

Ministerul Educației, Culturii și Cercetării al Republicii
Moldova
Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

RAPORT

Disciplina: Tehnologii Web
Lucrarea de laborator nr. 2

A elaborat:
st. gr. TI-215/TI-216

*Musteța Mihail
Pistrui Eutalia
Serbulenco Vitalie
Moscvin Vlad*

A verificat:
asit.univer,

A. Gaidarji

Chișinău – 2023

Scopul lucrării: Utilizarea ASP.NET pentru proiectare Web Site-urilor.

Sarcina de lucru:În Visual Studio, utilizând platforma ASP.NET să se includă conținutul paginii Web

Noțiuni teoretice despre ASP.NET

ASP.NET este un framework de dezvoltare web dezvoltat de Microsoft pentru a crea aplicații web dinamice, scalabile și robuste. Iată câteva noțiuni teoretice despre ASP.NET:

- **Arhitectură:** ASP.NET este construit pe baza arhitecturii Model-View-Controller (MVC) și/sau Model-View-ViewModel (MVVM). Această arhitectură ajută la separarea codului de prezentare și a logicii de afaceri, ceea ce face aplicațiile mai ușor de dezvoltat și întreținut.
- **Limbaje de programare:** ASP.NET oferă suport pentru o varietate de limbaje de programare, cum ar fi C#, Visual Basic și F#. De asemenea, permite dezvoltatorilor să folosească limbaje de marcare, cum ar fi HTML, CSS și JavaScript.
- **Biblioteci de cod:** ASP.NET vine cu o serie de biblioteci de cod predefinite, care ajută la dezvoltarea rapidă a aplicațiilor. Aceste biblioteci includ biblioteca de clase de bază .NET (.NET Framework sau .NET Core), biblioteca de cod Entity Framework pentru lucru cu baze de date și multe altele.
- **Autentificare și autorizare:** ASP.NET oferă suport pentru autentificarea și autorizarea utilizatorilor. Aceasta poate fi realizată prin intermediul diverselor metode, cum ar fi autentificarea bazată pe formulare sau autentificarea bazată pe token-uri.
- **Securitate:** Securitatea este un aspect important în dezvoltarea aplicațiilor web. ASP.NET oferă suport pentru diverse mecanisme de securitate, cum ar fi criptarea și semnătura digitală a datelor.

Utilizarea ASP.NET: *Account controller*

În ASP.NET, un controler de cont este de obicei responsabil pentru gestionarea conturilor de utilizator și autentificarea. Aceasta include sarcini precum înregistrarea de noi utilizatori, conectarea și deconectarea, resetarea parolelor și gestionarea profilurilor de utilizator.

Controlerul de cont este adesea o parte a cadrului ASP.NET Identity, care oferă un set de API-uri și componente pentru gestionarea autentificării și autorizării utilizatorilor. Cadrul Identity este conceput pentru a funcționa cu diverși furnizori de stocare, cum ar fi SQL Server, pentru a stoca informații despre utilizator.

Controlorul de cont utilizează de obicei diferite metode de acțiune pentru a efectua diferite sarcini. De exemplu, metoda Register poate fi utilizată pentru a gestiona noile înregistrări de utilizatori, în timp ce metoda Login poate fi utilizată pentru a gestiona autentificarea utilizatorilor. Aceste metode sunt de obicei decorate cu atribute, cum ar fi atributul [HttpPost], pentru a specifica metoda de solicitare HTTP folosită pentru a le invoca.

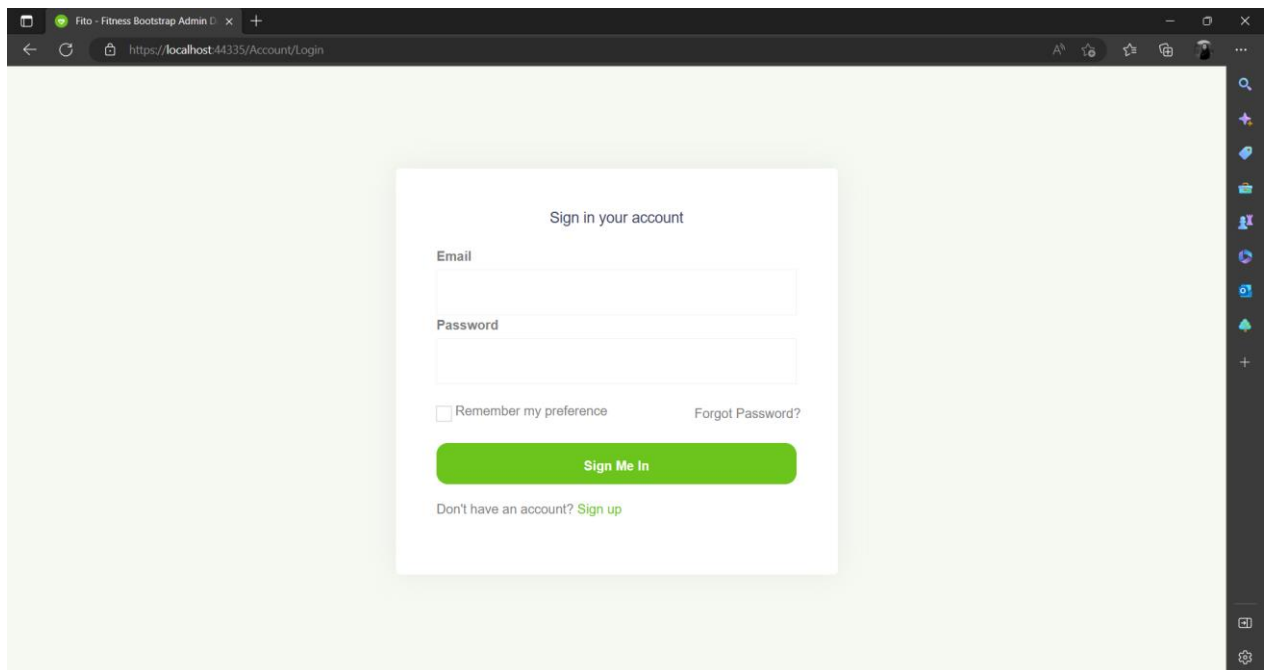


Figura 1 – Pagina de Log in

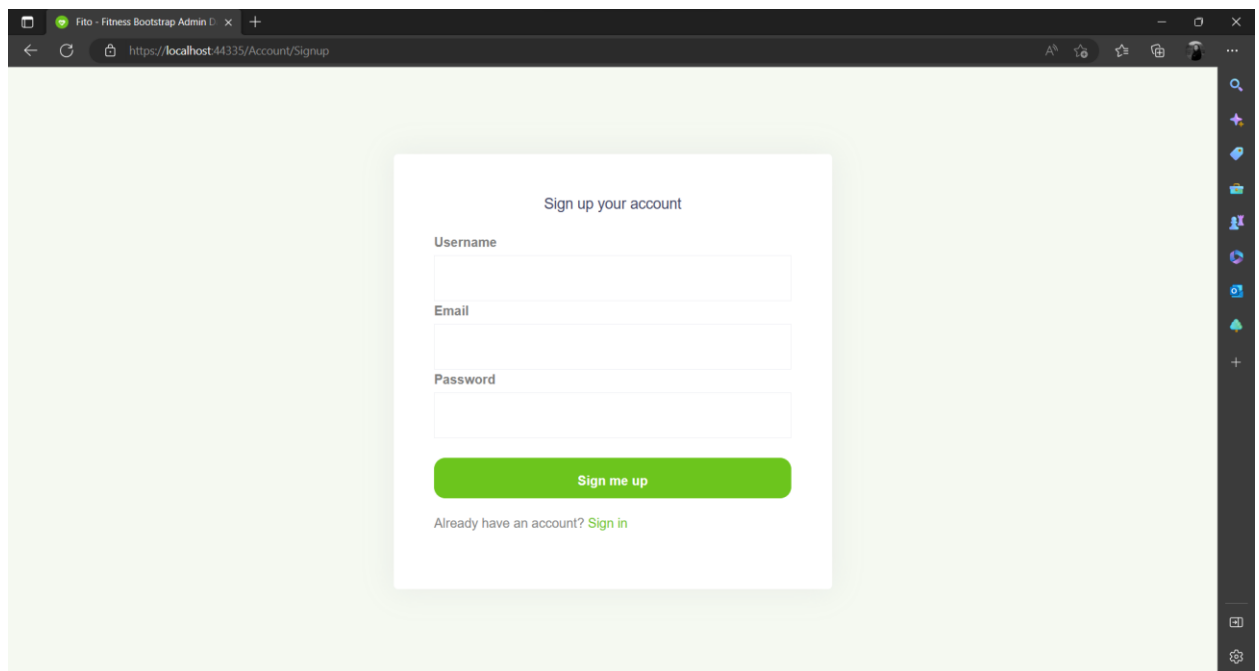


Figura 2 – Pagina de Sing Up

Codul sursă din proiectul nostru, folosind ASP.NET pentru account controller:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Helpers;
using System.Web.Mvc;
using System.Web.Security;
using FitnessServices.web.Models;

namespace FitnessServices.web.Controllers
{
    public class AccountController : Controller
    {
        FitnessServices_DBEntities db = new FitnessServices_DBEntities();
        // GET: Account
        public ActionResult Login()
        {
            return View();
        }
        public ActionResult Signup()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Login(UsersTable userinfo)
        {
            bool userExist = db.UsersTables.Any(user => user.Email == userinfo.Email
            && user.Password == userinfo.Password);
            UsersTable u = db.UsersTables.FirstOrDefault(user => user.Email ==
            userinfo.Email && user.Password == userinfo.Password);

            if (userExist)
            {
                FormsAuthentication.SetAuthCookie(u.Username, false);
                return RedirectToAction("Index", "Home");
            }

            ModelState.AddModelError("", "Invalid Email or Password");

            return View();
        }
        [HttpPost]
        public ActionResult Signup(UsersTable userinfo)
        {
            //userinfo.Password = Crypto.Hash(userinfo.Password, "SHA256");
            db.UsersTables.Add(userinfo);
            db.SaveChanges();
            return RedirectToAction("Login");
        }
    }
}
```

```

    }
    public ActionResult Signout()
    {
        FormsAuthentication.SignOut();
        return RedirectToAction("Login");
    }
}
}

```

Utilizarea ASP.NET: *Home controller*

Un controler Home în ASP.NET este o clasă care este responsabilă pentru gestionarea solicitărilor HTTP care sunt legate de pagina de pornire a unui site web sau a unei aplicații web. Controlerul Home conține de obicei metode care corespund diferitelor acțiuni care pot fi efectuate pe pagina de pornire, cum ar fi afișarea paginii de pornire, procesarea unei trimiteri a unui formular sau gestionarea autentificării unui utilizator.

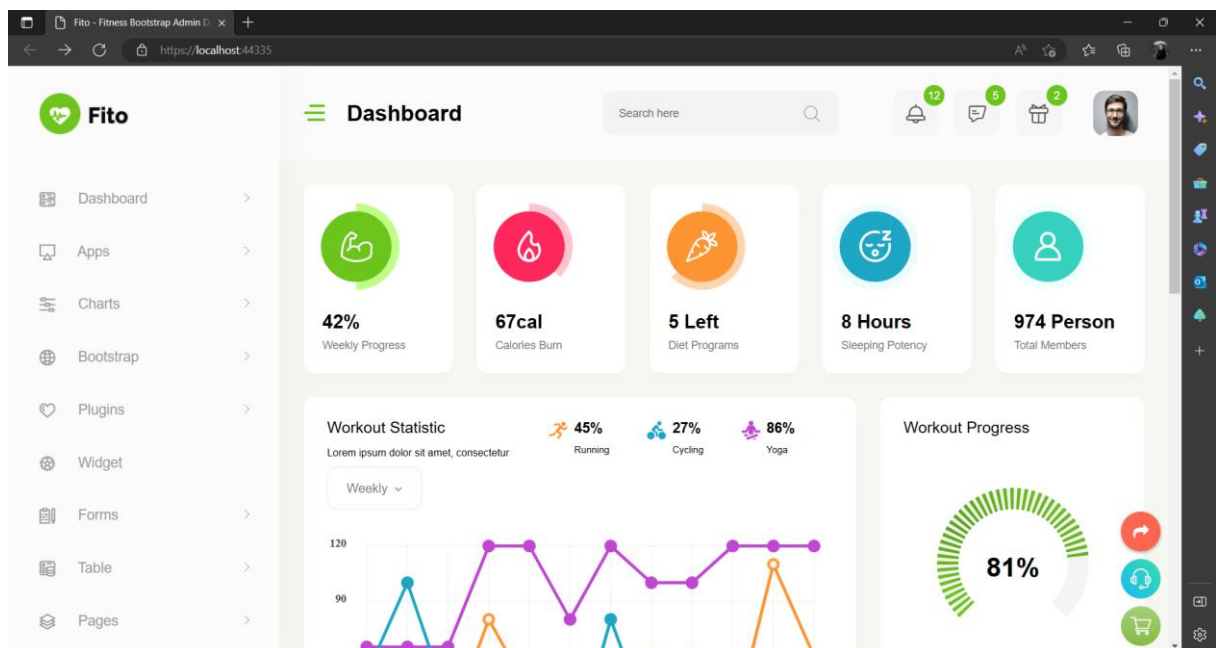


Figura 3 – Pagina Home

Codul sursă din proiectul nostru, folosind ASP.NET pentru home controller:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace FitnessServices.web.Controllers

```

```
{  
    public class HomeController : Controller  
    {  
        // GET: Home  
        public ActionResult Index()  
        {  
            return View();  
        }  
    }  
}
```

Concluzii:

În concluzie putem să spunem că ASP.NET este un cadru de aplicații web popular care oferă o gamă largă de caracteristici pentru construirea de aplicații web scalabile și sigure. Este utilizat pe scară largă de dezvoltatori pentru a crea aplicații web dinamice, inclusiv site-uri de comerț electronic, site-uri de rețele sociale și aplicații la nivel de întreprindere. Unele dintre avantajele cheie ale utilizării ASP.NET includ caracteristicile sale de înaltă performanță, scalabilitate și securitate. Cadrul este optimizat pentru a rula pe platforma Windows și are o gamă largă de instrumente și biblioteci care facilitează dezvoltarea aplicațiilor web complexe. În plus, ASP.NET oferă o serie de beneficii pentru dezvoltatori, inclusiv o comunitate mare și activă, documentație și asistență extinsă și o gamă de instrumente și pluginuri terțe care pot fi utilizate pentru a-și extinde funcționalitatea.

Concluzii:

Activitatea noastră actuală reprezintă un pas important în această direcție amintit cum se utilizează documentele **html**, **css** și **js** pentru a face pagini web front-end-ul în special a fost creat cu ajutorul unei machete potrivită pentru tema **Prestarea serviciilor fitness** însă nici una nu mi-a satisfăcut cerințele care sunt următoarele: să utilizeze bootstrap 5, să fie gratis, să aibă un design plăcut și curat fără o mulțime de elemente inutile care ar distra atenția de la produsele oferite pe pagina principală. astfel am economisit mai puțin timp pentru aspect, și am investit în realizarea scopului propus ce pune accentul pe partea de **back-end**.

Partea de **back-end** a fost realizată prin intermediul limbajului **css** , **html** și **js** deoarece este una dintre cele mai bune alegeri pentru programarea, acestea necesită un calcul intensiv. Deci am fost nevoit să îmi fac propria machetă inspirându-mă după un model care mi-a plăcut înstă avea o versiune învechită de bootstrap, diferite versiuni fiind de cele mai multe ori incompatibile iar portarea lor la o versiune mai nouă fiind mai dificilă decât crearea de la zero.

Faptul că am utilizat **ejs** va permite integrarea mai ușoară în laboratoarele viitoare a paginii în backend, care presupun că are posibilități similare cu **ejs** doar sintaxa fiind diferită.