

## FACULTATEA TEHNOLOGII INFORMAȚIONALE ȘI STATISTICĂ ECONOMICĂ

**DEPARTAMENTUL**

## TEHNOLOGIA INFORMAȚIEI ȘI MANAGEMENT INFORMAȚIONAL

# PROIECT DE AN

# Tema: Web aplicație pentru un restaurant

**Specialitatea 0613.6 Informatica Aplicată**

## Autor:

Student gr. ,

*învăţământ cu frecvenţă*

## Mititiuc Eduard

*(semnătura)*

## Conducător ştiinţific:

conf. univ., dr. **Lozan Victoria**

*(semnătura)*

## Chişinău – 2023

Introducere 3

1. Revoluția digitală din domeniului HORECA 5

1.1 Concepte generale despre domeniul HORECA. 5

1.2. Evoluția domeniului HORECA 8

1.3. Digitalizarea domeniului HORECA 10

2. Extinderea web aplicațiilor printre restaurante. 11

2.1. Istoria și structura web aplicațiilor 15

2.2. Importanța aplicațiilor web 23

2.2. Metode de implementare cu Java 25

Concluzii 34

Referințe bibliografie 37

Anexe 38

**Introducere**

În ultimii ani, restaurantele au fost profund afectate de revoluția digitală care a avut loc în domeniul HORECA (Hoteluri, Restaurante și Catering). O schimbare semnificativă în modul în care funcționează restaurantele a fost adusă de apariția și dezvoltarea web aplicațiilor pentru comandarea produselor alimentare. Aceste aplicații au devenit tot mai populare în rândul consumatorilor, oferindu-le o modalitate comodă și eficientă de a-și comanda mâncarea preferată de la restaurantele dorite, fără a mai fi nevoie să se deplaseze fizic la locație.

Motivul pentru care web aplicațiile au câștigat o popularitate crescândă în domeniul HORECA este în strânsă legătură cu nevoile și preferințele clienților moderni. Într-o lume agitată și mereu conectată, clienții caută soluții rapide și eficiente pentru a-și satisface nevoile alimentare. Așadar, prin intermediul web aplicațiilor, clienții pot comanda mâncare de la restaurantele preferate cu doar câteva click-uri, fără a fi necesară vizitarea fizică a localului. Aceasta oferă o mare comoditate, în special în contextul vieții aglomerate și stresante din ziua de azi.

Restaurantele, la rândul lor, au adoptat rapid aceste aplicații pentru a-și îmbunătăți serviciile și pentru a răspunde cererii clienților. Prin intermediul acestor platforme digitale, restaurantele pot promova meniurile, pot facilita plasarea comenzilor și pot gestiona livrările într-un mod mai eficient. Mai mult decât atât, web aplicațiile oferă restaurantelor posibilitatea de a colecta date despre preferințele și comportamentul clienților, ceea ce le permite să își personalizeze ofertele și să ofere recomandări mai relevante.

Astfel, conform unei analize detaliate și a unui studiu realizat de portalul cnb.com, „pandemia globală a impus restaurantele să se adapteze rapid la noile condiții impuse de restricțiile de mobilitate și distanțare socială. În situașia dată, unica soluție a fost dezvoltarea și implementarea web aplicațiilor pentru a-și menține afacerea activă și pentru a continua să deservească clienții”. Deci, din informația dată concluzionăm că doar prin intermediul web aplicațiilor, restaurantele au putut să ofere clienților o modalitate sigură și convenabilă de a comanda mâncare, eliminând nevoia de a vizita fizic locația și de a intra în contact direct cu angajații. Această măsură a contribuit la reducerea riscului de răspândire a virusului și la menținerea siguranței atât a clienților, cât și a personalului. Anume pandemia globală a scos la evidență cât de importante sunt tehnologiile digitale și a demonstrat lumii întregi ca web aplicațiile deja au devenit o adevărată necesitatea pentru industria HORECA.

Așa dar, web aplicațiile au oferit restaurantelor posibilitatea de a gestiona mai eficient comenzile și livrările, adaptându-se cerințelor și restricțiilor impuse de pandemie. Prin intermediul acestor platforme digitale, restaurantele au putut să controleze procesul de preluare a comenzilor, să organizeze rutele de livrare și să mențină comunicarea fluidă cu clienții.

În plus, web aplicațiile au contribuit la menținerea relației cu clienții fideli și la atragerea unor noi clienți, prin intermediul ofertelor personalizate, a promoțiilor și a recenziilor pozitive. Într-un moment în care restricțiile impun limitări semnificative asupra activității restaurantelor, dezvoltarea web aplicațiilor a fost o modalitate vitală de a menține conexiunea și interacțiunea cu publicul.

Motivația alegerii acestei teme este legată de importanța și impactul pe care web aplicațiile le au asupra industriei alimentare. Astfel de aplicații oferă multiple avantaje atât pentru restaurante, cât și pentru clienți. În plus, restaurantele pot să-și extindă afacerea și să ajungă la un număr mai mare de clienți, în timp ce clienții beneficiază de comoditatea de a comanda mâncare de oriunde și în orice moment dorit.

Tema acestei lucrări are un grad de noutate considerabil, având în vedere că digitalizarea în domeniul HORECA a luat amploare în ultimul deceniu și continuă să se dezvolte rapid, în special, după cum a fost deja afirmat, odată cu apariția pandemiei. Într-adevăr, în prezent, majoritatea restaurantelor utilizează web aplicații pentru a-și facilita procesul de comandă și pentru a oferi o experiență mai bună clienților lor, chiar dacă pandemia, în prezen, nu mai există oficial. Cu toate acestea, este important să înțelegem mai în profunzime modul în care aceste aplicații sunt implementate și modul în care influențează relația dintre restaurante și clienți.

Scopul acestei lucrări este de a investiga extinderea web aplicațiilor în industria HORECA, în special a restaurantelor, și de a analiza impactul lor asupra industriei alimentare. Obiectivele generale ale lucrării includ:

* Identificarea importanței aplicațiilor web în cadrul restaurantelor și analizarea beneficiilor aduse de utilizarea acestora.
* Examinarea componentelor cheie ale unei web aplicații pentru comandarea mâncării și evaluarea modului în care acestea contribuie la îmbunătățirea experienței clienților.
* Investigarea diferitelor metode de implementare a aplicațiilor web în restaurante și a tehnologiilor necesare

Metodologia de cercetare folosită în această lucrare va implica analiza literaturii de specialitate, studii de caz și interviuri cu profesioniști din domeniul HORECA. Aceste surse de informație ne vor ajuta să înțelegem mai bine impactul aplicațiilor web în restaurante și să obținem o perspectivă amplă asupra subiectului.

Această lucrare este formată din 2 capitole și 6 subcapitole, după cum urmează:

* Capitolul 1: Revoluția digitală din domeniul HORECA aduce o analiză detaliată a transformărilor digitale care au avut loc în industria HORECA. Acest capitol explorează conceptele generale despre domeniul HORECA, evidențiind evoluția sa în timp și impactul digitalizării asupra acestuia. În acest context, subcapitolul 1.1 prezintă aspecte esențiale și definitorii ale industriei HORECA, subcapitolul 1.2 explorează evoluția sa în raport cu tehnologia și cerințele pieței, în timp ce subcapitolul 1.3 se concentrează asupra procesului de digitalizare din industria HORECA.
* Capitolul 2: Implementarea web aplicațiilor este destul de dificilă și necesită foarte multe cunoștințe de programare. Astfel, subcapitolul 2.1 prezintă istoria web aplicațiilor de-a lungul deceniilor și nivelul la care acestea au ajuns în ziua de azi. Subcapitolul 2.2 prezintă importanța și impactul pe care acestea le au asupra economiei și umanității, dar în special asupra industriei HORECA, iar copitolul 2.3 detaliază tehnologiile disponibile pentru a crea asemenea aplicații, fiind pus accentul anume pe utilizarea limbajului de programare Java și a tehnologiile create cu acesta.

Astfel, Capitolul 1 și Capitolul 2 explorează, în mod consecvent, aspecte cheie ale transformării digitale în industria HORECA. Capitolul 1 se concentrează pe evoluția generală și digitalizarea domeniului, în timp ce Capitolul 2 se concentrează pe aplicarea specifică a tehnologiilor web în restaurante, componentele și metodele de implementare ale aplicațiilor web.

1. **REVOLUȚIA DIGITALĂ DIN DOMENIUL HORECA**

**1.1 Concepte generale despre domeniul HORECA**

Domeniul HORECA, acronim pentru Hoteluri, Restaurante și Catering, reprezintă o parte esențială a industriei ospitalității și alimentației publice. Acest sector are o contribuție semnificativă în economia globală și a devenit o parte integrantă a vieții noastre de zi cu zi.

Unul dintre conceptele cheie în industria HORECA este experiența oferită clienților. Hotelurile, restaurantele și serviciile de catering se străduiesc să creeze experiențe memorabile pentru oaspeți și clienți. Aceasta implică atât aspecte vizibile, cum ar fi calitatea mâncării și a băuturilor, confortul și curățenia în hoteluri, cât și aspecte nevizibile, cum ar fi atmosfera, ospitalitatea și serviciul impecabil. Oaspeții așteaptă să fie tratați cu respect și să li se ofere o experiență plăcută în timp ce se bucură de serviciile și produsele oferite de industria HORECA.

Pe de altă parte, un alt concept important în acest domeniu este calitatea. Indiferent de segmentul în care activează, hotelurile, restaurantele, localurile etc. sunt evaluate în funcție de calitatea oferită. Așadar, calitatea mâncării, a serviciilor, a facilităților și a produselor este un factor decisiv în succesul unei afaceri HORECA. Este esențial ca aceste unități să ofere produse și servicii de înaltă calitate pentru a-și păstra și atrage clienții. Astfel, se pune un accent deosebit pe selectarea ingredientelor proaspete și de înaltă calitate și formarea personalului în domeniu.

Istoria HORECA este strâns legată de evoluția umanității și a nevoilor sale în ceea ce privește alimentația și ospitalitatea. În această lucrare, vom explora evoluția și importanța restaurantelor(și nu numai) de-a lungul timpului în cadrul domeniului HORECA.

Astăzi, restaurantele continuă să fie un element esențial al industriei HORECA și al culturii noastre. Ele reflectă diversitatea culturală, oferă o varietate de opțiuni gastronomice și contribuie la economia locală și la turism. De asemenea, restaurantele reprezintă locuri de întâlnire și de socializare, unde oamenii se pot bucura de mâncare și băutură într-un cadru plăcut și ospitalier.

Printre altele, restaurantele reprezintă una dintre cele mai vizibile și populare componente ale domeniului HORECA. Acestea sunt locuri unde oamenii se adună pentru a se bucura de mâncare delicioasă, a socializa și a crea amintiri. În continuare, se vor enumera și descrie câteva concepte generale despre restaurante și importanța lor în societatea noastră.

Un concept cheie în industria restaurantelor este diversitatea. Restaurantele pot fi clasificate în funcție de bucătăriile pe care le oferă, de specificul mâncărurilor și de stilul serviciului. Există restaurante cu specific italian, chinezesc, mexican, indian, mediteranean și multe altele. Această diversitate în ofertă permite oamenilor să experimenteze și să descopere bucătării din întreaga lume, aducând în fiecare restaurant o notă unică. În plus, fiecare restaurant are propriul său farmec și atrage oameni cu preferințe și gusturi diferite.

Un alt concept important în restaurante este creativitatea culinară. Bucătarii din restaurante sunt adesea considerați artiști ai bucătăriei. Ei își pun amprenta asupra preparatelor, folosind ingredientele în mod inovator și combinând arome și texturi în moduri neașteptate. Creativitatea culinară nu se limitează doar la crearea unor preparate delicioase, ci și la prezentarea acestora într-un mod atractiv și artistic. Astfel, restaurantele devin locuri în care mâncarea devine o experiență estetică și senzorială.

Un alt aspect esențial în restaurante este serviciul de calitate. Personalul din restaurante joacă un rol crucial în oferirea unei experiențe plăcute pentru clienți. Ospitalitatea, amabilitatea și profesionalismul sunt calități căutate în angajații din domeniul HORECA. Un serviciu impecabil nu se rezumă doar la preluarea comenzilor și livrarea acestora, ci și la capacitatea de a răspunde rapid la cerințele și necesitățile clienților. Personalul trebuie să fie bine instruit și să fie capabil să ofere recomandări, să răspundă la întrebări și să ofere asistență într-un mod prietenos și eficient.

Mai mult ca atât, un element foarte important o reprezintă atmosfera. Așadar, ambianța și decorul pot influența în mod semnificativ experiența clienților. Fie că este vorba de un restaurant elegant și sofisticat sau de unul casual și relaxant, atmosfera trebuie să fie plăcută și să se potrivească cu stilul și tema restaurantului. Muzica de fundal, iluminarea și designul interior sunt elemente cheie în crearea unei atmosfere care să atragă și să ofere confort clienților.

Deci, restaurantele sunt mult mai mult decât simple locuri unde oamenii mănâncă. Ele reprezintă puncte de întâlnire, locuri de socializare și aduc oamenii împreună în jurul unei mese. Restaurantele sunt locuri unde se sărbătoresc evenimente speciale, se creează amintiri și se împărtășesc experiențe. Ele reprezintă o expresie a culturii, a gastronomiei și a ospitalității unei comunități.

În același timp, domeniul HORECA are o influență semnificativă asupra economiei globale. Acest sector aduce multiple contribuții prin generarea de locuri de muncă, stimularea turismului, impulsarea industriei alimentare și promovarea creșterii economice. Unul dintre aspectele majore ale HORECA este crearea de locuri de muncă. Industria ospitalității și alimentației publice oferă oportunități de angajare pentru o gamă largă de lucrători, de la bucătari și ospătari la personal de curățenie și management. Restaurantelor și hotelurilor le revine responsabilitatea de a oferi formare și dezvoltare profesională angajaților, contribuind astfel la crearea de locuri de muncă stabile și la creșterea nivelului de ocupare a forței de muncă.

HORECA are, de asemenea, un impact semnificativ asupra industriei turismului. Hotelurile și restaurantele sunt destinații de primă importanță pentru turiști, călători de afaceri și vizitatori. Acestea contribuie la atragerea de turiști într-o anumită regiune, aducând venituri și investiții. Oferind servicii de cazare și masă de calitate, sectorul HORECA joacă un rol esențial în crearea unei imagini pozitive a destinațiilor turistice și în stimularea creșterii economice.

Mai mult ca atît, HORECA influențează, de asemenea, industria alimentară. Restaurantelor, de exemplu, le revine responsabilitatea de a achiziționa și utiliza produse alimentare de înaltă calitate. Acest lucru generează o cerere constantă de produse agricole și alimentare, contribuind la dezvoltarea sectorului agricol și la susținerea producătorilor locali. Prin intermediul lanțului de aprovizionare, HORECA stimulează industria alimentară și generează valoare economică. Efectul enorm asupra altor sectoare al HORECA este, de asemenea, demn de menționat. Prin natura sa, acest sector generează o serie de efecte economice în alte industrii. De exemplu, hotelurile și restaurantele necesită servicii de întreținere și curățenie, ceea ce duce la crearea de locuri de muncă în industria serviciilor. De asemenea, aceste unități economice fac achiziții de bunuri și servicii, cum ar fi mobilierul, echipamentele de bucătărie, produsele de curățenie sau soluțiile software, sprijinind astfel alte sectoare și întreprinderi.

În plus, domeniul HORECA din Republica Moldova reprezintă o componentă importantă a economiei și a industriei turismului în această țară. Cu un potențial bogat în resurse naturale și culturale, Republica Moldova atrage tot mai mulți turiști și investitori în industria ospitalității și a alimentației publice.

Cu toate acestea, este important de menționat că industria HORECA din Republica Moldova se confruntă și cu provocări, cum ar fi concurența inegală, necesitatea de a menține standarde ridicate de calitate și adaptarea la cerințele în continuă schimbare ale turiștilor și consumatorilor locali. Cu toate acestea, cu o varietate de atracții turistice, gastronomie autentică și ospitalitate caldă, domeniul HORECA din Republica Moldova continuă să se dezvolte și să contribuie la dezvoltarea economică și turistică a țării. Iată câteva date concrete care ilustrează modul în care HORECA a ajutat economia Republicii Moldova:

* Crearea de locuri de muncă: Conform siteului de statistici oficiale statistica.gov.md, industria HORECA din Republica Moldova a generat “un număr semnificativ de locuri de muncă. Potrivit datelor din 2020, sectorul ospitalității și alimentației publice a angajat aproximativ 83.000 de persoane, reprezentând aproximativ 7% din forța totală de muncă a țării”. Aceasta oferă oportunități de angajare pentru tineri și pentru cei care caută o carieră în acest domeniu.
* Turismul și atragerea de turiști: Domeniul HORECA a fost un factor important în dezvoltarea turismului în Republica Moldova. Țara are numeroase obiective turistice, cum ar fi mănăstiri, cetăți, peisaje naturale și podgorii, care atrag turiști din întreaga lume. Prin furnizarea de servicii de cazare și masă de calitate, hotelurile și restaurantele din Republica Moldova au contribuit la creșterea numărului de turiști și la creșterea veniturilor din turism. Astfel, în 2018, conform siteului noi.md, numărul străinilor care au vizitat Moldova a crescut într-un an aproape cu 16%, constituind 15058 de persoane”.
* Promovarea produselor locale: Industria HORECA a sprijinit promovarea produselor locale din Republica Moldova. Restaurantele și hotelurile din țară au susținut producătorii locali prin achiziționarea de alimente și băuturi locale. Acest lucru a avut un impact pozitiv asupra agricultorilor și producătorilor locali, stimulând dezvoltarea sectorului agricol și susținând economia locală.
* Investiții și dezvoltare: Dezvoltarea domeniului HORECA a atras și investiții străine în Republica Moldova. Hotelurile și restaurantele au fost obiective pentru investitori, ceea ce a dus la creșterea infrastructurii turistice și la îmbunătățirea calității serviciilor.
* Creșterea sectorului alimentar: Domeniul HORECA a contribuit la creșterea sectorului alimentar în Republica Moldova. Restaurantele, cafenelele și serviciile de catering au nevoie de aprovizionare constantă cu produse alimentare de calitate, ceea ce a stimulat cererea și a creat noi oportunități pentru producătorii locali de a-și promova produsele.

Toate aceste aspecte arată că domeniul HORECA a avut un impact semnificativ asupra economiei Republicii Moldova, contribuind la creșterea economică, dezvoltarea turismului, crearea de locuri de muncă și promovarea produselor locale. Prin atragerea de turiști și investiții, HORECA a adus beneficii semnificative pentru dezvoltarea țării.

Republica Moldova găzduiește o varietate de restaurante care oferă o experiență culinară diversă și autentică. Aceste restaurante sunt recunoscute pentru preparatele lor delicioase, atenția acordată ingredientelor locale și atmosfera primitoare. Iată câteva exemple de restaurante din Republica Moldova, care, confor portalului diez.md, sunt printre cele mai renumite și prestigioase din țară:

* Restaurantul "La Butuc", situat în Chișinău, este renumit pentru bucătăria sa tradițională moldovenească. Meniul oferă o gamă largă de preparate locale, inclusiv mâncăruri cu carne, produse lactate și plăcinte tradiționale. Atmosfera rustică și servirea ospitalieră adaugă o notă autentică experienței culinare.
* Restaurantul "Symposium", localizat în Orheiul Vechi, oferă o combinație unică între gastronomie și istorie. Situat într-o peșteră, restaurantul oferă o experiență originală, unde vizitatorii pot savura preparate tradiționale moldovenești într-un cadru rustic și autentic.
* Restaurantul "Castel Mimi", situat în orașul Bulboaca, se află în incinta unei castele impresionante și este cunoscut pentru bucatele sale rafinate și prezentarea artistică a preparatelor. Meniul combină influențe moldovenești și internaționale, oferind o selecție variată de preparate culinare.
* Restaurantul "Propaganda", localizat în centrul Chișinăului, este renumit pentru designul său neconvențional și pentru preparatele moderne. Atmosfera vibrantă, combinată cu un meniu inovator și creativ, a făcut ca acest loc să devină popular printre tineri și entuziaști ai gastronomiei.
* Restaurantul "Vatra Neamului", situat în Chișinău, oferă preparate tradiționale românești și moldovenești într-un cadru autentic și plin de tradiție. Meniul include preparate clasice precum sarmale, mămăligă, plăcinte și deserturi tradiționale.

Acestea sunt doar câteva exemple de restaurante din Republica Moldova, iar industria alimentației publice din țară oferă multe alte opțiuni deosebite. Este demn de menționat că multe restaurante din Republica Moldova pun un accent deosebit pe utilizarea ingredientelor locale și pe promovarea bucătăriei tradiționale. Această abordare nu doar că oferă o experiență autentică turiștilor, dar sprijină și economia locală și producătorii agricoli din regiune.

Dar, e foarte important să remarcăm că chiar și turiștii apreciază restaurantele din Prepublica Moldova, fapt ce demonstrează importanța și rolul pe care îl au pentru dezvoltarea economiei și turismului. Într-adevăr, perspectiva străinilor asupra restaurantelor din Moldova poate varia în funcție de experiențele individuale și de așteptările personale. Cu toate acestea, există anumite aspecte comune pe care le pot observa sau aprecia în restaurantele din Moldova. De exemplu, mulți străini apreciază autenticitatea și unicitatea bucătăriei moldovenești. Mâncărurile tradiționale, cum ar fi mămăliga, sarmalele, plăcintele și vinurile locale, pot fi percepute ca fiind unice și interesante pentru străini care doresc să experimenteze cultura culinară locală. În general, Moldova a devenit o destinație populară pentru gastroturism, atrăgând vizitatori străini dornici să exploreze și să descopere bucătăria locală. Aceștia apreciază autenticitatea și ingredientele proaspete utilizate în preparatele locale, precum și experiența culinară autentică pe care o pot trăi în restaurantele moldovenești. Dar, mai ales Moldova este renumită pentru producția sa de vinuri de calitate. Străinii pasionați de vinuri pot fi încântați de selecția și diversitatea vinurilor moldovenești disponibile în restaurante. De asemenea, aceștia pot fi interesați să viziteze cramele și podgoriile din regiune pentru a descoperi mai multe despre producția de vin din Moldova. Mai mult ca atât, Restaurantele din Moldova pot oferi o atmosferă tradițională și autentică, cu decor specific, muzică și elemente culturale. Aceasta poate fi apreciată de străini ca o experiență autentică a culturii moldovenești. Pe de altă parte, străinii pot observa ospitalitatea și serviabilitatea personalului din restaurantele moldovenești. Un serviciu cald și prietenos poate contribui la experiența generală și poate face ca străinii să se simtă bineveniți și apreciați.Este important de menționat că percepția străinilor poate varia, iar aceste aspecte pot fi doar o perspectivă generală. În final, experiența individuală într-un restaurant din Moldova depinde de mai mulți factori, cum ar fi selecția restaurantului, calitatea mâncării și serviciului, precum și așteptările personale ale fiecărui străin în parte.

Astfel, restaurantele din Republica Moldova reprezintă un pilon important al domeniului HORECA, contribuind direct la bunăstarea și prosperitatea țării, iar rolul acestora devine din ce în ce mai vizibil în societate.

**1.2 Evoluția domeniului HORECA**

Istoria restaurantelor poate fi urmărită până în antichitate, când locurile publice de întâlnire unde se servea mâncare și băutură au început să apară în diferite civilizații. De exemplu, în Roma antică existau taberne și hanuri în care călătorii puteau găsi adăpost și hrană. Asemenea locuri erau frecventate și de localnici pentru a socializa și a se bucura de gustări și băuturi.

În Evul Mediu, taverna a devenit o componentă importantă a vieții urbane. Aceste locuri de întâlnire erau în principal frecventate de oameni obișnuiți și de călători și ofereau adăpost, mâncare și băutură. În timp, au apărut și localuri specializate, precum ospătării și hanuri, care serveau mâncăruri și băuturi specifice.

Cu trecerea timpului, restaurantele au început să se dezvolte ca entități distincte și rafinate. Primul restaurant modern este considerat "La Grande Taverne de Londres", deschis în Paris în 1782. Acesta a introdus un meniu fix și a oferit o selecție diversă de preparate culinare. În același timp, restaurantele de lux au început să apară în marile orașe europene, oferind experiențe culinare sofisticate și servicii de înaltă calitate. Apoi, în secolul al XIX-lea, restaurantele au devenit o parte integrantă a vieții urbane și ale culturii gastronomice. Parisul a fost un centru important al acestui fenomen, cu restaurante renumite precum Le Grand Véfour și Maxim's. Aceste locuri au atras oameni din toate părțile lumii și au devenit puncte de întâlnire pentru artiști, scriitori și elitele sociale.

Odată cu revoluția industrială și creșterea urbanizării, restaurantele s-au răspândit pe scară largă în întreaga lume. În Statele Unite, de exemplu, restaurantele au devenit parte a peisajului urban în secolul al XIX-lea, iar orașe precum New York și Chicago au devenit cunoscute pentru varietatea și inovația în domeniul gastronomiei. În secolul al XX-lea, restaurantul a devenit o afacere bine dezvoltată și diversificată. De la restaurante fast-food și lanțuri de restaurante până la restaurante cu stele Michelin și localuri inovatoare, industria restaurantelor a cunoscut o creștere rapidă și o diversificare a ofertei. Restaurantul a devenit un loc nu doar pentru a mânca, ci și pentru a experimenta bucătăria internațională, a descoperi noi arome și a explora creativitatea culinară.

Evoluția domeniului HORECA ân Republica Moldova a cunoscut schimbări semnificative în ultimele decenii, odată cu procesul de modernizare economică și deschiderea catre influențele occidentale. Domeniul HORECA a devenit din ce ân ce mai important pentru economia țării, contribuind la dezvoltarea turismului și la creșterea nivelului de ocupare a forței de muncă.

În anii post-sovietici, sectorul HORECA din Republica Moldova a fost caracterizat de o infrastructură redusă și de servicii limitate. Cu toate acestea, odată cu deschiderea economiei și crearea de legături comerciale cu țările din Uniunea Europeană, s-au produs schimbări semnificative. Au apărut hoteluri moderne, restaurante cu bucătării internaționale, cafenele și baruri, oferind o gamă variată de opțiuni pentru localnici și turiști. Influența occidentală s-a reflectat și în stilul de gestionare a unităților HORECA. S-au introdus standarde internaționale de calitate și servicii, iar personalul a fost instruit în conformitate cu aceste standarde. Astfel, atât calitatea serviciilor, cât și oferta gastronomică s-au îmbunătățit semnificativ.

În plus, un aspect important în evoluția domeniului HORECA în Republica Moldova a fost creșterea numărului de turiști și vizitatori străini. Țara a devenit o destinație din ce în ce mai atractivă pentru turiști, datorită peisajelor pitorești, patrimoniului cultural bogat și tradițiilor autentice. Această creștere a cererii a determinat apariția de noi hoteluri, restaurante și cafenele pentru a satisface nevoile turiștilor.

De asemenea, au avut loc și schimbări în preferințele consumatorilor moldoveni. Oamenii au devenit mai interesați de experiențe culinare diverse, de calitatea serviciilor și de ambianța oferită de unitățile HORECA. Astfel, s-a înregistrat o creștere a numărului de restaurante specializate, cum ar fi cele cu specific internațional (italian, japonez, mexican) sau cele care promovează bucătăria tradițională moldovenească. Un alt factor care a influențat evoluția domeniului HORECA în Republica Moldova a fost dezvoltarea tehnologiei și a serviciilor de livrare la domiciliu. În ultimii ani, aplicațiile de food delivery și comenzile online au devenit tot mai populare, permitând oamenilor să comande mâncare de la restaurantele preferate în confortul propriei locuințe. Acest trend a dus la apariția unor noi afaceri în domeniul HORECA, specializate exclusiv pe livrarea de mâncare.Totodată, sectorul HORECA din Republica Moldova a început să-și dezvolte și propria identitate culinară, promovând produsele locale și rețetele tradiționale. Astfel, au apărut restaurante și cafenele care utilizează ingrediente locale și susțin producătorii autohtoni, oferind astfel o experiență autentică pentru vizitatori.

Pe de altă parte, creșterea numărului de turiști, dezvoltarea infrastructurii și îmbunătățirea serviciilor au contribuit la transformarea Republicii Moldova într-o destinație turistică din ce în ce mai atractivă. Cu toate acestea, există încă mult potențial pentru dezvoltarea ulterioară a sectorului HORECA, în special prin promovarea patrimoniului cultural și gastronomic moldovenesc. Restaurantele din Republica Moldova au cunoscut o evoluție notabilă în ultimele decenii, și cu siguranță acestea vor evolua chiar și mai mult. De altfel, întregul domeniu HORECA are o dentință de a se moderniza și de a se adapta noilor valori umane caracteristice perioadei. În general, tendințele din industria HORECA se schimbă constant pentru a răspunde cerințelor și preferințelor în continuă evoluție ale consumatorilor. Oamenii sunt din ce în ce mai preocupați de sănătatea și bunăstarea lor, ceea ce a dus la o cerere crescută pentru opțiuni de mâncare sănătoasă și alternative dietetice. Restaurantele și cafenelele își extind meniurile pentru a include opțiuni vegetariene, vegane, fără gluten și fără lactate, și promovează ingrediente proaspete și organice. Acest lucru demonstrează clienților că restaurantele, de exemplu, se gândesc activ la clienții lor și fac tot posibilul pentru a le satisface nevoile și pretențiile. În plus, consumatorii devin mai conștienți de impactul pe care îl au asupra mediului și societății. Astfel, tendința în industria HORECA este de a adopta practici sustenabile, cum ar fi utilizarea ingredientelor locale și de sezon, reducerea deșeurilor, reciclarea și donarea excesului de alimente către organizații caritabile. Mai mult ca atât, svansurile tehnologice au un impact semnificativ în industria HORECA. De la comenzi online și aplicații de livrare la automatizarea proceselor de comandă și plată în restaurante, tehnologia îmbunătățește eficiența și experiența clienților. De asemenea, apar noi inovații în domeniul echipamentelor de bucătărie și a sistemelor de gestionare a restaurantelor. Dar, consumatorii încă caută experiențe culinare unice și personalizate. Astfel, restaurantele și hotelurile se concentrează pe oferirea de meniuri personalizate, evenimente tematice, sesiuni de gătit interactiv sau degustări de vin pentru a crea experiențe memorabile și captivante. Acest lucru este facilitat de faptul că oamenii sunt din ce în ce mai deschiși către descoperirea bucătăriilor și culturilor culinare diverse. Restaurantele internaționale și evenimentele gastronomice aduc în prim-plan o gamă variată de mâncăruri și arome din întreaga lume, oferind o experiență gustativă diversificată. Iată de ce, prezentarea vizuală a mâncărurilor joacă un rol important în atragerea clienților. Bucătarilor le place să creeze meniuri interactive, mâncăruri spectaculoase și prezentări artistice pentru a oferi o experiență gastronomică memorabilă. Deci, este evident că proprietarii de afaceri din acest domeniu trebuie să fie la curent cu aceste tendințe și să le adapteze în mod corespunzător pentru a satisface cerințele și preferințele în continuă schimbare ale clienților.

Deci, putem afirma că restaurantele din Republica Moldova au evoluat semnificativ în ultimele decenii, adaptându-se cerințelor și preferințelor clienților, diversificându-și oferta culinară și îmbunătățind calitatea serviciilor. Continuarea promovării bucătăriei tradiționale, a ingredientelor locale și a inovației culinare va juca un rol esențial în dezvoltarea ulterioară a industriei restaurantelor în Republica Moldova.

**1.3 Digitalizarea domeniului HORECA**

În ultimele decenii, umanitatea a fost martora unei revoluții tehnologice fără precedent. Dezvoltarea accelerată a tehnologiilor, în special a celor digitale, a schimbat în mod fundamental modul în care trăim, lucrăm și interacționăm cu lumea din jurul nostru. De la invenția primului computer, dezvoltarea tehnologiilor digitale a fost caracterizată de o creștere enormă. În anii 1940, primele calculatoare erau mari și greoaie, având o capacitate de procesare limitată. Cu toate acestea, odată cu avansul tehnologic, computerele au devenit tot mai mici, mai puternice și mai accesibile. Progresele în domeniul microprocesoarelor, a memoriei și a tehnologiilor de comunicații au deschis calea către dispozitive mai rapide și mai inteligente.

Trebuie să recunoaștem că tehnologiile digitale au influențat toate aspectele vieții noastre, de la modul în care comunicăm și lucrăm până la felul în care ne petrecem timpul liber. Internetul și platformele digitale ne-au adus oportunități nelimitate de acces la informație, comunicare instantanee și colaborare globală. Tehnologiile mobile ne-au transformat telefoanele în adevărate centrale multimedia, permițându-ne să accesăm oricând și de oriunde o gamă largă de servicii și aplicații.

În plus, sectorul economic a fost puternic afectat de dezvoltarea tehnologiilor digitale. Inovațiile digitale au generat noi modele de afaceri și au transformat sectoare întregi, cum ar fi comerțul electronic, publicitatea online și serviciile financiare. Automatizarea și inteligența artificială au îmbunătățit eficiența și productivitatea, permițând companiilor să inoveze și să se adapteze rapid la cerințele pieței. Educația a fost, de asemenea, revoluționată de tehnologiile digitale. Accesul la platforme online de învățare și resurse digitale a democratizat educația, permițând oamenilor din întreaga lume să acceseze informație și să învețe în ritmul lor. Tehnologiile digitale, precum realitatea virtuală și augmentată, au adus noi metode de învățare și au îmbunătățit interacțiunea dintre profesori și elevi. Într-adevăr, dezvoltarea tehnologiilor digitale a adus o serie de avantaje semnificative pentru societate, economie și cultură. Suntem conectați mai mult ca niciodată, iar tehnologia a devenit o parte indispensabilă a vieții noastre de zi cu zi. Cu toate acestea, este important să abordăm în mod responsabil și etic aceste tehnologii, pentru a maximiza beneficiile și a minimiza impactul negativ. Pe măsură ce continuă să avanseze, tehnologiile digitale vor continua să modeleze lumea în care trăim și vor deschide noi orizonturi de inovație și descoperire.

Există, desigur, și provocări asociate cu dezvoltarea tehnologiilor digitale. Una dintre ele ste securitatea cibernetică, deoarece în contextul digital există riscuri legate de furtul de date și de atacurile cibernetice. Protecția confidențialității și gestionarea datelor personale au devenit o preocupare majoră în această eră digitală și ignorarea lor poate avea chiar consecințe colosale, fie că este vorba de o persoană fizică, sau de o persoană juridică. Iată de ce, protejarea datelor în era digitală prezintă o provocare continuă și complexă. Într-adevăr, cu creșterea dependenței noastre de tehnologie și de stocarea datelor în mediul online, se pune o mai mare presiune asupra protecției acestor informații sensibile. Amenințările cibernetice evoluează rapid și devin tot mai sofisticate. Hackerii și infractorii cibernetici folosesc tehnici avansate pentru a obține acces la datele noastre și a le exploata în diverse moduri. Aceasta include atacuri de tip phishing, malware, ransomware și alte tactici de hacking. În ciuda acestui fapt, cantitatea de date pe care o generăm și o stocăm este în continuă creștere. Aceasta include informații personale, financiare, de afaceri și alte tipuri de date sensibile, pe care suntem nevoiți să le furnizăm când facem comenzi online. Gestionarea și protejarea acestui volum mare de date prezintă o provocare în sine, deoarece necesită soluții scalabile și eficiente de securitate, de care, din păcate, adesea nici cele mai mari întreprinderi nu dispun. În ciuda acestui fapt, datele noastre sunt acum mai mobile ca niciodată. Utilizăm dispozitive multiple, cum ar fi telefoane mobile, tablete și laptopuri, și accesăm informații de pe rețele Wi-Fi publice sau prin conexiuni la distanță. Aceasta crește riscul ca datele să fie interceptate sau accesate fără autorizație. Totuși, e important să menționăm că există o creștere semnificativă a reglementărilor privind protecția datelor, cum ar fi Regulamentul General privind Protecția Datelor (GDPR) din Uniunea Europeană și alte legi similare la nivel global. Conformitatea cu aceste reglementări poate fi complexă și costisitoare pentru organizații, mai ales în contextul datelor transfrontaliere. De fapt, e interesant că o mare parte a problemelor de securitate a datelor provine din erori umane. Acest lucru poate include utilizarea de parole slabe, accesarea site-urilor nesigure, trimiterea accidentală a informațiilor sensibile prin e-mail sau pierderea sau furtul dispozitivelor mobile. Educația și conștientizarea utilizatorilor sunt aspecte critice în protejarea datelor. Deci, chiar și cu cele mai bune măsuri de securitate, există întotdeauna riscul unui incident de securitate, ce poate avea loc si din cauza necunoașterii sau neatenției. Este important să existe planuri și proceduri de răspuns la incidente pentru a gestiona și a remedia rapid astfel de situații și pentru a minimiza impactul asupra datelor și sistemelor, căci protejarea datelor necesită o abordare multidimensională.

Dezvoltarea tehnologiilor a impus domeniul HORECA să facă anumite schimbări semnificative pentru a rămâne competitive și a se adapta la noile cerințe ale pieței. Această evoluție tehnologică a influențat modul în care afacerile din domeniul HORECA gestionează procesele interne, interacționează cu clienții și își promovează produsele și serviciile. De exemplu, implementarea plății cu cardul în sectorul HORECA a avut loc treptat, în special în ultimii ani. Această schimbare a fost susținută de avansul tehnologic și de creșterea cererii din partea consumatorilor pentru opțiuni de plată mai moderne și mai flexibile. Există mai multe motive-cheie pentru care HORECA a adoptat plata cu cardul:

* Rapiditate și eficiență: Plata cu cardul permite procesarea rapidă a tranzacțiilor și elimină necesitatea de a manipula numerar. Acest lucru poate duce la o experiență mai rapidă și mai eficientă pentru clienți, reducând timpul de așteptare la casele de plată.
* Siguranță și securitate: Plata cu cardul oferă un nivel sporit de securitate atât pentru consumatori, cât și pentru agenții economici din sectorul HORECA. Riscul de furt, pierdere sau contrafacere a banilor cash este eliminat, iar tranzacțiile sunt protejate prin intermediul protocolului de securitate al plăților electronice.
* Ușurința în administrare: Implementarea plății cu cardul în HORECA reduce nevoia de a gestiona numerarul în locații precum restaurante, hoteluri sau cafenele. Aceasta poate simplifica procesul de gestiune financiară și elimină riscul erorilor umane legate de manipularea banilor cash.
* Creșterea vânzărilor: Oferirea opțiunii de plată cu cardul în HORECA poate atrage un număr mai mare de clienți și poate crește volumul tranzacțiilor. Mulți consumatori preferă să plătească cu cardul pentru că este mai comod și elimină necesitatea de a avea bani cash asupra lor.
* Îmbunătățirea experienței clienților: Plata cu cardul oferă o experiență mai modernă și mai fluidă pentru clienți. Aceasta poate contribui la îmbunătățirea imaginii și reputației afacerilor din sectorul HORECA, oferind o impresie pozitivă consumatorilor.
* Ușurința în urmărirea tranzacțiilor: Prin intermediul plăților cu cardul, agenții economici din HORECA pot obține o evidență electronică clară a tranzacțiilor și pot avea acces la rapoarte detaliate. Aceasta facilitează procesul de urmărire a veniturilor și poate contribui la o gestionare mai eficientă a afacerii.
* Adaptarea la preferințele consumatorilor: Într-o eră digitală în care majoritatea persoanelor dețin cel puțin un card de debit sau credit, implementarea plății cu cardul în HORECA este o necesitate pentru a rămâne în pas cu preferințele și comportamentul de consum al clienților.

Avantajele implementării plății cu cardul în sectorul HORECA sunt multiple și contribuie la modernizarea și eficientizarea afacerilor din acest domeniu, în timp ce oferă clienților o experiență mai convenabilă și sigură.

Pe de altă parte, și rețelele sociale au jucat un rol semnificativ în sprijinirea și promovarea industriei HORECA. Aceste platforme digitale au oferit oportunități considerabile pentru afacerile din acest sector, permițându-le să interacționeze cu clienții existenți și potențiali într-un mod mai direct și eficient. Iată cum rețelele sociale au ajutat HORECA:

* Promovare și marketing: Rețelele sociale au oferit un canal de promovare eficient pentru afacerile din industria HORECA. Restaurantele, hotelurile și cafenelele pot utiliza platformele precum Facebook, Instagram, Twitter și LinkedIn pentru a-și promova meniurile, serviciile, ofertele speciale și evenimentele. Aceasta le permite să ajungă la un public mai larg și să genereze interes și participare din partea comunității locale și a turiștilor.
* Comunicare și relaționare cu clienții: Rețelele sociale oferă o modalitate directă și interactivă de a comunica cu clienții. HORECA poate utiliza aceste platforme pentru a răspunde întrebărilor, a primi feedback, a rezolva probleme și a construi o relație mai apropiată cu publicul său. Astfel, afacerile pot obține informații valoroase despre preferințele și nevoile clienților, ceea ce le permite să se adapteze și să îmbunătățească serviciile și experiența oferită.
* Crearea de conținut vizual atractiv: Industria HORECA se bazează mult pe aspectul vizual al produselor și serviciilor sale. Rețelele sociale oferă o platformă excelentă pentru a crea și distribui conținut vizual atractiv, cum ar fi fotografii și videoclipuri care prezintă preparate culinare delicioase, decoruri deosebite sau experiențe unice în locațiile respective. Acest conținut atrăgător poate stârni interesul și dorința de a vizita și experimenta localul respectiv.
* Recomandări și influențatori: Pe rețelele sociale, oamenii își împărtășesc experiențele și preferințele, oferind astfel recomandări pentru locații HORECA. Aceste recomandări pot veni de la prieteni, familie sau influențatori digitali. Prin colaborarea cu influențatori relevanți, restaurantele și hotelurile pot beneficia de o expunere mai mare și pot atrage noi clienți prin recomandările acestora.
* Programare și rezervări online: Multe platforme de rețele sociale permit afacerilor din HORECA să ofere opțiuni de programare și rezervări online. Prin intermediul acestor funcționalități, clienții pot face rezervări de mese sau camere de hotel într-un mod simplu și comod. Aceasta nu numai că facilitează procesul pentru clienți, ci și pentru afaceri, eliminând necesitatea de a utiliza intermediari sau de a gestiona rezervările prin telefon.
* Monitorizarea concurenței și tendințelor: Rețelele sociale oferă posibilitatea de a monitoriza activitatea concurenților și tendințele din industrie. Prin intermediul acestor platforme, afacerile din HORECA pot urmări ce fac competitorii lor, cum interacționează cu clienții și ce strategii de marketing adoptă. Aceasta le permite să rămână în pas cu schimbările și să ia decizii informate pentru a-și îmbunătăți afacerea.

Dar, poate fi observat că, în ultimii ani, industria HORECA a început să adopte tot mai mult web aplicațiile pentru a spori eficiența și a oferi o experiență mai bună clienților. Aceste aplicații web au devenit esențiale pentru gestionarea și operarea afacerilor din sectorul HORECA. Iată câteva exemple de roluri și beneficii ale web aplicațiilor în industria dată:

* Prezentare și rezervări online: Prin intermediul web aplicațiilor, restaurantele, hotelurile și cafenelele pot avea un site web atractiv și interactiv, care să ofere informații despre serviciile, meniurile, facilitățile și ofertele speciale. De asemenea, clienții pot utiliza aplicațiile pentru a face rezervări online, eliminând astfel necesitatea de a apela telefonic sau de a se deplasa personal pentru a face o rezervare.
* Meniuri și comenzi online: Web aplicațiile permit afacerilor din HORECA să ofere clienților posibilitatea de a vizualiza meniurile și de a plasa comenzi online. Acest lucru face procesul de comandă mai rapid și mai convenabil pentru clienți, iar pentru afaceri, elimină riscul de erori umane și poate duce la o gestionare mai eficientă a comenzilor.
* Procesarea plăților: Web aplicațiile permit integrarea sistemelor de plată online, ceea ce facilitează procesarea plăților pentru servicii și produse. Aceasta oferă clienților opțiunea de a plăti cu cardul sau prin alte metode electronice, reducând astfel necesitatea manipulării banilor cash și eliminând riscul de furt sau pierdere.
* Gestiunea stocurilor și a inventarului: Web aplicațiile pot fi utilizate pentru gestionarea stocurilor și a inventarului în industria HORECA. Aceste aplicații oferă funcționalități pentru monitorizarea și actualizarea nivelului stocurilor, gestionarea datelor privind furnizorii și prețurile, precum și generarea rapoartelor relevante pentru luarea deciziilor în ceea ce privește gestionarea stocurilor și achizițiile.
* Gestiunea relației cu clienții (CRM): Web aplicațiile CRM permit afacerilor din HORECA să gestioneze și să urmărească interacțiunile cu clienții într-un mod mai organizat și eficient. Aceste aplicații oferă funcționalități precum gestionarea bazei de date a clienților, urmărirea preferințelor și a istoricului de cumpărături, gestionarea feedback-ului și oferirea de oferte personalizate. Aceasta ajută la crearea și menținerea unei relații solide cu clienții și la generarea de loialitate față de brand.
* Marketing și promovare: Web aplicațiile oferă instrumente puternice pentru activități de marketing și promovare în industria HORECA. Prin intermediul acestor aplicații, afacerile pot gestiona campanii de email marketing, promovarea pe rețele sociale, generarea de coduri de reducere și alte tactici de marketing pentru a atrage și fideliza clienții.

Deși digitalizarea aduce multiple beneficii, există și câteva probleme și provocări pe care industria HORECA le poate întâmpina în procesul de adoptare a tehnologiei și de implementare a soluțiilor digitale. Iată câteva dintre aceste probleme:

* Costurile inițiale și de întreținere: Implementarea și menținerea unor soluții digitale în industria HORECA implică costuri inițiale semnificative. Achiziționarea de echipamente, dezvoltarea de aplicații personalizate sau integrarea cu sisteme existente poate fi costisitoare. În plus, există costuri continue pentru întreținerea și actualizarea tehnologiilor adoptate.
* Nevoia de instruire și adaptare: Digitalizarea necesită o adaptare și instruire corespunzătoare a personalului din industria HORECA. Angajații trebuie să învețe să utilizeze noile echipamente și aplicații, să înțeleagă procesele digitale și să fie capabili să gestioneze eventualele probleme tehnice. Aceasta poate necesita timp și resurse suplimentare.
* Dependența de infrastructura și conectivitate: Pentru a funcționa eficient, soluțiile digitale necesită o infrastructură adecvată și o conexiune stabilă la internet. În anumite zone sau în timpul unor evenimente, calitatea conexiunii la internet poate fi scăzută sau poate lipsi. Aceasta poate crea probleme în funcționarea și utilizarea corectă a soluțiilor digitale în industria HORECA.
* Protecția datelor și securitatea: Odată cu digitalizarea, crește și importanța protejării datelor și asigurarea securității informațiilor sensibile. Industria HORECA gestionează o gamă largă de date personale și financiare ale clienților. Astfel, există riscul de încălcări ale securității datelor, cum ar fi furtul de identitate, fraudă sau atacuri cibernetice. Astfel, este necesară implementarea unor măsuri solide de securitate pentru a proteja aceste informații.
* Dependenta excesivă de tehnologie: În timp ce digitalizarea poate aduce multiple beneficii, există riscul de a deveni prea dependenți de tehnologie. Apariția unor probleme tehnice sau a unei întreruperi a sistemului poate afecta funcționarea afacerilor și relația cu clienții. De asemenea, dependența de tehnologie poate reduce interacțiunea umană și poate afecta experiența personalizată oferită clienților.
* Schimbarea comportamentului consumatorilor: Odată cu digitalizarea și accesul facil la informații online, comportamentul consumatorilor se schimbă. Aceștia devin mai exigenți, au așteptări mai mari în ceea ce privește
* serviciile digitale și au tendința de a alege opțiuni mai rapide și mai convenabile.Astfel, industria HORECA trebuie să fie în pas cu aceste schimbări și să se adapteze pentru a rămâne competitivă. În caz contrar, întrepridnerele riscă să piardă clienți și să piradă foarte mult di venituri, ceea ce poate duce până la faliment.

Deci,concluzionăm că domeniul HORECA mereu a fost și încă este un pilon important al economiei unui stat, care se află într-o continuă evoluție. Ca și multe alte domenii, acesta a fost influențat de dezvoltarea rapidă a tehnologiilor. Dar, deși digitalizarea aduce numeroase avantaje pentru industria HORECA, există și anumite provocări și probleme pe care trebuie să le abordeze în procesul de adoptare a tehnologiei. Cu o planificare și gestionare corespunzătoare, aceste dificultăți pot fi depășite, iar digitalizarea poate contribui la creșterea și dezvoltarea afacerilor din domeniu.

1. **EXTINDEREA WEB APLICAȚIILOR PRINTRE RESTAURANTE**

**2.1 Istoria și structura web aplicațiilor**

De la crearea primului site web în anii 90 până în prezent, web-ul și aplicațiile web au parcurs o evoluție remarcabilă, devenind o parte indispensabilă a vieții noastre cotidiene. În acest capitol, se va prezenta istoria lor, urmărind progresul tehnologic și impactul pe care l-au avut asupra societății și economiei.

Întăi de toate, în anii 90, World Wide Web (WWW) a fost creat de către Tim Berners-Lee, marcând începutul unei noi ere în comunicare și schimbul de informații. Primele site-uri web au fost statice, cu conținut static și cu puține interacțiuni posibile. Cu toate acestea, a fost o revoluție în sine, oferind acces la informații globale într-un mod ușor și rapid. Odată cu progresele în tehnologia web, au apărut noi tehnologii și standarde care au permis dezvoltarea aplicațiilor web interactive și complexe. Conform wikipdia, “Un exemplu notabil este introducerea limbajului de programare JavaScript de către Brendan Eich în 1995, care a oferit posibilitatea de a crea funcționalități dinamice și interacțiuni în timp real pe paginile web. După. în anii 2000, a început era aplicațiilor web 2.0, marcând tranziția de la site-uri statice la aplicații web sociale și colaborative.” Platforme precum Facebook, YouTube și Wikipedia au devenit populare și au deschis calea pentru utilizatorii obișnuiți să contribuie la conținutul web-ului și să interacționeze în mod activ unii cu alții. O altă dezvoltare semnificativă în istoria aplicațiilor web este apariția cloud computing-ului. Prin intermediul serviciilor web, aplicațiile pot fi rulate și accesate de pe servere remote, fără a necesita instalarea locală pe dispozitivul utilizatorului. Aceasta a facilitat colaborarea și partajarea informațiilor într-un mod mai eficient și a deschis ușa pentru aplicații web mai scalabile și mai flexibile.

O mare revoluție tehnologică a reprezentat-o apariția telefoanelor mobile. Odată cu popularizarea smartphone-urilor, aplicațiile mobile au devenit o parte integrantă a experienței utilizatorului. Dezvoltatorii de aplicații web au trebuit să se adapteze și să creeze aplicații mobile responsive și optimizate pentru diferite platforme și dimensiuni de ecran. A apărut un ecosistem vast de aplicații mobile, cu magazine deaplicații care au devenit o piață înfloritoare pentru dezvoltatori.

Cu toate acestea, în prezent, asistăm la o expansiune continuă a aplicațiilor web și la o creștere rapidă a tehnologiilor emergente, cum ar fi inteligența artificială (AI), realitatea virtuală (VR) și internetul obiectelor (IoT). Aceste tehnologii aduc funcționalități noi și captivante în domeniul aplicațiilor web, cum ar fi asistenții virtuali, experiențe immersive și interconectivitatea dispozitivelor.

Istoria aplicațiilor web demonstrează progresul remarcabil al tehnologiei și modul în care aceasta a schimbat modul în care interacționăm și lucrăm în mediul online. De la site-uri web statice la aplicații web interactive, de la colaborarea în platformele 2.0 la era aplicațiilor mobile, aplicațiile web au devenit o parte indispensabilă a vieții noastre moderne. Cu noile inovații și tehnologii care apar în fiecare zi, viitorul aplicațiilor web este promițător și plin de oportunități nelimitate pentru dezvoltatorii și utilizatorii deopotrivă.

În general, wikipedia definește o web aplicație ca „un program care rulează într-o arhitectură client-server folosind tehnologiile deschise World Wide Web”. Aceste aplicații sunt proiectate pentru a oferi utilizatorilor funcționalități și servicii interactive, precum interacțiunea cu bazele de date, crearea și partajarea conținutului, procesarea tranzacțiilor și multe altele. Ele sunt dezvoltate folosind tehnologii web standard, cum ar fi HTML, CSS și JavaScript, și pot fi accesate de pe diferite dispozitive conectate la internet.

Caracteristicile specificice web aplicațiilor, conform portalului thinksys.com, sunt:

* Interactivitate: Web aplicațiile permit utilizatorilor să interacționeze în timp real cu conținutul și funcționalitățile oferite. Aceasta înseamnă că utilizatorii pot introduce informații, pot primi rezultate imediate și pot personaliza experiența în funcție de preferințe.
* Accesibilitate: Web aplicațiile sunt accesibile de pe orice dispozitiv conectat la internet și care are un browser web. Utilizatorii pot accesa aplicațiile de pe computere, laptopuri, telefoane mobile sau tablete, fără a fi nevoie de instalarea unei aplicații native.
* Actualizări ușoare: Deoarece web aplicațiile rulează pe server, actualizările și îmbunătățirile pot fi implementate centralizat și distribuite automat utilizatorilor. Astfel, utilizatorii beneficiază întotdeauna de ultimele funcționalități și corecții de erori fără a fi necesară actualizarea manuală.
* Stocare și gestionare a datelor: Web aplicațiile pot interacționa cu baze de date și pot stoca și gestiona informații specifice utilizatorilor. Aceasta permite personalizarea conținutului și a funcționalităților în funcție de preferințele individuale.
* Integrare și interoperabilitate: Web aplicațiile pot fi integrate cu alte aplicații și servicii web, permițând transferul de date și interoperabilitatea între diferite platforme și sisteme.

Este evident că importanța acestora este deosebită, așa cum ele au reușit să influențeze modul în care oamenii socializează. interacționează, au provocat schimbări în difertie domenii și au modificat modul de gândire. Printre principalele roluri ale acestora se numără:

* Accesibilitate globală: Web aplicațiile oferă posibilitatea de a accesa informațiile și serviciile de oriunde în lume, contribuind la conectarea globală și la înlăturarea barierelor geografice.
* Ușurință în utilizare: Datorită interfețelor user-friendly și a accesibilității, web aplicațiile sunt mai ușor de utilizat pentru utilizatori în comparație cu aplicațiile tradiționale. Ele nu necesită instalare și pot fi utilizate rapid și eficient.
* Eficiență și productivitate: Web aplicațiile permit automatizarea proceselor și manipularea datelor într-un mod eficient. Ele sporesc productivitatea și facilitează colaborarea între utilizatori, indiferent de locație.
* Scalabilitate și actualizări ușoare: Web aplicațiile sunt ușor de scalat pentru a gestiona un număr crescut de utilizatori și pot fi actualizate centralizat, permițând dezvoltatorilor să adauge funcționalități noi și să corecteze erori în mod rapid și eficient.
* Costuri reduse: Utilizarea web aplicațiilor poate reduce costurile pentru utilizatori și întreprinderi. Utilizatorii nu trebuie să investească în hardware sau software specializat, iar întreprinderile pot economisi bani prin gestionarea centralizată a aplicațiilor și actualizărilor.

Acum, este foarte important să se enumere și componentele unei astfel de aplicații. În general, o aplicație web este alcătuită dintr-o varietate de componente care lucrează împreună pentru a oferi o experiență interactivă și funcționalități complexe utilizatorilor.

Aceste componente sunt grupate în 2 domenii enorme, și anume Backend si Frontend.

Frontend este termenul folosit în domeniul dezvoltării software pentru a se referi la partea vizibilă și interactivă a unei aplicații sau a unui site web. Este responsabil de prezentarea și interacțiunea utilizatorului cu interfața grafică și include toate elementele cu care utilizatorii interacționează direct, cum ar fi designul, aspectul vizual, animațiile și funcționalitățile interactive. De obicei, dezvoltarea frontend implică utilizarea limbajelor de programare precum HTML (HyperText Markup Language) pentru structurarea și definirea elementelor web, CSS (Cascading Style Sheets) pentru stilizarea și aspectul vizual al acestora și JavaScript pentru implementarea funcționalităților interactive și manipularea datelor. Un dezvoltator frontend are rolul de a traduce designul și cerințele utilizatorului în codul HTML, CSS și JavaScript necesar pentru a crea interfața și experiența utilizatorului. Acesta se asigură că aplicația sau site-ul web arată și funcționează în mod corespunzător pe diferite dispozitive și browsere, având în vedere aspecte precum responsive design (design adaptiv la diferite dimensiuni de ecran) și optimizarea performanței. Pe scurt, frontend-ul se concentrează pe partea vizuală și interactivă a unei aplicații sau a unui site web, oferind utilizatorilor o experiență plăcută și intuitivă în utilizarea acestora.a. HTML (HyperText Markup Language): HTML este limbajul de marcare utilizat pentru structurarea și organizarea conținutului unei pagini web. Acesta definește elementele structurale ale unei pagini, precum titluri, paragrafe, imagini, link-uri și altele. Deci, principalele componente a Frontend-ului sunt:

* CSS (Cascading Style Sheets): CSS este folosit pentru a formata și stiliza elementele HTML, oferind aspectul și designul paginilor web. Prin intermediul CSS, dezvoltatorii pot controla culorile, fonturile, dimensiunile, poziționarea și alte aspecte vizuale ale aplicației.
* HTML (HyperText Markup Language): HTML este un limbaj de marcare utilizat pentru structurarea și afișarea conținutului unei pagini web. Este un standard fundamental în dezvoltarea web și este folosit pentru a defini structura și organizarea conținutului unei pagini, inclusiv text, imagini, link-uri, tabele, formulare și multe altele.HTML utilizează o sintaxă bazată pe etichete (tag-uri) care sunt plasate în interiorul documentului HTML și care descriu și definesc elementele și structura paginii. Fiecare etichetă HTML are o semnificație specifică și poate avea atribute care furnizează informații suplimentare despre element.De exemplu, o etichetă HTML de bază este `<h1>`, care definește un titlu mare sau o antet în cadrul paginii. Alte exemple de etichete HTML includ `<p>` pentru paragrafe, `<a>` pentru link-uri, `<img>` pentru imagini și `<table>` pentru tabele.HTML este interpretat de către browser și afișat utilizatorului ca o pagină web interactivă. El oferă structura de bază a unei pagini web, iar stilizarea și aspectul vizual sunt adăugate ulterior cu ajutorul CSS (Cascading Style Sheets), iar funcționalitatea interactivă este implementată cu JavaScript. Prin utilizarea etichetelor și a structurii corecte în HTML, dezvoltatorii web pot crea pagini web bine organizate, accesibile și compatibile cu diferite browsere și dispozitive.
* JavaScript: JavaScript este un limbaj de programare care adaugă interactivitate și funcționalități avansate la o aplicație web. Cu ajutorul JavaScript, dezvoltatorii pot manipula și controla elementele HTML, pot efectua validări de date, pot crea animații, pot gestiona evenimente și pot comunica cu serverul prin intermediul cererilor asincrone.

Chiar dacă există multe alte tehnologii noi, ca Framework-uri, ele oricum își au rădăcinile de la tehnologiile descrise mai sus.

Backend, cunoscut și sub denumirea de back-end sau partea serverului, se referă la partea nevizibilă a unei aplicații software sau a unui site web. Este responsabilă de procesarea și gestionarea datelor, logicii de afaceri, securității și comunicării cu alte componente ale sistemului, precum baze de date, servere și servicii externe. În dezvoltarea software, partea de backend se concentrează pe dezvoltarea și implementarea funcționalităților care rulează în spatele scenei și care nu sunt vizibile utilizatorilor. Acest lucru include gestionarea bazelor de date, logica de afaceri, procesarea datelor, autentificarea și autorizarea utilizatorilor, gestionarea sesiunilor și securitatea aplicației.

Pentru a construi partea de backend a unei aplicații sau a unui site web, dezvoltatorii folosesc diferite tehnologii și limbaje de programare, precum Python, Java, PHP, Ruby, C# și altele. Aceste tehnologii permit dezvoltatorilor să creeze servere web, să interacționeze cu bazele de date, să implementeze logica de afaceri și să gestioneze comunicarea cu alte sisteme sau servicii.

Partea de backend este responsabilă de a furniza datele și informațiile necesare frontend-ului (partea vizibilă) pentru a fi prezentate și utilizate de către utilizatori. Aceasta include extragerea, prelucrarea și pregătirea datelor, gestionarea cererilor și răspunsurilor, asigurarea securității și gestionarea erorilor.Deci, partea de backend reprezintă fundația unei aplicații sau a unui site web, oferind funcționalitatea și gestionând aspectele nevizibile pentru utilizatori. Este responsabilă de procesarea și gestionarea datelor, logica de afaceri și interacțiunea cu alte componente ale sistemului. Termenul de back-end face trimitre directă la următarele tehnologii:

* Server: Un server este componenta care primește și gestionează cererile utilizatorilor și furnizează răspunsuri corespunzătoare. Serverul poate fi implementat folosind diverse tehnologii, cum ar fi Node.js, PHP, Python, Java sau altele.
* Bază de date: O bază de date este utilizată pentru stocarea și gestionarea datelor aplicației. Aceasta poate fi o bază de date relațională (de exemplu, MySQL, PostgreSQL) sau o bază de date non-relațională (de exemplu, MongoDB, Firebase). Baza de date permite persistența datelor și facilitează interacțiunea cu informațiile utilizatorului.
* API (Application Programming Interface): API-urile sunt utilizate pentru a permite comunicarea între diferite componente ale unei aplicații, inclusiv între frontend și backend. Ele definesc metodele și formatele de date pentru transmiterea informațiilor între aplicații sau între aplicație și server.

Pe de altă parte, o web aplicație mereu implică interacțiunea cu utilizatorul cu scopul de a colecta sau oferi date. Interacțiunea directă cu utilizatorul se face prin următoarele elemente:

* Formulare: Formularele permit utilizatorilor să introducă și să transmită date către aplicație. Acestea sunt utilizate pentru înregistrare, autentificare, introducerea de informații și multe altele. Dezvoltatorii trebuie să gestioneze validarea și prelucrarea datelor introduse în formulare.
* Elemente interactive: Elementele interactive, cum ar fi butoanele, link-urile, meniurile și elementele de navigare, facilitează interacțiunea utilizatorului cu aplicația și navigarea prin diferitele pagini și funcționalități.
* Animatiții și efecte vizuale: Animațiile și efectele vizuale pot îmbunătăți experiența utilizatorului prin adăugarea de elemente interactive, tranziții și alte efecte estetice. Acestea pot fi create folosind CSS, JavaScript sau biblioteci șp framework-uri specializate.

Deoarece datele sunt foarte prețioase, orice web aplicație necesită o securitate foarte bine pusă la punct. Securitatea unei web aplucații se bazează pe următoarele componente de bază:

* Autentificare și autorizare: Pentru a proteja datele utilizatorilor și a limita accesul la anumite funcționalități, aplicațiile web utilizează sisteme de autentificare și autorizare. Acestea permit utilizatorilor să își creeze conturi, să se autentifice și să primească permisiuni specifice în funcție de rolul lor în aplicație.
* Protecția datelor: Securitatea datelor este o componentă vitală a unei aplicații web. Prin utilizarea protocoalelor de criptare și a metodelor adecvate de stocare și transmitere a datelor, se asigură că informațiile utilizatorilor sunt protejate împotriva accesului neautorizat.

Componentele unei aplicații web lucrează împreună pentru a crea o experiență interactivă și funcțională pentru utilizatori. Frontend-ul se ocupă de prezentarea și interacțiunea cu utilizatorul, în timp ce backend-ul gestionează procesarea datelor și comunicarea cu serverul. Interacțiunea cu utilizatorul, animațiile și efectele vizuale adaugă valoare experienței, iar securitatea protejează datele și integritatea aplicației. Într-un efort comun, aceste componente construiesc fundația pentru o aplicație web de succes în era digitală.

Pe de altă parte, o web aplicație ar fi imposibilă fără un server. Servereleweb joacă un rol fundamental în furnizarea conținutului și serviciilor online. Un server web este un sistem de calcul care primește și gestionează cererile primite de la clienți (utilizatori) și furnizează răspunsurile corespunzătoare. Acesta oferă pagini web, fișiere, aplicații și alte resurse solicitate de clienți prin intermediul protocolului de transfer de hipertext (HTTP).

Funcționarea unui server web se bazează pe:

* Cererea și răspunsul HTTP: Când un utilizator accesează o pagină web sau trimite o cerere către un server web, aceasta este înregistrată sub forma unei cereri HTTP. Cererea conține informații despre resursa solicitată, cum ar fi URL-ul și tipul de cerere (GET, POST, PUT, DELETE).
* Procesarea cererii: Serverul web primește cererea HTTP și o procesează. Acesta caută resursa solicitată și decide cum să răspundă în funcție de cerințele utilizatorului și configurația sa. Serverul poate efectua operațiuni precum citirea dintr-o bază de date, generarea dinamică a conținutului sau transmiterea de fișiere statice.
* Răspunsul HTTP: După procesarea cererii, serverul web generează un răspuns HTTP care conține informațiile solicitate sau rezultatele cererii. Răspunsul poate fi o pagină HTML, un fișier CSS, un document JSON sau orice alt conținut specificat în cerere.

Există mai multe tipuri de servere web, dintre care cele mai comune, conform IBM.com, sunt:

* Apache HTTP Server: „Apache este un server web gratuit și open-source, utilizat pe scară largă în întreaga lume. Este cunoscut pentru fiabilitatea sa și flexibilitatea în gestionarea diverselor cereri și configurații”.
* Nginx: Nginx este un server web și proxy invers cu o arhitectură orientată pe evenimente. Este recunoscut pentru performanța sa ridicată și capacitatea de a gestiona un număr mare de cereri simultan.
* Microsoft Internet Information Services (IIS): „IIS este un server web dezvoltat de Microsoft pentru sistemele de operare Windows. Este integrat în mod nativ cu platforma Windows și oferă suport extins pentru tehnologii Microsoft, cum ar fi ASP.NET”.

În general, serverele web nu sunt destinate strict web aplicațiilor. Ele sunt folosite pe larg și pentru multe alte scopuri. Importanța serverelor web constă în:

* Furnizarea conținutului online: Serverele web permit accesul la pagini web, aplicații și alte resurse de pe internet. Ele joacă un rol crucial în furnizarea conținutului și serviciilor online pecare le utilizăm în viața de zi cu zi.
* Gestionarea traficului și scalabilitatea: Serverele web pot gestiona un număr mare de cereri simultan și pot asigura funcționarea în condiții de trafic intens. Acestea pot fi configurate pentru a permite scalabilitatea orizontală, adăugând sau eliminând servere pentru a răspunde cerințelor crescute.
* Securitatea și confidențialitatea datelor: Serverele web joacă un rol crucial în protejarea datelor utilizatorilor. Ele pot utiliza protocoale de criptare și certificate SSL/TLS pentru a asigura conexiuni sigure și pentru a proteja informațiile sensibile.

Trebuie să menționăm că în ultimii ani serviciile de Cloud au devenit foarte populare, astfel încât web aplicațiile au ajuns sa fie stocate folosind serviciile de Cloud. Înainte, acestea rulau pe simple mașini fizice locale conectate în rețea care aveau un server central cu care comunicau. Totuși, această metodă este una învechită și nesigură. Termenul "computing cloud" sau "cloud computing" se referă la furnizarea de servicii și resurse IT, precum stocare, calcul și acces la aplicații, prin intermediul internetului. În esență, cloud computing reprezintă utilizarea unor servere și infrastructuri remote pentru a depozita și procesa date, în loc să se utilizeze echipamente și resurse fizice locale.Într-un mediu de cloud computing, serviciile și resursele sunt furnizate de către un furnizor de servicii cloud, care gestionează infrastructura hardware și software. Utilizatorii pot accesa și utiliza aceste servicii prin intermediul internetului, fără a avea nevoie de cunoștințe tehnice avansate sau de a deține echipamente costisitoare.

Conform companiei Microsoft, există trei modele principale de cloud computing:

* Software as a Service (SaaS): „Furnizorul de servicii cloud pune la dispoziție aplicații software gata de utilizare prin intermediul internetului”. Utilizatorii pot accesa și utiliza aceste aplicații fără a fi nevoie să le instaleze sau să le gestioneze pe propriile dispozitive. Exemple de SaaS includ platforme de colaborare online, servicii de e-mail și aplicații de management al relațiilor cu clienții.
* Platform as a Service (PaaS): „Furnizorul de servicii cloud oferă o platformă de dezvoltare și un mediu de execuție pentru dezvoltatorii de software”. Aceasta permite dezvoltatorilor să creeze, să testeze și să ruleze aplicații fără a fi nevoie să se ocupe de aspectele infrastructurale subiacente, cum ar fi gestionarea serverelor sau a bazei de date. Exemple de PaaS includ servicii de dezvoltare web și medii de execuție pentru aplicații.
* Infrastructure as a Service (IaaS): ‚Furnizorul de servicii cloud pune la dispoziție resurse de infrastructură, cum ar fi servere virtuale, rețele și stocare, care pot fi utilizate de către utilizatori pentru a-și construi și gestiona propriile aplicații și medii de lucru.” Utilizatorii au control total asupra sistemelor lor, de la sistemul de operare până la aplicațiile instalate. Exemple de IaaS includ servicii de stocare în cloud și servicii de calcul scalabil.

Prin utilizarea cloud computing, utilizatorii și companiile pot beneficia de scalabilitate, flexibilitate, accesibilitate și eficiență sporită. De asemenea, aceștia pot reduce costurile asociate cu achiziționarea și întreținerea echipamentelor fizice, precum și cu gestionarea infrastructurii și a serviciilor IT.

Utilizarea cloud-ului pentru o aplicație aduce numeroase beneficii și este considerată o alegere strategică pentru mulți dezvoltatori și companii. Iată câteva dintre importante importanțe ale cloud-ului pentru o aplicație:

* Scalabilitate și flexibilitate: Cloud-ul oferă posibilitatea de a scala resursele și capacitatea de procesare în funcție de necesitățile aplicației. Astfel, poți crește sau reduce rapid resursele în funcție de traficul sau cererea aplicației, asigurând performanță și disponibilitate.
* Disponibilitate și reziliență: Cloud-ul oferă o infrastructură redundantă și distribuită, ceea ce asigură disponibilitatea ridicată a aplicației. Datorită distribuirii geografice a serverelor și redundanței, aplicația poate fi accesibilă în mod continuu, chiar și în cazul unor probleme tehnice sau întreruperi planificate.
* Backup și recuperare în caz de dezastru: Cloud-ul oferă servicii de backup automat și recuperare în caz de dezastru. Datele și aplicația pot fi copiate și salvate în siguranță în cloud, asigurând o protecție împotriva pierderii datelor în cazul unor evenimente neașteptate, cum ar fi defectele hardware sau întreruperile de curent.
* Costuri reduse: Utilizarea cloud-ului elimină nevoia de a investi în echipamente hardware costisitoare și infrastructură fizică. Plătești doar pentru resursele și serviciile de care ai nevoie, ceea ce poate reduce semnificativ costurile inițiale și permite o scalabilitate în funcție de nevoile aplicației.
* Accesibilitate și colaborare: Cloud-ul permite accesul la aplicație și date de oriunde, prin intermediul internetului. Acest lucru facilitează colaborarea între echipe, oferind posibilitatea de a lucra împreună pe aceeași aplicație și de a partaja informații în mod eficient.
* Actualizări și mentenanță ușoară: Prin utilizarea cloud-ului, actualizările de software și mentenanța sistemului pot fi realizate în mod centralizat și ușor de gestionat. Astfel, poți beneficia de cele mai recente funcționalități și actualizări de securitate fără a fi necesară intervenția manuală pe fiecare instanță a aplicației.
* Securitate: Furnizorii de servicii cloud investesc în securitatea infrastructurii și a datelor. Aceștia pun la dispoziție măsuri de securitate avansate, cum ar fi criptarea datelor, autentificarea în doi pași și monitorizarea constantă, pentru a asigura confidențialitatea și integritatea informațiilor.

Deci, utilizarea cloud-ului pentru o aplicație oferă avantaje semnificative, atât din punct de vedere tehnic, cât și din punct de vedere al costurilor și eficienței. Prin externalizarea infrastructurii și serviciilor în cloud, dezvoltatorii pot se concentra pe dezvoltarea funcționalităților aplicației și pe îmbunătățirea experienței utilizatorilor.

Dar, cel mai important component al oricărei web aplicații râmăn a fi datele. De fapt, datele sunt cele care oferă importanță oricărei aplicații și totul se rotește în jurul lor, iată de ce este necesar ca orice web aplicație să aibă o bază de date cu care comunică. Baza de date poate fi utilizată pentru a stoca date despre utilizatori, conținutul aplicației, configurări, informații despre produse sau orice altă informație relevantă pentru funcționalitatea aplicației. Iată câteva motive cheie pentru care o aplicație web are nevoie de o bază de date:

* Stocarea și gestionarea datelor: O bază de date permite aplicației să stocheze date într-un mod organizat și structurat. Aceasta facilitează gestionarea și accesul la informații într-un mod eficient și fiabil. De exemplu, o aplicație de comerț electronic poate utiliza o bază de date pentru a stoca detalii despre produse, inventar, prețuri și informații despre clienți.
* Persistența datelor: O bază de date asigură persistența datelor, ceea ce înseamnă că informațiile sunt păstrate chiar și după închiderea sau repornirea aplicației. Acest lucru este crucial pentru a evita pierderea datelor și pentru a asigura coerența și integritatea informațiilor.
* Manipularea datelor: Baza de date oferă funcționalități puternice pentru manipularea și interogarea datelor. Prin intermediul limbajului de interogare a bazei de date (de exemplu, SQL), aplicația poate executa operațiuni precum inserarea, actualizarea, ștergerea și interogarea datelor într-un mod eficient.
* Securitatea datelor: Bazele de date oferă mecanisme de securitate pentru a proteja informațiile stocate. Aceasta include autentificare, autorizare și criptare a datelor sensibile. Utilizarea unei baze de date asigură că datele sunt protejate și accesibile numai utilizatorilor autorizați.
* Eficiență și performanță: O bază de date bine proiectată și optimizată poate contribui la îmbunătățirea performanței aplicației. Indexarea și optimizarea interogărilor pot ajuta la obținerea rezultatelor mai rapide și la gestionarea eficientă a volumului mare de date.

Modul în care poate fi folosită o bază de date într-o aplicație web depinde de nevoile și cerințele specifice ale aplicației. În general, aplicația web poate utiliza baza de date pentru a efectua următoarele operațiuni:

* Salvarea datelor: Aplicația poate folosi baza de date pentru a salva și actualiza datele furnizate de utilizatori sau generate de aplicație, cum ar fi informațiile despre utilizatori, setările, conținutul creat, informații despre produse etc.
* Recuperarea și afișarea datelor: Aplicația poate interoga baza de date pentru a recupera și afi
* șa datele în funcție de cerințele utilizatorilor. De exemplu, o aplicație de blog poate recupera articolele salvate în baza de date și le poate afișa pe site.
* Manipularea datelor: Aplicația poate permite utilizatorilor să interacționeze cu datele prin intermediul interfeței sale. De exemplu, utilizatorii pot adăuga, edita sau șterge date din baza de date, cum ar fi adăugarea unui produs într-un coș de cumpărături sau actualizarea informațiilor personale.
* Analiza și raportarea datelor: Baza de date poate fi utilizată pentru a stoca și analiza date, generând rapoarte și statistici relevante pentru aplicație. Aceasta poate oferi informații valoroase despre utilizatori, comportamentul acestora și performanța aplicației.
* Este important ca dezvoltatorii de aplicații web să proiecteze și să implementeze o structură corespunzătoare a bazei de date, să utilizeze practici de securitate adecvate și să optimizeze interogările pentru a asigura funcționalitatea eficientă și securitatea datelor în cadrul aplicației web.

Există foarte multe Sisteme de Gestiune a bazelor de date, dar cel utilizat în proiectul respectiv este MySQL. MySQL oferă o serie de avantaje și funcționalități care îl fac un instrument puternic și popular în dezvoltarea aplicațiilor. Iată câteva motive pentru care MySQL este considerat excelent:

* Ușurință de utilizare: MySQL este cunoscut pentru ușurința sa de utilizare și administrare. Are o sintaxă simplă și intuitivă, care permite dezvoltatorilor și administratorilor de bază de date să lucreze eficient cu el. Este, de asemenea, bine documentat și are o comunitate activă care oferă suport și resurse.
* Performanță ridicată: MySQL este optimizat pentru a oferi o performanță ridicată în gestionarea și interogarea bazelor de date. Folosește tehnici avansate de indexare, optimizare a interogărilor și caching, ceea ce duce la timp de răspuns rapid și eficiență în manipularea datelor.
* Securitate: MySQL oferă opțiuni puternice de securitate pentru a proteja datele stocate în baza de date. Include funcționalități precum autentificare, criptare, gestionarea privilegiilor și auditarea acțiunilor utilizatorilor. De asemenea, beneficiază de actualizări regulate de securitate pentru a aborda eventualele vulnerabilități.
* Suport pentru limbajele de programare: MySQL este compatibil cu o gamă largă de limbaje de programare și tehnologii. Acest lucru permite dezvoltatorilor să integreze ușor bazele de date MySQL în aplicațiile lor, indiferent de tehnologia folosită.
* Open-source și comunitate activă: MySQL este un produs open-source, ceea ce înseamnă că este disponibil gratuit și beneficiază de o comunitate mare și activă de dezvoltatori. Aceasta asigură dezvoltarea continuă a produsului, corectarea erorilor și furnizarea de suport prin intermediul forumurilor și resurselor online.

Deci, putem vedea că o web aplicație are foarte multe componente care contribuie la funcționarea corectă a acesteia, dar fiecare dintre ele este foarte importantă, iar configurarea greșită a uneia poate avea consecințe dezastruoase.

**2.2 Importanța aplicațiilor web**

În era digitală în care trăim, aplicațiile web au devenit indispensabile pentru societatea noastră. Ele ne oferă o gamă largă de servicii și funcționalități care ne simplifică viața și ne permit să interacționăm cu lumea într-un mod mai eficient

Una dintre cele mai evidente și răspândite utilizări ale aplicațiilor web este în domeniul comunicării și colaborării. Platforme precum Facebook, Twitter, Instagram și LinkedIn ne permit să păstrăm legătura cu prietenii, familia și colegii noștri de oriunde în lume. Aceste aplicații web au transformat modul în care ne conectăm și comunicăm, aducându-ne împreună într-un mod mai rapid și mai eficient decât înainte. De asemenea, aplicațiile web de colaborare, cum ar fi Google Docs, Dropbox și Trello, facilitează munca în echipă, indiferent de distanța fizică dintre membrii acesteia. Aceste platforme permit partajarea și editarea simultană a documentelor, îmbunătățind comunicarea și eficiența în cadrul proiectelor și a activităților de afaceri.

O altă contribuție semnificativă a aplicațiilor web este accesul ușor și rapid la informație. Cu motoarele de căutare precum Google și Bing, putem obține răspunsuri la întrebări și căutări complexe în câteva secunde. Astfel, aplicațiile web ne permit să căutăm și să obținem informații despre orice subiect dorit, îmbogățindu-ne cunoștințele și ajutându-ne în procesul de învățare.

De asemenea, există numeroase aplicații web care oferă conținut educațional, cum ar fi platformele de învățare online precum Coursera, edX și Udemy. Aceste aplicații web permit accesul la cursuri și materiale educaționale de la instituții de învățământ de top din întreaga lume, oferind oportunități de dezvoltare personală și profesională fără a fi limitați de locație sau timp.

Aplicațiile web joacă, de asemenea, un rol crucial în gestionarea activităților noastre zilnice. De exemplu, aplicațiile de banking online ne permit să verificăm

soldul contului, să efectuăm plăți și să gestionăm finanțele personale cu ușurință și securitate. Aceste aplicații ne economisesc timpul și efortul de a merge la o bancă fizică și ne oferă un grad mai mare de control asupra finanțelor noastre.

De asemenea, aplicațiile web de planificare și organizare, cum ar fi Google Calendar și Todoist, ne ajută să ne organizăm programul și să ne gestionăm sarcinile într-un mod eficient. Aceste aplicații ne permit să setăm și să urmărim obiective, să stabilim priorități și să ne asigurăm că nu uităm nicio activitate importantă.

Totuși, trebuie să recunoaștem că web aplicațiile au adus schimbări uriașe în sistemul bancar. În ziua de azi este posibil de a face tranzacții în doar câteva secunde, iar banii cache sunt mult mai puțin utilizați. Băncile practic devin virtuale, ceea ce ne ușurează viața foarte mult. Dacă ne referim la sfera bancară, un rol din ce în ce mai important îl au MicroServiciile. În ultimii ani, conceptul de microservicii a devenit tot mai popular și influent în dezvoltarea aplicațiilor web. Această abordare arhitecturală a câștigat teren datorită beneficiilor pe care le aduce în dezvoltarea, implementarea și scalarea aplicațiilor. Microserviciile au început să devină populare odată cu nevoia de a dezvolta aplicații mai flexibile, scalabile și ușor de întreținut. Arhitecturile monolitice, care implicau construirea și implementarea unei aplicații ca un întreg monolit, au început să întâmpine provocări în fața cerințelor în continuă schimbare ale industriei IT. Pe măsură ce aplicațiile web au devenit mai complexe și au avut nevoie de scalabilitate și modularitate, dezvoltatorii au căutat soluții alternative. Microserviciile au apărut ca răspuns la această nevoie, promovând dezvoltarea aplicațiilor ca un set de servicii independente, fiecare gestionând o funcționalitate specifică.

Conform siteului microservices.io, „microserviciile reprezintă un stil arhitectural în care o aplicație este împărțită în componente mai mici și independente, numite servicii.” Aceste servicii sunt autonome și comunică între ele prin intermediul interfețelor bine definite, cum ar fi API-uri (Interfețe de Programare a Aplicațiilor). Caracteristica cheie a microserviciilor este faptul că fiecare serviciu este responsabil pentru o funcționalitate specifică și poate fi dezvoltat, implementat și scalat independent. Această abordare modulară permite dezvoltatorilor să lucreze în paralel și să se concentreze pe dezvoltarea serviciilor specifice, fără a afecta celelalte componente ale aplicației. Se spune că microserviciile reprezintă viitorul dezvoltării aplicațiilor web, și anume din mai multe motive:

* Scalabilitate și flexibilitate: „Microserviciile permit scalarea și dezvoltarea independentă a fiecărui serviciu în funcție de necesități.” Aceasta oferă flexibilitate în adaptarea aplicației la cerințele în schimbare și permite o gestionare eficientă a traficului și a volumului mare de date.
* Îmbunătățirea vitezei de dezvoltare: Arhitectura bazată pe microservicii permite echipei de dezvoltare să lucreze în mod autonom asupra serviciilor lor specifice, fără a afecta dezvoltarea celorlalte componente. Aceasta accelerează procesul de dezvoltare și implementare și permite lansarea rapidă a actualizărilor și îmbunătățirilor.
* Reziliență și stabilitate: Într-o arhitectură bazată pe microservicii, dacă un serviciu eșuează, celelalte servicii rămân în continuare disponibile. Aceasta asigură o reziliență sporită a aplicației și previne opririle complete în cazul unor probleme tehnice.
* Integrare ușoară și tehnologii variate: „Fiecare microserviciu poate fi dezvoltat folosind tehnologii diferite și poate fi integrat cu ușurință în aplicații existente. Acest lucru permite folosirea celor mai bune tehnologii și instrumente pentru fiecare serviciu în parte și favorizează abordarea orientată către domeniu.”
* Îmbunătățirea experienței utilizatorului: Microserviciile permit dezvoltarea de interfețe și servicii specializate pentru a îndeplini nevoile specifice ale utilizatorilor. Aceasta duce la o experiență personalizată și îmbunătățită pentru utilizatori.

Iată de ce, Microserviciile au devenit populare în dezvoltarea aplicațiilor. În general, se spune că acestea reprezintă viitorul dezvoltării aplicațiilor web, ducând la aplicații mai rapide, mai reziliente și mai ușor de întreținut. Cu avantajele lor în ceea ce privește scalabilitatea, viteza de dezvoltare și integrarea, microserviciile aduc o abordare modernă și eficientă în construirea aplicațiilor web de succes.

**Metode de implementare cu Java**

Web aplicatiile moderne pot fi create cu ajutorul limbajelor de programare de ultimă generație și cu un set divers de tehnologii. Acestea permit dezvoltatorilor să creeze aplicații web avansate, cu interfețe interactive și funcționalități complexe. Cu uneltele și framework-urile moderne disponibile, este posibil să se implementeze rapid și eficient funcționalități precum autentificare, comunicare în timp real și gestionare a bazelor de date. De asemenea, limbajele de programare moderne facilitează crearea de cod modular și ușor de întreținut, ceea ce duce la o dezvoltare eficientă și la o extensibilitate sporită a aplicațiilor. Prin utilizarea limbajelor de programare moderne și a tehnologiilor adecvate, se deschid numeroase posibilități pentru crearea de aplicații web inovatoare, rapide și scalabile. Astfel, dezvoltatorii pot profita de avantajele oferite de aceste limbaje și tehnologii, pentru a crea aplicații web de succes, capabile să satisfacă cerințele și așteptările utilizatorilor.

Crearea de aplicații web cu Java oferă o platformă puternică și flexibilă pentru dezvoltatori. Java este un limbaj de programare popular și este susținut de o vastă comunitate și o varietate de instrumente și framework-uri. Java este un limbaj de programare orientat pe obiect, creat de către Sun Microsystems (achiziționată ulterior de către Oracle Corporation) în anul 1995. A devenit extrem de popular în dezvoltarea aplicațiilor enterprise, mobile și web, fiind folosit pe scară largă în industrie. Evident, la momentul actual, există mii de limbaje de programare, dar ceea ce face Java să fie special sunt următoarele puncte forte pe care le posedă:

* Portabilitate: Una dintre caracteristicile cheie ale Java este portabilitatea sa. Codul scris în Java poate fi compilat într-un format intermediar cunoscut sub numele de "bytecode", care poate fi rulat pe orice platformă care dispune de o mașină virtuală Java (JVM). Aceasta înseamnă că aplicațiile Java pot fi rulate pe diverse sisteme de operare, fără a necesita modificări semnificative ale codului sursă.
* Obiecte și orientare pe obiect: Java este un limbaj de programare orientat pe obiect, ceea ce înseamnă că toate entitățile dintr-un program sunt reprezentate prin obiecte. Aceasta facilitează dezvoltarea de aplicații modulare, ușor de întreținut și extensibile.
* Platforma Java: Pe lângă limbajul de programare propriu-zis, Java include și o platformă extinsă cunoscută sub numele de Java Platform, Standard Edition (Java SE). Aceasta oferă biblioteci și framework-uri puternice, care facilitează dezvoltarea de aplicații desktop, web și mobile.
* Securitate: Java a fost proiectat cu un accent puternic pe securitate. Mașina virtuală Java (JVM) include mecanisme avansate de securitate, cum ar fi sandboxing-ul, care limitează accesul aplicațiilor Java la resursele sistemului, protejând astfel utilizatorii de eventualele amenințări.
* Comunitate puternică: Java beneficiază de o comunitate mare și activă de dezvoltatori. Există numeroase resurse online, forumuri și biblioteci open-source disponibile, care facilitează învățarea și dezvoltarea în Java.
* Utilizare extinsă: Java este folosit într-o varietate de domenii, inclusiv dezvoltarea de aplicații enterprise, aplicații mobile Android, servicii web, aplicații financiare și multe altele. Este un limbaj popular în industrie și oferă

oportunități ample de angajare pentru dezvoltatori Java.

Java rămâne unul dintre cele mai populare și influente limbaje de programare, datorită caracteristicilor sale puternice, portabilității și versatilității în dezvoltarea aplicațiilor de diferite tipuri și complexități.

Crearea unei aplicații web în Java include următorii pași:

* Configurarea mediului de dezvoltare: Pentru a începe dezvoltarea aplicațiilor web cu Java, este necesar să se configureze mediul de dezvoltare. Acesta constă în instalarea Java Development Kit (JDK) și a unui IDE (Integrated Development Environment) precum Eclipse, IntelliJ IDEA sau NetBeans. JDK oferă toate bibliotecile și instrumentele necesare pentru a dezvolta și executa aplicații Java.
* Selectarea unui framework web: Există numeroase framework-uri web disponibile pentru dezvoltarea aplicațiilor Java. Câteva exemple populare sunt Spring MVC, JavaServer Faces (JSF) și Play Framework. Aceste framework-uri oferă structuri și biblioteci care facilitează dezvoltarea aplicațiilor web, inclusiv gestionarea rutelor URL, interacțiunea cu baza de date și manipularea cererilor HTTP.
* Proictarea și implementarea aplicației:După ce s-a selectat framework-ul, următorul pas este proiectarea și implementarea aplicației. Acest lucru implică definirea arhitecturii aplicației, a structurii bazei de date și a logicii de business. De obicei, un framework web oferă un sistem de gestionare a dependențelor și un set de instrumente pentru a organiza codul și a facilita dezvoltarea.
* Crearea paginilor web: În aplicațiile web, interacțiunea cu utilizatorii se realizează prin intermediul paginilor web. În Java, se utilizează JavaServer Pages (JSP), JavaServer Faces (JSF) sau chiar JavaScript și HTML pentru a crea interfețele utilizator. Aceste tehnologii permit încorporarea logicii Java în paginile web și manipularea datelor.
* Implementarea funcționalităților: În funcție de cerințele aplicației, se implementează diferite funcționalități cum ar fi autentificarea utilizatorilor, gestionarea sesiunilor, interacțiunea cu baza de date, procesarea formularelor și gestionarea cererilor HTTP. Framework-ul ales va oferi adesea biblioteci și unelte pentru a facilita aceste sarcini.
* Testarea și depanarea: Este important să se testeze aplicația în mod regulat pentru a se asigura că funcționează corect și că nu există erori sau vulnerabilități. Se poate utiliza instrumente de testare automate, cum ar fi JUnit, și să se efectueze teste manuale pentru a verifica funcționalitatea și performanța aplicației.
* Implementarea și gazduirea: După ce s-a finalizat dezvoltarea și testarea aplicației, se poate trece la implementarea acesteia pe un server web. Există mai multe opțiuni pentru gazduirea aplicațiilor Java, cum ar fi Apache Tomcat, Jetty sau serverele de aplicații Java Enterprise Edition (Java EE).

Totuși, crearea de web aplicații folosind pur limbajul Java este extrem de dificlă, iată de ce se folosesc diferite tehnologii, ca framework-urile. Cel mai popular dintre acestea este Spring. Spring Framework este un framework Java open-source, conceput pentru a dezvolta și construi aplicații de înaltă calitate, robuste și scalabile. A fost creat inițial de către Rod Johnson în anul 2003 și de atunci a devenit unul dintre cele mai populare framework-uri pentru dezvoltarea aplicațiilor Java. Principiile pe care Spring Framework se bazeză, conform documentației oficiale, sunt:

* Inversiunea de control (IoC) și injecția de dependențe: „Unul dintre principiile de bază ale Spring Framework este inversiunea de control și injecția de dependențe. Aceasta presupune că obiectele nu mai sunt responsabile de crearea și gestionarea dependențelor lor, ci aceasta este preluată de către containerul Spring.” Acest mecanism permite o cuplare redusă între componentele aplicației și facilitează testarea unitară și modularitatea.
* Modularitate și flexibilitate: Spring Framework este conceput modular, ceea ce înseamnă că se pot folosi doar componentele necesare pentru aplicație. Acest lucru oferă flexibilitate în construcția și configurarea aplicației, iar dezvoltatorii pot beneficia de caracteristicile specificate fără a fi nevoiți să utilizeze întregul framework.
* Spring Boot: „Spring Boot este o extensie a Spring Framework care facilitează crearea rapidă și ușoară a aplicațiilor standalone. Aceasta integrează o configurație implicită și reduce nevoia de configurare manuală, permițând dezvoltatorilor să se concentreze mai mult pe logica aplicației decât pe aspectele de configurare.” Spring Boot aduce beneficii în dezvoltarea aplicațiilor web, microserviciilor și aplicațiilor bazate pe cloud.
* Gestionarea tranzacțiilor: Spring Framework oferă suport puternic pentru gestionarea tranzacțiilor în aplicații. Acesta integrează un mecanism de tranzacționare declarativă, care permite definirea și gestionarea tranzacțiilor prin intermediul adnotărilor sau fișierelor de configurare. Acest lucru face mai ușoară implementarea și gestionarea operațiunilor tranzacționale în aplicații.
* Integrare cu alte tehnologii: Spring Framework oferă integrare facilă cu alte tehnologii și framework-uri din ecosistemul Java. De exemplu, permite integrarea simplă a bazelor de date prin intermediul JDBC sau ORM-uri precum Hibernate, gestionează interacțiunea cu serviciile web prin intermediul RESTful APIs, oferă suport pentru autentificare și autorizare prin intermediul framework-ului Spring Security și multe altele.
* Suport pentru testare: Spring Framework facilitează testarea unitară și testarea integrată a aplicațiilor prin intermediul unor mecanisme și biblioteci dedicate. Acesta oferă suport pentru testarea componentelor individuale, testarea controlerelor web, simularea dependențelor și multe altele.

În principiu, Spring se bazează pe Front Controller. Front Controller este un pattern arhitectural utilizat în dezvoltarea aplicațiilor web pentru a gestiona cererile primite de la client și a distribui aceste cereri către componente specifice pentru prelucrare. În cadrul Spring Framework, Front Controller este implementat prin intermediul modulului Spring MVC (Model-View-Controller). Spring MVC oferă o soluție robustă și flexibilă pentru dezvoltarea aplicațiilor web bazate pe Front Controller. Implementarea Front Controller în Spring MVC, conform documentației oficiale, implică următorii pași:

* Definirea configurării: Pentru a utiliza Spring MVC, trebuie să se configureze aplicația prin intermediul unui fișier de configurare sau prin adnotări. Acesta poate specifica scanarea componentelor, setările de bază și interacțiunea cu alte module sau tehnologii.
* Definirea controlerului: „Controlerul reprezintă componenta responsabilă de gestionarea cererilor primite de la client. În Spring MVC, un controler poate fi definit prin intermediul unei clase care este marcată cu adnotarea “@Controller”. Această clasă poate conține metode care sunt marcate cu adnotarea „@RequestMapping” pentru a specifica ruta URL la care răspund”.
* Definirea metodelor de prelucrare a cererilor: Metodele din controler pot fi definite pentru a prelucra cererile primite de la client. Aceste metode pot primi parametri, cum ar fi parametrii URL sau datele trimise prin intermediul cererii, și pot returna modele sau vederi pentru a fi afișate în răspunsul către client.
* Configurarea vederilor: În Spring MVC, vederile sunt utilizate pentru a afișa datele către utilizator. Acestea pot fi în formate diferite, cum ar fi pagini JSP, șabloane Thymeleaf sau fișiere JSON/XML. Configurarea vederilor implică specificarea modului în care acestea trebuie să fie generate și afișate către utilizator.
* Configurarea modului de gestionare a cererilor: În Spring MVC, este necesară configurarea modului de gestionare a cererilor prin intermediul unui fișier de configurare sau prin adnotări. Acesta poate specifica metodele de gestionare a cererilor, intercepția cererilor și aplicarea regulilor de securitate.
* Lansarea aplicației: După ce configurarea este finalizată, aplicația poate fi lansată pe un server web care suportă Java, cum ar fi Apache Tomcat sau Jetty. Aceasta va trata cererile primite de la client și le va dirija către controlerul corespunzător în cadrul Front Controller-ului din Spring MVC.

Prin utilizarea Front Controller-ului în Spring MVC, dezvoltatorii pot beneficia de o gestionare eficientă a cererilor, o separare clară a responsabilităților și o arhitectură modulară și scalabilă. Spring Framework oferă numeroase facilități și funcționalități pentru a simplifica dezvoltarea aplicațiilor web bazate pe Front Controller și pentru a sprijini dezvoltatorii în crearea de aplicații robuste și flexibile.

Dar, în ziua de azi, aplicațiile web folosesc pe larg REST api. Spring REST (Representational State Transfer) este o parte importantă a framework-ului Spring care facilitează dezvoltarea serviciilor web RESTful. REST este un stil arhitectural utilizat pentru a construi aplicații web scalabile și interoperabile prin utilizarea protocoalelor și a operațiunilor HTTP. Spring REST oferă un suport puternic pentru crearea, consumarea și gestionarea serviciilor web RESTful. Iată câteva avantaje, pe care, conform documentației oficiale, ni le oferă Spring REST:

* Crearea serviciilor web RESTful: Cu ajutorul Spring REST, se pot defini ușor endpoint-uri pentru a crea servicii web RESTful. Acesta oferă adnotări precum “@RestController”, „@RequestMapping” și „@GetMapping” pentru a specifica rutele URL și metodele HTTP suportate de serviciu.

De exemplu, se poate defini un controller REST în Spring astfel:

„ @RestController

@RequestMapping("/api")

public class MyRestController {

@GetMapping("/resource")

public ResponseEntity<String> getResource() {

// Logica pentru returnarea resursei

}

} „.(Vezi Anexa 1 pentru un Controller complet)

* Serializarea și deserializarea datelor: „Spring REST facilitează serializarea și deserializarea automată a datelor în formatele JSON, XML sau alte formate acceptate de HTTP. Acest lucru se realizează prin intermediul convertorilor de mesaje (message converters) incluse în Spring, care transformă obiectele Java în formatul dorit și viceversa”.
* Gestionarea erorilor și codurilor de stare HTTP: Spring REST oferă mecanisme pentru gestionarea și returnarea codurilor de stare HTTP și a mesajelor de eroare în mod coerent și flexibil. Aceasta include adnotări precum „@ResponseStatus” pentru a specifica codurile de stare ale răspunsului, precum și posibilitatea de a defini handler-e pentru gestionarea excepțiilor și erorilor în mod centralizat.
* Validarea datelor de intrare: Spring REST integrează validarea datelor de intrare prin intermediul adnotărilor precum „@Valid” și „@RequestBody”. Aceste adnotări permit definirea regulilor de validare și oferă mecanisme pentru a verifica și gestiona erorile de validare într-un mod consistent.
* Documentarea serviciilor REST: Spring REST facilitează documentarea serviciilor web RESTful prin intermediul instrumentelor precum Swagger sau Spring REST Docs. Acestea permit generarea automată a documentației API-ului și a exemplelor de utilizare, ceea ce facilitează colaborarea cu echipele de frontend și documentarea clară a serviciilor oferite.
* Testarea serviciilor REST: Spring REST oferă suport pentru testarea unitară și testarea integrată a serviciilor REST prin intermediul bibliotecii MockMvc. Aceasta permite simularea cererilor HTTP și a răspunsurilor, facilitând testarea în izolare a serviciilor și a logicii asociate.

Acum apare necesitatea de a comunica cu o bază de date, și aici ne ajută o tehnologie numită Hibernate. Hibernate este un framework ORM (Object-Relational Mapping) pentru limbajul de programare Java, care facilitează interacțiunea cu bazele de date relaționale. A fost creat pentru a simplifica dezvoltarea aplicațiilor care utilizează persistența datelor, eliminând necesitatea scrierii de cod repetitiv și gestionarea manuală a operațiunilor de bază de date. Iată câteva aspecte-cheie despre Hibernate, care sunt oferite și de documentația oficială:

* Mapping obiect-relațional: „Hibernate facilitează maparea obiectelor Java către tabelele și coloanele din baza de date relațională. Acest lucru permite dezvoltatorilor să lucreze cu obiecte și să se concentreze pe logica de afaceri, fără a fi nevoie să se ocupe de detaliile de nivel inferior ale bazei de date. Maparea obiect-relațională se realizează prin intermediul fișierelor de configurare XML sau prin adnotări directe în clasele Java.”
* CRUD (Create, Read, Update, Delete):Hibernate oferă suport pentru operațiunile CRUD, precum crearea, citirea, actualizarea și ștergerea datelor din baza de date. Aceste operațiuni sunt simplificate prin intermediul metodelor și a interfețelor furnizate de Hibernate, fără a fi nevoie de scrierea manuală a SQL-ului.
* Lazy loading și eager loading: Hibernate gestionează automat încărcarea lenesă (lazy loading) și încărcarea imediată (eager loading) a datelor asociate. Aceasta înseamnă că obiectele asociate vor fi încărcate din baza de date numai atunci când sunt necesare, ceea ce poate îmbunătăți performanța aplicației prin reducerea cantității de date accesate.
* Interogări HQL și Criteria API: „Hibernate oferă un limbaj de interogare propriu numit HQL (Hibernate Query Language), care permite scrierea de interogări complexe într-un mod orientatpe obiecte”. De asemenea, Hibernate oferă și API-ul Criteria, care permite construirea programatică a interogărilor folosind expresii și criterii.
* Tranzacții: Hibernate gestionează tranzacțiile în mod automat, oferind suport pentru tranzacții atomice și izolate. Aceasta înseamnă că modificările făcute în cadrul unei tranzacții vor fi salvate în baza de date numai atunci când tranzacția este confirmată. Hibernate integrează suportul pentru tranzacții prin intermediul API-urilor Java, cum ar fi Java Transaction API (JTA) sau JDBC Transaction API.
* Caching: Hibernate oferă suport pentru caching, ceea ce permite stocarea în memorie a obiectelor citite din baza de date. Acest lucru poate îmbunătăți semnificativ performanța aplicației prin reducerea acceselor la baza de date și prin obținerea mai rapidă a datelor din cache.

Hibernate este un instrument puternic și versatil pentru gestionarea persistenței datelor în aplicațiile Java. Prin intermediul Hibernate, dezvoltatorii pot lucra cu obiecte Java familiare și pot beneficia de o abstracție eficientă a operațiunilor de bază de date.

Totuși, apare problema că web aplicația noastră va ncesita foarte multe resurse externe, cum ar fi librării deja implementate. Pentru a rezolva problema dată, se foloseste un instrument puternic numit Maven. Maven este o unealtă utilă de gestionare a proiectelor în limbajul de programare Java. A fost creat pentru a simplifica procesul de dezvoltare, construire și gestionare a dependențelor în aplicațiile Java. Maven se bazează pe conceptul de convenție peste configurare, ceea ce înseamnă că urmează un set de convenții și structuri implicite, eliminând nevoia de configurare manuală extensivă. Iată câteva aspecte-cheie pe care se bazează Maven și care sunt accentuate de compania Apache, care a creat tehnologia:

* Structura proiectului: „Maven impune o structură standardizată a proiectului, ceea ce facilitează colaborarea între dezvoltatori și menținerea unei organizări coerente”. Conform convențiilor Maven, resursele, codul sursă, testele și alte fișiere asociate sunt plasate în directoarele corespunzătoare, cum ar fi “src/main/java” pentru codul sursă principal și `src/test/java` pentru testele unitare.
* Fișierul de configurare pom.xml: Proiectele Maven sunt definite prin intermediul unui fișier de configurare XML numit „pom.xml” (Project Object Model). Acest fișier conține informații despre proiect, cum ar fi dependențele, plugin-urile utilizate, configurările specifice și alte detalii relevante. Pom.xml servește drept o sursă centralizată de informații și configurări pentru întregul proiect.
* Gestionarea dependențelor: „Maven facilitează gestionarea dependențelor proiectului. Prin intermediul fișierului pom.xml(Vezi Anexa 2), se pot specifica dependențele externe necesare pentru aplicație”. Maven va căuta și descărca automat dependențele necesare din repositoriile Maven central sau din alte repo-uri configurate. Acest lucru simplifică semnificativ procesul de adăugare și actualizare a dependențelor într-un proiect.
* Ciclul de viață al construirii: Maven definește un ciclu de viață al construirii standard, care include o serie de faze precum „clean”, „compile”, „test”, „package” și”`install’. Fiecare fază are un set de obiective (goal-uri) asociate care sunt executate în ordine specificată. Ciclul de viață al construirii asigură o abordare consistentă și repetabilă pentru generarea, testarea și distribuirea aplicației.
* Plugin-uri: Maven oferă suport pentru plugin-uri, care permit extinderea funcționalității și adăugarea de sarcini suplimentare în cadrul procesului de construire. „Maven vine cu un set de plugin-uri implicite, cum ar fi plugin-urile pentru compilare, testare, ambalare și distribuire. De asemenea, se pot utiliza plugin-uri personalizate pentru a satisface cerințele specifice ale proiectului”.
* Integrare cu mediile de dezvoltare: Maven este bine integrat cu mediile de dezvoltare populare, cum ar fi Eclipse și IntelliJ IDEA. Aceste IDE-uri oferă suport nativ pentru Maven, inclusiv instrumente pentru importul, actualizarea și construirea proiectelor Maven direct din interfața IDE-ului.

Maven a devenit o alegere populară în comunitatea Java datorită abordării sale coerente de gestionare a proiectelor și dependențelor. Simplifică procesul de construire și distribuire a aplicațiilor Java, permițând dezvoltatorilor să se concentreze mai mult pe logica de afaceri și mai puțin pe aspectele tehnice.

Pe de altă parte, HTML și CSS nu pot comunica direct cu componentele de backend, însă Java oferă e tehnologie numită JSP ce poate rezolva această problemă. JSP (JavaServer Pages) este o tehnologie Java utilizată pentru crearea de pagini web dinamice. Este o alternativă populară la scrierea de cod HTML static, deoarece permite incorporarea de cod Java în pagini web, ceea ce facilitează generarea de conținut dinamic în funcție de datele și logica de afaceri. Conform siteului de tutoriale tutorialspoint.com, JSP se bazează pe următoarele componente și principii logice:

* Separarea logicii de prezentare: „JSP permite separarea logicii de prezentare de codul Java prin intermediul șabloanelor. Codul HTML static poate fi inclus direct în fișierele JSP”, iar blocurile de cod Java pot fi inserate utilizând tag-urile „<% ... %>}. Aceasta permite dezvoltatorilor să lucreze cu ușurință cu ambele aspecte ale aplicației web. (Vezi Anexa 3)
* Etichete JSP: JSP furnizează un set de etichete predefinite (tag-uri) care facilitează interacțiunea cu codul Java și generarea de conținut dinamic. Aceste etichete pot fi utilizate pentru a itera prin liste de obiecte, a gestiona condiții, a include alte fișiere JSP și multe altele. De exemplu, eticheta „<c:forEach>” poate fi utilizată pentru a itera printr-o colecție și a afișa dinamic elementele în pagină.
* Expresii JSP: „Expresiile JSP permit afișarea valorilor sau rezultatelor de calcul într-un mod simplu și direct în pagină”. Aceste expresii pot fi utilizate pentru a afișa variabile, valori returnate de metode Java sau evaluări de expresii matematice. Expresiile JSP sunt scrise utilizând sintaxa „${...}”.
* Obiectul implicit request, session și application: JSP oferă acces la obiecte implicite precum „request”, „session” și „application”, care reprezintă contextul și starea curentă a aplicației web. Aceste obiecte pot fi utilizate pentru a accesa și manipula datele transmise între client și server sau pentru a partaja date între diferite componente ale aplicației.
* Suport pentru tehnologii și framework-uri: „JSP se integrează bine cu alte tehnologii și framework-uri Java, cum ar fi Servlets, JavaBeans și framework-ul Spring”. Aceasta permite dezvoltatorilor să construiască aplicații web complexe și scalabile utilizând o combinație de aceste tehnologii.
* Extensibilitate: JSP este o tehnologie extensibilă, permițând dezvoltatorilor să creeze propriile etichete JSP personalizate sau să utilizeze etichete JSP furnizate de alte biblioteci sau framework-uri. Aceasta facilitează dezvoltarea de funcționalități specifice și reutilizabile în cadrul proiectelor JSP.

Orice web aplicație trebuie neapărat să ruleze pe un server. Există foarte multe opțiuni, dar cea mai populară, care de altfel este folosită în acest proiect este Tomcat. Tomcat este un server web open-source și un container servlet, dezvoltat de Apache Software Foundation. Este una dintre cele mai populare soluții pentru implementarea aplicațiilor Java web, oferind suport pentru tehnologiile Java Servlet și JavaServer Pages (JSP). Tomcat, conform documentației oficiale, oferă următoarele avantaje:

* Server web și container servlet: „Tomcat acționează ca un server web, permițând deservirea de pagini și fișiere statice prin protocolul HTTP”. În același timp, este un container servlet care furnizează mediul de execuție pentru aplicațiile bazate pe tehnologiile Java Servlet și JSP. Acesta preia cererile HTTP primite și le direcționează către servlet-uri și pagini JSP corespunzătoare.
* Configurare simplă și ușurință în utilizare: Tomcat este cunoscut pentru configurația sa simplă și ușurința în utilizare. Configurarea se realizează prin intermediul fișierului „server.xml”, care permite specificarea porturilor, alocarea resurselor și alte setări relevante. În plus, „Tomcat vine cu o interfață web (Manager App) care permite administrarea și gestionarea aplicațiilor instalate”.
* Suport pentru specificațiile Java EE: „Tomcat implementează specificațiile Java Servlet și JSP, oferind un mediu pentru dezvoltarea și rularea aplicațiilor” Java web. Acesta respectă specificațiile Java EE (Enterprise Edition) și oferă suport pentru funcționalități avansate, cum ar fi pool-uri de conexiuni la baze de date, sesiuni distribuite și autentificare.
* Integrare cu alte tehnologii: Tomcat se integrează bine cu alte tehnologii Java și framework-uri populare. De exemplu, poate fi utilizat în combinație cu framework-ul Spring, Hibernate sau Struts pentru dezvoltarea aplicațiilor web. De asemenea, poate fi configurat pentru a funcționa împreună cu servere proxy invers, precum Apache HTTP Server.
* Scalabilitate și performanță: Tomcat oferă o scalabilitate și o performanță bune, permițând gestionarea unui număr mare de cereri și conexiuni simultane. Acesta utilizează un model de thread-uri pentru a gestiona cererile, permițând procesarea paralelă și eficientă a acestora. În plus, Tomcat include și suport pentru clustering, ceea ce permite distribuirea aplicațiilor pe mai multe instanțe Tomcat pentru a asigura redundanță și scalabilitate în medii de producție.
* Comunitate activă și suport: Tomcat beneficiază de o comunitate activă de dezvoltatori și utilizatori, care oferă suport și actualizări continue. Există o documentație bogată și numeroase resurse online disponibile pentru a ajuta dezvoltatorii să înceapă și să rezolve problemele întâmpinate în utilizarea Tomcat.

Totuși, trebuie să remarcăm că crearea unei web aplicații rareori e posibilă dintr-o singură încercare. Adesea, este necesar de a reveni la anumite configurații anterioare, de a le revedea, de a le combina și de a avea o copie de rezerva pentru fiecare versiune. Crearea unui folder sau arhive pentru fiecare versiune ar fi foarte costisitoare de timp și resurse, iată de ce, la crearea web aplicațiilor se utilizează pe larg tehnologii numite Version Control. Conform documentației oficiale a aplicației Git, “Version Control este un sistem care urmărește și gestionează modificările aduse într-un set de fișiere de-a lungul timpului. Este utilizat în dezvoltarea software pentru a permite colaborarea între mai mulți dezvoltatori și pentru a păstra istoricul schimbărilor făcute în codul sursă”. Un sistem de Version Control stochează o copie a întregului proiect și urmărește fiecare modificare adusă în fișierele acestuia. Acesta înregistrează informații precum cine a efectuat schimbarea, când a fost făcută și ce modificări specifice au fost adăugate sau eliminate. Aceste informații sunt stocate într-o bază de date sau într-un sistem distribuit, în funcție de tipul de VC utilizat. Există diferite tipuri de sisteme de Version Control, dar unul dintre cele mai populare și larg utilizate este Git. Git este un sistem de Version Control distribuit, ceea ce înseamnă că fiecare dezvoltator are o copie completă a întregului istoric al proiectului local pe propriul său calculator.

Git permite dezvoltatorilor să creeze branch-uri separate pentru a lucra la funcționalități noi sau la repararea problemelor, fără a afecta ramura principală a codului. Apoi, „aceste branch-uri pot fi integrate înapoi în ramura principală prin procesul de merging. În plus, Git utilizează un staging area sau index pentru a permite dezvoltatorilor să selecteze și să pregătească modificările pe care doresc să le commită”. Această funcționalitate permite o mai mare flexibilitate și control asupra modificărilor care vor fi incluse în următorul commit. Totodată, facilitează colaborarea între dezvoltatori prin posibilitatea de a partaja modificările între ei. Deci, dezvoltatorii pot sincroniza codul lor cu proiectul principal și pot rezolva conflicte în cazul în care mai mulți dezvoltatori au adus modificări incompatibile în același fișier. Aceste aspecte permit Git să păstreaze un istoric complet al schimbărilor aduse într-un proiect, inclusiv informații despre autor, dată și mesajul asociat fiecărei modificări. Aceasta permite o vizualizare detaliată a evoluției proiectului și o ușurință în identificarea și corectarea erorilor. Defapt, Git oferă o gamă largă de comenzi și funcționalități, care permit dezvoltatorilor să lucreze eficient cu sistemul de Version Control. Este un instrument puternic și versatil, utilizat în mod frecvent de echipe de dezvoltare software de diferite dimensiuni și în diferite domenii.

**Concluzii**

Efecutarea tezei de an despre crearea unei aplicații web pentru un restaurant a fost o experiență extrem de valoroasă pentru mine. Această sarcină mi-a oferit oportunitatea de a dezvolta noi abilități și cunoștințe în domeniul dezvoltării aplicațiilor web. În primul rând, am învățat să utilizez diverse tehnologii și framework-uri pentru a construi o aplicație web funcțională. În al doilea rând, am lucrat cu limbajul de programare Java și am explorat framework-uri precum Spring și Hibernate, care sunt esențiale în dezvoltarea aplicațiilor web robuste și scalabile.

Pe de altă parte, pe parcursul proiectului, am avut ocazia de a învăța despre arhitectura aplicațiilor web, inclusiv concepte precum Front Controller, Spring MVC și REST. Am înțeles importanța separării logicii de prezentare și a logicii de afaceri și am aplicat aceste principii în dezvoltarea aplicației mele.

De asemenea, am învățat să utilizez baze de date relaționale și să implementez persistența datelor în aplicație prin intermediul framework-ului Hibernate și a tehnologiei JPA. Am înțeles importanța unei baze de date bine proiectate și am implementat scheme adecvate pentru gestionarea informațiilor despre meniu, rezervări și alte aspecte specifice unui restaurant.

Un alt aspect valoros al acestui proiect a fost experiența practică în gestionarea dependențelor și construirea aplicației cu ajutorul Maven. Am învățat să configurez fișierul pom.xml și să adaug dependențele necesare pentru proiectul meu, facilitând astfel dezvoltarea și actualizarea aplicației.

Pe lângă aspectele tehnice, această teză de mi-a permis să învățat, de asemenea, despre importanța planificării și organizării unui proiect. Am învățat să îmi structurăm codul și să utilizez cele mai bune practici de programare pentru a obține un cod clar, modular și ușor de întreținut.

Mai mult ca atât, studierea diferitor resurse mi-a permis să aflu că web aplicațiile sunt indispensabile în industria HORECA, în special în cazul restaurantelor. Acestea aduc o serie de beneficii semnificative pentru atât proprietarii de restaurante, cât și pentru clienți. Pentru proprietarii de restaurante, o aplicație web oferă o modalitate eficientă de a interacționa cu clienții și de a gestiona diferite aspecte ale afacerii lor. Prin intermediul unei aplicații web, proprietarii pot afișa meniul, programul de funcționare și informații despre locație, facilitând accesul și vizibilitatea pentru potențialii clienți. De asemenea, aplicația web poate permite clienților să facă rezervări online, eliminând nevoia de apeluri telefonice sau vizite personale în restaurant. Aceasta poate ajuta la gestionarea eficientă a fluxului de clienți și la evitarea supraaglomerării sau a timpilor de așteptare prelungiți. În plus, o aplicație web poate oferi proprietarilor de restaurante informații valoroase despre preferințele clienților, feedback-ul și recenziile acestora. Prin intermediul aplicației, pot fi colectate date relevante care pot fi utilizate pentru a îmbunătăți experiența clienților și pentru a adapta meniul sau serviciile în funcție de nevoile lor. Am concluzionat că web aplicațiile sunt indispensabile în industria HORECA, mai ales pentru restaurante. Acestea facilitează interacțiunea dintre restaurant și clienți, îmbunătățesc eficiența operațională și contribuie la o experiență mai bună pentru toți cei implicați. Un restaurant care adoptă o aplicație web beneficiază de un avantaj competitiv semnificativ și poate oferi o experiență mai modernă și personalizată pentru clienți.

Pe de altă parte, m-am convins că Java este o alegere excelentă pentru crearea de aplicații web datorită puterii și flexibilității sale. Cu toate acestea, trebuie menționat că învățarea Java poate fi o provocare pentru mulți începători. Java este un limbaj de programare robust și complex, care necesită o înțelegere profundă a conceptelor și a structurii sale. Este necesară o investiție semnificativă de timp și efort pentru a învăța sintaxa Java, principiile de bază ale programării orientate pe obiect și alte concepte avansate asociate acestuia. Cu toate acestea, dificultatea de a învăța Java nu trebuie să descurajeze pe nimeni. Există o mulțime de resurse disponibile, cum ar fi tutoriale, documentație oficială și comunități online active, care pot ajuta la învățarea Java. În plus, există numeroase cadre de lucru și biblioteci Java, cum ar fi Spring, Hibernate și Maven, care simplifică dezvoltarea aplicațiilor web și reduc complexitatea. Odată ce obstacolele inițiale sunt depășite și bazele Java sunt înțelese, avantajele devin evidente. Java oferă un nivel ridicat de portabilitate, permițând dezvoltarea aplicațiilor care rulează pe o gamă largă de platforme. De asemenea, Java oferă o performanță excelentă și scalabilitate, făcându-l potrivit pentru aplicații web de dimensiuni mari și complexe.

În plus, Java are o comunitate activă și o industrie vastă, ceea ce înseamnă că există oportunități de carieră abundente pentru dezvoltatorii Java. Aprofundarea cunoștințelor și experiența în dezvoltarea de aplicații Java web pot fi extrem de valoroase și pot deschide uși în diverse industrii și companii. Deci, în ciuda dificultăților inițiale de învățare, investiția în învățarea limbajului Java pentru dezvoltarea de aplicații web merită efortul. Cu timp și răbdare, această limbă poate deveni un instrument puternic în mâinile unui dezvoltator, oferindu-i posibilitatea de a crea aplicații web complexe, scalabile și robuste.

Din păcate, în timpul realizării tezei de an, am întâmpinat anumite dificultăți în implementarea Spring AOP. Această tehnologie a fost complet nouă pentru mine, iar procesul de învățare și înțelegere a sa a necesitat timp suplimentar de studiu și explorare. Spring AOP (Aspect-Oriented Programming) este o abordare de programare care permite separarea preocupărilor transversale, cum ar fi logarea, gestionarea tranzacțiilor sau securitatea, de logica de bază a aplicației. Implementarea cu succes a Spring AOP implică înțelegerea conceptelor-cheie, cum ar fi aspecte, join points, pointcuts și advice-uri. În timpul studiului tehnologiei Spring AOP, am descoperit că necesită o perspectivă diferită asupra programării și o înțelegere aprofundată a conceptelor și a modului în care acestea sunt integrate în cadrul aplicației. Am realizat că trebuie să identific corect punctele de alăturare (join points) și să creez puncte de tăiere (pointcuts) pentru a aplica consiliere (advice) în locurile potrivite din codul aplicației. În plus, configurarea corectă a Spring AOP necesită familiarizarea cu aspectele și notarea corectă a acestora în fișierele de configurare. Această parte a implementării poate fi sensibilă și poate necesita un proces de depanare și ajustare pentru a obține comportamentul dorit. Iată de ce, Spring AOP nu a fost implementat în proiectul final.

Cu toate acestea, în ciuda provocărilor întâmpinate, am privit această experiență ca pe o oportunitate de a învăța și de a crește în domeniul programării. Am înțeles că este normal să întâmpini dificultăți în implementarea unei tehnologii noi și că este nevoie de răbdare și perseverență pentru a înțelege și a aplica conceptele în mod corect.

Astfel, am continuat să studiez și să explorez Spring AOP în timpul realizării tezei mele de an, fiind conștient de faptul că o înțelegere solidă și o experiență practică în această tehnologie vor fi benefice pentru viitoarele proiecte și dezvoltarea mea ca programator. Această experiență m-a motivat să continui să explorez și să învăț mai mult despre Spring AOP și alte tehnologii relevante, pentru a-mi dezvolta cunoștințele și abilitățile în domeniul programării.

În final, finalizarea acestei teze de an mi-a oferit satisfacția de a avea un proiect complet funcțional și gata făcut. A fost o oportunitate de a aplica cunoștințele acumulate într-un context real și de a-mi demonstra abilitățile în dezvoltarea de aplicații web. Efecutarea tezei de an despre crearea unei aplicații web pentru un restaurant a fost o experiență valoroasă, care mi-a permis să capăt noi abilități, cunoștințe și să am un proiect gata făcut. Această experiență m-a pregătit și m-a înarmat cu cunoștințele necesare pentru a aborda și a rezolva provocări în dezvoltarea de aplicații

**Bibliografie**

* [8 Big Advantages of Using MySQL | Datamation](https://www.datamation.com/storage/8-major-advantages-of-using-mysql/)
* [Forța de muncă în Republica Moldova: ocuparea și șomajul în trimestrul II 2020 (gov.md)](https://statistica.gov.md/ro/forta-de-munca-in-republica-moldova-ocuparea-si-somajul-in-9430_3341.html)
* [Frontend vs Backend - GeeksforGeeks](https://www.geeksforgeeks.org/frontend-vs-backend/)
* [Git - About Version Control (git-scm.com)](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control)
* [HoReCa - tendințele și provocările unui domeniu esențial din industria ospitalității - International Work Finder (work-finder.eu)](https://work-finder.eu/horeca-tendinte-provocari/)
* [Hibernate Getting Started Guide (jboss.org)](https://docs.jboss.org/hibernate/orm/6.2/quickstart/html_single/)
* [How Has Covid-19 Affected Restaurants | City National Bank (cnb.com)](https://www.cnb.com/business-banking/media/looking-forward/how-the-pandemic-affected-restaurants.html)
* [JSP - Quick Guide (tutorialspoint.com)](https://www.tutorialspoint.com/jsp/jsp_quick_guide.htm)
* [Maven – Maven Getting Started Guide (apache.org)](https://maven.apache.org/guides/getting-started/index.html)
* [Numărul turiștilor care vizitează Moldova prin intermediul agențiilor de turism, în creștere (noi.md)](https://noi.md/md/societate/numarul-turistilor-care-viziteaza-moldova-prin-intermediul-agentiilor-de-turism-in-crestere)
* [Server (computing) - Wikipedia](https://en.wikipedia.org/wiki/Server_(computing))
* [Spring Data JPA - Reference Documentation](https://docs.spring.io/spring-data/jpa/docs/current/reference/html/)
* [Spring Framework Reference Documentation](https://docs.spring.io/spring-framework/docs/4.3.x/spring-framework-reference/htmlsingle/)
* [Spring REST Docs](https://docs.spring.io/spring-restdocs/docs/3.0.0/reference/htmlsingle/)
* [Web Application Architecture: Working,Components,Types,Trends 2022 (thinksys.com)](https://www.thinksys.com/development/web-application-architecture-complete-guide/)
* [Web application - Wikipedia](https://en.wikipedia.org/wiki/Web_application)
* [Web MVC framework (spring.io)](https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html)
* [What are microservices?](https://microservices.io/)
* [What is a REST API? | IBM](https://www.ibm.com/topics/rest-apis)
* [What Is Cloud Computing? | Microsoft Azure](https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-cloud-computing)
* [16 restaurante din Moldova unde poți mânca delicios când ești într-o călătorie - #diez](https://diez.md/2021/03/04/16-restaurante-din-moldova-unde-puteti-manca-delicios-cand-esti-intr-o-calatorie/)

**Anexe**

**Anexa 1**

Exemplu de REST API prin REST Controller

@RestController  
@RequestMapping("/api")

public class RestAPIMethods {  
 private static final Logger logger = LoggerFactory.getLogger(RestAPIMethods.class);  
 public static List<Customer> getCustomers(){  
 SessionFactory sessionFactory = new Configuration().configure().addAnnotatedClass(User.class).addAnnotatedClass(Customer.class).buildSessionFactory();  
 logger.info("A sessionFactory was successfully created.");  
 Session session = sessionFactory.getCurrentSession();  
 logger.info("Session was successfully created.");  
  
 session.beginTransaction();  
 logger.info("Transaction for session was successfully started");  
  
 Query query = session.createQuery("SELECT firstName,lastName,email,job,orders FROM Customer ORDER BY firstName,lastName");  
  
 List<Customer> customers = query.getResultList();  
 session.getTransaction().commit();  
 logger.info("Session was successfully committed");  
 session.close();  
 logger.info("Transaction for session was successfully closed");  
  
 return customers;  
 }  
 public static Customer getCustomer(int id){  
 SessionFactory sessionFactory = new Configuration().configure().addAnnotatedClass(User.class).addAnnotatedClass(Customer.class).buildSessionFactory();  
 logger.info("A sessionFactory was successfully created.");  
 Session session = sessionFactory.getCurrentSession();  
 logger.info("Session was successfully created.");  
  
 session.beginTransaction();  
  
 Customer customer = session.get(Customer.class,id);  
 session.getTransaction().commit();  
 logger.info("Session was successfully committed");  
 session.close();  
 logger.info("Transaction for session was successfully closed");  
  
 return customer;  
 } }

**Anexa 2**

Exemplu de configurație pentru fișierul pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>Project</artifactId>  
 <packaging>war</packaging>  
 <version>1.0-SNAPSHOT</version>  
 <name>Project Maven Webapp</name>  
 <url>http://maven.apache.org</url>  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>3.8.1</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.0.31</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.validation</groupId>  
 <artifactId>validation-api</artifactId>  
 <version>2.0.1.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>org.hibernate.validator</groupId>  
 <artifactId>hibernate-validator</artifactId>  
 <version>6.0.21.Final</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>3.1.0</version>  
 <scope>provided</scope>  
 </dependency>

**Anexa 3**

Exemplu de configurare a unei pagini JSP

<!DOCTYPE HTML>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  
<%@taglib prefix="form" uri="http://www.springframework.org/tags/form"%>  
<%@page isELIgnored="false" %>  
  
<html>  
  
<head>  
 <title>Login</title>  
 <meta name="viewport" content="width=device-width, initial-scale=1.0" />  
 <meta charset="UTF-8"/>  
 <link rel="stylesheet" type="text/css" href="<c:url value="/resources/CSS/styles.css"/>">  
</head>  
<body>  
  
  
<nav class>  
 <div class="buttons">  
 <a href="home">Home</a>  
 <a href="about">About Us</a>  
 <a href="order">Order</a><img src="<c:url value="/resources/Images/buy.png"/>" class="icon" alt="Img"/>  
 <a href="login">Log in</a>  
 <a href=""><img src="<c:url value="/resources/Images/facebook.png"/>" class="facebook" alt="Img"/></a>  
 <a href=""><img src="<c:url value="/resources/Images/instagram.png"/>" class="instagram" alt="Img"/></a>  
 </div>  
  
</nav>

.......