Ziua 1

1)Analiza temei: Inchirierea Automobilelor

Tema:

Tema „Închirierea mașinilor” face parte din domeniul aplicațiilor web interactive, care au ca scop facilitarea accesului utilizatorilor la servicii comerciale prin intermediul internetului. În contextul actual, serviciile de închiriere auto sunt din ce în ce mai căutate de persoane care au nevoie temporară de un mijloc de transport, fie pentru uz personal, fie pentru călătorii de afaceri sau vacanțe.

Aplicația propusă simulează un sistem simplificat de închiriere, oferind utilizatorului posibilitatea de a selecta dintr-o listă de mașini disponibile, fiecare având asociat un tarif fix pe zi. Utilizatorul poate introduce perioada dorită de închiriere, exprimată în zile, iar aplicația va calcula automat costul total pe care acesta îl are de plătit.

2) Intelegerea contextului si problemei pe care il abordeaza tema

Am ales tema „Închirierea mașinilor” pentru că este o problemă des întâlnită în viața de zi cu zi. Mulți oameni au nevoie, la un moment dat, să închirieze o mașină — fie pentru o vacanță, fie pentru o călătorie de afaceri sau pentru că mașina personală nu este disponibilă. În general, procesul de închiriere poate fi complicat sau consumator de timp, mai ales când trebuie să mergi fizic la o agenție sau să cauți informații din surse diferite.

Prin această aplicație, am vrut să creez o soluție simplă, practică și ușor de folosit, care să le permită utilizatorilor să vadă ce mașini sunt disponibile, cât costă și cât ar plăti în funcție de câte zile le închiriază. Practic, aplicația rezolvă problema lipsei de transparență și ușurință în calculul prețului, fără să fie nevoie de asistență din partea unei persoane.

M-am gândit la această aplicație ca la un punct de pornire. Deși este una simplă, ea ar putea fi extinsă ușor: cu opțiuni pentru asigurare, reduceri, rezervări sau chiar plată online. În plus, dezvoltarea acestui proiect m-a ajutat să înțeleg mai bine cum pot folosi HTML, CSS și JavaScript pentru a crea ceva util în viața reală, nu doar teorie.

În concluzie, problema pe care am vrut să o abordez este lipsa unei metode rapide și clare prin care cineva poate afla cât îl costă închirierea unei mașini. Aplicația oferă o soluție digitală care simplifică tot procesul.

3) Analiza a 3 platforme

1. **)**Rentalcars.com – este una dintre cele mai mari platforme globale de inchiriere auto. Nu inchiriaza direct, ci functioneaza ca un comparator de oferte – adica aduna date de la mai multe companii de inchirieri auto , cum ar fi Europcar, Hertz, si isi arata care sunt cele mai bune preturi in functie de locatie, data si tip de masina, este disponibila in circa 50.000 de locatii in intrega lume.

AVANTAJE : Poti compara oferte de la mai multe firme dintr-un singur loc;

-Este disponibila in foarte multe tari si limbi de vorbire;

-Ai filtre utile: tip de masina, pretul acestuia , kilometri mersi, si asigurarea;

-Deseori apar reduceri sau oferte special, mai ales daca rezervi din timp

DEZAVANTAJE : Pot fi preturi ascunse , mai marite , mai mari decat pe site-ul online;

-Politicile diferă de la un furnizor la altul, iar uneori nu sunt foarte clare pe site.

-Dacă există probleme cu rezervarea, soluționarea acestora poate fi greoaie din

cauza faptului că platforma este un intermediar între tine și firma de închirieri.

2 )Turo - este o platformă ceva mai recentă, dar care a câștigat rapid popularitate, în special în SUA, Canada și câteva țări din Europa de Vest. Este bazată pe modelul peer-to-peer, adică oamenii obișnuiți își pun mașinile personale la dispoziție pentru închiriere.

AVANTAJE : Ai sansa sa inchiriezi o masina de pe aproape , fara sa pleci undeva departe

-Se pot găsi oferte mai ieftine decât la firmele clasice, în special pentru închirieri de scurtă durată.

-Poti gasi masini deosebite , cum ar fi ceva rar , scump , luxos sau retro

DEZAVANTAJE : Fiind bazată pe utilizatori individuali, calitatea serviciului poate varia mult,

unii proprietari sunt seriosi si punctuosi , alti mai putin , depinde de om;

-Nu le poti gasi in toata lumea , in multe tari Turo nu exista;

-Asigurarea și suportul în caz de probleme depind de politica Turo, care nu este întotdeauna la fel de clară ca la companiile tradiționale;

3 ) Autonom este o firmă de închirieri auto 100% românească, cu o prezență puternică în majoritatea orașelor mari din România. Este cunoscută pentru seriozitatea serviciului și pentru că are o gamă destul de variată de mașini, ofera servicii de inchiriere auto, in peste 30 de orase din Romania, Ungaria si Serbia

AVANTAJE : Platforma si serviciul sunt bine organizate,iar angajati sunt profesionali si amabili;

-Găsești puncte de preluare în aproape toate orașele mari și la aeroporturi, ceea ce e foarte util pentru turiști sau pentru cei care vin în țară;

-Oferă și leasing , nu doar închiriere pe termen scurt, ceea ce poate fi util pentru firme sau freelanceri;

DEZAVANTAJE : Prețurile pot fi ceva mai mari decât la unele platforme internaționale, mai ales în sezonul turistic;

-Oferta de mașini este destul de bună, dar nu la fel de variată ca la agregatori internaționali;

-În orașele mai mici sau în afara sezonului, disponibilitatea poate fi mai redusă;

**Concluzie :**

Fiecare platformă are punctele ei forte și limitele ei. Dacă urmează o călătorie internațională și vrei să compari rapid ofertele mai multor firme, ***Rentalcars.com*** este foarte eficientă. Pentru o experiență diferită și posibil mai ieftină, în locuri unde este disponibil, ***Turo*** poate fi o alternativă interesantă, mai ales pentru pasionații de mașini. Iar dacă te afli în România și vrei siguranță și profesionalism, ***Autonom*** este, după părerea mea, una dintre cele mai bune alegeri.

Alegerea finală ține de context: durata închirierii, locul în care te afli, bugetul disponibil și nivelul de confort pe care îl dorești.

Ziua 2

1)Crearea unui cont GitHub si cum functioneaza platforma

GitHub este o platforma foarte folosita in lumea programarii, mai ales cand vine vorba de proiecte la care lucreaza mai multe persoane. Practic e un loc unde poti urca codul tau , il poti salva in siguranta si poti colabora cu alti colegii de clasa. Folosesti un sistem numit Git, care te ajuta sa tii evidenta tuturor modificarilor facute intr-un proiect.

Primul pas pentru a începe să folosești GitHub este crearea unui cont. Este simplu și gratuit. Accesezi site-ul oficial – <https://github.com> – și dai click pe butonul „Sign up”. Vei completa un formular cu:

* un **username** (nume de utilizator),
* o **adresă de email** ,
* o **parolă** (pe care so tii minte)

Dupa asta vei primi un cod pe gmailul utilizat, si il vei folosi pentru inregistrare, odata cu asta esti gata de lucru si poti incepe sa creezi proiecte

Dupa preferinta poti schimba numele , fotografia de profil si o descriere cu ce preferi

GitHub folosește **Git**, un sistem de control al versiunilor care îți permite să salvezi și să revii la diferite etape ale unui proiect. Cu GitHub, poți gestiona ușor codul scris

2)  Ce este Git, care sunt principiile de funcționalitate.

Git este un sistem de control al versiunilor, folosit în principal în programare pentru a ține evidența modificărilor făcute într-un proiect. Practic, cu Git poți salva „etape” ale muncii tale, ca să te poți întoarce oricând la o versiune anterioară dacă apare vreo problemă. Ce mi s-a părut util e că nu lucrezi direct pe fișierele finale, ci ai mereu o copie sigură și bine organizată a tot ce ai schimbat.

Git a fost creat de Linus Torvalds, același om care a creat și Linux, și este foarte rapid și eficient. Ce îmi place la el e că te ajută să lucrezi fără să-ți fie teamă că strici ceva, pentru că poți oricând reveni la o versiune mai veche sau poți compara modificările.

3) Studierea comenzilor Git

Gitul are multe comenzi usoare , care se scriu in panelul “Termal” – Terminal , si nu e nevoie de programamere avansata pentru a putea a le stii si utiliza

1. Git init – acesta comanda se foloseste pentru a crea un nou repository Git local .

2) Git clone - cu această comandă poți copia un proiect care este deja pe internet ,de exemplu pe GitHub, în calculatorul tău. Git clone + linkul care doresti sa copie .

3) Git status - Această comandă îți arată ce fișiere ai modificat, ce fișiere sunt pregătite pentru commit și ce fișiere noi există. E foarte utilă ca să vezi „starea” repo-ului.

4) Git add – Cu acesta comanda , spui lui Git ce fișiere vrei să salvezi la următorul commit. Poți adăuga un fișier sau toate fișierele

5) Git push - Comanda git push trimite modificările tale de pe calculator pe serverul online de exemplu, GitHub

4) Conectarea dispozitivului local la contul Github.

După ce mi-am creat un cont GitHub și am început să învăț cum funcționează Git, mi-am dat seama că trebuie să conectez calculatorul meu la cont ca să pot lucra mai eficient. Astfel, pot încărca și sincroniza proiectele fără să mai folosesc metode complicate.

### ****1. Instalarea Git****

Primul lucru pe care l-am făcut a fost să instalez Git. L-am descărcat de pe site-ul oficial (git-scm.com), am urmat pașii de instalare și apoi am verificat dacă funcționează scriind în terminal:

* git --version

Mi-a apărut versiunea instalată, deci știam că totul e ok.

### ****2. Configurarea datelor mele****

Git are nevoie de un nume și un email, ca să știe cine face modificările. Am rulat aceste două comenzi:

* git config --global user.name "Numele meu"
* git config --global user.email "emailulmeu@email.com"

Am folosit exact aceleași date ca în contul GitHub, ca să nu fie confuzii.

### ****3. Crearea unei chei SSH****

Pentru a face legătura între laptopul meu și contul GitHub, am folosit o cheie SSH (o metodă sigură de autentificare). Am creat cheia cu comanda:

* ssh-keygen -t ed25519 -C "emailulmeu@email.com"

Am apăsat Enter la fiecare întrebare și cheia s-a salvat automat în folderul ascuns .ssh.

### ****4. Adăugarea cheii în GitHub****

Am deschis fișierul cu cheia publică folosind:

* cat ~/.ssh/id\_ed25519.pub

Apoi am copiat tot conținutul și m-am dus pe site-ul GitHub, la „Settings > SSH and GPG keys”. Acolo am dat click pe „New SSH key”, am pus un nume (ca să știu de pe ce dispozitiv e) și am lipit cheia.

### ****5. Verificarea conexiunii****

Pentru a verifica dacă totul a fost setat corect, am scris:

* ssh -T git@github.com

Dacă totul e în regulă, GitHub îți zice un mesaj de salut cu numele tău de utilizator.

5)Analiza Stack-ului Tehnologic: VSCode, JS, HTML, CSS, Git

VSCode (Visual Studio Code) – Editor de Cod

VSCode este un editor de cod sursă dezvoltat de Microsoft, folosit pe scară largă în comunitatea dezvoltatorilor web datorită caracteristicilor sale puternice și personalizabile.

JavaScript (JS) – Limbaj de Programare

JavaScript este un limbaj de programare esențial pentru dezvoltarea web, permițând implementarea de interactivitate pe paginile web.

HTML (HyperText Markup Language) – Limbaj de Marcare

HTML este limbajul de marcare utilizat pentru a structura și a organiza conținutul unei pagini web, definind elementele vizibile pentru utilizator.

CSS (Cascading Style Sheets) – Limbaj de StilizareCSS este utilizat pentru a controla aspectul vizual al paginilor web, incluzând layout-uri, culori, fonturi și altele.

Git – Sistem de Control al Versiunii

Git este un sistem de control al versiunilor distribuit, folosit pentru a urmări modificările codului sursă pe măsură ce proiectele evoluează.

Ziua 3

Crearea unei pagini WEB

1. Programarea web. Ce este HTML

### ****1. Programarea web. Ce este HTML****

HTML (HyperText Markup Language) este limbajul de bază folosit pentru crearea și structurarea paginilor web. El definește elementele vizibile pe o pagină: titluri, paragrafe, imagini, linkuri etc. HTML nu este un limbaj de programare propriu-zis, ci unul de marcare (markup), adică folosește „etichete” pentru a organiza conținutul.

#### **a. Interacțiunea dintre client și server**

Când un utilizator accesează un site, are loc o comunicare între:

1. **Client** (ex: browserul tău web – Chrome, Firefox)
2. **Server** (calculatorul care găzduiește site-ul)

Clientul face o **cerere** (request), iar serverul răspunde cu o **pagină web** (răspuns – response), care conține fișiere HTML, CSS, JavaScript etc.

#### **b. Arhitectura aplicației web-based**

O aplicație web-based este structurată în general pe:

1. **Frontend**: tot ce vede utilizatorul (HTML, CSS, JavaScript)
2. **Backend**: partea din spate care gestionează date, baze de date, autentificare etc. (scris în limbaje precum PHP, Node.js, Python, Java)
3. **Bază de date**: stochează informații (ex: MySQL, MongoDB)

Aplicațiile moderne folosesc modelul **client-server**, uneori cu intermediere prin API-uri.

#### **c. Limbajul HTML**

Include:

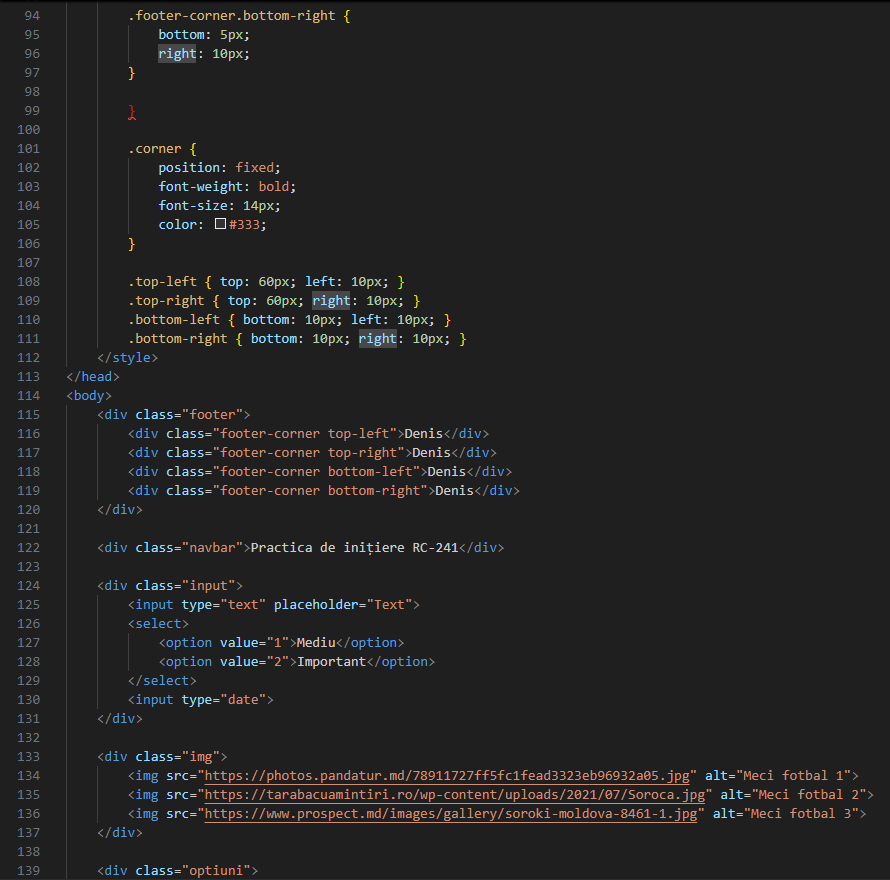
1. **Directive uzuale**: <html>, <head>, <body>, <title>, <h1>–<h6>, <p>, <br>, <hr>
2. **Ancore (linkuri)**: <a href="link.html">Text</a>
3. **Imagini**: <img src="imagine.jpg" alt="Descriere">
4. **Culoare fundal**: prin CSS, de ex. background-color: lightblue;
5. **Meta taguri**: pentru SEO, charset, responsive design: <meta charset="UTF-8">

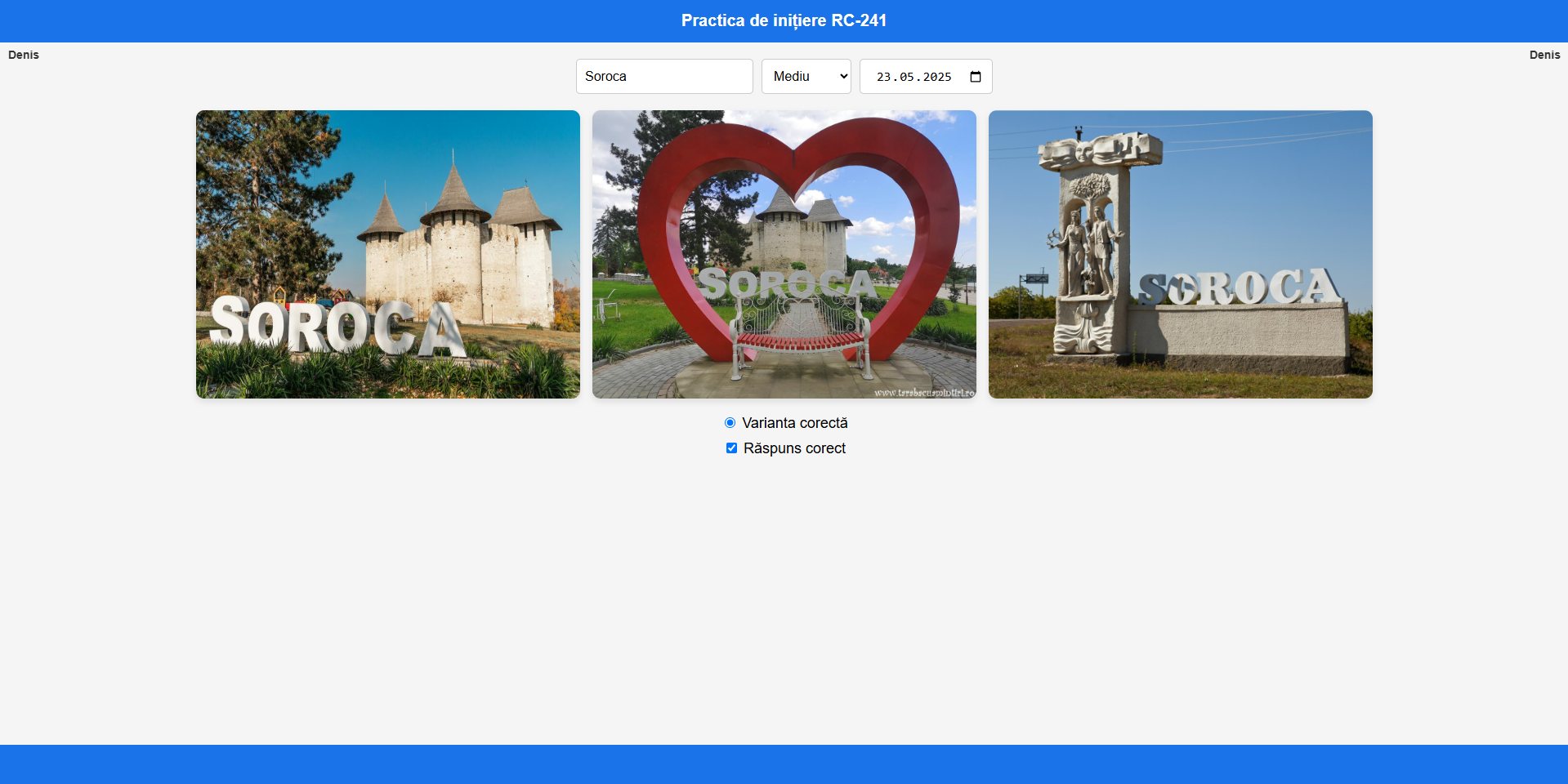
#### **d. Formularele în HTML, Tabelele, Cadrele (Frames)**

1. **Formulare**: permit trimiterea datelor către server. Exemple de taguri: <form>, <input>, <textarea>, <select>, <button>
2. **Tabele**: afișează date în rânduri și coloane: <table>, <tr>, <td>, <th>
3. **Cadre (frames)**: folosite în trecut pentru a împărți pagina în secțiuni independente cu <frameset> și <frame>, dar acum sunt învechite și înlocuite cu CSS și iframe-uri moderne.

Fig.1 – inceputul / style

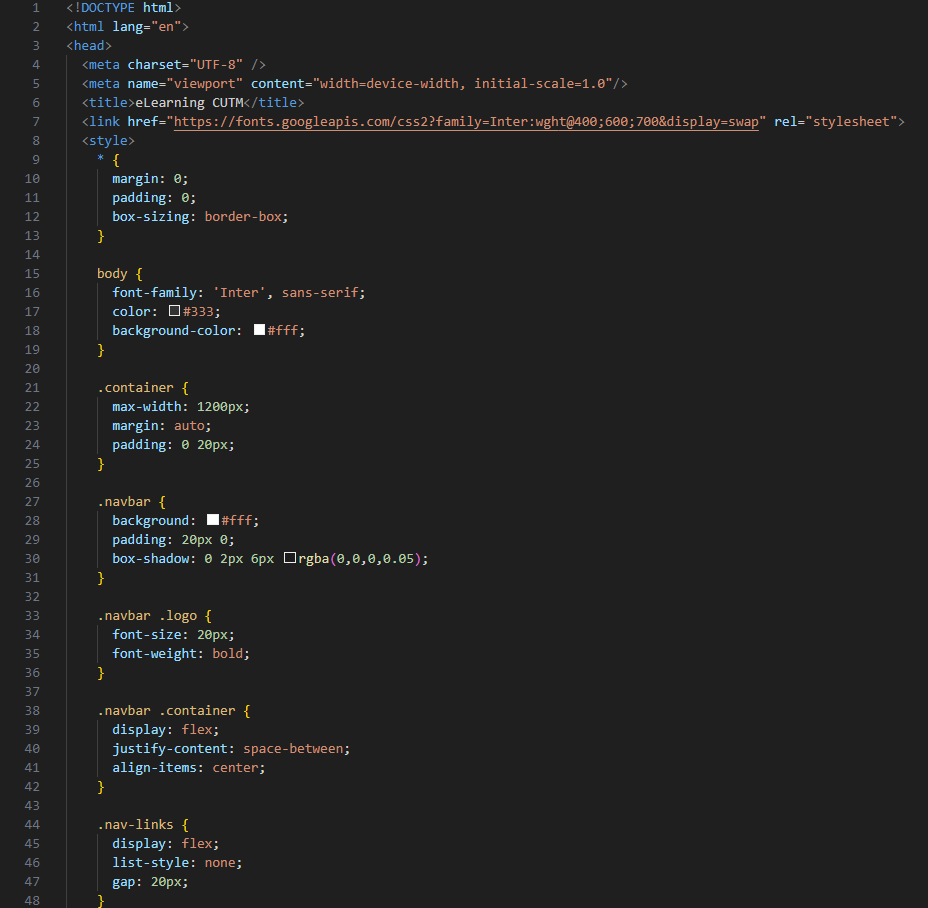


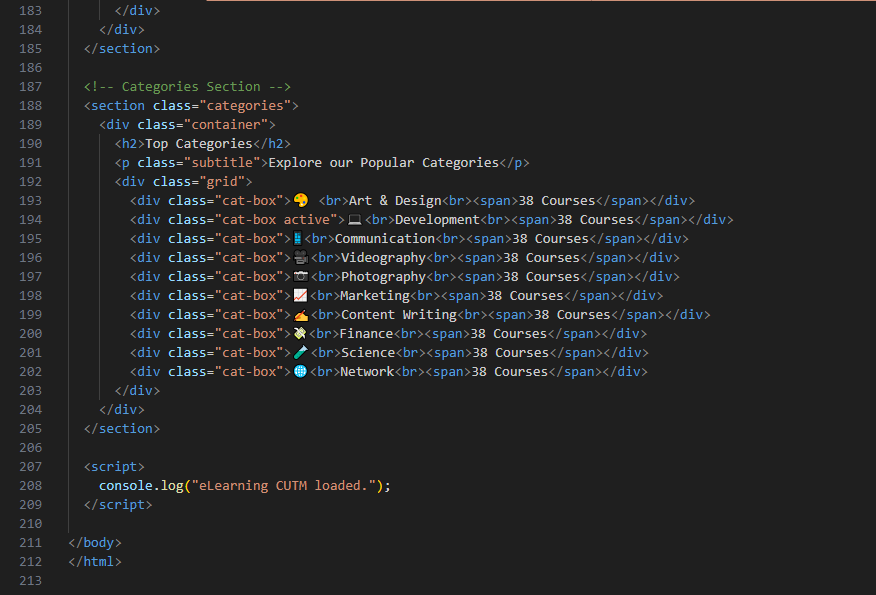
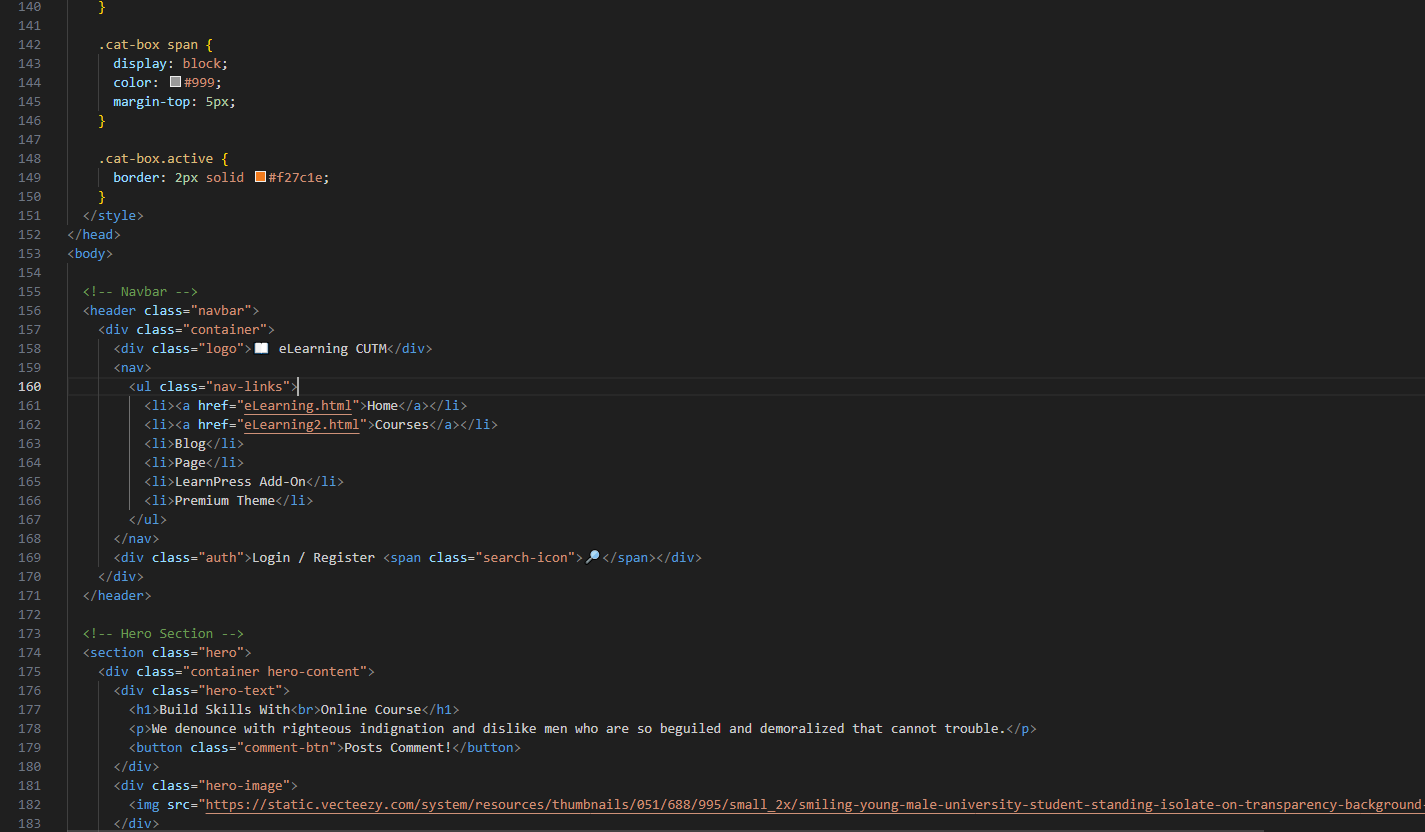
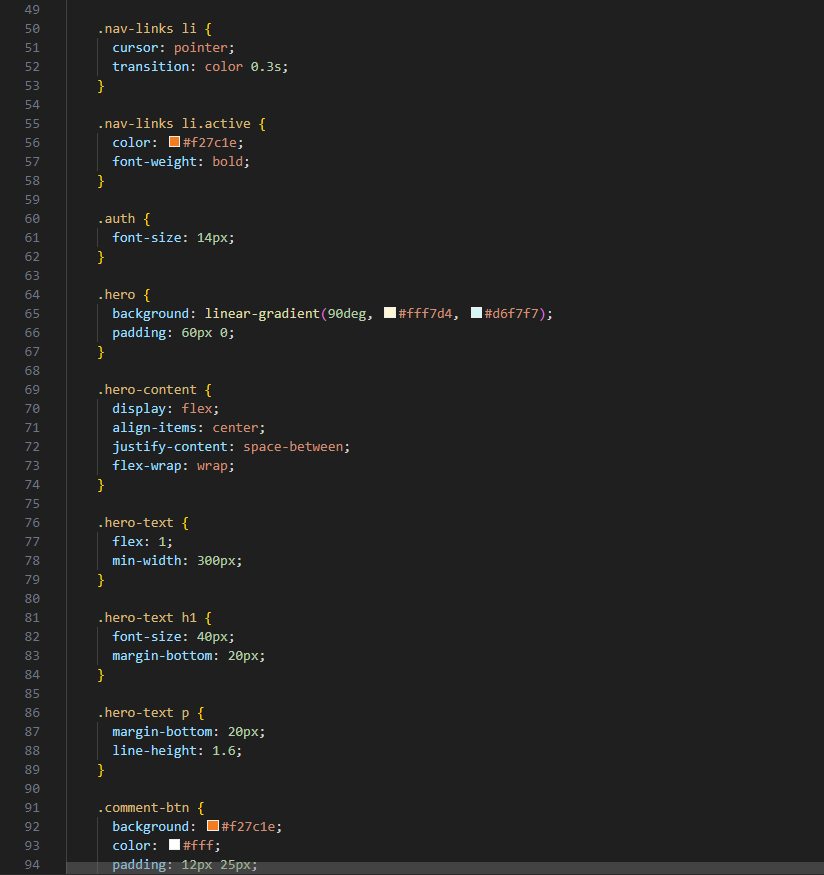
fig 2 - stylefig 3 – imaginile / codulfig 4 – sfarsitul



Ziua 4

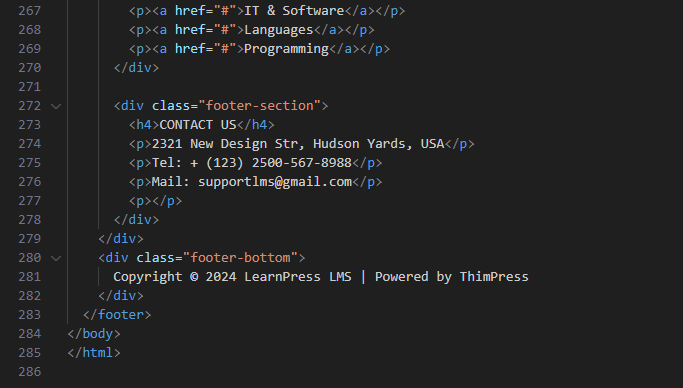
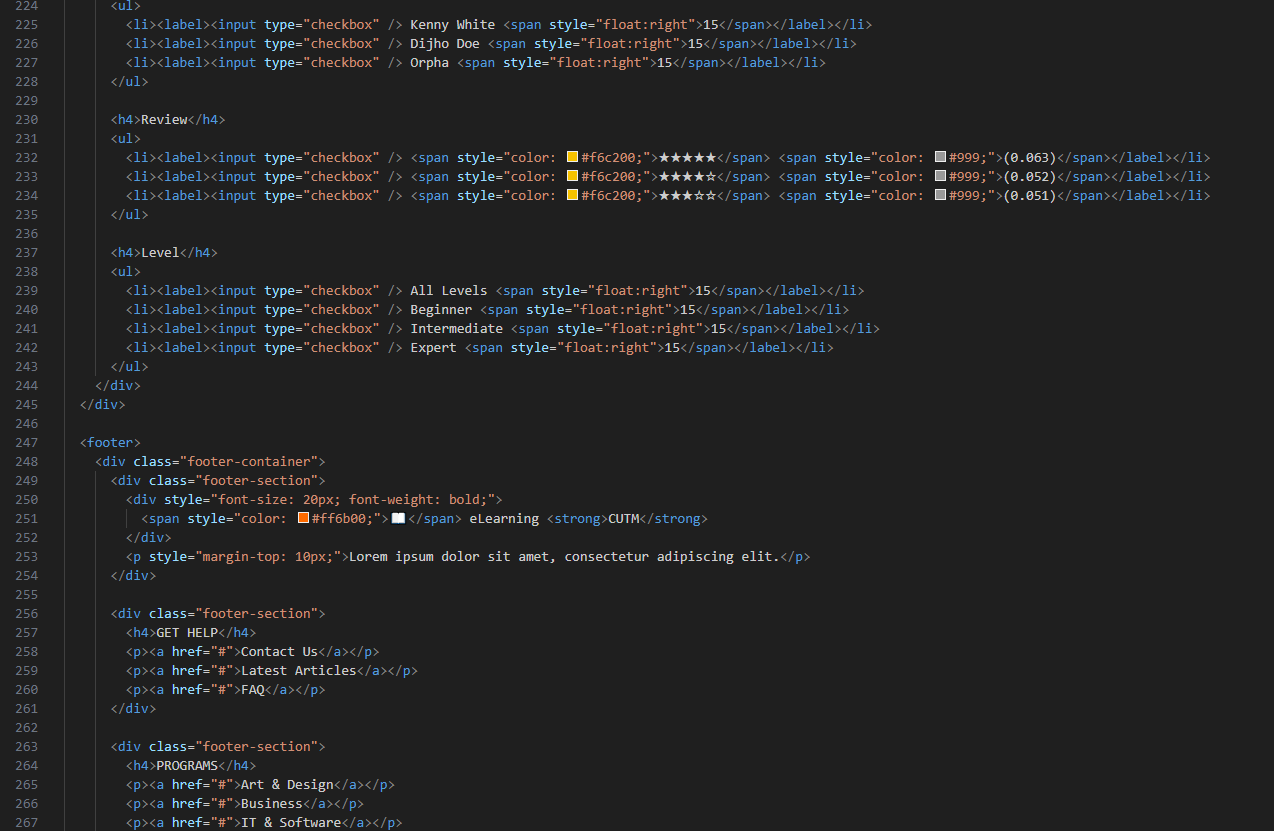
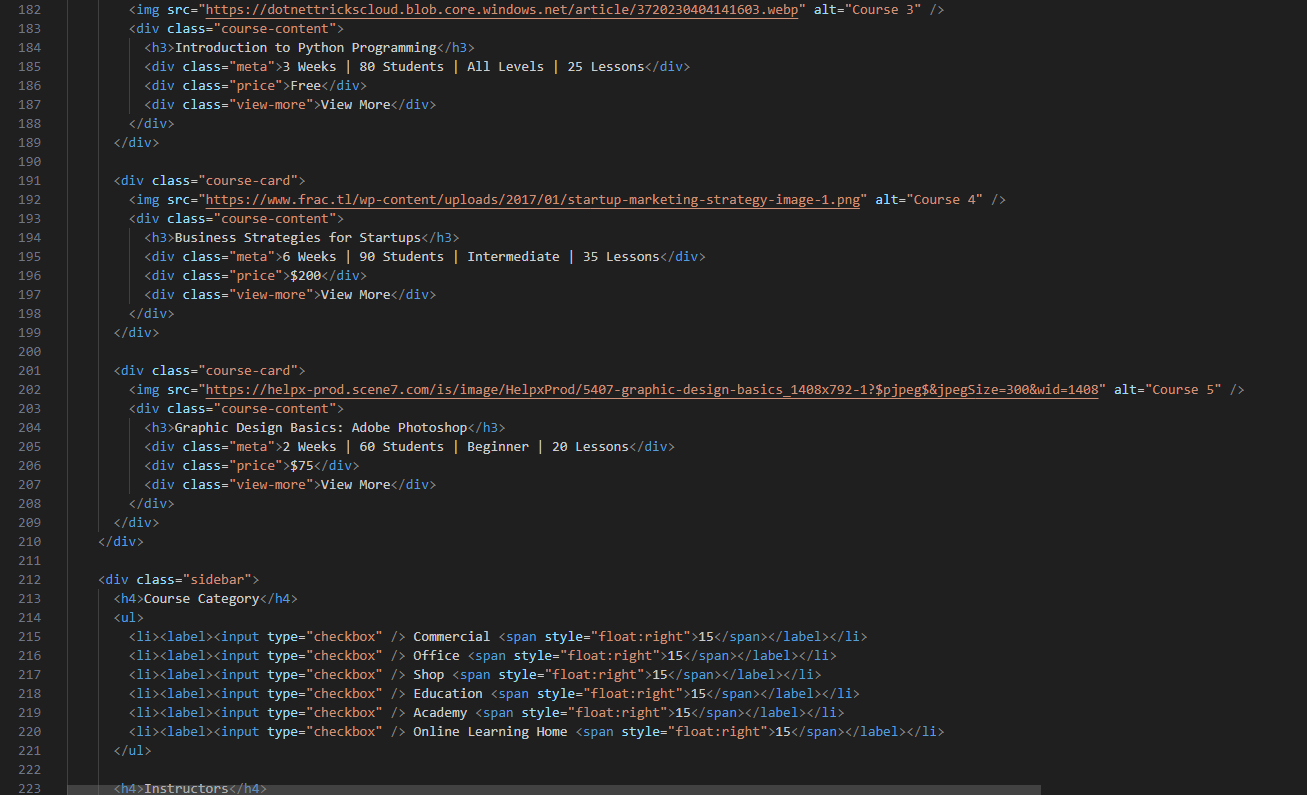
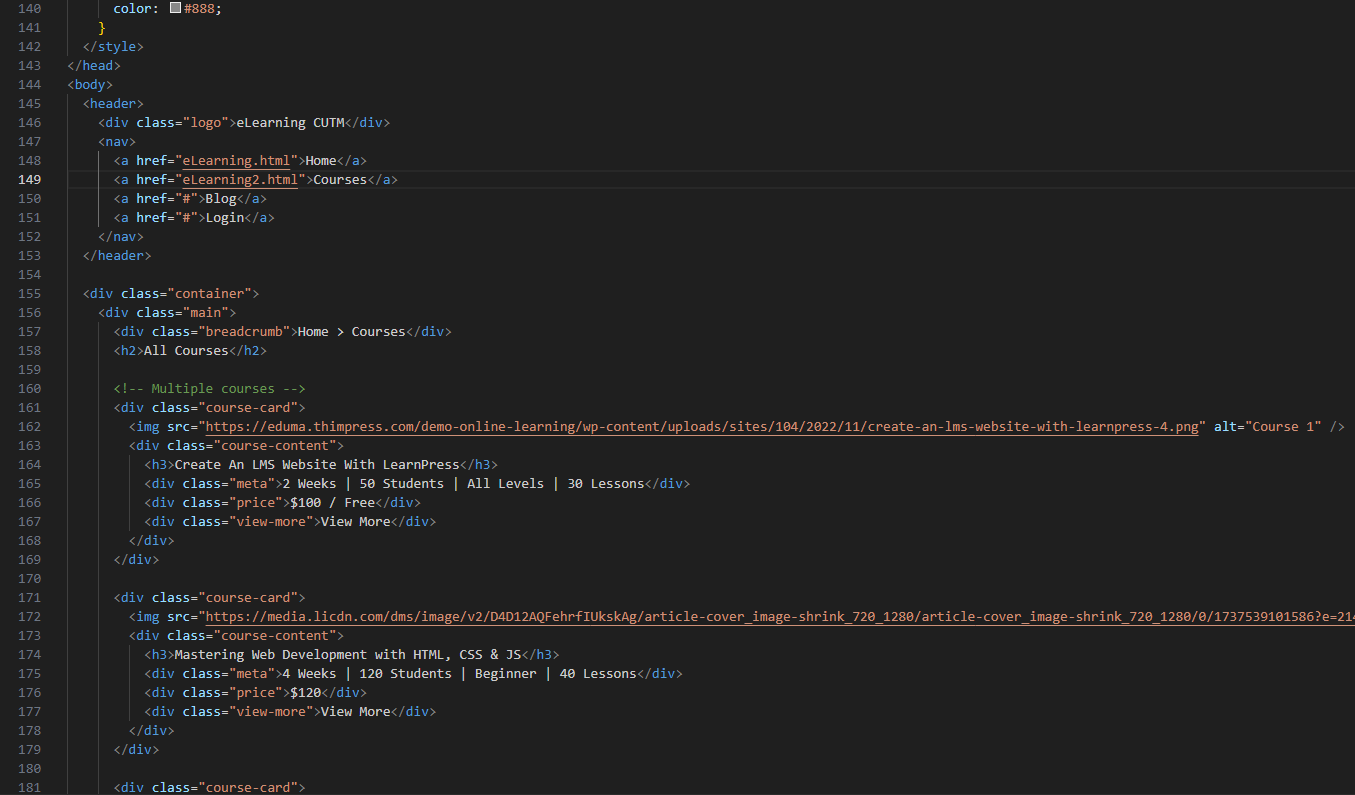
eLearning.html



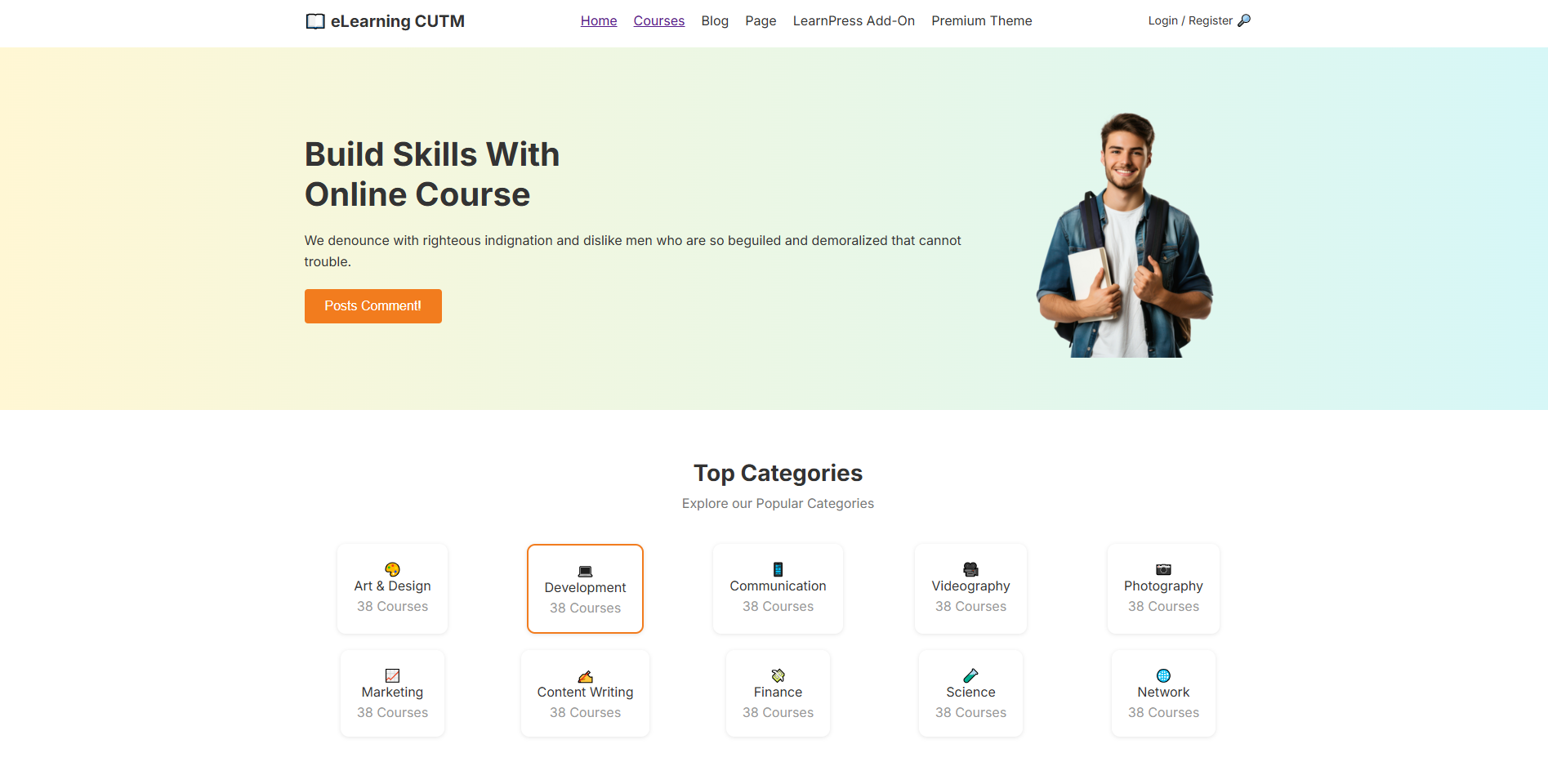


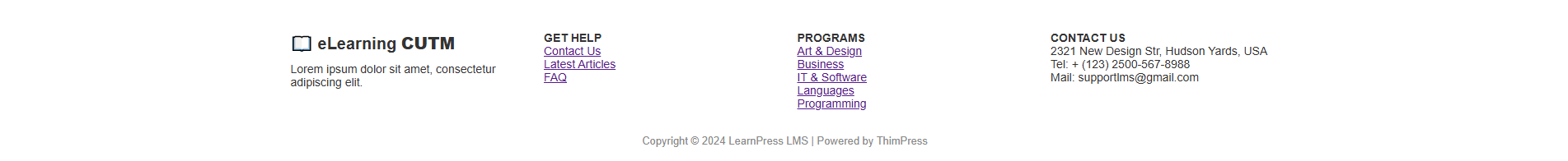
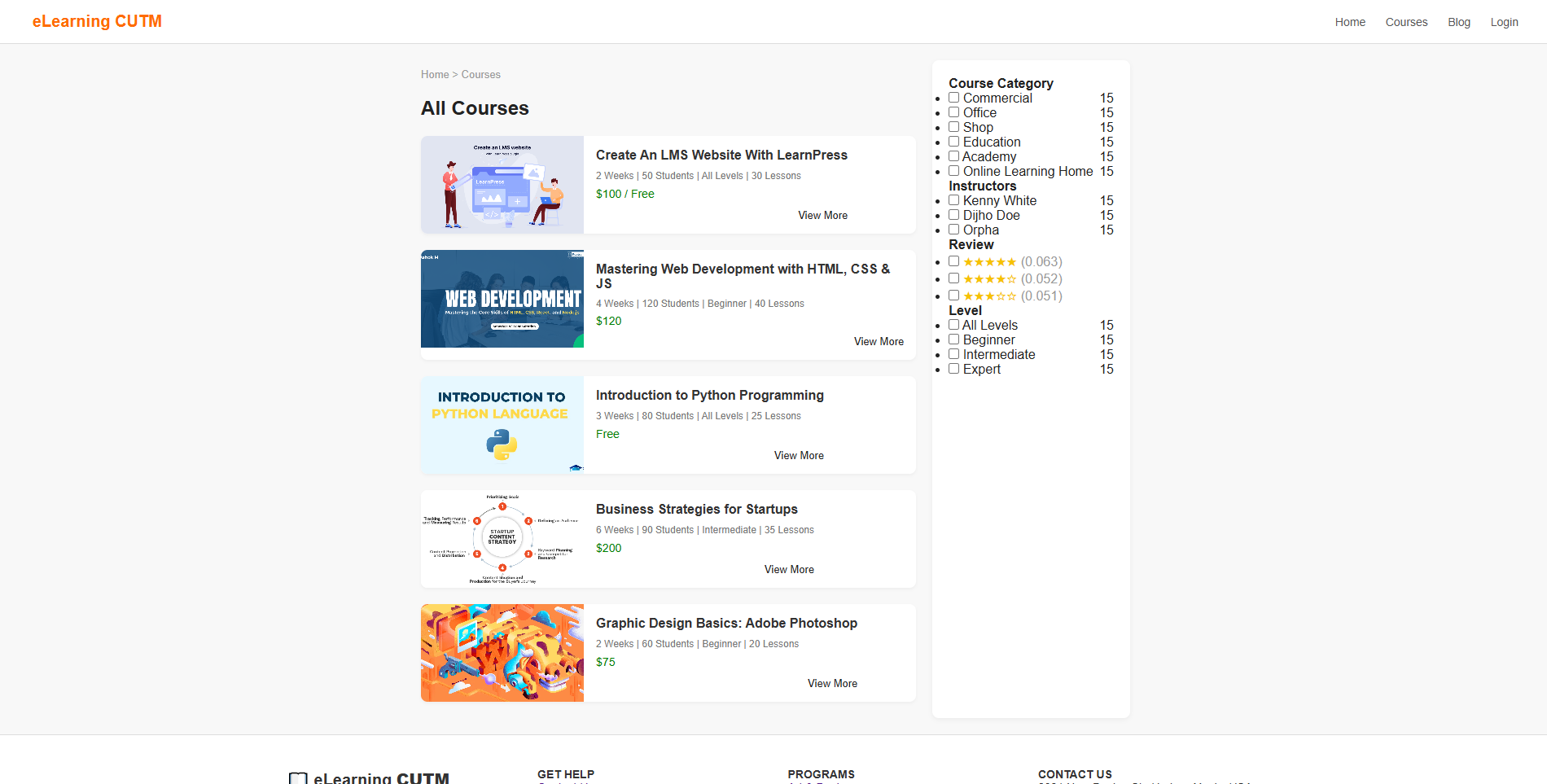
eLearning2.html





Rezultatul :





Concluzie :

Prima pagină (Landing Page / Homepage**)**

Scop principal:  
Aceasta este pagina principală de prezentare a platformei. Rolul ei este să capteze atenția utilizatorului și să prezinte pe scurt beneficiile, oferta educațională și categoriile disponibile.

Elemente cheie**:**

1. Navbar elegant cu logo, meniu de navigare și autentificare.
2. Secțiunea Hero cu un mesaj motivațional, un buton de acțiune („Posts Comment!”) și o imagine sugestivă (student).
3. Secțiune de categorii: afișează cele mai populare domenii (ex: Art & Design, Development, Marketing etc.) într-o grilă de 5x2, bine structurată vizual.
4. Design vizual modern, cu culori calde și font „Inter”, ceea ce oferă o identitate vizuală prietenoasă și atractivă.

Utilitate**:**  
Perfectă pentru atragerea utilizatorilor noi și oferirea unei priviri de ansamblu asupra platformei.

A doua pagină (Courses Listing Page**)**

Scop principal**:**  
Aceasta este o pagină de listare a cursurilor oferite de platformă, împreună cu un sidebar care permite filtrarea după categorii, instructori, recenzii și niveluri.

Elemente cheie:

1. Breadcrumb navigation: indică clar poziția utilizatorului pe site.
2. Listă detaliată de cursuri, fiecare cu imagine, titlu, descriere scurtă, durată, număr de studenți, nivel și preț.
3. Sidebar cu filtre (checklisturi): permite utilizatorului să selecteze tipuri de cursuri în funcție de preferințe.
4. Footer informativ și bogat, care include informații despre companie, programe, suport și date de contact.

Design:  
Ușor diferit ca stil față de prima pagină – folosește fonturi clasice („Arial”), culori mai neutre și structură tipică unei platforme educaționale. Este axată mai mult pe funcționalitate decât pe atracție vizuală.

Utilitate:  
Ideală pentru utilizatorii deja interesați de conținut și care vor să navigheze printre cursuri.

Concluzie generală:

Cele două pagini se completează reciproc și împreună formează un parcurs coerent pentru utilizator: de la descoperirea platformei (homepage), la explorarea ofertei educaționale detaliate (courses page).

1. Prima pagină este orientată spre branding și prezentare vizuală.
2. A doua pagină este orientată spre conținut și funcționalitate.

Este recomandat să existe o coerență mai mare între stiluri (fonturi, culori, iconografie), astfel încât experiența utilizatorului să fie uniformă. De asemenea, integrarea unei bare de căutare, filtre responsive pe mobil și o interacțiune mai dinamică ar putea îmbunătăți semnificativ experiența generală.

Ziua 5

1. Ce este **.env**  file ?

Un fișier .env este un fișier text simplu folosit pentru a salva variabile de configurare sau variabile de mediu ale unei aplicații. De obicei, este folosit pentru a păstra date sensibile sau importante, cum ar fi parole, chei secrete, link-uri către baze de date, porturi sau alte informații care țin de configurarea aplicației. Aceste variabile pot fi apoi accesate în codul sursă, fără a fi nevoie să fie scrise direct în fișierele de cod.

Fișierul .env este util mai ales pentru a păstra separarea între codul aplicației și datele de configurare. De exemplu, dacă dezvolt o aplicație local și apoi o urc pe un server de producție, pot folosi același cod, dar cu un fișier .env diferit, care conține valorile potrivite pentru acel mediu (precum o altă bază de date sau alt port).

Un alt motiv pentru care este folosit fișierul .env este acela că permite păstrarea în siguranță a unor date sensibile. În loc să scriu în cod o parolă sau o cheie API, pot scrie în .env ceva de genul SECRET\_KEY=ceva\_secret și apoi să o accesez din cod ca o variabilă. Astfel, dacă partajez codul cu cineva sau îl urc pe GitHub, pot exclude fișierul .env din versiunea publică și astfel nu expun datele sensibile.

Acest fișier este folosit frecvent în proiecte cu Node.js, Laravel, Django și alte framework-uri. De regulă, pentru ca un proiect să recunoască fișierul .env, trebuie să folosim o bibliotecă specială (cum ar fi dotenv în Node.js sau python-dotenv în Python) care știe să încarce aceste variabile în mediul aplicației.

În concluzie, fișierul .env este foarte important în dezvoltarea web modernă pentru că ajută la o mai bună organizare, securitate și flexibilitate a aplicației.

2) Ce este Json , unde se foloseste ?

**JSON** (JavaScript Object Notation) este un format de stocare și schimb de date care este ușor de citit pentru oameni și de procesat de calculatoare. Este bazat pe o structură simplă de tip cheie-valoare, asemănătoare cu obiectele din JavaScript, dar este folosit în foarte multe limbaje de programare.

JSON se scrie ca un text simplu, cu date organizate în perechi de forma "cheie": valoare, și este structurat cu acolade {} pentru obiecte și paranteze pătrate [] pentru liste. De exemplu:

{

"nume": "Andrei",

"varsta": 22,

"student": true,

"materii": ["Matematica", "Informatica"]

}

Unde se folosește JSON:

1. *Aplicații web* – Este unul dintre cele mai folosite formate pentru a face schimb de date între client (de exemplu un browser) și server. De exemplu, când o pagină web trimite un formular, răspunsul de la server poate fi în format JSON.
2. *Stocare locală* – Unele aplicații folosesc fișiere JSON pentru a salva date local, cum ar fi configurații, preferințe ale utilizatorului sau date temporare.
3. *API-uri* – Majoritatea serviciilor web (precum cele de la Google, Facebook, sau orice aplicație modernă) folosesc JSON pentru a trimite și primi informații între aplicații.
4. *Baze de date* – Unele baze de date moderne, cum ar fi MongoDB, salvează datele intern în format JSON (sau un format derivat din el, cum ar fi BSON).
5. *Fișiere de configurare* – Multe aplicații sau proiecte de programare (mai ales în JavaScript, Node.js, Python etc.) folosesc fișiere .json pentru configurări, cum ar fi package.json în Node.js.

**JSON** este foarte apreciat pentru că este simplu. E un mod eficient de a trimite informații între sisteme sau aplicații care vorbesc limbi diferite, dar înțeleg același format**.**

**Home.html**

**<!DOCTYPE html>**

**<html lang="ro">**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1.0"/>**

**<title>Pagina Principală</title>**

**<style>**

**/\* Stiluri comune \*/**

**:root {**

**--bg-color: #0d1b2a;**

**--navbar-bg: #1b263b;**

**--accent: #0077b6;**

**--text: #ffffff;**

**--input-bg: #415a77;**

**--input-border: #778da9;**

**--hover: #00b4d8;**

**}**

**body {**

**margin: 0;**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**min-height: 100vh;**

**background: linear-gradient(135deg, #1f1f1f, #2c3e50);**

**font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;**

**flex-direction: column;**

**padding-top: 70px;**

**box-sizing: border-box;**

**}**

**/\* Navbar \*/**

**.navbar {**

**position: fixed;**

**top: 0;**

**width: 100%;**

**background-color: var(--navbar-bg);**

**padding: 12px 0;**

**box-shadow: 0 4px 10px rgba(0, 0, 0, 0.4);**

**z-index: 1000;**

**}**

**.navbar-content {**

**width: 90%;**

**max-width: 1200px;**

**margin: 0 auto;**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**}**

**.navbar .left a {**

**margin: 0 10px;**

**color: var(--text);**

**text-decoration: none;**

**font-size: 16px;**

**padding: 8px 14px;**

**border-radius: 5px;**

**transition: background-color 0.3s;**

**}**

**.navbar .left a:hover {**

**background-color: var(--hover);**

**}**

**.navbar .right {**

**font-size: 18px;**

**font-weight: bold;**

**color: var(--accent);**

**}**

**/\* Form container \*/**

**.form-container {**

**background-color: var(--navbar-bg);**

**padding: 40px 30px;**

**border-radius: 12px;**

**box-shadow: 0 8px 20px rgba(0, 0, 0, 0.5);**

**width: 340px;**

**margin: 60px 20px;**

**animation: fadeIn 0.6s ease-out;**

**text-align: center;**

**}**

**.form-container h2 {**

**margin-bottom: 20px;**

**}**

**p {**

**color: var(--text);**

**font-size: 16px;**

**}**

**button {**

**width: 100%;**

**padding: 12px;**

**margin-top: 20px;**

**background-color: var(--hover);**

**color: white;**

**font-size: 16px;**

**border: none;**

**border-radius: 8px;**

**cursor: pointer;**

**transition: background-color 0.3s, transform 0.2s;**

**}**

**button:hover {**

**background-color: #0056b3;**

**transform: scale(1.03);**

**}**

**/\* Footer \*/**

**.footer {**

**background-color: var(--navbar-bg);**

**color: var(--text);**

**width: 100%;**

**text-align: center;**

**padding: 20px 10px;**

**margin-top: auto;**

**box-shadow: 0 -4px 10px rgba(0, 0, 0, 0.4);**

**font-size: 14px;**

**}**

**.footer p {**

**margin: 8px 0;**

**}**

**.footer .contact-info {**

**font-size: 16px;**

**}**

**/\* Animations \*/**

**@keyframes fadeIn {**

**from { opacity: 0; transform: translateY(20px); }**

**to { opacity: 1; transform: translateY(0); }**

**}**

**/\* Responsive \*/**

**@media (max-width: 600px) {**

**.navbar-content {**

**flex-direction: column;**

**gap: 10px;**

**text-align: center;**

**}**

**.form-container {**

**width: 90%;**

**}**

**}**

**</style>**

**</head>**

**<body>**

**<!-- Navbar -->**

**<div class="navbar">**

**<div class="navbar-content">**

**<div class="left">**

**<a href="acasa.html">My Project</a>**

**</div>**

**<div>**

**<a href="register.html">Register</a>**

**<a href="login.html">Log in</a>**

**<a href="logout.html">Log out</a>**

**</div>**

**</div>**

**</div>**

**<!-- Content -->**

**<div class="form-container">**

**<h2>Salut!</h2>**

**<p>Please choose one of the options below:</p>**

**<button onclick="location.href='register.html'">Register</button>**

**<button onclick="location.href='login.html'">Log in</button>**

**<button onclick="location.href='logout.html'">Log out</button>**

**</div>**

**<!-- Footer -->**

**<div class="footer">**

**<p>&copy; 2025 Inchirierea Automobilelor</p>**

**<div class="contact-info">**

**<p>Telefon: 062011064</p>**

**<p>Ne găsiți în Chișinău, Moldova</p>**

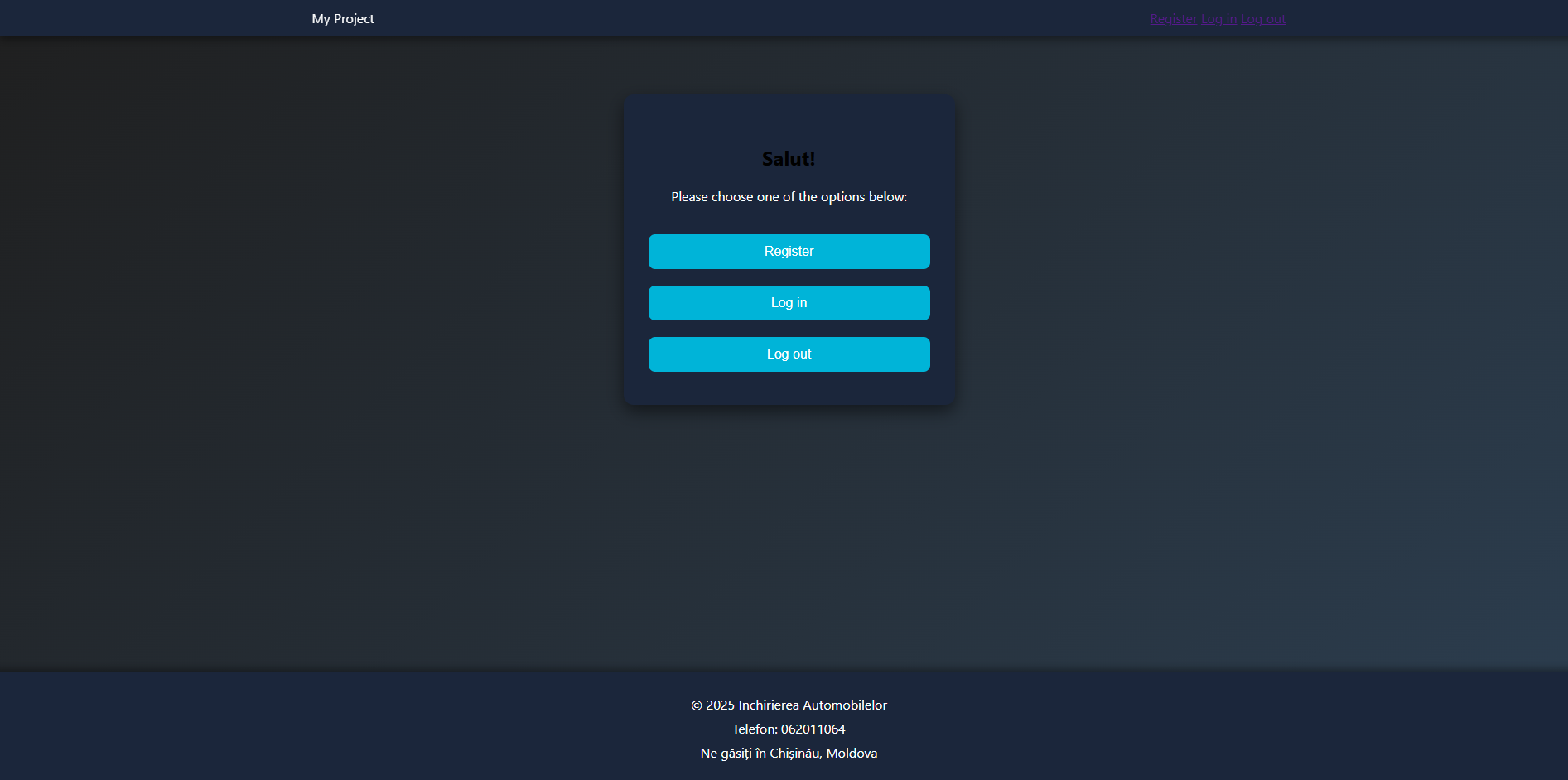
**</div>**

**</div>**

**</body>**

**</html>**

Rezultatul :



Register.html

<!DOCTYPE html>

<html lang="ro">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Înregistrare</title>

<style>

:root {

--bg-color: #0d1b2a;

--navbar-bg: #1b263b;

--accent: #0077b6;

--text: #ffffff;

--input-bg: #415a77;

--input-border: #778da9;

--hover: #00b4d8;

}

body {

margin: 0;

padding-top: 70px;

background-color: var(--bg-color);

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

color: var(--text);

display: flex;

flex-direction: column;

align-items: center;

min-height: 100vh;

box-sizing: border-box;

}

/\* Navbar \*/

.navbar {

position: fixed;

top: 0;

width: 100%;

background-color: var(--navbar-bg);

padding: 12px 0;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.4);

z-index: 1000;

}

.navbar-content {

width: 90%;

max-width: 1200px;

margin: 0 auto;

display: flex;

justify-content: space-between;

align-items: center;

}

.navbar .left a {

margin: 0 10px;

color: var(--text);

text-decoration: none;

font-size: 16px;

padding: 8px 14px;

border-radius: 5px;

transition: background-color 0.3s;

}

.navbar .left a:hover {

background-color: var(--hover);

}

.navbar .right {

font-size: 18px;

font-weight: bold;

color: var(--accent);

}

/\* Form container \*/

.form-container {

background-color: var(--navbar-bg);

padding: 40px 30px;

border-radius: 12px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.5);

width: 340px;

margin: 60px 20px;

animation: fadeIn 0.6s ease-out;

text-align: center;

}

.form-container h2 {

margin-bottom: 20px;

}

form {

display: flex;

flex-direction: column;

}

input[type="text"],

input[type="password"] {

background-color: var(--input-bg);

border: 1px solid var(--input-border);

border-radius: 8px;

padding: 12px;

margin-bottom: 15px;

font-size: 15px;

color: var(--text);

}

input:focus {

outline: none;

border-color: var(--hover);

}

input::placeholder {

color: #cbd5e1;

}

button {

background-color: var(--accent);

color: white;

border: none;

border-radius: 8px;

padding: 12px;

font-size: 16px;

cursor: pointer;

transition: background-color 0.3s, transform 0.2s;

}

button:hover {

background-color: var(--hover);

transform: scale(1.03);

}

/\* Footer \*/

.footer {

background-color: var(--navbar-bg);

color: var(--text);

width: 100%;

text-align: center;

padding: 20px 10px;

margin-top: auto;

box-shadow: 0 -4px 10px rgba(0, 0, 0, 0.4);

font-size: 14px;

}

.footer p {

margin: 8px 0;

}

/\* Animations \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(20px); }

to { opacity: 1; transform: translateY(0); }

}

/\* Responsive \*/

@media (max-width: 600px) {

.navbar-content {

flex-direction: column;

gap: 10px;

text-align: center;

}

.form-container {

width: 90%;

}

}

</style>

</head>

<body>

<!-- Navbar -->

<div class="navbar">

<div class="navbar-content">

<div class="left">

<a href="register.html">Register</a>

<a href="login.html">Log in</a>

<a href="logout.html">Log out</a>

</div>

<div class="right"></div>

<a href="home.html">My Project</a>

</div>

</div>

<!-- Register Form -->

<div class="form-container">

<h2>Înregistrare</h2>

<form action="/register" method="POST">

<input type="text" name="username" placeholder="Nume utilizator" required>

<input type="password" name="password" placeholder="Parolă" required>

<button type="submit">Înregistrează-te</button>

</form>

</div>

<!-- Footer -->

<div class="footer">

<p>&copy; 2025 Inchirierea Automobilelor</p>

<p>Telefon: 062011064</p>

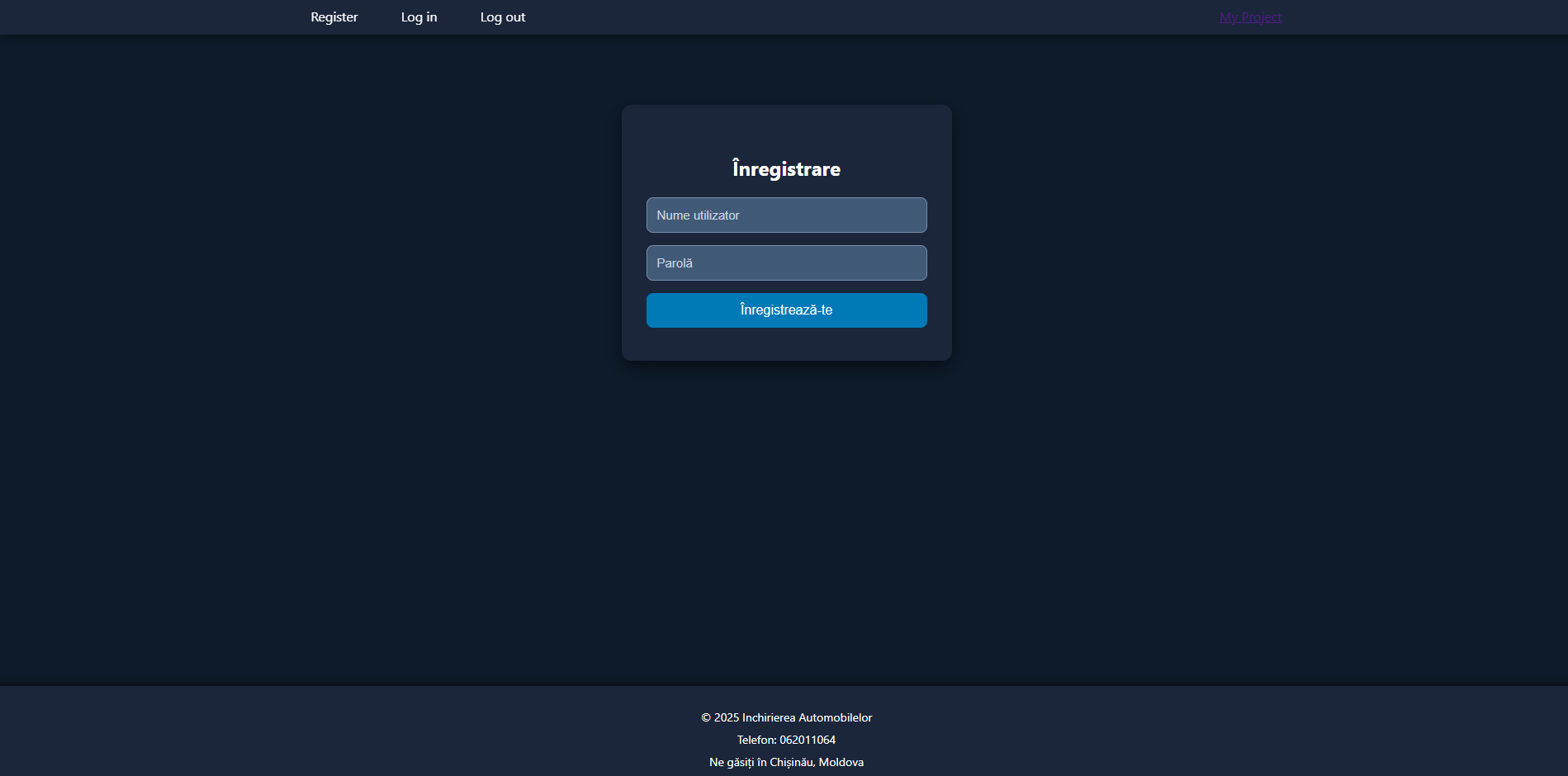
<p>Ne găsiți în Chișinău, Moldova</p>

</div>

</body>

</html>

Rezultatul :



Login.html

<!DOCTYPE html>

<html lang="ro">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Log in</title>

<style>

:root {

--bg-color: #0d1b2a;

--navbar-bg: #1b263b;

--accent: #0077b6;

--text: #ffffff;

--input-bg: #415a77;

--input-border: #778da9;

--hover: #00b4d8;

}

body {

margin: 0;

padding-top: 70px;

background-color: var(--bg-color);

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

color: var(--text);

display: flex;

flex-direction: column;

align-items: center;

min-height: 100vh;

box-sizing: border-box;

}

/\* Navbar \*/

.navbar {

position: fixed;

top: 0;

width: 100%;

background-color: var(--navbar-bg);

padding: 12px 0;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.4);

z-index: 1000;

}

.navbar-content {

width: 90%;

max-width: 1200px;

margin: 0 auto;

display: flex;

justify-content: space-between;

align-items: center;

}

.navbar .left a {

margin: 0 10px;

color: var(--text);

text-decoration: none;

font-size: 16px;

padding: 8px 14px;

border-radius: 5px;

transition: background-color 0.3s;

}

.navbar .left a:hover {

background-color: var(--hover);

}

.navbar .right {

font-size: 18px;

font-weight: bold;

color: var(--accent);

}

/\* Form container \*/

.form-container {

background-color: var(--navbar-bg);

padding: 40px 30px;

border-radius: 12px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.5);

width: 340px;

margin: 60px 20px;

animation: fadeIn 0.6s ease-out;

text-align: center;

}

.form-container h2 {

margin-bottom: 20px;

}

form {

display: flex;

flex-direction: column;

}

input[type="text"],

input[type="password"] {

background-color: var(--input-bg);

border: 1px solid var(--input-border);

border-radius: 8px;

padding: 12px;

margin-bottom: 15px;

font-size: 15px;

color: var(--text);

}

input:focus {

outline: none;

border-color: var(--hover);

}

input::placeholder {

color: #cbd5e1;

}

button {

background-color: var(--accent);

color: white;

border: none;

border-radius: 8px;

padding: 12px;

font-size: 16px;

cursor: pointer;

transition: background-color 0.3s, transform 0.2s;

}

button:hover {

background-color: var(--hover);

transform: scale(1.03);

}

/\* Footer \*/

.footer {

background-color: var(--navbar-bg);

color: var(--text);

width: 100%;

text-align: center;

padding: 20px 10px;

margin-top: auto;

box-shadow: 0 -4px 10px rgba(0, 0, 0, 0.4);

font-size: 14px;

}

.footer p {

margin: 8px 0;

}

/\* Animations \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(20px); }

to { opacity: 1; transform: translateY(0); }

}

/\* Responsive \*/

@media (max-width: 600px) {

.navbar-content {

flex-direction: column;

gap: 10px;

text-align: center;

}

.form-container {

width: 90%;

}

}

</style>

</head>

<body>

<!-- Navbar -->

<div class="navbar">

<div class="navbar-content">

<div class="left">

<a href="register.html">Register</a>

<a href="login.html">Log in</a>

<a href="logout.html">Log out</a>

</div>

<div class="right"></div>

<a href="home.html">My Project</a>

</div>

</div>

<!-- Login Form -->

<div class="form-container">

<h2>Log in</h2>

<form action="/login" method="POST">

<input type="text" id="username" name="username" placeholder="Introdu numele de utilizator" required>

<input type="password" id="password" name="password" placeholder="Introdu parola" required>

<button type="submit">Log in</button>

</form>

</div>

<!-- Footer -->

<div class="footer">

<p>&copy; 2025 Inchirierea Automobilelor</p>

<p>Telefon: 062011064</p>

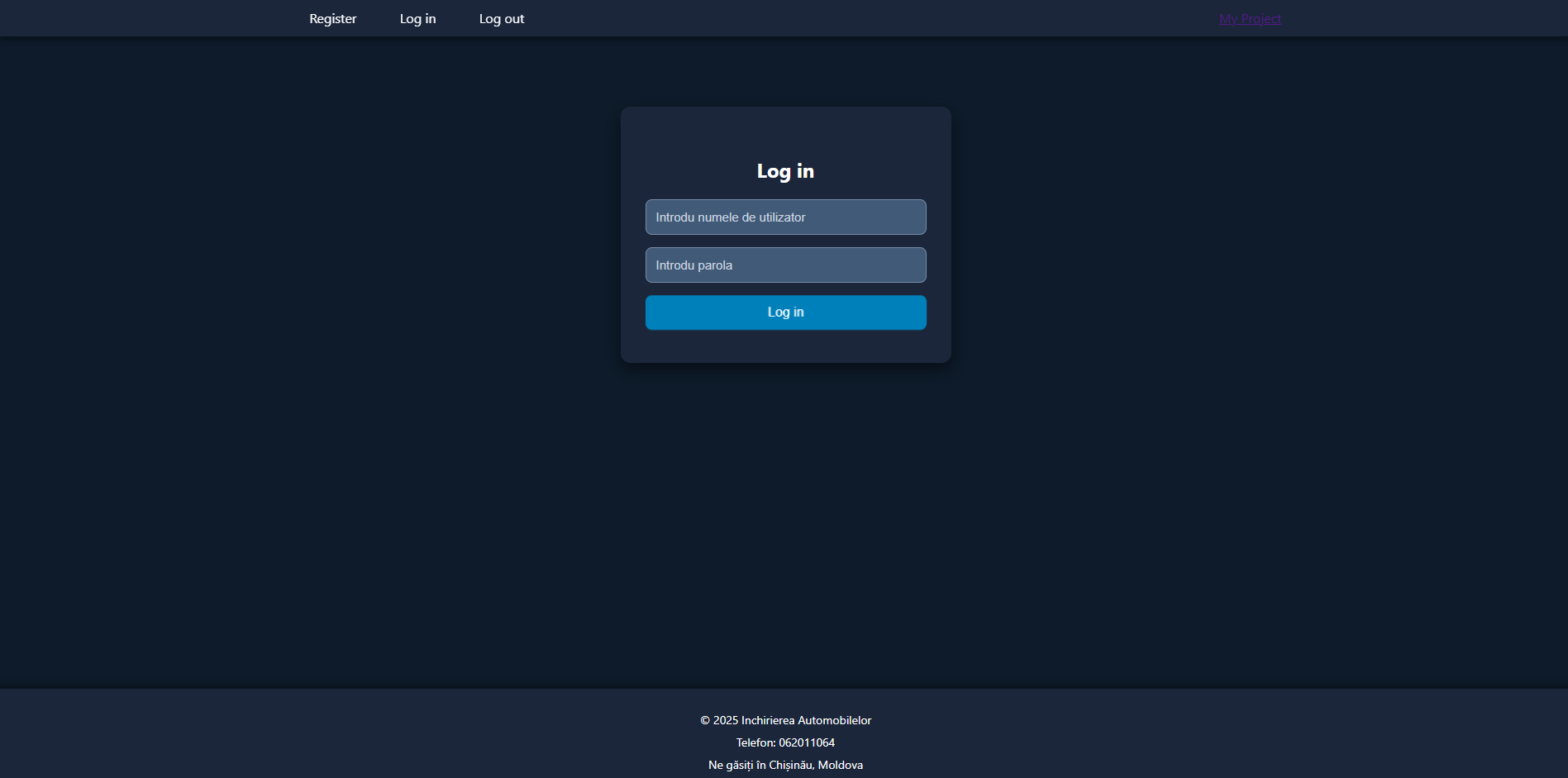
<p>Ne găsiți în Chișinău, Moldova</p>

</div>

</body>

</html>

Rezultatul :



Logout.html

<!DOCTYPE html>

<html lang="ro">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Log out</title>

<style>

:root {

--bg-color: #0d1b2a;

--navbar-bg: #1b263b;

--accent: #0077b6;

--text: #ffffff;

--input-bg: #415a77;

--input-border: #778da9;

--hover: #00b4d8;

}

body {

margin: 0;

padding-top: 70px;

background-color: var(--bg-color);

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

color: var(--text);

display: flex;

flex-direction: column;

align-items: center;

min-height: 100vh;

box-sizing: border-box;

}

/\* Navbar \*/

.navbar {

position: fixed;

top: 0;

width: 100%;

background-color: var(--navbar-bg);

padding: 12px 0;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.4);

z-index: 1000;

}

.navbar-content {

width: 90%;

max-width: 1200px;

margin: 0 auto;

display: flex;

justify-content: space-between;

align-items: center;

}

.navbar .left a {

margin: 0 10px;

color: var(--text);

text-decoration: none;

font-size: 16px;

padding: 8px 14px;

border-radius: 5px;

transition: background-color 0.3s;

}

.navbar .left a:hover {

background-color: var(--hover);

}

.navbar .right {

font-size: 18px;

font-weight: bold;

color: var(--accent);

}

/\* Form container \*/

.form-container {

background-color: var(--navbar-bg);

padding: 40px 30px;

border-radius: 12px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.5);

width: 340px;

margin: 60px 20px;

animation: fadeIn 0.6s ease-out;

text-align: center;

}

.form-container h2 {

margin-bottom: 20px;

}

p {

color: var(--text);

font-size: 16px;

}

/\* Footer \*/

.footer {

background-color: var(--navbar-bg);

color: var(--text);

width: 100%;

text-align: center;

padding: 20px 10px;

margin-top: auto;

box-shadow: 0 -4px 10px rgba(0, 0, 0, 0.4);

font-size: 14px;

}

.footer p {

margin: 8px 0;

}

/\* Animations \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(20px); }

to { opacity: 1; transform: translateY(0); }

}

/\* Responsive \*/

@media (max-width: 600px) {

.navbar-content {

flex-direction: column;

gap: 10px;

text-align: center;

}

.form-container {

width: 90%;

}

}

</style>

</head>

<body>

<!-- Navbar -->

<div class="navbar">

<div class="navbar-content">

<div class="left">

<a href="register.html">Register</a>

<a href="login.html">Log in</a>

<a href="logout.html">Log out</a>

</div>

<div class="right"></div>

<a href="home.html">My Project</a>

</div>

</div>

<!-- Log out Confirmation -->

<div class="form-container">

<h2>Ați fost delogat cu succes!</h2>

<p>Vă mulțumim că ați utilizat serviciile noastre.</p>

</div>

<!-- Footer -->

<div class="footer">

<p>&copy; 2025 Inchirierea Automobilelor</p>

<p>Telefon: 062011064 | Chișinău, str. Testului 123</p>

</div>

</body>

</html>

Rezultatul :

