UNIVERSITATEA “LUCIAN BLAGA” DIN SIBIU

FACULTATEA DE INGINERIE

DEPARTAMENTUL DE CALCULATOARE ŞI INGINERIE ELECTRICĂ

**PROIECT DE DIPLOMĂ**

Îndrumător: Dr. ing. Ioana COFARU Absolvent:

Migea Andreea-Maria

Specializarea Calculatoare

* Sibiu, 2023 –

UNIVERSITATEA “LUCIAN BLAGA” DIN SIBIU

FACULTATEA DE INGINERIE

DEPARTAMENTUL DE CALCULATOARE ŞI INGINERIE ELECTRICĂ

# APLICAȚIE WEB ADMINISTRARE PENSIUNE

Îndrumător: Dr. ing. Ioana COFARU Absolvent:

Migea Andreea-Maria

Specializarea Calculatoare

**Cuprins**

[APLICAȚIE WEB ADMINISTRARE PENSIUNE 2](#_Toc134904628)

[**1.** **Prezentare 2-3pg** 5](#_Toc134904629)

[**1.1** **Noțiuni introductive** 6](#_Toc134904630)

[**1.2** **Motivația alegerii temei** 6](#_Toc134904631)

[**1.3** **Scop și obiective** 7](#_Toc134904632)

[**1.4** **Realizări similare acestei teme** 7](#_Toc134904633)

[**2** **Teorie și tehnologii utilizate** 7](#_Toc134904634)

[**2.1** **Baze de date** 7](#_Toc134904635)

[**2.1.1** **SQL** 8](#_Toc134904636)

[**2.1.2** **SQL Server Management Studio** 10](#_Toc134904637)

[**2.2** **ASP.NET** 10](#_Toc134904638)

[**2.2.1** **Web Forms** 11](#_Toc134904639)

[**2.2.2** **Diferența dintre MVC și Web Forms** 12](#_Toc134904640)

[**2.2.3** **Limbajul C#** 13](#_Toc134904641)

[**2.2.4** **Visual Studio** 14](#_Toc134904642)

[**2.3** **HTML** 15](#_Toc134904643)

[**2.4** **Bootstrap** 17](#_Toc134904644)

[**2.5** **CSS** 17](#_Toc134904645)

[**2.6** **JavaScript** 18](#_Toc134904646)

[**3** **Realizarea aplicației 45-55pg** 19](#_Toc134904647)

[**3.1** **Mediul de dezvoltare** 19](#_Toc134904648)

[**3.2** **Diagrama UML a bazei de date** 20](#_Toc134904649)

[**3.3** **Arhitectura bazei de date** 21](#_Toc134904650)

[**3.4** **Design-ul aplicației** 22](#_Toc134904651)

[**3.5** **Dezvoltarea propriu-zisă a aplicației** 23](#_Toc134904652)

[**3.5.1** **Funcționalitatea aplicției** 23](#_Toc134904653)

[**3.5.2** **Autentificare Admin** 23](#_Toc134904654)

[**3.5.2.1** **Gestionarea rezervarilor de către admin** 23](#_Toc134904655)

[**3.5.2.2** **Gestionarea utilizatorilor** 23](#_Toc134904656)

[**3.5.2.3** **Gestionarea categoriilor de camere** 23](#_Toc134904657)

[**3.5.3** **Autentificare Utilizator** 23](#_Toc134904658)

[**3.5.3.1** **Procesul de rezervare de către utilizator** 23](#_Toc134904659)

[**4** **Concluzii și dezvoltări ulterioare** 24](#_Toc134904660)

[**4.1** **Concluzii** 24](#_Toc134904661)

[**4.2** **Dezvoltări ulterioare** 24](#_Toc134904662)

[**5** **Bibliografie** 25](#_Toc134904663)

1. **Prezentare 2-3pg**
   1. **Noțiuni introductive**

(despre calatorii, importanta calatoriilor, a pensiunilor(locul unde te cazezi))

* 1. **Motivația alegerii temei**

Motivația pe care am avut-o în momentul în care m-am decis să fac o aplicație web pentru administrarea unei pensiuni a fost în urma interacțiunilor din călătoriile mele, astfel am dorit o facilitare a procesului de cazare, cu o interacțiune cât mai puțină cu proprietarul, o transparență în ceea ce privește disponibilitatea în așa fel încât călătorul care dorește să fie cazat să iși rezerve singur camera. Utilizând această metodă interactivă prin care utilizatorii își pot rezerva un loc de cazare într-o pensiune, generează automat un nivel ridicat de rapiditate când vine vorba de timpul necesar, atât al clientului, cât și al celui care oferă camera pentru cazare, este redus de la un nivel destul de înalt, cauzat de nevoia de a introduce manual datele clientului, fie pe o foaie, sau un dosar, la un nivel mult mai scăzut.

Am călătorit destul de mult, în special prin țara noastră și am observat mari deficiențe în ceea ce privește digitalizarea acestui proces de rezervare a unui sejur. În ultimii ani totuși se observa o creștere a celor care s-au axat mai mult pe partea aceasta de facilitare a demersului în ceea ce privește rezervările, însa încă mulți nu cred ca este un lucru benefic.

Eu consider ca ideea de creare a unei aplicații web pentru o pensiune este utilă nu doar în scop informativ pentru clienți, ci mai ales în mod administrativ pentru proprietar. În acest fel, proprietarul are o organizare mult mai bună a cazărilor/camerelor ceea ce duce și la o viziune mai bună asupra profitului, dar asta nu este singurul lucru benefic. Timpul celui care deține proprietatea dată spre cazare este de asemenea redus semnificativ, deoarece procesul de rezervare se rezumă doar la a elimina cererile de cazare neconforme, sau neplatite, de a asigura un bun sejur pentru persoanele care își fac prezența în urma rezervării, și de a gestiona camerele pentru o fluidizare cât mai eficientă a ideii de „book in/book out”. Însă când vine vorba de sejur deja plătit nu mai are această influență legată de gestionarea și eliminarea cazărilor neplătite, singurul lucru pe care trebuie să-l facă este reprezentat de primitea clientului într-un spațiu exact așa cum el a fost descris în prezentarea camerei sau a pensiunii.

Așadar, în opinia mea aplicația web de administrare a pensiunii este o unealtă de care orice prorietar nu ar trebuii să se lipseasca, fiind un lucru benefic atât pentru el cât și pentru creșterea satisfacției utilizatorilor.

* 1. **Scop și obiective**

Scopul proiectului meu de diplomă este acela de a dezovlta o aplicație de administrare a unei pensiuni, aplicație care este una web cu web forms, care să permită proprietarilor să gestioneze eficient rezervarile, evidența si alte activitați administrative legate de afacerea lor. În plus, aplicația se dorește să aiba o interfața atractivă si intuitivă, usor de folosit, care să permită clientelei să faca rezervări.

Principalele obiective pe care doresc să le indeplinesc în realizarea proiectului meu de diploma sunt:

* Login
* Register
* G
* A
* Dsa
* D
* as
  1. **Realizări similare acestei teme**

Câteva exemple de aplicații similare de pe piața:

aici

Propunerea mea ar aduce o valoare adaugată și ar fi mai atractivă pentru proprietarii de pensiune li clienți, deoarece dispune de o interfată prietenoasa și foarte intuitivă.

De asemenea chiar dacă există multe alte exemple asemanătoare, aplicația aceasta oferă multe beneficii înglobate, tot ce ai nevoie în momentul în care cauți o cazare undeva, prin intermediul aplicației poți să rezervi dintr-un singur loc cazare, masă și toate celalte beneficii pe care le oferă pensiunea fără a fi nevoit să cauți în alta parte informații.

1. **Teorie și tehnologii utilizate**
   1. **Baze de date**

Bazele de date reprezintă niște colecții foarte bine organizate de date care sunt depozitate de regulă într-un calculator. În ceea ce privește structurarea acestor date, ele se regăsesc grupate în tabele(linii si coloane), fapt ce ajută foarte mult atunci când se dorește să se modifice anumite lucruri în baza de date.

Astfel, formatul electronic ne permite un acces eficient la date, pentru a putea defini, gestiona tabelele în funcție de preferințe. La aceasta gestiune ne ajută limbajul de interogare (SQL), limbaj care are un rol important pentru scriere, dar totodata și interogarea datelor.

În ceea ce priveste evoluția, bazele de date la început erau simple, cele cu structura ierahica, dar nu permitea flexibilitatea pe care o permit în zilele de astazi. Mulți oameni confundă foile de calcul cu bazele de date, dar există diferențe majore în ceea ce le privește pe cele doua: accesul la datele din tabele este un exemplu foarte important, la foile de calcul este doar un singur utilizator care poate gestiona si interpreta, spre deosebire de bazele de date unde accesul este permis mai multor persoane.

Un alt factor decisiv în ceea ce privește diferența dintre cele doua este cantitatea care poate să fie stocată în bazele de date este semnificativ mai mare decât în foile de calcul. Există 6 tipuri a acestor colecții de date: relaționale, orientate pe obiecte, distribuite, grafice, NoSQL, OLTP fiecare având caracteristici diferite, astfel în funcție de nevoie se alege un tip potrivit pentru a stoca datele. [1]

* + 1. **SQL**

Structured Query Language este denumirea completa pentru SQL și reprezintă un limbaj de programare utilizat în gestionarea bazelor de date relaționale care dețin seturi enorme de date. Acesta este folosit pentru a prelucra date, insera in tabele, ștergere sau editare a înregistrarilor din tabele. Chiar și crearea de baze de date sau de tabele sunt procese care se pot îndeplini cu ajutorul SQL, de asemenea și setarea unor permisiuni pentru vizualizare și permiterea aceeselor de catre alți utilizatori. [2]

Funcțiile care pot fi realizate cu ajutorul SQL sunt:

* Creare baza de date
* Ștergere baza de date
* Creare tabel în baza de date
* Ștergere tabel
* Adăugare/ștergere/editare a coloanei intr-un tabel
* Inserare într-un tabel
* Editare înregistrari într-un tabel
* Ștergere dintr-un tabel

Un avantaj major al SQL este acela că permite accesarea și modificarea informațiilor din tabele prin interogare. Deși majoritatea îl foloseste pentru gestiune, mulți nu știu și faptul că SQL poate efectua și analize detaliate și generare de rapoarte minuțioase în ceea ce privește datele structurate. Accesul rapid la informații este o caracteristică importanta pe care utilizatorii care doresc să iși gestioneze un set de date structurat în baze de date se pot baza, deoarece informtațiile sunt la îndemana utilizatorului și oferă clientului o viziune ușor de înteles și clară. Odata cu asta crește și productivitatea prin atragerea clienților care doresc acces rapid la informații, la datele furnizate și totodata și securitate de nivel înalt.

Comenzile care pot fi utile in limbajul SQL, pentru baze de date sunt:

* CREATE DATABASE 🡪 această comandă este utilizată pentru inițierea și creearea unei baze de date
* CREATE TABLE 🡪 această comandă este folosită în mediul SQL pentru a realiza un tabel în interiorul bazei de date tocmai creeate
* ALTER 🡪 comanda aceasta este utilizată pentru a realiza modificări diverse asupra tabelei disponibile în baza de date
* DROP 🡪 reprezintă o comandă destructivă, rolul ei fiind de a elimina un tabel creeat din baza de date. Această comandă, fiind ireversibilă, poate duce la pierderea neintenționată a datelor

Comenzile utile pentru gestiunea inregistrarilosunt:

* SELECT 🡪 comanda „Select” reprezintă una dintre cele mai utilizate din SQL, rolul ei fiind de a extrage anumite date din baza de date, fiind utilizată mereu impreună cu anumiți parametrii pentru a filtra elementele dorite
* INSERT 🡪 această comandă servește la inserarea datelor în interiorul tabelei specificate în corpul comenzii
* DELETE 🡪 idem cu comanda „DELETE TABLE”, această comandă reprezintă o comandă destructivă, rolul ei fiind de a șterge o anumită linie de date din tabela specificată
* UPDATE 🡪 utilizând această metodă se pot executa operațiuni de modificare a datelor din tabela bazei de date, datele fiind modificate în funcție de anumiți parametrii specificați.
  + 1. **SQL Server Management Studio**

SQL Server Management Studio este un instrument dezvoltat de către cei de la Microsoft pentru a facilita accesul utilizatorilor oferind o interfată grafică ușor de utilizat împreuna cu un set de instrumente care de asemenea ușureaza operațile asupra bazelor de date.

Interfața grafica este foarte ușor de folosit, chiar dacă nu se cunoaște în totalitate limbajul SQL în SSMS se pot efectua operații de gestionare cum ar fi inserare, editare, ștergere. Tot de aici, se poate permite sau nu accesul clienților la anumite date de către utilizator sau pe lângă funcțiile predefinite se pot instala alte tool-uri care ușureaza și mai mult munca utilizatorilor.

Panoul principal conține o structură arboreșcentă al bazelor de date și permite vizualizarea tabelelor și a înregistrarilor. Se oferă și informații despre conectare sau statusul bazei de date dacă este sau nu conectată și totodata regasim și un panou unde se scriu interogarile SQL, comenzile amintite mai sus de creare, ștergere, editare și multe altele.

SSMS oferă un mediu propice pentru a lucra simultan cu mai multe baze de date. Se respectă modelul relațional al bazelor de date, organizare sub formă de tabele care conțin înregistrari. Acestea din urmă pot fi:

• tipuri de date întregi

• caractere

• data calendaristică și ora

• data calendaristică

* + 1. **Diagrama UML a bazei de date**

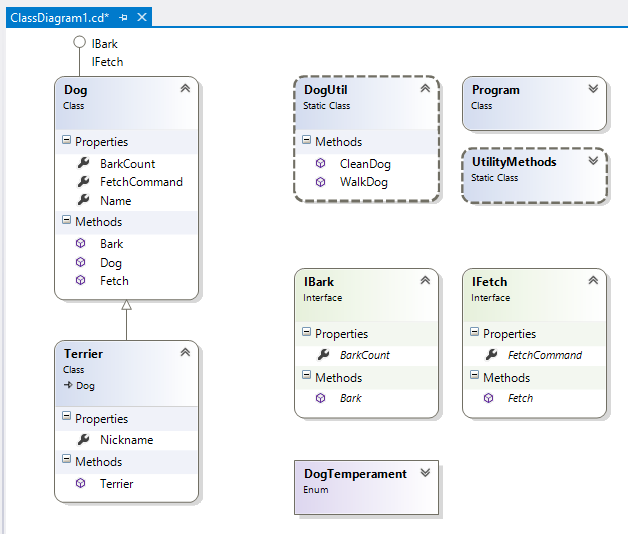
Diagrama UML reprezintă un limbaj standardizat de tip grafic care este utilizat în modelarea și în reprezentarea diferitelor aspecte grafice ale unităților software. Acest tip de limbaj a fost dezvoltat cu ideea de a permite programatorilor să recunoască și să partajeze într-un mod mult mai ușor structura , interactiunile și funcționalitatea acestor unități software.

Aceste diagrame oferă un tip de abordare vizuala diferit, în sensul că descrie componentele unei unități software prin reprezentarea lor în elemente vizuale ale claselor, obiectelor și a relaților dintre acestea. Modul de utilizare a diagramei UML poate fi împărțit în diferite etape, acestea fiind definite de procesul de dezvoltare soft. Aceste etape pornesc de la analiza și proiectarea incipientă până la implementarea finală a sistemului. Astfel se creează un mijloc de transfer de informații între părțile implicate în dezvolare, într-un mod clar și ușor de înțeles.

Modul de diferențiere a acestor diagrame este definit utilizând o serie de notații, simboluri care sunt standaardizate pentru o ușoară reprezentare a dei;feritelor relații din sistem într-un mod consistent.

Simbolurile utilizare includ:

* Clase
* Obiecte
* Moșteniri
* Evenimente
* Stări
* Asociații



Figură 1 Exemplu Diagramă UML [3]

* 1. **ASP.NET**

ASP.NET, o platformă ajutătoare pentru dezvoltarea de aplicații web s-a dezvoltat foarte mult de-a lungul timpului devenind foarte cunoscută în domeniul IT. Acesta utilizează un ciclu de execuție pentru fiecare pagină, care este compus din faze diferite și anume inițializare, instanțiere, restabilirea și ce e mai important: menținerea stării. În acest ciclu de viață al paginii utilizatorul poate adăuga cod astfel încat să personalizeze paginile după bunul plac sau dezvoltatorul modifică după cerintele clientului, asfet încat el să fie cât mai aproape de dorințele lui.

Flexibilitatea acestei platforme impresioneaza orice dezvoltator dornic de a crea pagini web. Aceasta permite crearea și adăugarea de controlere prin instrumente specifice care ajuta foarte mult utilizatorul, cum ar fi Visual Studio, mediu de dezvoltare care ajută și mai mult un dezvoltator prin paleta de instrumente numeroase pe care le deține în acest scop.

|  |
| --- |
|  |
| Figură 2 Componentele ASP.NET [4] |

* + 1. **Web Forms**

Web Forms este considerat cel mai vechi model ASP.NET, el are pagini web scrie ca o combinație de HTML, împreuna cu un control de tip server, și de asemenea codul de server. Compilarea și execuția Web Froms se face pe server și genereaza HTML prin intermediul căruia putem avea o afișare a conținului paginii web.

Astfel, penrtu a construi pagini web complexe și bine dezvoltate, formularele web dețin sute de controale web și componente specifice orientate spre utilizator care asigură un mediu propice pentru crearea site-urilor web. Pentru a putea dezvolta, Microsoft ne propune mediul de dezvoltare Visual Studio amintit mai sus, intâlnit atât în varianta gratis cat și cu plata.

Specific lucrului cu web forms este faptul că orice control de server ar trebui să apară între tagurile <form></form>. De asemenea tag-ul de deschidere<form> trebuie să aibă runat=”server” pentru a funcționa în parametrii optimi și a crea formularul web dorit. Formularele web din ASP.NET oferă o gamă largă de controale ale serverului, care sunt obiecte ce rulează când pagina este solicitată și genereaza marcaje în browser. Aceste controale includ atât elemente similare cu cele HTML, precum butoane și casete de text, dar și controale specializate pentru conectarea la surse de date și afișarea datelor.

Controalele cele mai des intâlnite în web forms sunt urmatoarele:

* Label - pentru a afisa anumit text
* TextBox – creaza un text de intrare in formular
* Button – se creaza efectiv un buton
* Hyperlink – creaza un hyperlink, în momentul în care se dă click se deschide pagina respectivă
* DropDownList – se creaza o listă din care se pot selecta valori predefinite
* ListBox – o listă
* DataGrid – un tabel unde se afișează date
* DataList – o listă unde sunt afișate datele, difera de DataGrid pentru ca acesta nu este un tabel
* CheckBox – se creaza o casuță care permite ceva cand este selectată
* Image – se pot adăuga imagini

În plus, ASP.NET permite crearea de pagini master, ceea ce permite definirea unei structuri și comportament standard pentru toate paginile dintr-o aplicație web. Când utilizatorii solicită paginile de conținut, acestea se combina cu pagina master pentru a produce o ieșire care combină structura paginii master cu conținutul specific din pagina de conținut.

* + 1. **Diferența dintre MVC și Web Forms**

ASP.NET are două versiuni principale: ASP.NET Web Forms și ASP.NET MVC. Ambele ediții oferă abordări diferite ale dezvoltării aplicațiilor web:

* Formularele Web ASP.NET sunt mai ușor de utilizat, oferă o abordare simplă și mai rapidă a dezvoltării aplicațiilor web. Utilizează o abordare bazată pe evenimente, în care evenimentele sunt gestionate de server și pot fi legate de elementele UI
* ASP.NET MVC este o abordare mai sofisticată și mai puternică a dezvoltării de aplicații web. Utilizează modelul arhitectural Model-View-Controller, unde modelul reprezintă datele, controlerul se ocupă de evenimente și logica de afaceri, iar vizualizarea afișează datele și oferă interacțiunea utilizatorului.

|  |  |
| --- | --- |
| ASP.NET | ASP .NET MVC |
| * Controlere de server incorporate * Suport viewstate * Programare bazată pe evenimente * Dezvoltare rapidă a aplicațiilor | * Dezvoltare bazată pe testare și reuzabilitate * Performanță crescută * Control crescut asupra HTML * Dispunere la extindere |

Figură 3 Diferența dintre Web Forms și MVC

* + 1. **Limbajul C#**

C# reprezinta limbajul de programare care se bucura de un real succes de când s-a lansat. Acesta a fost lansat în jurul anilor 2000 și vine în ajutorul programatorilor care realizeaza aplicații desktop, sau web, dar și windows sau chiar IOS și Android. Deși a fost creat de către cei de la Mictosoft, poate fi folosit și pe sisteme cum ar fi Macintosh sau Linux.

C# este asemanator cu limbajele de programare C sau C++, dar are și caracteristici comune cu Java, cu toate acestea C# are foarte multe avantaje față de celealte limbaje amintite fiind unul dintre cele mai appreciate și folosite limbaje de pe piața în momentul actual. Legatura limbajului C# cu C provine de la faptul că C# a preluat de la C multe din cuvintele cheie ale metodelor și ale opratorilor, în general ceea ce privește sintaxa de scriere. Dacă vorbim despre C++, limbajul C# moștenește modelul de orientare pe obiecte, crearea de obiecte și mostenirea acestora. Limbajul este unul de actualitate și pentru fapul că ajută la crearea de jocuri video și jocuri VR. Dezvoltarea jocurilor se face printr-un framework special secific C# numit Unity care dispune de efecte grafice, de fotografie pentru programatori astfel încât, cu ajutorul realității virtuale 3D se pot realiza priecte execpționale. Alt beneficiu pe care îl are C# constă în deținerea de instrumente specific pentru o usoară dezvoltare a unor aplicații web, prin intermediul ASP.NET.

În ceea ce privește dificultatea, acest limbaj nu se încadreaza în categoria celor dificile, este unul ușor de folosit de aceea și procentul celor care îl folosesc este unul foarte mare, acesta fiind unul din motive, dar nu doar acesta. Versatilitatea și mobilitatea sunt alte avantaje care fac acest limbaj placut programatorilor. Oricine poate învata cu ușurință limbajul C# pe cont propriu, cu ajutorul carților sau a diferitelor tutoriale de programare, începand de la sintaxa și mai apoi crescând dificultatea tratează anumite probleme pe diferite grade de dificultate. Important de reținut este și faptul că mediul de dezvoltare unde se poate programa in limbaj C# este Visual Studio, acesta fiind cel mai versatil și ușor de folosit.

A picture containing text, screenshot, line, font

Description automatically generated

Figură 4 Proprietăți limbaj C# [5]

* + 1. **Visual Studio**

Visual Studio este un mediu unde se pot dezvolta aplicații, a fost dezvoltat de catre cei de la Microsoft. Multitudinea de limbaje de programare în care se pot crea aplicații este impresionantă, și anume C, C++, C#, Python și multe altele.

Acest instrument de dezvoltare software este folositor pentru a putea implementa aplicații web, mobile, android, aplicații desktop sau chiar jocuri. Versatilitatea acestui mediu de dezvoltare în ceea ce privește caracteristicile și instrumentele ajutatoare programatorului îi face pe aceștia să apreceieze foarte mult Visual Studio și să îl foloseasca într-un numar mare. IntelliSense este unul din motivele pentru care dezvoltatorii aleg să îl folosească pentru că acesta completează (dă sugestii de auto completare) în ceea ce privește informațiile parametrilor, lista membrilor, informații despre variabile sau chiar sugereaza dupa ce s-au tastat câteva litere ce anume ar putea folosi programatorul.

Un alt motiv pentru care Visual Studio este atat de apreciat în rândul dezvoltatorilor software este debugging-ul, un instrument prin care se poate face depanare pe un anume proiect. Astfel, sunt aflate foarte ușor erorile de cod, chiar prin poziționarea unor puncte prin care poți întrerupe într-o anumita porîiune de cod pentru a fi mai exacta înțelegerea sau găsirea erorilor. Nu în ultimul rând Visual Studio are o interfața foarte ușor de folosit, este ușor de intuit cum anume se folosește, cum se creaza proiecte, unde se gasesc anumite funcții care ajută programatorul chiar și numerotarea liniilor de cod sau afișarea tututor directoarelor împreuna cu fișierele corespunzatoare în partea dreaptă. [6]

* 1. **HTML**

HTML, abreviere de la HyperText Markup Language este un limbaj prin care se face o legatură mai ușoară între un browser și site-ul web, oferind ajutor în ceea ce privește afișarea informațiilor pe site, improtant de reținut că este posibilă crearea de pagini web dar nu și gestionarea design-ului. Dacă vorbim despre structura unei pagini html discutăm despre elemente de bază, cum ar fi:

* Tag-uri
* Elemente
* Atribute

Tag-urile sunt acele informații care apar în fișierul html între simbolurile “<” și “>” și unele tag-uri au o parte de deschidere si una de închidere. Cel de deschidere are numele între simbolurile amintite mai devreme și cel de închidere conține semnul “/” la începutul numelui respectivului tag. Cele mai utilizate tag-uri sunt:

* <head> </head> = acest tag este folosit în paginile html și este indispensabil, conține informații despre pagină spre exemplu importările CSS sau formatarea începutului paginii web
* <body></body = tagul de body este de asemenea unul care nu lipsește din documentul html pentru ca aici se vor plasa informațiile despre conținutul paginii web, el poate conține si alte taguri fie complexe cu deschidere și închidere sau simple pentru a crea o organizare mai bună a paginii
* <div></div> = tagul div, unul din cele mai importante tag-uri în ceea ce privește organizarea codului, el permite gruparea in blocuri logice urmand a fi stilizat cu CSS
* <form></form> = form este un tag care crează un formular prin care se trimit informații spre pagina web
* <ul></ul> = tag-ul ul creaza lista neordonata cu puncte, rezultatul poate fi regasit in pagina web dorită

Așa cum există tag-urile pereche, regăsim și tag-uri simple care nu dețin o pereche. Acestea sunt fără conținut, mai sunt întâlnite și sub denumirea de elemente vide, închiderea se regasește în tag. Un exemplu pentru tag simplu ar fi <hr /> sau <br />.

Important pentru a folosi html trebuie ca tot codul să se regăsească între tagul <html> </html> astfel nu se va putea recunoaște codul pentru construcția site-ului.

Elementele sunt informațiile regăsite între tag-uri. Acestea pot fi denumiri de titluri, de tabele, etichete care se doresc a fi afișate pe pagina web, spre exemplu <h1> titlul </h2>

Atributele reprezintă calea prin care elementele sunt modificate in html, de asemenea se ofera prin intermediul lor si informații suplimentare desptre un tag oarecare folosit în document. În general dac aun tag nu are niciun atribut, browser-ul va lua valorile implicite pe care acel tag le are, dar în situația în care se dorește personalizarea paginii prin adaugarea de atribute un exemplu în acest sens ar fi urmatorul<body bgcolor=”blue”> prin care se atribuie culoarea de fundal cu ajutorul lui bgcolor.

A screenshot of a computer code

Description automatically generated with low confidence

Figură 5 Structura pagina HTML [7]

Așa cum se observă și în figura 4, un document HTML are o structură bine definită și anume o zonă de antet, una de corp ele fiind încorporate între tag-urile de <html></html>. Programatorii de obicei respectă această structură, astfel în zona de antet regasită între <head> și </head> se adauga tituluri pentru pagina sau declarații de fisiere CSS folosite pentru design, spre deosebire de corp unde regăsim tot ce se dorește a se afla in conținutul paginii web.

* 1. **Bootstrap**

Site-urile dezvoltate de către programatori nu trebuie sa conțină doar funționalitatea pe care o au, desi ele funcționează și astfel este foarte importanta si partea vizuala, front-endul, imaginea de ansamblu pe care pagina web o are. În acest sens în jurul anilor 2011 a fost lansat un framework de front-end și anume Bootstrap care avea ca sens dezvoltarea aplicațiilor web pe partea de interfață. Creat inițial de Twitter, acesta este unul dintre cele mai populare și utilizate framework-uri de acest gen. Bootstrap folosește HTML, CSS și JavaScript pentru a permite dezvoltatorilor să creeze site-uri web ușor de utilizat, rapide și bine structurate.

Bootstrap poate fi utilizat într-o varietate de moduri, fie că dezvoltatorii doresc să înceapă de la zero și să construiască un site web personalizat sau să utilizeze una dintre multe teme și șabloane disponibile în Bootstrap. Acest lucru face ca Bootstrap să fie o opțiune populară pentru dezvoltatorii web de toate nivelele de experiență. Bootstrap vine cu o varietate de caracteristici și componente care îi permit să fie utilizat pentru crearea unor site-uri web complet funcționale și personalizate. Unele dintre aceste caracteristici și componente includ:

* Grid: Bootstrap folosește un sistem de grilă flexibil care permite dezvoltatorilor să creeze site-uri web responsive care se adaptează la diferite dimensiuni ale ecranului.
* Componente UI: Bootstrap vine cu o serie de componente UI, cum ar fi butoane, meniuri de navigare, bare de progres, carusele și multe altele.
* Modale: Bootstrap include suport pentru modale, care permit dezvoltatorilor să creeze ferestre pop-up cu conținut personalizat.
* Formulare: Bootstrap vine cu stiluri predefinite pentru elementele de formular, cum ar fi casetele de text, casetele de selectare și butoanele radio și de verificare.
* Suport pentru CSS personalizat: Bootstrap este ușor de personalizat și permite dezvoltatorilor să își adapteze stilurile CSS pentru a se potrivi nevoilor lor.
  1. **CSS**

CSS cunoscut sub denumirea întreagă Cascading Style Sheets reprezintă limbajul de stilizare folosit pentru a descrie aspectul vizual astfel încat prezentarea paginilor web să fie mult mai ușor de înteles de către utilizatori. Ajutându-se de CSS, dezvoltatorii web pot formata elemente HTML: text,dimensiune și culoare a textulu,i imagini sau scalarea acestora după dorințele fiecăruia, schimbat culoare de fundal sau chiar adaugare de poze în schimbul culorii și multe alte elemente de design în așa fel încat aspectul vizual va fi mai atractiv experiența de utilizare mai ușoara si placută celor care vor folosi aplicația. Un avantaj major al dezvoltării design-ului cu ajutorul CSS-ului este permiterea separarii conținutului de prezentare.

Cu ajutorul utilizării fișierelor CSS separate, dezvoltatorii sunt capabili de a crea un aspect vizual facil de modificat în întreaga pagina web, fără să fie nevoiți să modifice fiecare pagină în parte. De aici rezultă că întreținerea site-urilor web e mai ușoară și e foarte eficientă. CSS-ul se bucură de o sintaxă simplă, intuitivă, iar dezvoltatorii sunt capabili sa defineasca stilul elementelor HTML cu ajutorul selecțiilor de clasă, ID-uri și de asemenea și alte metode pentru o mai bună identificare a elementelor. Stilurile CSS ajută dezvoltatorii web în controlul culorilor, fonturilor, a dimensiunilor și multe alte proprietăți în ceea ce privește elementele HTML. Cu ajutorul CSS-ului, programatorilor li se ușurează munca în ceea ce privește efectele de animație și tranzițiile pentru a adăuga un nivel suplimentar de dinamism și interactivitate la site-unuri web.

\*rescris\*CSS-ul a evoluat și pe masură ce au trecut anii se bucură de multe caracteristici noi, inclusiv suport pentru design responsive și pentru flexbox și grid layout-uri. Cu suportul continuu pentru noile caracteristici și tehnologii web, CSS-ul rămâne unul dintre cele mai importante limbaje de programare pentru dezvoltatorii web.

Așadar, CSS este un limbaj de stilizare fundamental pentru dezvoltarea de site-uri web și aplicații web. Prin separarea conținutului de prezentare și utilizarea unei sintaxe intuitive, CSS permite dezvoltatorilor web să creeze aspecte vizuale consistente și ușor de întreținut, iar prin adăugarea de efecte de animație și interactivitate, CSS poate transforma site-urile web în experiențe atractive și dinamice pentru utilizatori.

* 1. **JavaScript**

\*rescris\*JavaScript este un limbaj de programare de nivel înalt, care este utilizat în principal pentru a crea pagini web interactive și dinamice. Este unul dintre cele trei principale tehnologii utilizate în dezvoltarea paginilor web, alături de HTML și CSS. JavaScript este un limbaj de programare client-side, ceea ce înseamnă că este executat pe computerul utilizatorului, în browserul web, și nu pe server. Acest lucru îi permite să interacționeze cu utilizatorul și să ofere o experiență mai interactivă și dinamică.

JavaScript a fost creat inițial de către Brendan Eich la Netscape în anii 1990 și a devenit rapid unul dintre cele mai populare limbaje de programare din lume. Acesta a fost standardizat de către ECMA International și a devenit cunoscut sub numele de ECMAScript. De-a lungul anilor, au fost lansate mai multe versiuni ale ECMAScript, cu cel mai recent standard fiind ECMAScript 2021. JavaScript este un limbaj de programare versatil și flexibil, care poate fi utilizat pentru o varietate de aplicații, cum ar fi dezvoltarea de aplicații web, crearea de aplicații mobile, programarea server-side, crearea de jocuri și multe altele. Este, de asemenea, un limbaj de programare ușor de învățat, care este utilizat de mulți programatori și dezvoltatori web din întreaga lume.

Unul dintre cele mai importante aspecte ale JavaScript este faptul că acesta este un limbaj de programare orientat către obiecte. Acest lucru înseamnă că în loc să manipuleze direct elementele unei pagini web, programatorii pot crea obiecte care să reprezinte aceste elemente și să interacționeze cu ele. Aceste obiecte pot fi apoi utilizate pentru a schimba starea paginii web și pentru a crea interactivitate.

JavaScript oferă, de asemenea, o serie de funcții și metode predefinite, care pot fi utilizate pentru a efectua diferite operații. Aceste funcții includ, printre altele, funcții pentru manipularea elementelor paginii web, funcții pentru manipularea șirurilor de caractere, funcții pentru gestionarea evenimentelor și multe altele.

1. **Realizarea aplicației 45-55pg**
   1. **Mediul de dezvoltare**

Aplicația dezvoltată de mine de administrare a unei pensiuni este dezvoltată in Visual Studio, iar pașii care au fost urmați sunt unii foarte simpli care respectă o logică specifică unor aplicații de tip Asp.net, dupa cum urmează:

* Am creat un proiect nou de tip ASP.NET WebApplication în cadrul căruia există subfoldere precum View, Models
* În folderul View am creat subfolderele Admin si User. Adminul, după cum se observă și în figura 6, conține paginile AdminMaster, Categories.aspx, Categories.aspx.cs, Rooms.aspx, Rooms.aspx.cs, Users.aspx, Users.asp.cs. De asemena User conține UserMaster, Booking.aspx, Booking.aspx.cs. Mai exista paginile Login.aspx, Login.aspx.cs unde se realizeaza trierea in functie de rolul pe care il are cel care interactioneaza cu aplicatia.
* Pentru a putea realiza operațiile de adăugare/ștergere am creat baza de date locală în cadrul Visual Studio denumită PensiuneAsp.mdf

În final, după parcurgerea tuturor pașilor de mai sus și finisarea detaliilor, aplicația mea are urmatoarea structură:

A screenshot of a computer

Description automatically generated

Figură 6 Structură aplicație administrare pensiune

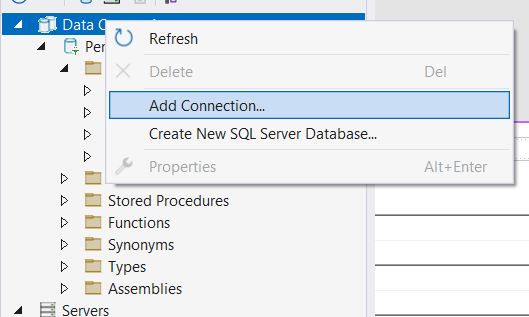
* 1. **Diagrama UML a bazei de date**

Aplicația web a fost gândită în așa fel încât să poată să fie îndeplinite toate oprațiunile necesare unei astfel de situații, de rezervare, de informare asupra unui sejur, dar și de o mentenanță și administrare mai precisă. Mediul de dezvoltare Visual Studio ajută la structurarea proiectului în aceste tipuri de diagrame, permite vizualizarea acestora în mod organizat după cum se observă in figura următoare:

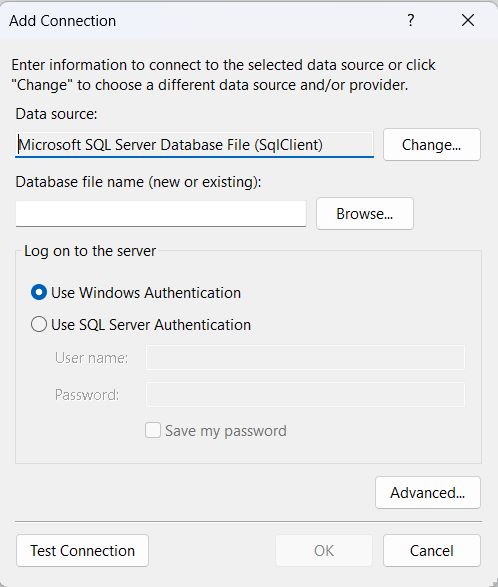
Diagrama exportata de visual

* 1. **Arhitectura bazei de date**

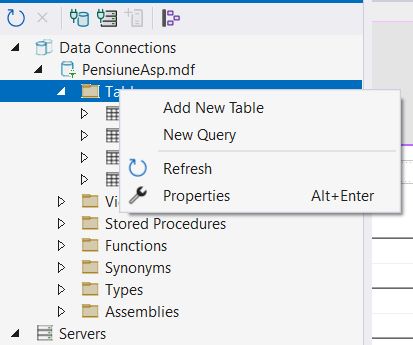
Proiectul meu de diplomă se bucură de o organizare și structurare a datelor în tabele în cadrul unei baze de date. Aceasta a fost denumită sugestiv “PensiuneAsp.mdf” si respectă proprietățiile unei baze de date obișnuite, ajutând la gestionarea mai ușoară a datelor în ceea ce privește rezervările de camere, masă, dar și organizarea în categorii a camerelor și de administrare a utilizatorilor. Crearea bazei de date a fost realizată local, în cadrul mediului de dezvoltare Visual Studio în felul următor:



\*adaugat nume figura



În capitolul 3.2 denumit Diagrama UML a bazei de date am prezentat organizarea completă, vizualizarea în ansamblu a bazei de date PensiuneAsp.mdf, tocmai de aceea în cele ce urmează o sa detaliez tabelele și proprietătile acesteia. Pentru a ajunge la structura finală a fost nevoie de creare de tabele unde sunt structurate datele în funtie de nevoie, cum ar fi utilizatori, categorii de camere, camerele de inchiriat și rezervarile. Crearea unui nou tabel în baza de date a fost facută în interiorul bazei de date, așa cum se observă în figura .....



Tabelul User (UserTable) dispune de urmatoarele câmpuri:

* UId 🡪 reprezintă id-ul utilizatorului, este capabil de autoincrementare, tipul acestuia fiind bineînteles int deoarece fiecare utilizator are un număr de id unic
* UName 🡪
* UPhone 🡪
* UGen 🡪
* UAdd 🡪
* UPass 🡪

Toate cele detaliate mai sus se pot observa în figura … poză extrasă exact din Visual Studio

A screenshot of a computer

Description automatically generated with medium confidence

Tabelul Room (RoomTable) – este desemnata in stocarea datelor cu privire la camerele care sunt in proprietate si conține urmatoarele campuri:

* Rid 🡪
* RName 🡪
* RCategory 🡪
* RLocation 🡪
* RCost 🡪
* RRemarks 🡪
* Status 🡪
* PretAtv 🡪
* PretMasa 🡪
* PretGhid 🡪
* PretBicicleta 🡪

Pentru o intelegere mai buna in figura ... se poate observa exact dispunerea lor in proiect

A screenshot of a computer

Description automatically generated with medium confidence

Tabelul Category (CategoryTable) este un tabel care este folosit pentru a organiza camerele disponibile pensiunii in categorii astfel utilizatorilor le este mai facil sa se orienteze spre nevoile fiecaruia. Acest tabel contine urmatoarele câmpuri:

* CatId 🡪
* CatName 🡪
* CatRemarks 🡪

A screenshot of a computer

Description automatically generated with medium confidence

Tabela Booking (BookingTable) este destinată stocării informațiilor despre rezervările efectuate în pensiune. Aceasta a fost gandită exact dupa nevoile clienților și al proprietarilor si are urmatoarele câmpuri:

* BId 🡪
* BDate 🡪
* BRoom 🡪
* User Id 🡪
* DateIn 🡪
* DateOut 🡪
* Amount 🡪

Figura … ilustrează aceste detalii în ceea ce privește BookinTable

A screenshot of a computer

Description automatically generated with medium confidence

În felul acesta am ajuns la urmatoarea structura a bazei de date PensiuneAsp:

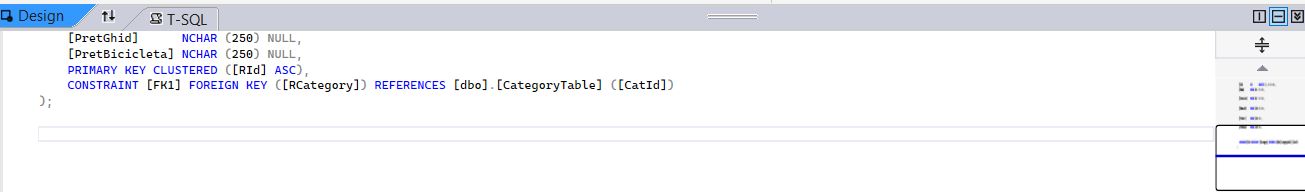
\*inlcouit cu poza pe alb

A screenshot of a computer

Description automatically generated

\*legaturi intre tabele\* fk

Roomtable fk

0

Bookingtable fk



* 1. **Design-ul aplicației**

Design-ul unei aplicații are un rol deosebit de important atât în vânzarea acesteia, dar și pentru simpla interacțiune cu aceasta. Performanța acestei aplicații consider eu că este mărită de către interfața prietenoasă prin faptul că un design optimizat și interactiv crește numărul de interacțiuni cu utilizatorii astfel crește popularitatea pensiunii deci aduce și un beneficiu proprietarului. Utilizarea aplicației de către utilizatori este importanta pentru cel ce deține pensiunea, deci s-a dorit o interfață a aplicației ușor de folosit, sa fie intuitive lucrurile pe care aceștia trebuie să le faca pentru a avea o bună interacțiune cu proprietarul. În ceea ce urmează o sa prezint design-ul pe care eu l-am gandit pentru proiectul meu de diplomă, incepand de la primul lucru pe care un utilizator îl vede cand accesează aplicația pana la cel mai mic detaliu. Așadar, în momentul în care un user doreste sa rezerve o camera, este dornic de a avea un sejur la pensiune regasește următoarea pagina:

A picture containing text, screenshot, house, sky

Description automatically generated

A screenshot of a login box

Description automatically generated with medium confidence

Important de reținut este că pentru a putea beneficia de sejur user-ul trebuie să își facă un cont, lucru care este foarte simplu de realizat. Pe pagina principala se află un buton care va conduce la un formular unde se va putea crea un cont nou pentru utilizatorii care sunt pentru prima dată pe site și doresc sa se bucure de o vacanță la pensiune. Formularul de înscriere a noilor membrii arată in felul urmator:

A screenshot of a computer

Description automatically generated with medium confidence

Odata ce fiecare membru dornic de vacanță are un cont poate reveni la pagina principală, cea din figura…… unde se poate autentifica cu username-ul si parola create anterior, dar nu trebuie să selecteze check box-ul corespunzător cu numele user, rolul lui fiind de utilizator. Interfata user-ului este create în idea ca orice utilizator să nu depună mult efort pentru a cauta informatii cruciale, sau pentru a idendifica cum anume sa faca rezervarea. Pagina corespunzatoare utilizatorului arată în felul următor: \*poza cu pagina de user dup ace afis pretul inainte de plata

Bineînteles ca utilizatorul este foarte important, dar pe același loc în rangul importanței se află și adminul, sau mai pe înteles proprietarul pensiunii. Acesta se bucură, de asemenea, de o pagină cu un design uimitor care îi facilitează treaba, scurtează timpul și interacțiunea cu utilizatorul, pagina destinată acestuia arată astfel:

* Pagina responsabilă de gestionarea camerelor si rezervărilor

A screenshot of a computer

Description automatically generated with medium confidence

* Administrarea categoriilor de camere

A screenshot of a computer

Description automatically generated

* Gestionarea utilizatorilor

A screenshot of a computer

Description automatically generated

* 1. **Dezvoltarea propriu-zisă a aplicației** 
     1. **Use-case aplicție**
     2. **Conexiunea cu baza de date – fisier functions**

+schema cu diagrama ce face admin si user

* + 1. **Autentificare Admin**
       1. **Parte vizuala**
       2. **Parte funcțională**
       3. **Gestionarea rezervarilor de către admin**
          1. **Parte vizuală**
          2. **Parte funcțională**
       4. **Gestionarea utilizatorilor**
          1. **Parte vizuală**
          2. **Parte funcțională**
       5. **Gestionarea categoriilor de camere**
    2. **Autentificare Utilizator**

* + - 1. **Procesul de rezervare de către utilizator**

1. **Concluzii și dezvoltări ulterioare**
   1. **Concluzii**
   2. **Dezvoltări ulterioare**
2. **Bibliografie**

* [1]<https://www.oracle.com/ro/database/what-is-database/>
* [2] <https://codecool.com/ro/blog/sql-incepatori/>
* [3] <https://csharp.2000things.com/2013/01/11/756-viewing-a-class-diagram-in-visual-studio-2012/>
* [4] <https://www.javatpoint.com/asp-net-web-form-introduction>
* [5] <https://static.javatpoint.com/csharp/images/csharp-features1.png>
* [6] <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
* [7] <http://eed.usv.ro/~cpalaghianu/resurse/informatica/html/h_2.htm>