

RAPORT 3 – Final

Proiect III-Instrumente pentru dezvoltarea programelor



Student: Draghici Andreea-Maria

Grupa: CR4.S1 A

Anul de studiu: IV

Specializarea: Calculatoare Romana

LIBRARY MANAGEMENT

Cuprins

1. Introducere
2. Proiectarea si implementarea finala
3. Concluzii

Introducere

Acest raport are scopul de a arata proiectarea finala a proiectului Library Management. Aplicatia pe care o implementez este o aplicatie web MVC folosind Entity Framework Core si ASP.Net Core, o aplicatie folosita pentru gestionarea activitatii unei biblioteci / librarii in format electronic.

La accesarea website-ului, orice utilizator poate observa in bara de navigare toate paginile, si anume: Home, About, Books, Contact, Register, Login, de unde se poate naviga pe oricare pagina, in ce ordine se doreste sau in functie de nevoile utilizatorului.

Proiectarea si implementarea finala

Acest sistem este format dintr-o singura parte, o aplicatie web, care va fi utilizata de bibliotecar, dar si de catre cititori. Aplicatia comunica constant cu baza sa de date.

Astfel spus, aplicatia permite doua tipuri de utilizatori:

- Utilizatorii care au rolul de user si pot vizualiza lista de carti in librerie de, pot sa trimita un ticket catre asistenta pentru utilizatori cu privire la orice eroare/ problema intalnita in timpul aplicatiei, pot cauta o carte anume din librerie.
2. Utilizatorii care au rolul de administrator si vor avea aceleasi functionalitati ca si utilizatori normali, dar pe langa asta vor mai avea si altele in plus:
- Administratorul poate adauga, crea, sterge sau edita carti din lista de carti sau date despre acestea.
 - Administratorul poate vizualiza lista de carti rezervate.
 - Administratorul poate modifica informatia starii unei carti anume din baza de date. (Ex: Daca cartea nu se mai afla pe stoc in biblioteca).
 - Administratorul poate actualiza informatii despre politica de confidentialitatea sau despre cartile rezervate.

Functionalitatile aplicatiei sunt urmatoarele:

- In primul rand, aplicatia are un formular de inregistrare, unde user-ul isi introduce datele pentru a-si crea contul. Acestia isi pot seta email-ul, parola, etc.
- Aplicatia dispune si de o pagina de login, unde utilizatori isi vor introduce email-ul si parola, iar daca sunt corecte, ei vor fi redirectionati catre pagina principala de Home.
- Aplicatia are o pagina de contact.
- Aplicatia are o pagina de About care ofera informatii despre site si activitatea acestuia.
- Un utilizator (indiferent de rol) poate accesa functionalitatea aplicatiei, numai dupa ce s-a logat.
- Aplicatia dispune si de o pagina numita Books, unde se gaseste lista cu carti in librerie.
- Aplicatia contine si pagina principala de Home, aceasta oferind posibilitatea utilizatorului de a naviga catre lista de carti din librerie, aceasta pagina mai ofera si informatii si fotografii despre cartile care au fost accesate de catre alti utilizatori, astfel incat noii utilizatori sa poata vedea care sunt cele mai recente carti.

Pentru partea de autentificare am utilizat functionalitatea Register din ASP.Net Core Identity. Fiecare utilizator care isi creeaza un nou cont de utilizatori, va primi rolul de user si poate avea acces la toate paginile mai putin la pagina de adaugare a cartilor, pagina de stergere a unei carti, pagina de editare a unei carti, aceste functionalitati sunt accesibile doar utilizatorului cu rol de administrator.

LIBRARYMANAGEMENT HOME BOOKS CONTACT ABOUT REGISTER LOGIN

REGISTER

CREATE A NEW ACCOUNT.

EMAIL
name@example.com

PASSWORD
password

CONFIRM PASSWORD
confirm your password

Register

© 2022 - LIBRARY MANAGEMENT BY ANDREEA DRAGHICI - [PRIVACY](#)

Fiecare utilizator care isi face un cont de utilizator isi poate introduce mail-ul si parola. Pentru introducerea datelor am folosit validari. Adresa de email trebuie sa aiba un format lizibil.

CREATE A NEW ACCOUNT.

USERNAME 'ADMIN@GMAIL.COM' IS ALREADY TAKEN.

EMAIL

qwer

THE EMAIL FIELD IS NOT A VALID E-MAIL ADDRESS.

PASSWORD

password

CONFIRM PASSWORD

confirm your password

Register

Pentru pagina de login, am utilizat functionalitatea Login din ASP.NET Core Identity. Fiecare utilizator care doreste sa se logheze isi poate introduce mail-ul si parola. Orice utilizator, indiferent de rol, poate accesa functionalitatea aplicatiei numai dupa ce s-a logat la aplicatie.

LIBRARYMANAGEMENT HOME BOOKS CONTACT ABOUT REGISTER LOGIN

LOG IN

USE A LOCAL ACCOUNT TO LOG IN.

EMAIL

client@gmail.com

PASSWORD

☐ REMEMBER ME?

Log in

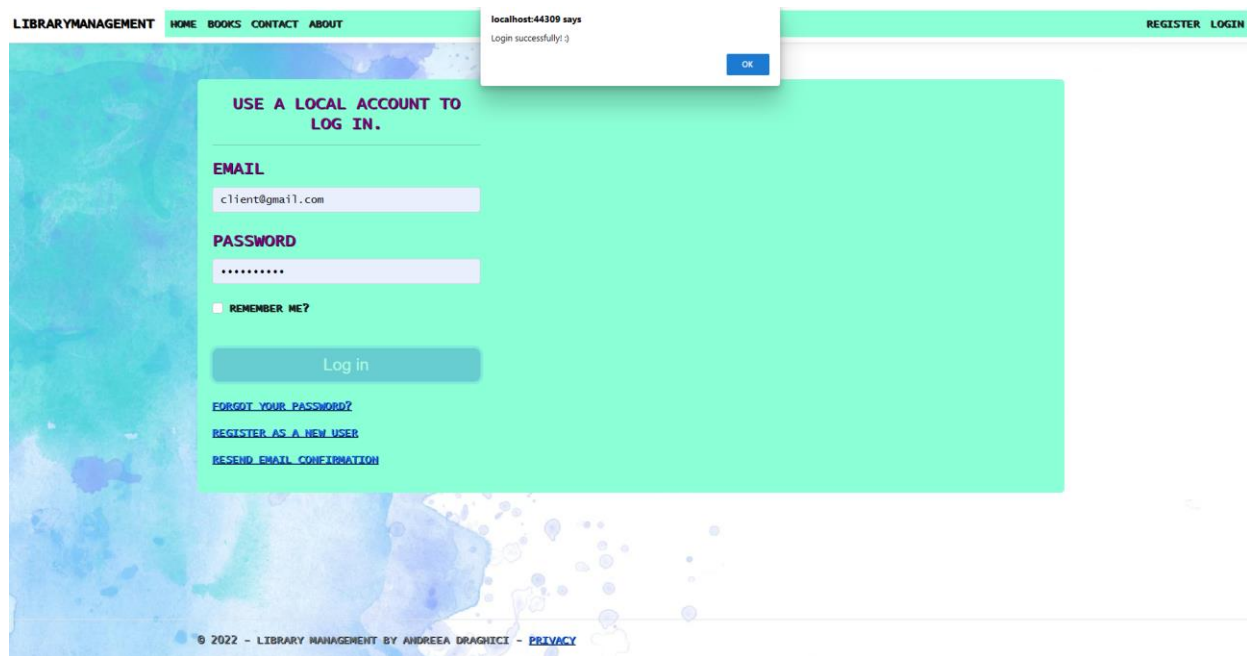
[FORGOT YOUR PASSWORD?](#)

[REGISTER AS A NEW USER](#)

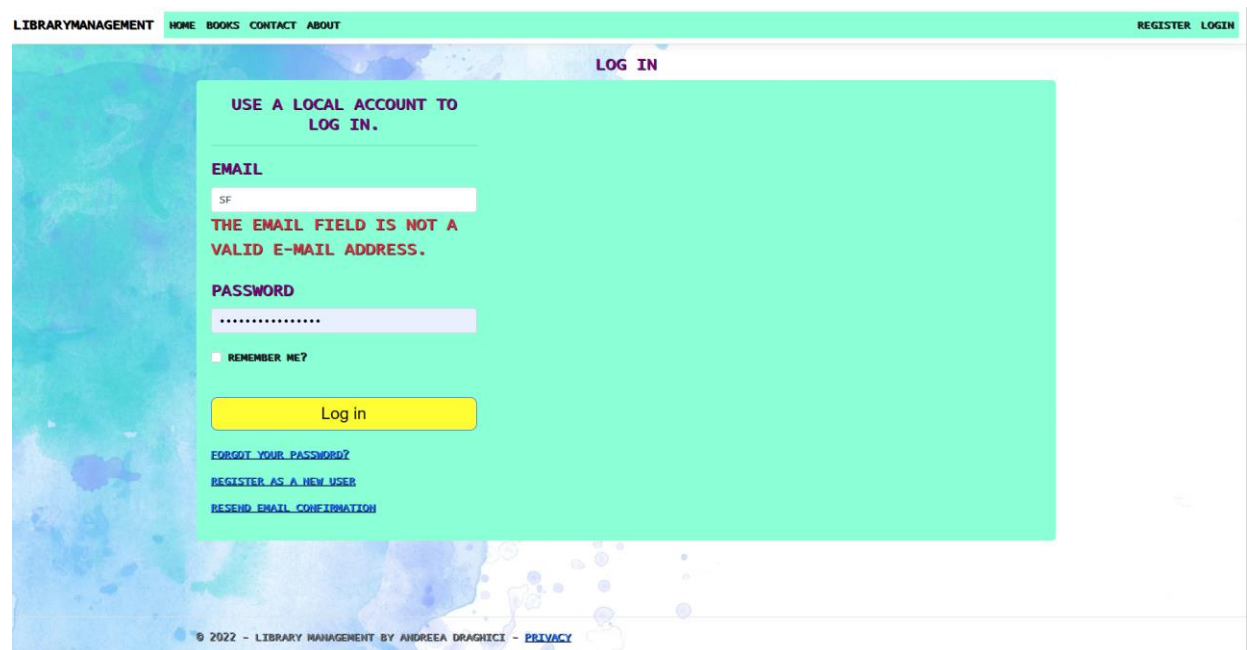
[RESEND EMAIL CONFIRMATION](#)

© 2022 - LIBRARY MANAGEMENT BY ANDREEA DRAGHICI - [PRIVACY](#)

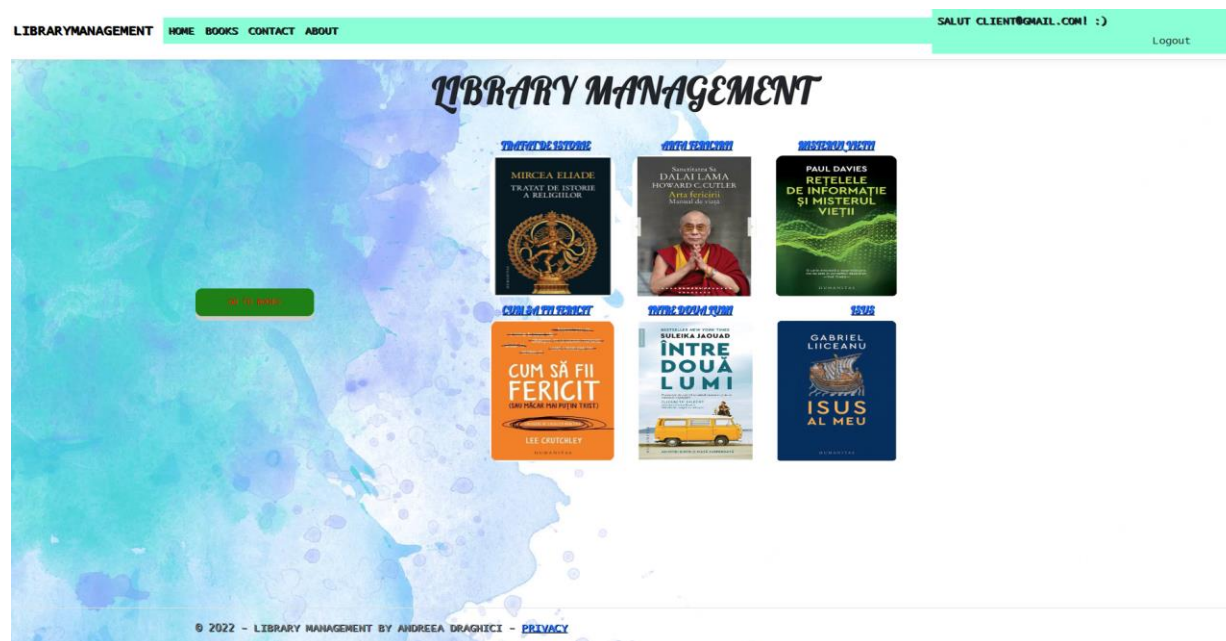
Pentru introducerea datelor am folosit validari, in momentul in care datele introduse de utilizatori sunt valide, in partea de sus a ecranului va aparea un pop-up alert cu mesajul "Login successfully!", care ii va confirma utilizatorului ca acesta s-a conectat cu succes. Pentru acest lucru, am folosit o functie in JavaScript care va fi apelata atunci cand utilizatorul isi introduce corect datele si apasa pe butonul log-in.



In cazul in care utilizatorul greseste datele cand doreste sa se logheze, va aparea urmatorul mesaj:



Dupa ce utilizatorul se conecteaza la aplicatie, acesta este redirectionat catre pagina de home, indiferent de rolul pe care acesta il are, iar email-ul acestuia cat este afisat in bara de navigare. Butonul de logout este si el activ acum. Utilizatorul normal poate naviga catre lista de carti din librerie, acest lucru fiind posibil accesand butonul de 'Go to library', pe langa aceasta functionalitate, utilizatorul mai are acces atat la informatiile, cat si la fotografiile despre cartile care au fost accesate de catre alti utilizatori.



Pentru a vizualiza informatii despre cartile accesate de alti utilizatori, utilizatorul trebuie sa acceseze link-urile care se afla deasupra fiecarui card din pagina de home, acesta va fi redirectionat catre o pagina web externa de pe un site ce contine mai multe informatii despre respectiva carte accesata.



Accesand link-ul “Misterul vietii” de deasupra cardului “Rețelele de informatie si misterul vietii”, utilizatorul a fost redirectionat catre o pagina ce contine mai multe detalii despre cartea respectiva.



Accesand pagina About, utilizatorul nostru poate vizualiza informatii despre activitatea aplicatiei noastre.



Utilizatorul are posibilitatea de a vizualiza informatii despre lista cu carti disponibile in librerie, nume, categorie, editura, id-ul carti din stoc. Utilizatorul poate cauta o anumite carti din librerie. Butoanele

Edit si Delete ofera functionalitate doar utilizatorului cu rolul de administrator, in cazul in care acestea sunt accesate de un utilizator cu rolul de user, va aparea un mesaj de forma 'Access denied'.

LIBRARYMANAGEMENT HOME BOOKS CONTACT ABOUT SALUT CLIENT@GMAIL.COM! :) Logout

CARTI

Cauta o carte...

NAME	PUBLISHER	CATEGORY_TYPE	STOCK_ID	MESSAGE_ID	
CARTE	CORINT	AVENTURA	1	0	DETAILS
TEST	CORINT	TEST	2	2	DETAILS

© 2022 - LIBRARY MANAGEMENT BY ANDREEA DRAGHICI - [PRIVACY](#)

Utilizatorul cu rolul de administrator poate adauga o noua carte in lista de carti, toate field-urile trebuie completate, daca administratorul nu a introdus date valide sau a uitat sa completeze unul sau mai multe field-uri, atunci operatia va esua si va primi mesaje de validare, acesta va ramane pe aceeasi pagina curenta pentru a incerca iarasi sa adauge date despre carte. Fiecare carte este identificata dupa nume, editura, categorie. Pentru a se inregistra cartea in baza de date, administratorul trebuie sa apese pe butonul de adauga.

LIBRARYMANAGEMENT HOME BOOKS CONTACT ABOUT SALUT ADMIN@GMAIL.COM! :) Logout

ADAUGA O NOUA CARTE

NAME

PUBLISHER

CATEGORY_TYPE

STOCK_ID

MESSAGE_ID

Adauga

[INAPOI LA PAGINA CU CARTI](#)

© 2022 - LIBRARY MANAGEMENT BY ANDREEA DRAGHICI - [PRIVACY](#)

Utilizatorul cu rolul de administrator poate edita anumite informatii despre o carte. La final, pentru a salva continutul despre o anumita carte, acesta trebuie sa apese pe butonul 'salveaza' si baza de date va fi actualizata, in cazul in care doreste sa se intoarca inapoi pe pagina unde se afla lista cu carti, acesta poate accesa butonul 'inapoi la pagina cu carti'.

The screenshot shows the 'EDITEAZA CARTEA' (Edit Book) form. The form contains several input fields for editing book details: NAME (set to 'Carte'), PUBLISHER (set to 'Corint'), CATEGORY_TYPE (set to 'Aventura'), STOCK_ID (set to '1'), and MESSAGE_ID (set to '0'). Below these fields is a yellow 'Salveaza' (Save) button and a light blue 'INAPOI LA PAGINA CU CARTI' (Back to Book List) button. The page has a green header with navigation links (HOME, BOOKS, CONTACT, ABOUT) and a user status bar showing 'SALUT ADMIN@GMAIL.COM! :)' and a 'Logout' button. The footer includes copyright information: '© 2022 - LIBRARY MANAGEMENT BY ANDREEA DRAGHICI - PRIVACY'.

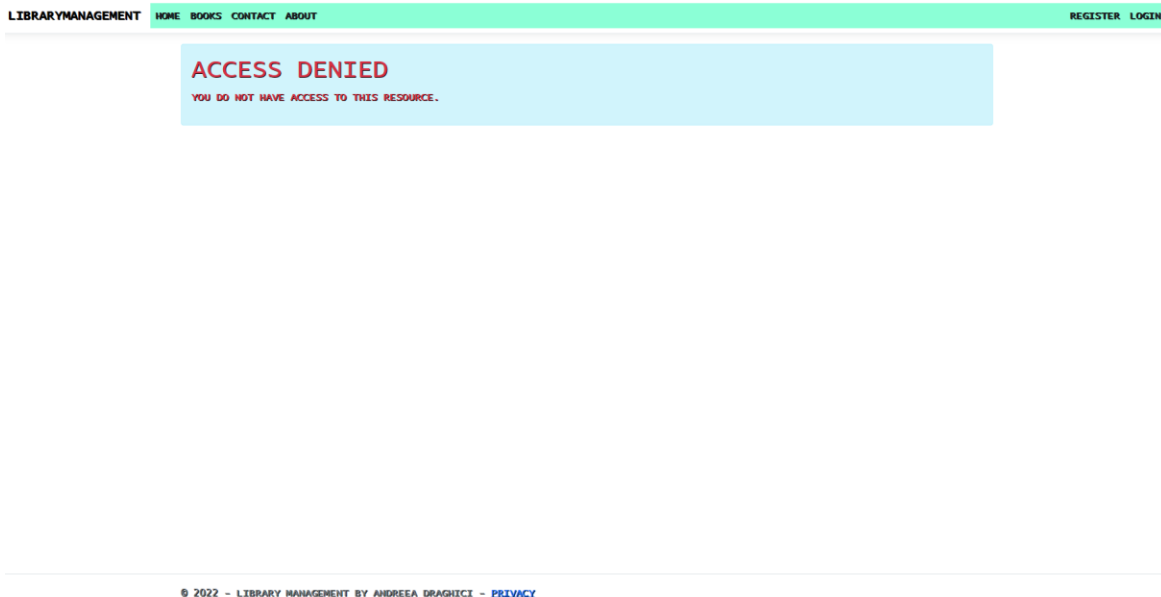
Daca se doreste ca o anumita carte sa fie exclusa din lista de carti, utilizatorul cu rolul de administrator poate realiza acest lucru alegand care carte sa fie stearsa din lista, apoi se apasa butonul 'sterge', iar cartea respectiva va fi exclusa din lista de carti si baza de date va fi actualizata, in cazul in care doreste sa se intoarca inapoi pe pagina unde se afla lista cu carti, acesta poate accesa butonul 'inapoi la pagina cu carti'.

The screenshot shows a confirmation dialog titled 'ARE YOU SURE YOU WANT TO DELETE THIS?'. It displays the details of the book to be deleted in a table:

NAME :	CARTE
PUBLISHER :	CORINT
CATEGORY_TYPE :	AVENTURA
STOCK_ID :	1
MESSAGE_ID :	0

Below the table are two buttons: a yellow 'Sterge' (Delete) button and a light blue 'INAPOI LA PAGINA CU CARTI' (Back to Book List) button. The page layout is consistent with the previous screenshot, including the green header and footer.

In cazul in care un utilizator cu rolul de user va accesa butonul de 'sterge' de pe pagina Books, acesta va fi redirectionat catre o pagina ce contine mesajul 'Access denied'.

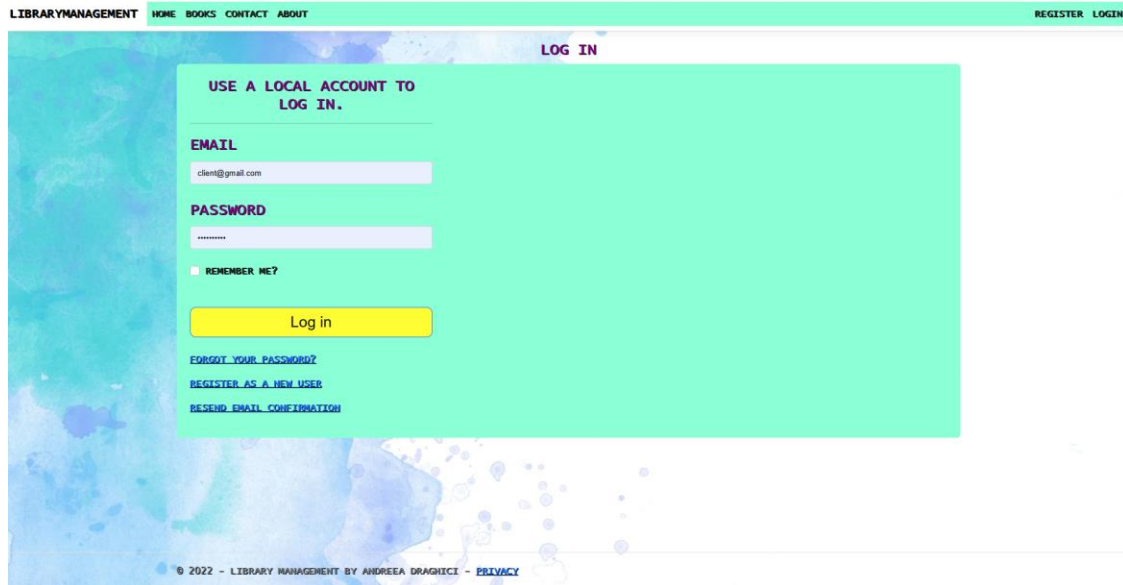


Utilizatorul poate trimite un ticket catre asistenta pentru utilizatori cu privire la orice problema intalnita in timpul aplicatiei. Dupa ce formularul va fi completat, se va apasa butonul ' trimite ' si pe mail se va primi formularul cu ticket-ul respectiv.

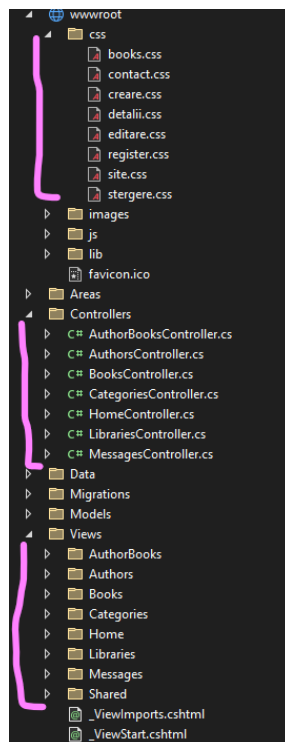
The screenshot displays a contact form titled 'CONTACTEAZA-NE' with the subtitle 'SWING BY FOR A CUP OF COFFEE, OR LEAVE US A MESSAGE:'. On the left side of the form is a large blue circular button with a white envelope icon and the text 'CONTACT US'. To the right of this button is a form with several input fields: 'NUME' (Name) with a text input field containing 'Numele tau..'; 'PRENUME' (First Name) with a text input field containing 'Prenumele tau..'; 'TARA' (Country) with a dropdown menu showing 'Australia'; and 'SUBIECT' (Subject) with a text input field containing 'Lasă un mesaj..'. At the bottom of the form is a green button labeled 'Trimite'.

Utilizatorul se poate deconecta apasand pe butonul de logout, apoi acesta este redirectionat catre pagina de login. Acesta este deconectat si nu va mai putea accesa toate paginile decat atunci cand se va conecta iar. Daca incearca sa acceseze o pagina si nu este conectat, acesta va fi redirectionat catre pagina de login, de specificat faptul ca in momentul cand utilizatorul este deconectat de la aplicatie,

acesta devine utilizator anonim si nu va mai putea accesa functionalitatea paginilor decat atunci cand se va conecta din nou.



Pentru partea de stilizare, am realizat codul in HTML si CSS si, de asemenea, pentru a putea fi accesata aceste paginile de catre utilizatori am create controllere care contin ruta acestora. Views contine fisierele HTML ale aplicatiei. Un fisier de tip view este un fisier cu extensia .cshtml care contine cod HTML si C#. Views contine cate un folder separat pentru fiecare controller al aplicatiei. De exemplu, toate fisierele .cshtml care vor fi randate de catre HomeController se gasesc in folderul View -> Home.

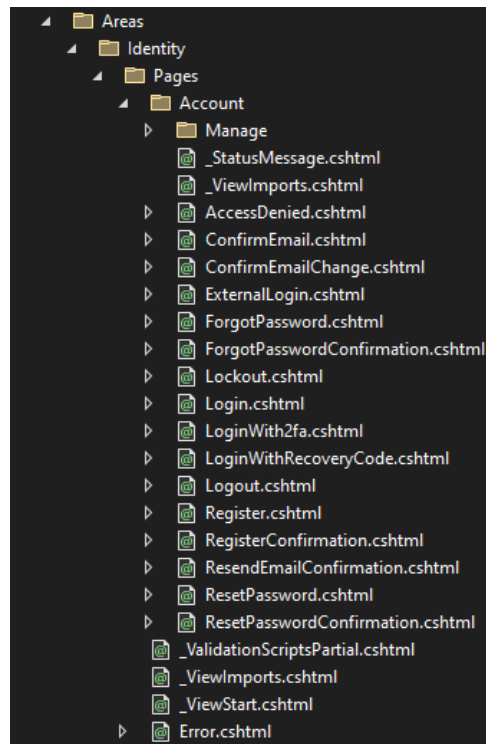


Folderul `wwwroot` in proiectul ASP.NET Core este tratat ca folder radacina (root). Fișierele statice pot sa fie stocate in orice folder din folderul radacina si accesate cu o cale relativa la acesta. Doar acele fișiere care se gasesc in folderul `wwwroot` pot sa fie servite (accesate) intr-o cerere http. Toate celelalte fișiere sunt blocate si nu pot fi accesate implicit (acest lucru se poate configura).

Este recomandat sa existe cate un folder separat pentru fiecare tip diferit de fișiere statice cum ar fi JavaScript, CSS, Imagini, librării client-side, etc.

Controlerele sunt creierul aplicației. Acestea procesează cererile primite, efectuează operațiuni pe datele modelului și selectează Vizualizări pentru a fi redade utilizatorului. Controlerele sunt stocate în folderul `Controlere` din rădăcina aplicației web. Acestea sunt practic clase C# ale căror metode publice sunt numite metode de acțiune.

Aspectele de securitate ar fi autentificarea si autorizarea pe roluri pentru utilizatorii ce folosesc aplicatia. Atat pentru autentificare, cat si pentru autorizare am folosit Identity.



Parola este stocata in baza de date folosind un algoritm de hashing inclus in ASP.NET Identity, iar aplicatia este protejata de SQL injection (SQLi) deoarece am folosit Entity Framework Core, acesta avand deja incluse mecanisme pentru prevenirea acestui tip de atac. Pe langa acest lucru, pentru pagina de Register si Login am adaugat si validari de date care vor specifica daca datele introduse respecta formatul precizat.

Am adaugat si partea de autorizare, astfel incat sa existe doua roluri pentru utilizatori. Exista doua tipuri de utilizatori, utilizatorii care au rol de administrator si utilizatorii care au rol de user. Diferenta

este ca utilizatorii cu rol de user nu vor putea avea acces la paginile destinate unui utilizator cu rol de administrator. Exista si tipul de utilizator anonim, acesta va fi nevoit sa se conecteze la un cont valid pentru a accesa paginile.

Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	PasswordHash	SecurityStamp	ConcurrencyStamp	PhoneNumber	PhoneNumberConfirmed	TwoFactorEnabled	LockoutEnd
54f-812f37b4da6	admin@gmail.com	ADMIN@GMAIL.COM	admin@gmail.com	ADMIN@GMAIL.COM	True	AQAAAAIAA...	2W3EY4VLU3YP...	6684eed3-f451...	NULL	False	False	NULL
756289b0-2b08...	client@gmail.com	CLIENT@GMAIL.COM	client@gmail.com	CLIENT@GMAIL.COM	True	AQAAAAIAA...	GBK74UKAWD...	5ceea1e7-13dc...	NULL	False	False	NULL

Tipul de utilizator anonim va fi nevoit sa se conecteze la un cont valid pentru a accesa paginile aplicatiei.

Id	Name	NormalizedName	ConcurrencyStamp
1	admin	ADMIN	NULL
2	client	CLIENT	NULL

UserId	RoleId
33bf3e8c-45b2...	1
756289b0-2b08...	2

Pentru a testa aplicatia, am utilizat testarea functionala, astfel am testat fiecare flow dupa implementarea sa.

Cateva exemple din test case-urile efectuate sunt urmatoarele:

➤ Test Case 1:

Title	Butonul de Login
Steps	1. Deschide aplicatia 2. In pagina de Login se introduc: Email-ul si parola 3. Se apasa butonul de login
Expected result	Dupa ce se apasa butonul de login, apare un popup care contine mesajul "Login successfully" ceea ce inseamna ca butonul respectiv este activ.

➤ Test Case 2:

Title	Butonul de Register
Steps	1. Deschide aplicatia 2. In pagina de Register se introduc: Email-ul, parola si iarasi parola pentru a fi confirmata 3. Se apasa butonul de register
Expected result	Dupa ce se apasa butonul de register, utilizatorul va fi inregistrat si conectat automat in aplicatie, iar credentialele acestuia vor fi salvate si stocate in baza de date.

➤ **Test Case 3:**

Title	Pagina de Home
Steps	1. Deschide aplicatia 2. Se introduce datele pentru conectare in aplicatie 3. Validam aceste date prin apasarea butonului de login 4. Apasam pe butonul de OK in popup-ul care ne confirma ca datele sunt valide
Expected result	Utilizatorul este redirectionat catre pagina de Home.

➤ **Test Case 4:**

Title	Introducere date invalide
Steps	1. Deschide aplicatia 2. Se introduce urmatoarele date invalide pentru conectare in aplicatie Email: draghici108@gmail.co Password: 1234 3. Apasam butonul login
Expected result	La introducerea datelor invalide, mesajul de form 'invalid login attempt' va aparea deasupra campului email pe font de culoare rosie.

Concluzii



- Aplicatia Library Management este usor de utilizat deoarece toate paginile pot fi accesate din bara de navigatie, ele fiind impartite pe sectiuni: Home, About, Contact, Books, Register, Login. Functionalitatea fiecărei pagini iarasi este usor de invatat si utilizat, deoarece pentru paginile ce contin field-uri care trebuie completate, am utilizat mesaje de validare astfel incat utilizatorul sa stie cum sa completeze aceste campuri cu date si informatii corecte.

- **Contrastul dintre text si background-ul paginilor face ca text-ul sa fie usor de inteles si descifrat, textul fiind vizibil pentru utilizatori. Paginile au un layout destul de friendly, ceea ce inseamna ca utilizatorul se poate concentra pe functionalitati fara sa se simta incarcat de complexitatea paginilor sau a aplicatiei.**
- **Aplicatia este accesibila indiferent de echipamentul hardware sau denumirea browser-ului pe care utilizatorul le are la indemana, astfel daca tehnologia evolueaza, continutul aplicatiei ramane accesibil.**