborator nr. 4
la disciplina Tehnologii Web

Tema: Entity Framework.

Studentul gr. TI-173: Heghea Nicolae
Conducător: asistent universitar, Rusu Cristian

Chișinău 2019

Cuprins

1. Scopul :

Familiarizarea cu Entity Framework, și crearea unei baze de date. Continuarea laboratorului 3.

2. Scurtă teorie

Entity Framework a fost lansat pentru prima dată în 2008, principalul mijloc de interacțiune dintre aplicațiile .NET și bazele de date relaționale ale Microsoft. Entity Framework este un Object Relational Mapper (ORM) care este un tip de instrument care simplifică maparea între obiectele din software-ul dvs. și tabelele și coloanele unei baze de date relaționale.

- Entity Framework (EF) este un cadru open source ORM pentru ADO.NET, care face parte din .NET Framework.
- Un ORM are grijă să creeze conexiuni de bază de date și să execute comenzi, precum și să ia rezultatele interogării și să concretizeze automat aceste rezultate ca obiecte ale aplicației.
- Un ORM ajută, de asemenea, să urmăriți modificările aduse acestor obiecte și, atunci când este instruit, va persista și aceste modificări înapoi în baza de date pentru dvs.

O bază de date este o colecție de date care este stocată în conformitate cu schema de date proiectată. Definiția unei baze de date nu trebuie confundată cu definirea unui sistem de gestionare a bazelor de date (SGBD).

SGBD – un set de instrumente software utilizate pentru crearea și gestionarea unei baze de date. Acest laborator utilizează unul dintre sistemele de gestionare a bazelor de date client-server – Microsoft SQL Server, dezvoltat de Microsoft.

3. Realizare

Pentru a vă conecta la proiect, *Entity Framework* trebuie să îl adăugați la Nuget. Pachetele se adaugă pentru bibliotecile de clasă *BusinessLogic* și *Web*. Pachetele Nuget sunt prezentate în Figura 1 (a, b) .

Conținutul proiectului :

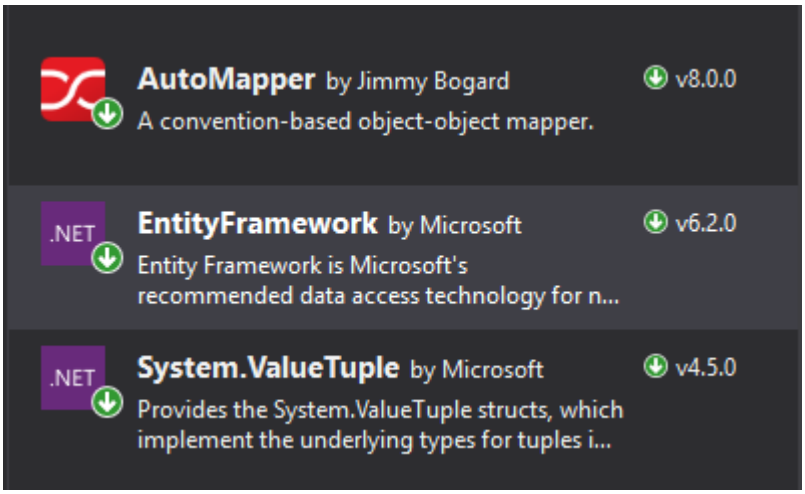


Figura 1 Pachetul *Entity Framework*

Următorul pas este crearea directorului DbModel în proiectul BusinessLogic menționat anterior, care va conține apoi clasa *UserContext*, moștenită de la clasa DbContext. Această clasă conține numele unei baze de date viitoare și cu ajutorul acesteia este accesată baza de date.

Conținutul clasei *UserContext* este :

```
using System.Data.Entity;
using TW.Hejea.Domain.Entities.User;

namespace TW.Hejea.BusinessLogic.DBModel
{
    class UserContext : DbContext
    {
        public UserContext() : base("name=TW.Hejea") // connectionstring name define in your
web.config
        {
        }

        public virtual DbSet<UDbTable> Users { get; set; }
    }
}
```

Apoi, utilizați caracteristica principală Entity Framework. Pentru aceasta, este creată o clasă UDbTable care va stoca câmpurile pe care le are fiecare utilizator al sistemului.

Conținutul fișierului *UDbTable.cs*

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using TW.Hejea.Domain.Enums;

namespace TW.Hejea.Domain.Entities.User
{
    public class UDbTable
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Id { get; set; }

        [Required]
        [Display(Name = "Username")]
        [StringLength(30, MinimumLength = 5, ErrorMessage = "Username cannot be longer than 30
characters.")]
        public string Username { get; set; }

        [Required]
        [Display(Name = "Password")]
        [StringLength(50, MinimumLength = 8, ErrorMessage = "Password cannot be shorter than 8
characters.")]
        public string Password { get; set; }

        [Required]
        [Display(Name = "Email Address")]
        [StringLength(30)]
        public string Email { get; set; }

        [DataType(DataType.Date)]
        public DateTime LastLogin { get; set; }

        [StringLength(30)]
        public string LasIp { get; set; }

        public URole Level { get; set; }
    }
}
```

Aceste câmpuri vor avea tabelul *UDbTable* din baza de date. După cum puteți vedea, adnotările C# sunt folosite pentru a descrie proprietățile acestor câmpuri.

Adnotarea *[Key]* indică faptul că acest câmp va fi cheia primară a tabelului proiectat în baza de date.

Adnotarea *[Required]*, la rândul său, indică faptul că acest câmp este necesar. În acest caz, numele de utilizator, parola și adresa de e-mail.

Adnotarea *[StringLength]*, care vă permite să specificați șirul maxim și minim pentru a scrie în câmpul în cauză.

După efectuarea modificărilor necesare la proiect, este necesar să se instaleze un SGBD în care va fi creată baza de date. Acest laborator utilizează SGBD-ul MS SQL Server.

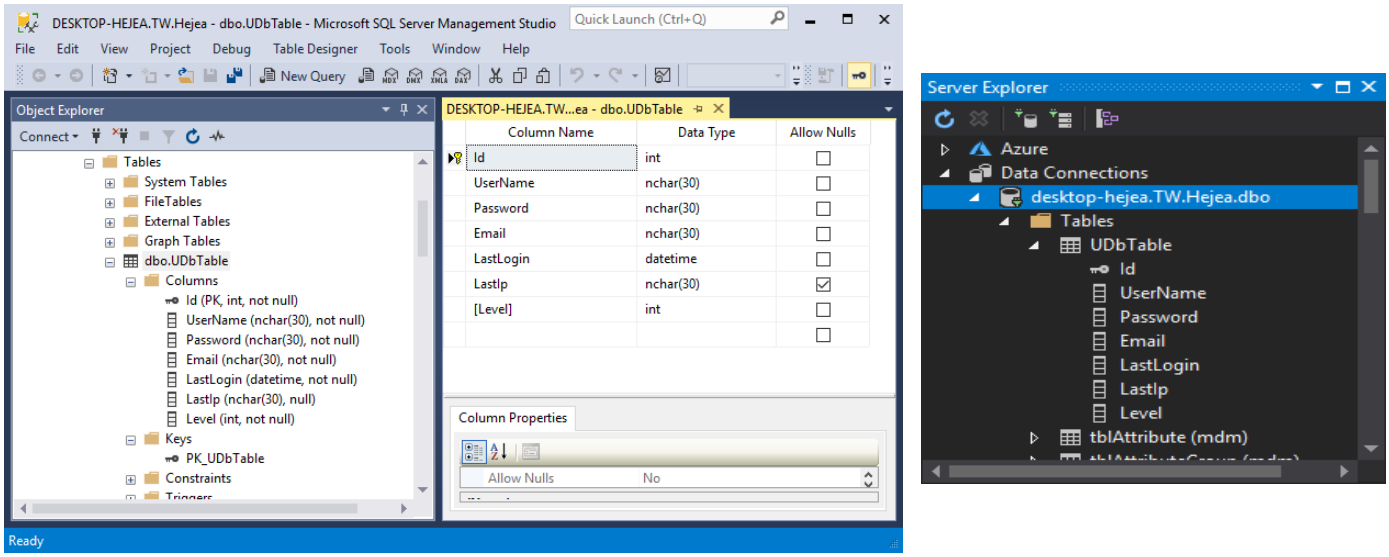


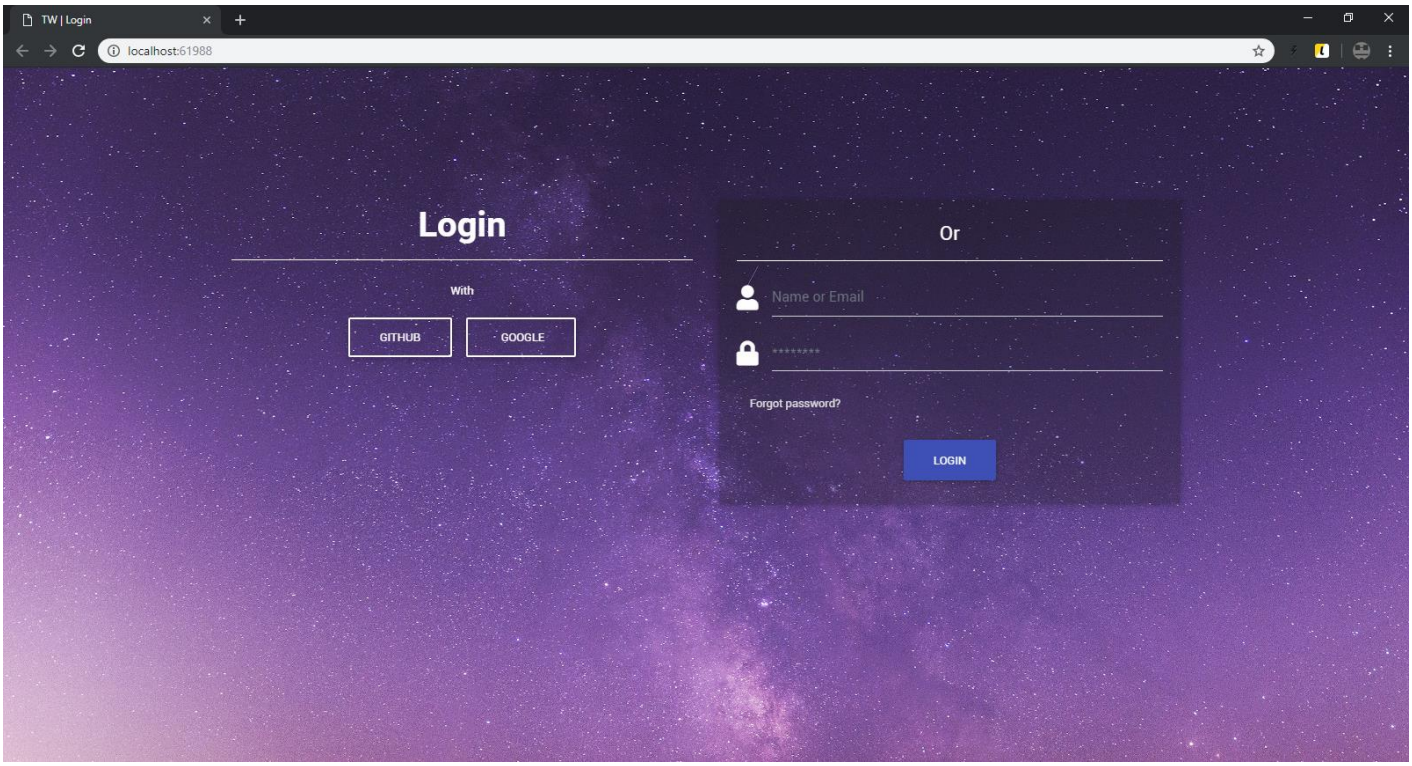
Figura 2 Tabelul *UdbTable* din baza de date creat în MS SQL Server

În *Web.config* am adăugat :

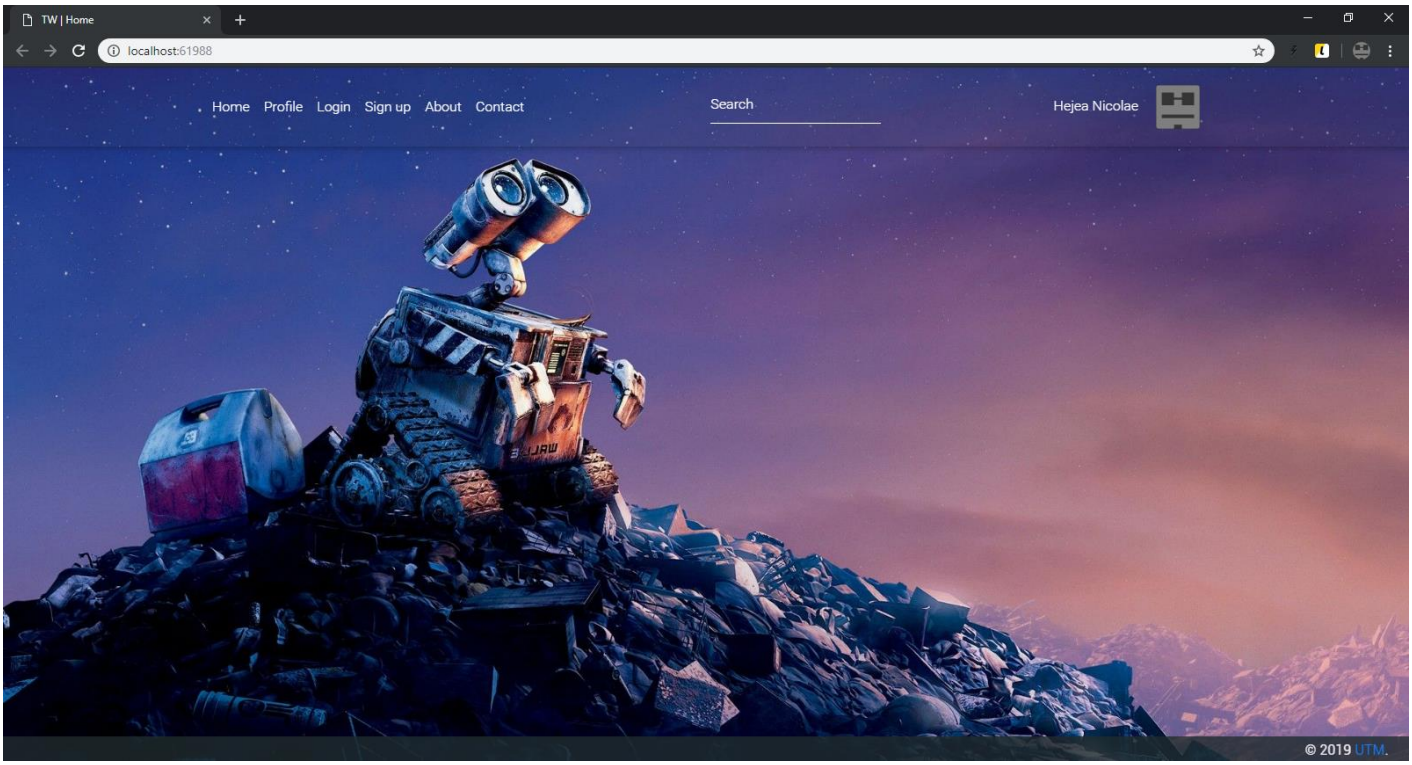
```
<connectionStrings>
    <remove name="LocalSqlServer" />
    <add name="TW.Hejea"
         connectionString="Data Source=DESKTOP-HEJEA;
         Initial Catalog=TW.Hejea;
         Integrated Security=True;
         MultipleActiveResultSets=True;
         App=EntityFramework"
         providerName="System.Data.SqlClient" />
  </connectionStrings>
```

4. Rezultat

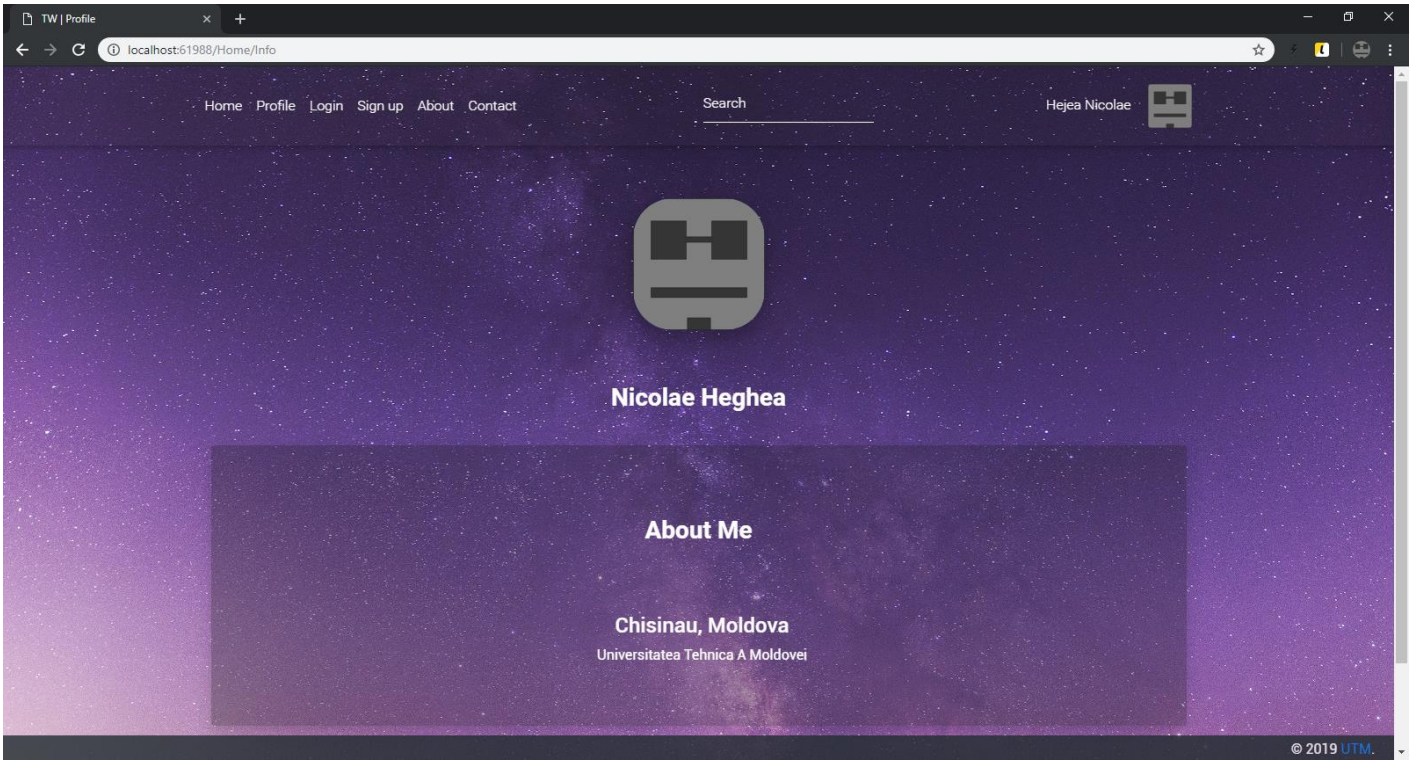
Pagina *login* :



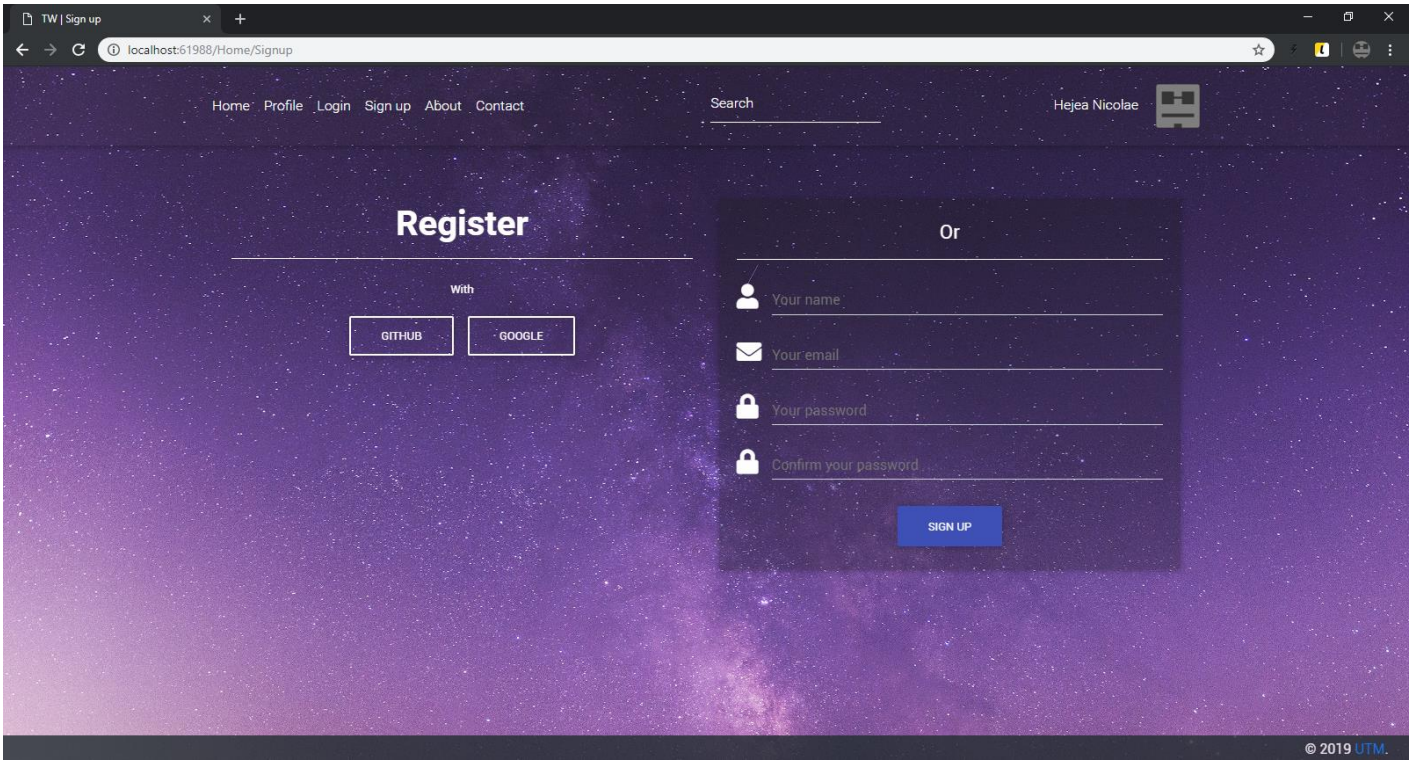
Pagina *home* :



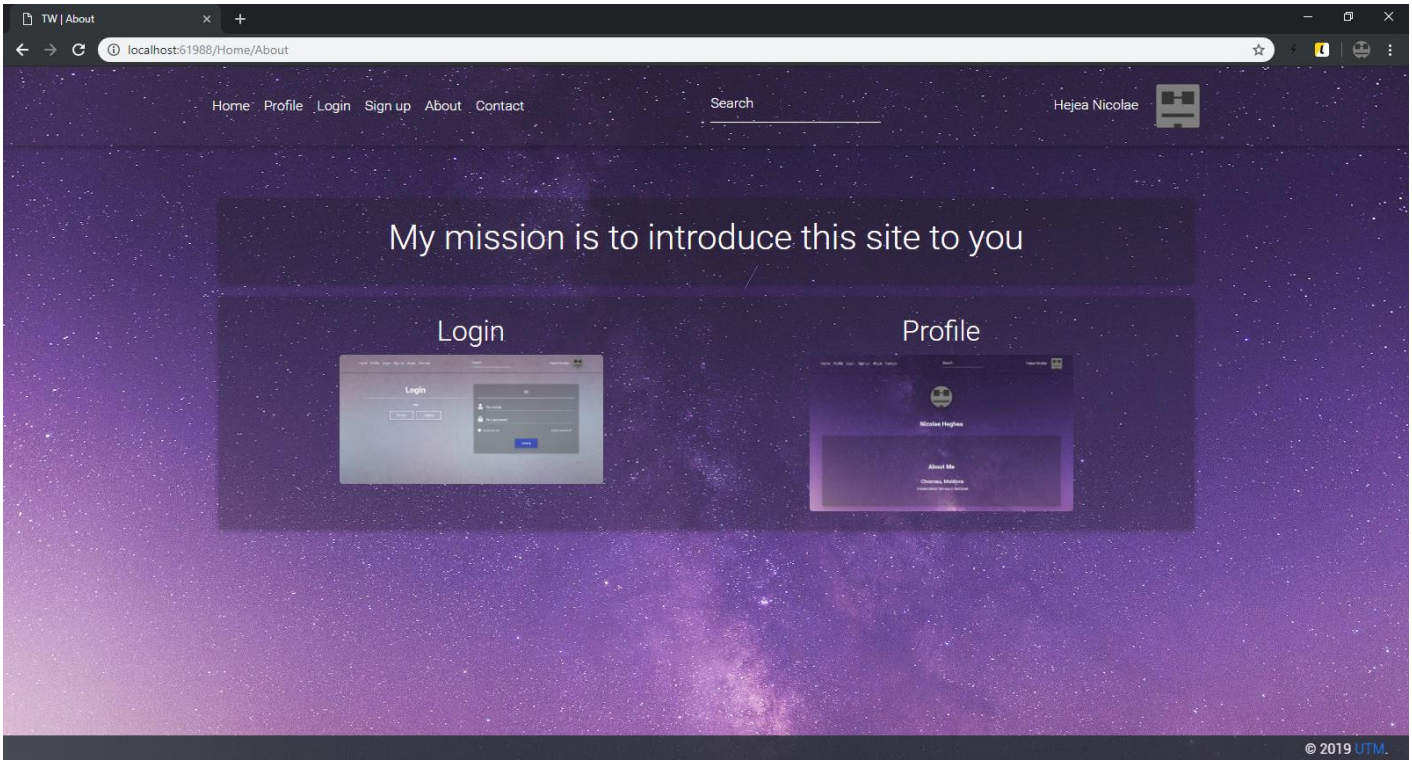
Pagina *profile* :



Pagina *sign up* :



Pagina *about* :



Pagina *contact* :