

**FACULTATEA TEHNOLOGII INFORMAȚIONALE**

**ȘI STATISTICĂ ECONOMICĂ**

**DEPARTAMENTULTEHNOLOGIA INFORMAȚIEI**

**ȘI MANAGEMENT INFORMAȚIONAL**

**PROIECT DE AN**

**Tema: Proiectarea unei aplicații pentru comandarea și livrarea bucatelor de la un restaurant.**

**Specialitatea 0613.5 Informatică Aplicată**

**Autor:**

Student gr. INFa-211,

*învăţământ cu frecvenţă*

**Mititiuc Eduard**

*(semnătura)*

**Conducător ştiinţific:**

lector. univ., dr. **Lozan Victoria**

*(semnătura)*

**Chișinău – 2023**

Cuprins

[INTRODUCERE 3](#_Toc135671061)

[1. REVOLUȚIA DIGITALĂ DIN DOMENIUL HORECA 5](#_Toc135671062)

[1.1 Concepte generale despre domeniul HORECA 5](#_Toc135671063)

[1.2 Evoluția domeniului HORECA 9](#_Toc135671064)

[1.3 Digitalizarea domeniului HORECA 12](#_Toc135671065)

[2. EXTINDEREA WEB APLICAȚIILOR PRINTRE RESTAURANTE 18](#_Toc135671066)

[2.1 Istoria și structura web aplicațiilor 18](#_Toc135671067)

[2.2 Importanța aplicațiilor web 25](#_Toc135671068)

[2.3 Metode de implementare cu Java 27](#_Toc135671069)

[CONCLUZII 35](#_Toc135671070)

[BIBLIOGRAFIE 38](#_Toc135671071)

[Anexe 39](#_Toc135671072)

### 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.

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 situația 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 că web aplicațiile deja au devenit o adevărată necesitatea pentru industria HORECA.

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 prezent, nu mai există oficial.

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.

- Capitolul 2: Implementarea web aplicațiilor este destul de dificilă și necesită foarte multe cunoștințe de programare.

### 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.

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, mediteraneean 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 modest 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.

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 atât, 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.

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 site-ului 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 site-ului 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, conform 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.

Dar, e foarte important să remarcăm că chiar și turiștii apreciază restaurantele din Republica 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 gastro-turism, 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. 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.

### 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.

Evoluția domeniului HORECA în Republica Moldova a cunoscut schimbări semnificative în ultimele decenii, odată cu procesul de modernizare economică și deschiderea către influențele occidentale. Domeniul HORECA a devenit din ce în 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, permițând 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.

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. De altfel, întregul domeniu HORECA are o tendință 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. 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, avansurile 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. 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. În plus, 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. 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. 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. 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.

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.

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.

- 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.

- 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.

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.

- 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.

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.

### 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.

Întâi 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 Java Script 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.

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.

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 Java Script, și pot fi accesate de pe diferite dispozitive conectate la internet.

Caracteristicile specifice 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.

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.

- 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.

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 Java Script pentru implementarea funcționalităților interactive și manipularea datelor. Un dezvoltator frontend are rolul de a traduce designul și cerințele utilizatorului în codul HTML, CSS și Java Script 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 browser, având în vedere aspecte precum responsive design (design adaptiv la diferite dimensiuni de ecran) și optimizarea performanței. 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. HTML este interpretat de către browser și afișat utilizatorului ca o pagină web interactivă

- Java Script: Java Script este un limbaj de programare care adaugă interactivitate și funcționalități avansate la o aplicație web. Cu ajutorul Java Script, 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.

Este important de menționat că anume aceste tehnologii au fost folosite pentru a crea propria web aplicație, așa cum ele sunt suficient de puternice pentru a crea o pagină web funcțională și bine aranjată. Astfel, pagina de start a aplicației este realizată cât mai simplist, oferind câteva butoane pentru a permite user-ului să întreprindă diferite acțiuni. Deci, în imaginea 2.1 (Anexa 1), odată ce user-ul dă click pe butonul “about us”, se va deschide o pagină nouă, ce poate fi văzută în imaginea 2.2 (Anexa 1)ce conține informații legate de site și restaurant care ar putea să-i fie utile pentru a cunoaște date generale despre funcționarea restaurantului.

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. În general, 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. Termenul de back-end face trimitere directă la următoarele 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.

Deoarece datele sunt foarte prețioase, orice web aplicație necesită o securitate foarte bine pusă la punct. Securitatea unei web aplicații 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.

Pe de altă parte, vedem că o web aplicație ar fi imposibilă fără un server. Serverele web 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.

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.

- Securitatea și confidențialitatea datelor: Serverele web joacă un rol crucial în protejarea datelor utilizatorilor.

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.

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.

- 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.

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, se poate 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.

- 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.

Totuși, pentru persoanele fizice, serviciile cloud încă rămân a fi destul de scumpe, motiv pentru care proiectul propriu a fost realizat pe memoria internă a calculatorului personal. Deci, fișierele de configurație, serverul, baza de date și toate instrumentele necesare au fost instalate local, ceea ce este suficient pentru a rula aplicația local.

Dar, cel mai important component al oricărei web aplicații rămân 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.

- 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.

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. De fapt, web aplicația creată folosește protocoale speciale pentru a comunica cu MySql, existând o dependență “com.mysql” ce ne oferă un Driver ce este capabil să creeze conexiuni cu MySql direct. În general, aplicația folosește 4 tabele create în baza de date. Primul tabel (Imaginea 2.3 Anexa 2)are denumirea “customer” și conține datele personale ale unui client, date ce sunt necesare pentru a efectua tranzacții. Mai apoi, există un tabel numit “user\_detail”(Imaginea 2.4 Anexa 2) care conține datele clienților pentru a se putea conecta în web aplicație, printre care numele de utilizator și parola acestuia. Este evident că în cazul unei web aplicații reale, parola va necesita un algoritm de criptare, dar folosirea unui asemenea mecanism de criptare necesită tehnologii avansate, de aceea nu a fost utilizat în cadrul acestui proiect. După, urmează tabelul „product” (Imaginea 2.5 Anexa 2) care conține produsele ce sunt comercializate de către restaurant, prețurile etc. Anume aceste date vor fi preluate pentru a putea crea lista de produse din meniu. În final, avem tabelul “Order\_list”(Imaginea 2.6 Anexa 2), care stochează comenzile efectuate de clienți.

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.

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.

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 site-ului 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. Iată de ce, Microserviciile au devenit populare în dezvoltarea aplicațiilor. 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.

### 2.3 Metode de implementare cu Java

Web aplicațiile 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.

Totuși, crearea de web aplicații folosind pur limbajul Java este extrem de dificilă, 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 bazează, 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.

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. În proiectul dat, configurarea acestui fișier poate fi văzută în Anexa 3

- 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”. O parte din Controllerul creat în proiect poate fi vizualizată în Anexa 4

- 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.

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 (Vezi Anexa 5 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”. De exemplu, folosirea endpoint-ului “/api/clients” va returna in forma JSON lista e clienți a restaurantului, dar, evident, va oferi doar date ce nu sunt sensibile. Un exemplu de rest request poate fi văzut în imaginea 2.7 din Anexa 6.

- 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. Este evident că un utilizator poate face anumite greșeli la introducerea datelor, de aceea au fost implementate sisteme de validare a datelor. De exemplu, în imaginea 2.8, din Anexa 7, putem vedea un exemplu de validare a formei în care user-ul introduce datele pentru a se înregistra. De asemenea, în imaginea 2.9, din Anexa 7, este vizibil că la apariția unei probleme de validare, user-ul va fi informat de aceasta prin interfața grafică.

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.” Efectuarea mapării se face prin crearea claselor, marcate ca entități prin intermediul anotației “@Entity”. Un exemplu al unei asemenea entități poate fi vizibil în Anexa 8.

- 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.

- 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 orientat pe 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. De exemplu, o tranzacție este efectuată când user-ul efectuează o comandă(Imaginea 2.10 Anexa 9), căci baza de date trebuie să înregistreze o comandă și produsele comandate.

Totuși, apare problema că web aplicația noastră va necesita foarte multe resurse externe, cum ar fi librării deja implementate. Pentru a rezolva problema dată, se folosește 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.

- Gestionarea dependențelor: „Maven facilitează gestionarea dependențelor proiectului. Prin intermediul fișierului pom.xml(Vezi Anexa 10), 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.

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 (Java Server 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 site-ului 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 11)

- 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. Această etichetă este utilizată când oferim o listă de date, fără a ști numărul de elemente prezent în ea.

- 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 „${...}”. De exemplu, pe pagina de profil(Imaginea 2.11 Anexa 12), clientul poate vedea datele propriului cont, și dacă dorește, le poate modifica. Utilizând pur HTML și CSS, ar fi imposibil să comunicăm cu Front Controllerul, dar JSP poate face acest lucru prin tag-urile și simbolurile speciale.

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”.

- 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.

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. Mai mult ca atât, 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. De fapt, 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. Este important să evidențiem că commit-urile conțin doar modificări realizate de noi în proiect, deci putem spune că proiectul este oarecum divizat în mai multe părți, așa cum poate fi văzut de exemplu în Imaginea 2.12 Anexa 13

### CONCLUZII

Efectuarea 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. Efectuarea 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

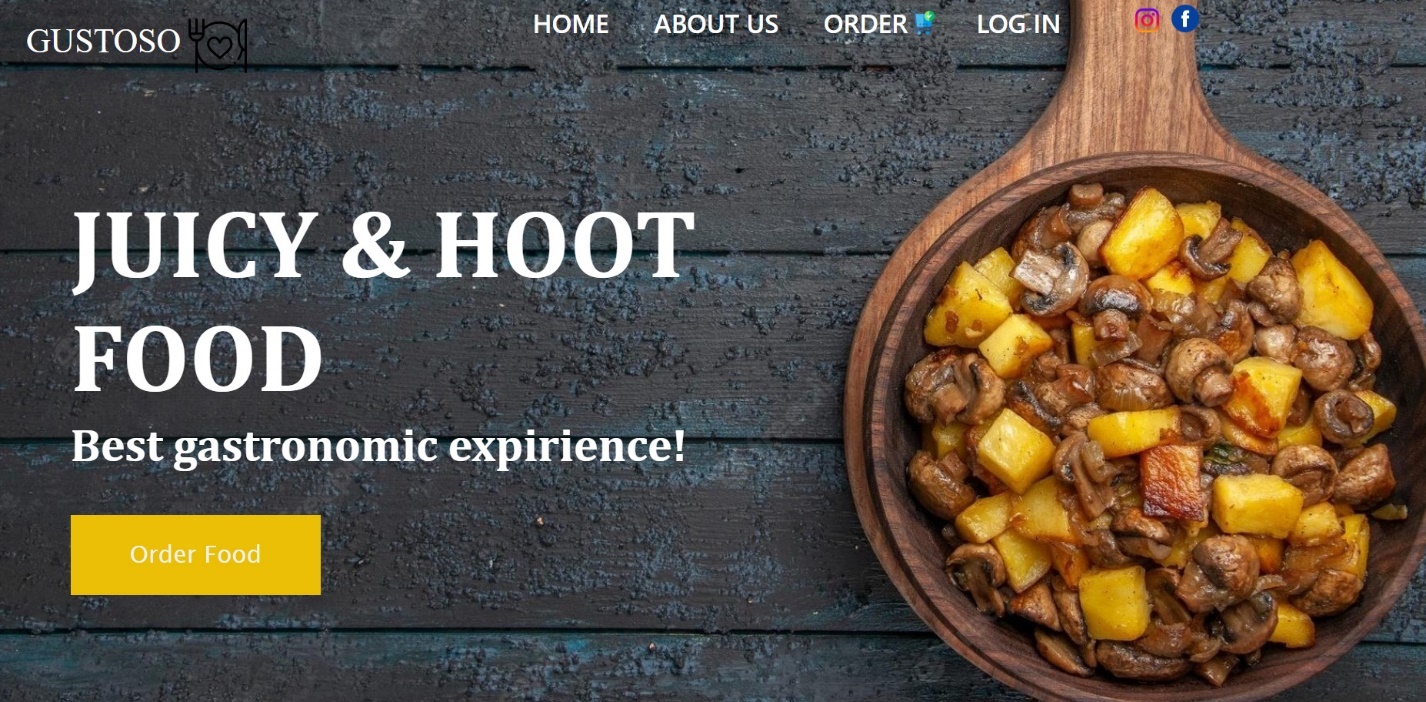
### BIBLIOGRAFIE

* [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/)

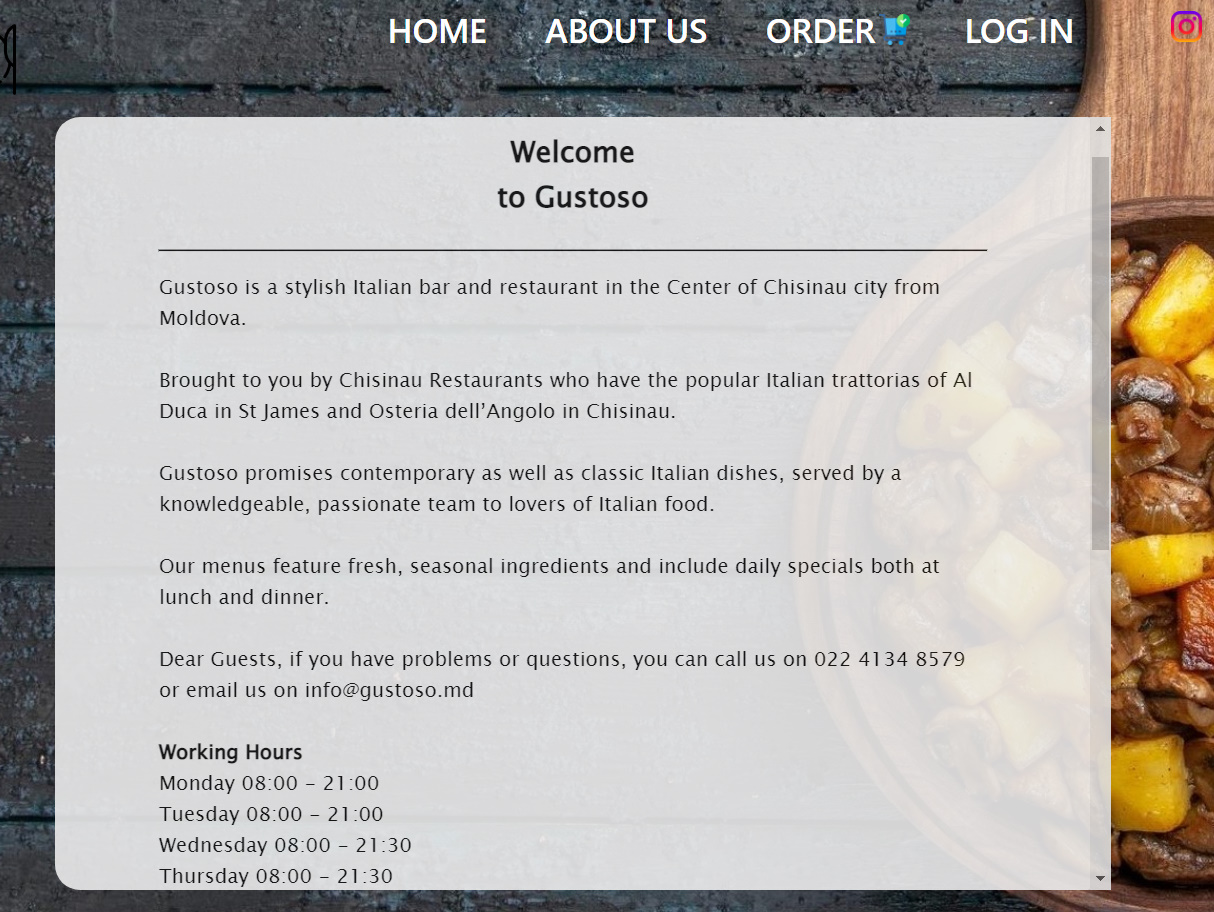
### Anexe

Anexa 1

Imaginea 2.1

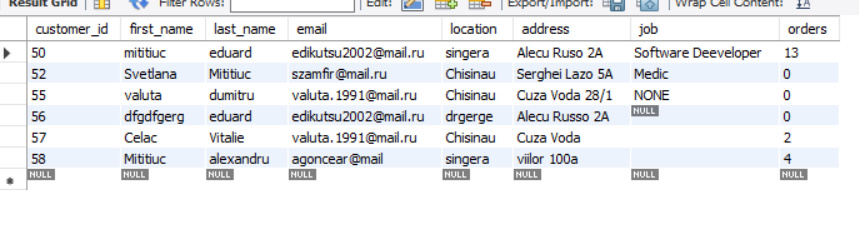


Imaginea 2.2

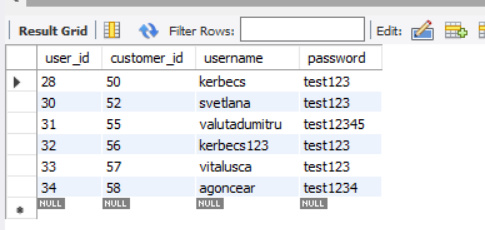


Anexa 2

Imaginea 2.3



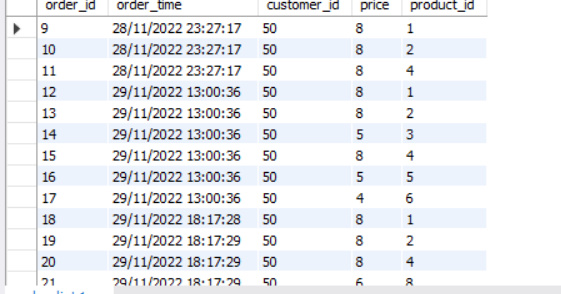
Imaginea 2.4



Imaginea 2.5



Imaginea 2.6



Anexa 3

<!DOCTYPE web-app PUBLIC  
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
 "http://java.sun.com/dtd/web-app\_2\_3.dtd" >  
  
<web-app>  
 <display-name>Archetype Created Web Application</display-name>  
 <context-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath\*:applicationContext.xml</param-value>  
 </context-param>  
 <listener>  
 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  
 </listener>  
 <servlet>  
 <servlet-name>dispatcher</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
   
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath\*:applicationContext.xml</param-value>  
 </init-param>  
   
 <load-on-startup>1</load-on-startup>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>dispatcher</servlet-name>  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
</web-app>

Anexa 4

@Controller  
@Validated  
@SessionAttributes(value = {"globalUser","infoUser","registerTest"})  
public class ControllerConfig {  
 private AnnotationConfigApplicationContext config = new AnnotationConfigApplicationContext(appContext.class);  
 private final Logger logger = LoggerFactory.getLogger(ControllerConfig.class);  
 @ModelAttribute("globalUser")  
 public User globalUser(){  
 User user = config.getBean("user",User.class);  
 user.setCustomer(config.getBean("customer", Customer.class));  
 return user;  
 }  
 @ModelAttribute("registerTest")  
 public UserRegistration userRegistration(){  
 return config.getBean("userRegistration",UserRegistration.class);  
 }  
 @RequestMapping("/")  
 public String showMainPage(@ModelAttribute("globalUser")User user,Model model, @ModelAttribute("registerTest")UserRegistration userLogged){  
 if(userLogged.isLogged()) {  
 model.addAttribute("userLogged",user);  
 return "homeLogged";  
 }  
 return "home";  
 }  
 @RequestMapping("/home")  
 public String showMainPage2(@ModelAttribute("globalUser")User user,Model model, @ModelAttribute("registerTest")UserRegistration userLogged){  
 if(userLogged.isLogged()) {  
 model.addAttribute("userLogged",user);  
 return "homeLogged";  
 }  
 return "home";  
 }  
 @RequestMapping("/login")  
 public String showLoginPage(@ModelAttribute("globalUser")User user, Model model,@ModelAttribute("registerTest")UserRegistration userLogged){  
 if(!userLogged.isLogged()) {  
 model.addAttribute("userTest", user);  
 return "login";  
 }  
 else {  
 return "home";  
 }  
 }  
}

......

Anexa 5

@RestController  
@RequestMapping("/api")  
public class RestControllerConfig {  
 private static final Logger logger = LoggerFactory.getLogger(RestControllerConfig.class);  
 @GetMapping("/customers")  
 public List<Customer> getCustomers(){  
 List<Customer> customers = RestAPIMethods.getCustomers();  
 if(customers==null) throw new ItemNotFoundException("No customers were found",HttpStatus.NOT\_FOUND);  
 return customers; }  
 @GetMapping("/customers/{id}")  
 public Customer customer(@PathVariable int id){  
 Customer customer = RestAPIMethods.getCustomer(id);  
 if(customer==null)  
 throw new ItemNotFoundException("Can't find customer with id "+id,HttpStatus.NOT\_FOUND);  
 return customerș}  
 @GetMapping("/customers/{firstName}/{lastName}")  
 public List<Customer> customerName(@PathVariable String firstName,@PathVariable String lastName){  
 List<Customer> customers = RestAPIMethods.getCustomer(firstName,lastName);  
 if(customers==null)  
 throw new ItemNotFoundException("Can't find customer(s) named "+firstName+" "+lastName,HttpStatus.NOT\_FOUND);  
 return customers;}  
 @GetMapping("/products")  
 public List<Product> getProducts(){  
 List<Product> productList = RestAPIMethods.getProducts();  
 if(productList==null)  
 throw new ItemNotFoundException("Can't find any products",HttpStatus.NOT\_FOUND);  
 return productList;}  
 @GetMapping("/orders/{id}")  
 public Order getOrder(@PathVariable int id){  
 Order order = RestAPIMethods.getOrder(id);  
 if(order==null)  
 throw new ItemNotFoundException("Can't find order with id "+id,HttpStatus.NOT\_FOUND);  
 return order;}  
 @PostMapping("/customers")  
 public Customer addCustomer(@RequestBody User user){  
 user.setUser\_id(0);  
 if(!RestAPIMethods.addUser(user)){  
 throw new ItemNotFoundException("Something is wrong with your data. Verify your data and try again", HttpStatus.BAD\_REQUEST);  
 }  
 return user.getCustomer();  
 }  
}

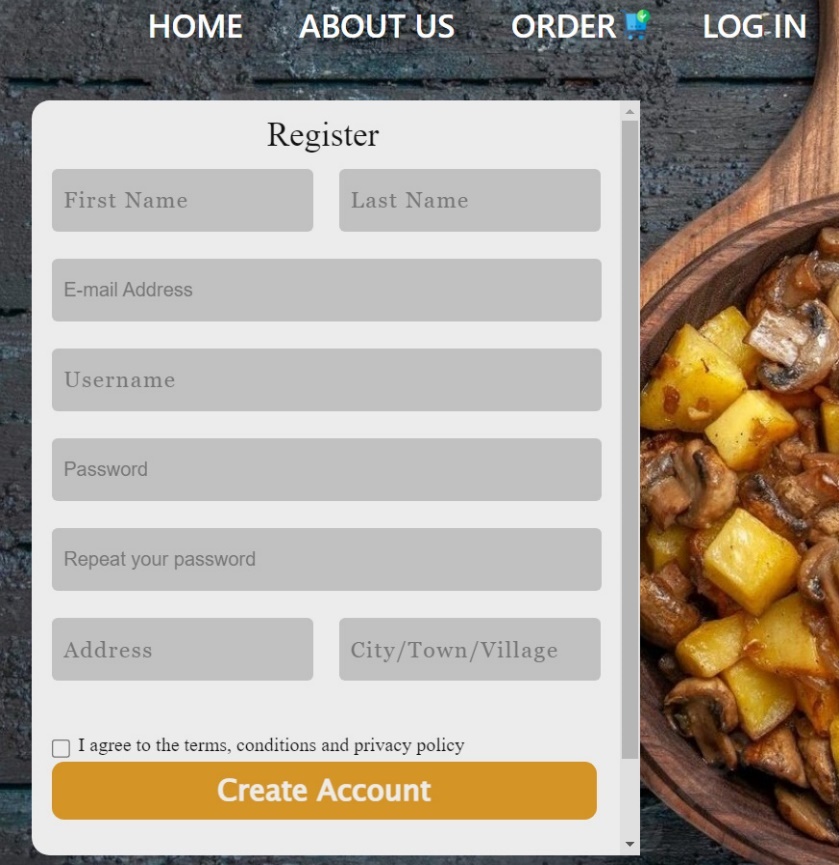
Anexa 6

Imaginea 2.7

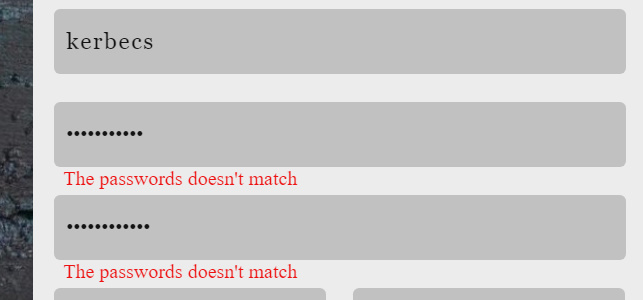


Anexa 7

Imaginea 2.8



Imaginea 2.9

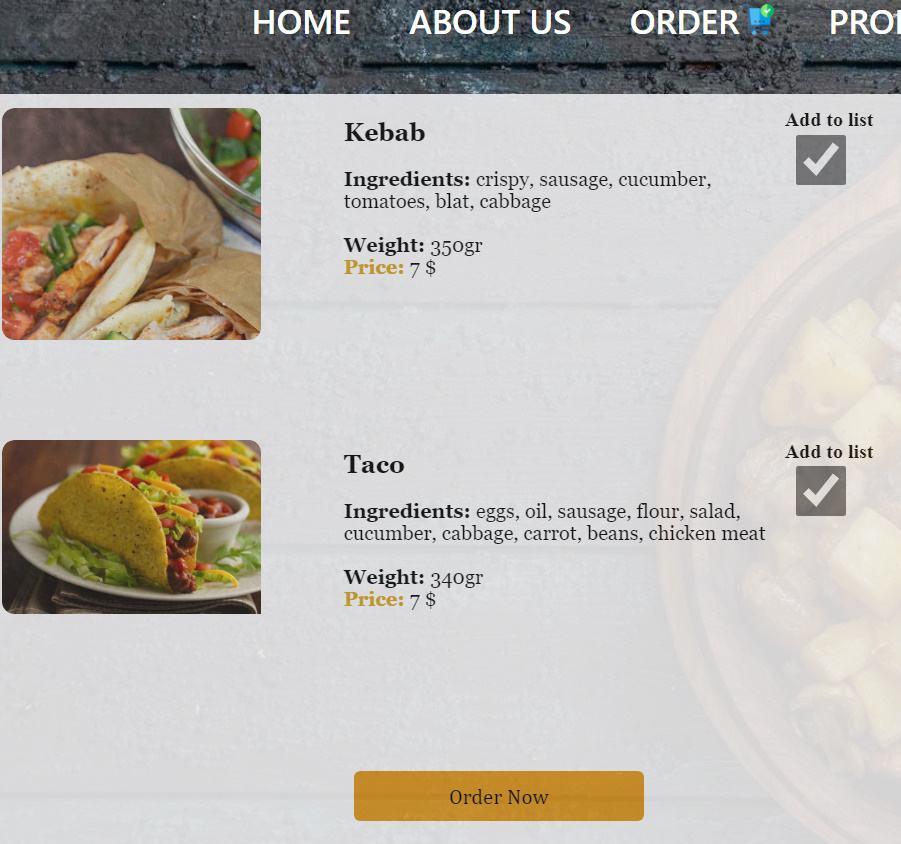


Anexa 8

@Entity  
@Table(name = "product")  
public class Product {  
 @Id  
 @GeneratedValue(strategy = GenerationType.IDENTITY)  
 @Column(name = "product\_id")  
 private int productId;  
  
 @Column(name = "title")  
 private String title;  
  
 @Column(name = "ingredients")  
 private String ingredients;  
  
 @Column(name = "price")  
 private int price;  
  
 @Column(name = "weight")  
 private int weight;

Anexa 9

Imaginea 2.10



Anexa 10

<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 11

<form:form action="loginCheck" modelAttribute="userTest">  
 <br><br><br>  
 <p class="invalid">${invalid}</p>  
 <br><br>  
 <font class="loginFont">Username</font><br>  
 <form:input type="text" class="fname user" placeholder="Enter your username" path="username"/>  
 <br>  
 <font class="required">&nbsp &nbsp &nbsp &nbsp &nbsp <form:errors path="username"/></font>  
 <br>  
 <font class="loginFont">Password</font><br>  
 <form:input type="password" class="lname pass" placeholder="Enter your password" path="password"/>  
 <br>  
 <font class="required"> &nbsp &nbsp &nbsp &nbsp &nbsp <form:errors path="password" /></font>  
  
  
 <br>  
 <input type="submit" class="create\_acc login" value="Login"/>  
 <br>  
 <br>  
 <input type="button" class="create\_acc login forgot" value="Reset Password "/>  
 <br>  
 <div class="notRegistered">  
 Don't have an account? <a href="register">Register now</a>  
 </div>  
  
</form:form>

Anexa 12

Imaginea 2.11



Anexa 13

Imaginea 2.12

