**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

****

**LUCRARE DE LICENŢĂ**

**Waitor: Aplicaţie pentru eficientizarea modului de lucru al**

**chelnerilor în restaurante**

Propusӑ de : **Sucurei Robert – Pavel**

**Iunie/Iulie, 2022**

Coordonator Ştiinţific

**Drd. Florin Olariu**

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

**Waitor: Aplicaţie pentru eficientizarea modului de lucru al chelnerilor în restaurante**

**Absolvent:**

**Sucurei Robert – Pavel**

**Iunie/Iulie, 2021**

**Coordonator Ştiinţific:**

**Drd. Florin Olariu**

Avizat,

Îndrumător Lucrare de Licență,

Titlul, numele si prenumele: Drd. Florin Olariu.

Data \_\_\_\_\_\_\_\_\_\_\_\_ Semnătura \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemntatul Sucurei Robert – Pavel cu domiciliul în România, jud. Vaslui, mun. Vaslui, str. Ion Luca Caragiale, nr. 32 născut la data de 29 iunie 1999, identificat prin CNP 1990629374520, absolvent al Universității „Alexandru Ioan Cuza” din Iași, Facultatea de Informatică specializarea Informatică, promoția 2022, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul: **„Waitor, aplicaţie pentru eficientizarea modului de lucru a chelnerilor în restaurante”**, elaborată sub îndrumarea dl. Drd. Florin Olariu, pe care urmează să o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice in vederea facilitării fasificării de către cumpărător a calității de autor al unei lucrări de licență, de diplomă sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, XX.06.2022      Semnătură student …………………………

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord cӑ Lucrarea de licență cu titlul **„Waitor, aplicaţie pentru eficientizarea modului de lucru a chelnerilor in restaurante”**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de testetc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, XX. 06. 2022 Absolvent *Sucurei Robert - Pavel*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

   (semnătura în original)

Cuprins

[Introducere 7](#_Toc106455426)

[1. Motivaţie 7](#_Toc106455427)

[2. Sursa de inspiraţie 7](#_Toc106455428)

[3. Contribuţii 9](#_Toc106455429)

[Aplicaţia 10](#_Toc106455430)

[1. Tehnologii folosite: 10](#_Toc106455431)

[I. Node.js 16.13.1 10](#_Toc106455432)

[II. Express.js 11](#_Toc106455433)

[III. React 11](#_Toc106455434)

[IV. Tachyons 12](#_Toc106455435)

[V. pgAdmin 4 13](#_Toc106455436)

[VI. PostgreSQL 14](#_Toc106455437)

[VII. Knex.js 15](#_Toc106455438)

[VIII. qrserver 15](#_Toc106455439)

[IX. Postman 16](#_Toc106455440)

[2. Arhitecturӑ: 17](#_Toc106455441)

[I. User: 17](#_Toc106455442)

[II. Restaurant 18](#_Toc106455443)

[III. Type 18](#_Toc106455444)

[IV. Preparate 19](#_Toc106455445)

[V. Purchase order 20](#_Toc106455446)

[VI. Purchase order line 20](#_Toc106455447)

[VII. Comment 21](#_Toc106455448)

[3. Modul de utilizare: 22](#_Toc106455449)

[I. Restaurant: 23](#_Toc106455450)

[II. Client: 27](#_Toc106455451)

[III. Comunicarea dintre client şi restaurant: 28](#_Toc106455452)

[IV. Cursul aplicaţiei: 33](#_Toc106455453)

[Concluzie 34](#_Toc106455454)

[1. Posibile funcţionalitӑţi noi 34](#_Toc106455455)

[2. Concluzie finalӑ 34](#_Toc106455456)

[Bibliografie 36](#_Toc106455457)

# Introducere

## Motivaţie

Secolul XXI (secolul vitezei) este o perioadӑ de timp în care majoritatea oamenilor dorește ca totul sӑ se întâmple cât mai repede şi cu cât mai puţin efort, unii oameni începând sӑ prefere mult mai mult interacţiunea om – aplicaţie, în favoarea interacţiunii om – om.

Restaurantele reprezintӑ acele locuri cunoscute si frecventate de majoritatea populaţiei lumii. Chiar dacӑ unele persoane aleg sӑ meargӑ la restaurant cu scop recreativ, iar altele merg doar pentru a servi masa, consider cӑ modul de lucru al chelnerilor cât şi comunicarea client – chelner poate fi imbunӑtӑţitӑ.

Astfel, motivaţia pe care am avut-o în dezvoltarea acestei lucrӑri se bazeazӑ pe eficientizarea modului de lucru al chelnerului în restaurante dar şi imbunӑtӑţirea experienţei clientului.

## **Sursa de inspiraţie**

Toatӑ aceastӑ idee a plecat de la multitudinea aplicaţiilor de livrare de mâncare precum (Bolt Food, Tazz, Glovo, Food Panda etc.), care m-au fӑcut sӑ îmi pun întrebarea: “De ce nu existӑ o aplicaţie asemӑnӑtoare în interiorul unui restaurant?” Prin acest gând, doresc sӑ spun cӑ cu ajutorul unei aplicaţii, chelnerul poate deveni un “curier” (precum cei angajati la Bolt Food, Tazz, Glovo, Food Panda etc.) al restaurantului în care lucreazӑ cu ajutorul aplicaţiei dezvoltate de mine.



*Figura 1 – Logouri firme de livrare mâncare1*

* Clientul poate comanda prin intermediul telefonului şi astfel chelnerul sӑ îi aducӑ produsele cerute direct la masӑ.
* Procesul de aşteptare a clientului pentru chelner atunci când comandӑ, când face anumite cereri sau când cere nota de platӑ se va micşora.
* Chelnerul işi poate administra timpul de lucru mult mai bine, nemaifiind obligat sӑ interacţioneze la fel de mult cu clienţii.

1 <https://www.anuntul.ro/anunt-angajare-livrator-foodpanda-VnjmXL>

## Contribuţii

În cadrul aplicaţiei pe care am dezvoltat-o, comenzile se fac în timp real în incinta restaurantului care utilizeazӑ platforma. Deşi experienţa clientului este în mare parte asemӑnatoare cu cea din restul aplicaţiilor deja existente pe piaţӑ (care au servit drept sursӑ de inspiraţie în realizarea acestui proiect), contribuţia majorӑ adusӑ de lucrarea mea constӑ în eficientizarea servirii comenzilor din restaurant. Printre aceste optimizӑri sunt prezente urmatoarele: chelnerul nu mai este nevoit sӑ supravegheze incinta restaurantului la fel de mult şi pe lângӑ aceasta, acesta este scutit de interacţiunile cu clienţii care reprezintӑ un mare consumator de timp. Erorile umane privind prelurea si procesarea comenzilor sunt reduse în mod semnificativ. În plus, timpul livrӑrii serviciilor este eficientizat deoarece efortul personalului este şi el diminuat. Aceastӑ funcţionalitate poate fi de asemenea de folos şi pentru clienţii care favorizeazӑ mai mult interacţiunea cu mediul digital.

Aceastӑ problemӑ pe care am observat-o în industria Horeca poate fi diminuatӑ cu uşurinţӑ pentru a îmbunӑtӑţi atât experienţa clientului cu restaurantele dar şi experienţa chelnerilor cu clienţii pe care îi servesc.

# Aplicaţia

## Tehnologii folosite:

### I. Node.js 16.13.1

Conform site-ului oficial2, Node.js reprezintӑ un mediu de execuţie JavaScript care executӑ cod JavaScript în afara unui navigator web, iar acesta funcţioneazӑ pe un singur thread şi se foloseşte de apeluri IN/OUT non-blocante.

Conform celor de la Section3, Node.js a fost creat de cӑtre Ryan Dahl în anul 2009 deoarece acesta nu era mulţumit de posibilitӑţile limitate pe care cel mai popular server web le avea in acel moment, acela fiind Apache HTTP4 Server.

Node.js se foloseşte de motorul de execuţie JavaScript V8 care a fost iniţial construit pentru browserul Google Chrome, ca mai apoi sӑ fie deschis publicului în anul 2008.



*Figura 2 - logo Node.js5*

2 <https://nodejs.org/en/about/>

3 <https://www.section.io/engineering-education/history-of-nodejs/>

4 Hypertext Transfer Protocol

5 <https://en.wikipedia.org/wiki/Node.js#/media/File:Node.js_logo.svg>

Am decis sӑ folosesc Node.js deorece popularitatea sa mi-a oferit o documentaţie foarte bine realizatӑ şi uşor de înţeles.

### II. Express.js

Conform site-ului oficial6, Express.js, sau mai simplu Express, reprezintӑ un framework de back end pentru Node.js.



*Figura 3 – logo Express.js7*

Am folosit Express.js pentru crearea serverului şi a endpointurilor necesare pentru buna funcţionare a aplicaţiei.

### III. React

Conform site-ului oficial8, React (cunoscut şi ca React.js sau ReactJS) este un framework de front end pentru crearea de interfeţe pentru utilizatorii unei aplicaţii web.

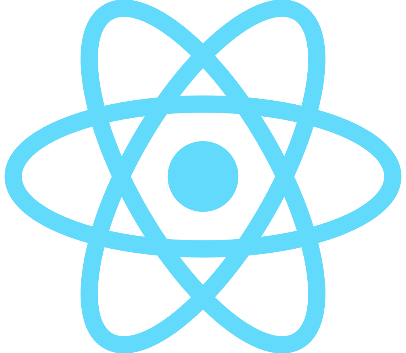
6 <https://expressjs.com/>

7 <https://en.wikipedia.org/wiki/Express.js#/media/File:Expressjs.png>

8 <https://reactjs.org/>

Conform celor de la Educationecosystem9, React a fost creat de cӑtre Jordan Walke, un inginer software de la Facebook, în anul 2011, acesta fiind inspirat de cӑtre XHP (care reprezintӑ o simplӑ componentӑ HTML10 creatӑ pentru limbajul de programare PHP11).

Cu timpul, React s-a dezvoltat din ce în ce mai mult, astfel, Facebook a decis sӑ îl facӑ deschis publicului in mai 2013.



*Figura 4 – logo React12*

Am hotӑrât sӑ folosesc React în dezvoltarea lucrӑrii mele deorece oferӑ o flexibiltate enormӑ în felul în care o interfaţӑ poate fi creatӑ, dar şi popularitatea sa şi documentaţiile bine realizate au fost unele dintre motive.

### IV. Tachyons

Conform site-ului oficial13, Tachyons reprezintӑ o colecţie de module mici care pot fi combinate, potrivite sau folosite independent.

9 <https://educationecosystem.com/guides/programming/react-js/history>

10 Hypertext Markup Language

11 PHP: Hypertext Preprocessor

12 <https://educationecosystem.com/guides/programming/react-js/history>

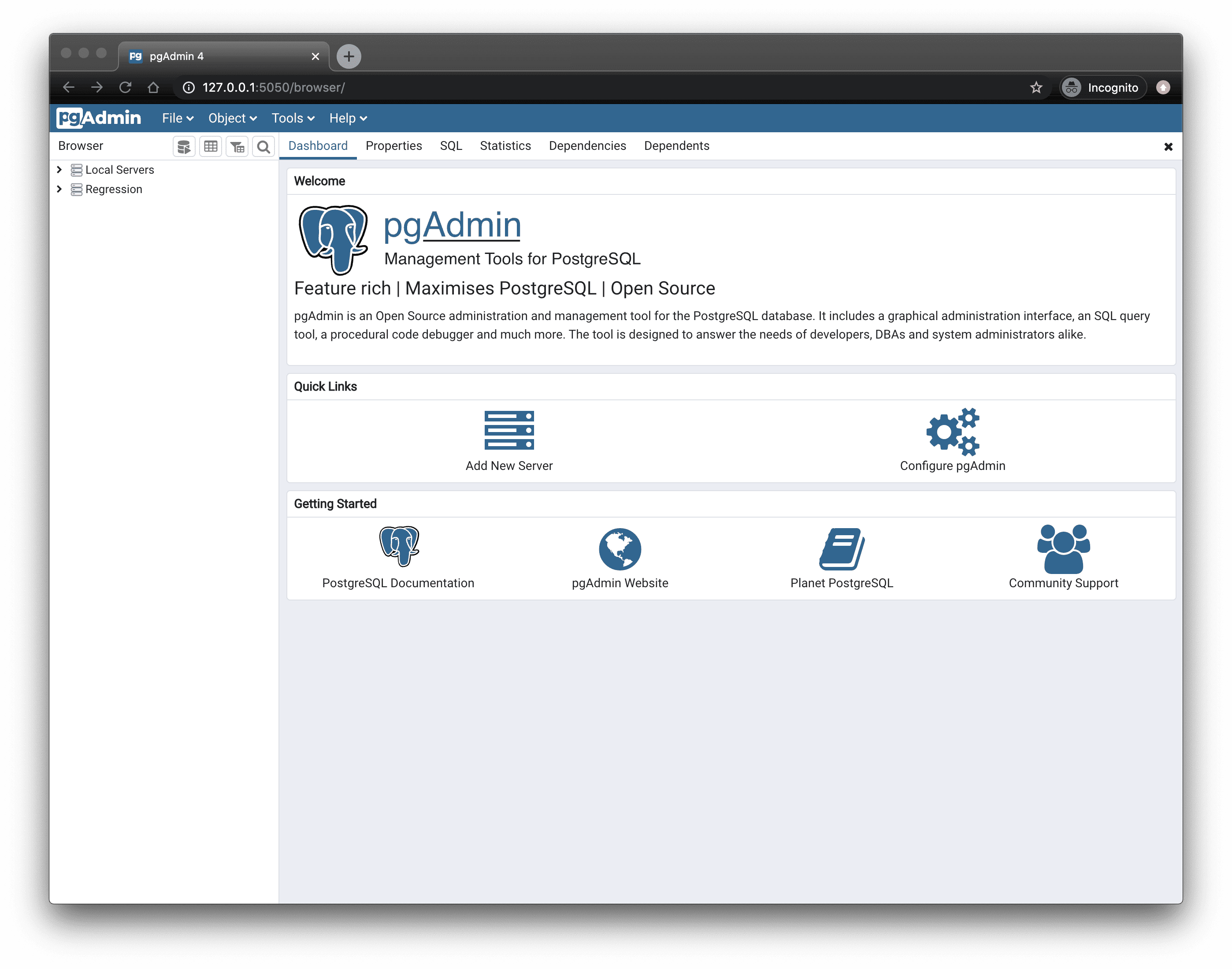
13 <https://tachyons.io/#principles>

Cu ajutorul acestei colecţii, front end-ul este 100% responsive14, indiferent de dispozitivul sau mӑrimea ecranului utilizatorului.

Am decis sӑ folosesc aceastӑ colecţie datoritӑ simplitӑţii oferite.

### V. pgAdmin 4

Conform site-ului oficial15, pgAdmin este un instrument de *management* pentru PostgreSQL. Acesta poate funcţiona atât ca aplicaţie desktop, cât şi ca aplicaţie web.



*Figura 5 – aplicaţia pgAdmin416*

14 Accesibil şi adaptabil pe orice fel de dispozitiv

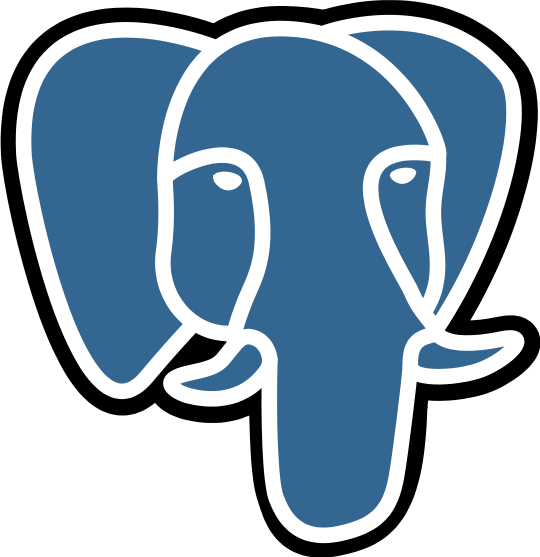
15 [https://www.pgadmin.org/faq/#](https://www.pgadmin.org/faq/)

16 <https://www.pgadmin.org/screenshots/#1>

Am decis sӑ folosesc aceastӑ aplicaţie datoritӑ simplitӑţii cu care pot fi manipulate tabelele si vizualizate datele.

### VI. PostgreSQL

Conform site-ului oficial17 PostgreSQL reprezintӑ o bazӑ de date relaţionalӑ care foloseşte si extinde limbajul SQL18.



*Figura 6 – logo PostgreSQL19*

Limbajul SQL a fost creat de cӑtre Universitatea Barkeley din California şi are mai mult de 30 de ani de dezvoltare activӑ, fiind lansat în anul 1986.

Am hotӑrât sӑ folosesc aceastӑ bazӑ de date datoritӑ instrumentului de *management* pentru PostgrSQL, pgAdmin4.

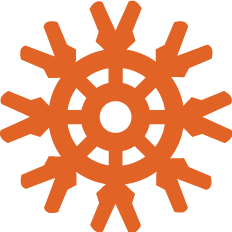
17 <https://www.postgresql.org/about/>

18 Structured Query Language

19 <https://www.postgresql.org/>

### VII. Knex.js

Potrivit site-ului official20, Knex reprezintӑ un instrument cu care pot fi construite într-un mod foarte intuitiv interogӑri SQL.



*Figura 7 – logo Knex.js21*

Am ales sӑ folosesc acest instrument deorece modul de funcţionare este extrem de asemӑnӑtor cu tehnologia Linq din limbajul de programare C# .NET cu care eram deja acomodat.

### VIII. qrserver

Conform site-ului oficial22, qrserver reprezintӑ un API23 cu care pot fi generate coduri QR24.

Un cod QR reprezintӑ un cod de bare bidimensional format cel mai adesea din pixeli albi şi negri.

20 <http://knexjs.org/>

21 <http://knexjs.org/>

22 <https://goqr.me/>

23 Application Programing Inteface

24 Quick Response

Modul de funcţionare reprezintӑ un apel cӑtre urmӑtorul URL25: “<http://api.qrserver.com/v1/create-qr-code/?data=>”, astfel textul ce se aflӑ dupӑ „=” va fi codat într-un cod QR.

Am folosit acest API pentru a genera coduri QR specifice fiecӑrei mese dintr-un restaurant.

### IX. Postman

Conform site-ului oficial26, Postman reprezintӑ o platformӑ pentru construit şi utilizat API-uri.

Site-ul oficial, la secţiunea “about-postman”27, ne spune cӑ Postman a fost creat de cӑtre Abhinav Asthana, care este CEO28-ul dar şi cofondatorul acestuia.

În momentul de faţӑ Postman este liderul mondial în ceea ce reprezintӑ tehnologie pentru testat API-uri.



*Figura 8 – logo Postman29*

Am ales sӑ folosesc Postman pentru a-mi testa într-un mod cât mai comod endpointurile aplicaţiei create de mine.

25 Uniform Resource Locators

26 <https://www.postman.com/>

27 <https://www.postman.com/company/about-postman/>

28 Chief Executive Officer

29 <https://www.postman.com/>

## Arhitecturӑ:

Aplicaţia funcţioneazӑ cu ajutorul mai multor tabele. Aceste tabele sunt:

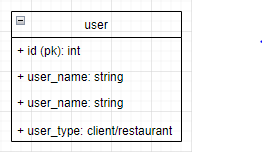
* user
* restaurant
* type
* preparate (in aplicaţie se gӑseşte sub numele de product)
* purchase order
* purchase order line
* comment

### I. User:

Reprezintӑ tabela care reţine utilizatorii inregistraţi în platformӑ. Utilizatorii se identificӑ unic prin cheia primarӑ “id”.

În momentul înregistrӑrii, utilizatorul va fi nevoit sӑ îşi introducӑ un nume de utilizator (trebuie sa fie unic faţӑ de alţi utilizatori) şi o parolӑ (aceasta va fi criptata şi mai apoi stocatӑ în baza de date).

Un utilizator poate fi de douӑ feluri: restaurant sau client.



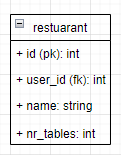
*Figura 9 – schema tabelei “user”*

### II. Restaurant

Reprezintӑ tabela care reţine restaurantele create de cӑtre utilizatorii de tip “restaurant”. Restaurantele se identificӑ unic prin cheia primarӑ “id”.

Cheia strӑinӑ “user\_id” pӑstreazӑ valoarea cheii primare “id” din tabela “user”. Astfel se va şti utilizatorul care deţine restaurantul.

Aceastӑ tabelӑ pӑstreazӑ şi numele restaurantului în coloana “name” dar şi numӑrul de mese existente în coloana “nr\_tables”.



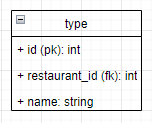
*Figura 10 – schema tabelei “restaurant”*

### III. Type

Reprezintӑ tabela care reţine tipurile de produse pe care un restaurant le vinde. Tipurile se identificӑ unic prin cheia primara “id”.

Cheia strӑinӑ “restaurant\_id” pӑstreazӑ valoarea cheii primare “id” din tabela ”restaurant”. Astfel se va şti ce tipuri de produse vinde restaurantul respectiv.

Aceastӑ tabelӑ pӑstreazӑ numele tipului în coloana “name”.



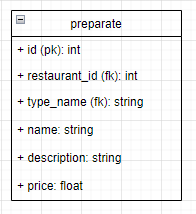
*Figura 11 – schema tabelei “type”*

### IV. Preparate

Reprezintӑ tabela care pӑstreazӑ produsele pe care un restaurant le vinde. Produsele se identificӑ unic prin cheia primarӑ “id”.

Conţine douӑ chei strӑine: “restaurant\_id” şi “type\_name”, în care se pӑstreazӑ valoarea cheii primare din tabela “restaurant”, respectiv valoarea coloanei “name” din tabela “type”, astfel se va şti ce produse vinde restaurantul şi din ce tip face parte acel produs.

De asemenea, aceastӑ tabelӑ are şi urmatoarele coloane: “name”, care pӑstreazӑ numele produsului vândut, “description”, care pӑstreazӑ o descriere despre produs şi “price” care pastreazӑ preţul produsului.



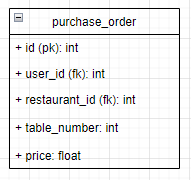
*Figura 12 – schema tabelei “preparate”*

### V. Purchase order

Reprezintӑ tabela în care sunt înregistrate comenzile din restaurant. Acestea se identificӑ unic prin cheia primarӑ “id”.

Conţine doua chei strӑine: “user\_id” şi “restaurant\_id”, în care se pӑstreazӑ valoarea cheii primare din tabela “user”, respectiv valoarea cheii primare din tabela “restaurant”, astfel se va şti ce user a fӑcut comanda şi la ce restaurant s-a fӑcut comanda.

Aceastӑ tabelӑ conţine şi coloanele “table\_number” care pӑstreazӑ numӑrul mesei la care clientul se aflӑ şi “price” care reprezintӑ preţul total al comenzii date.



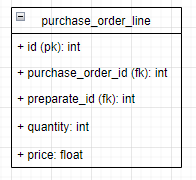
*Figura 13 – schema tabelei “purchase\_order”*

### VI. Purchase order line

Reprezintӑ tabela în care sunt înregistrate detaliile comenzii date de client. Acestea se identificӑ unic prin cheia primarӑ “id”.

Conţine doua chei strӑine: “purchase\_order\_id” şi “preparate\_id”, în care se pӑstreazӑ valoarea cheii primare din tabela “purchase\_order”, respectiv valoarea cheii primare din tabela “preparate”, astfel se va şti la ce comandӑ sunt aceste detalii şi ce produs a fost comandat.

Aceastӑ tabelӑ conţine şi coloanele “quantity” care pӑstreazӑ de câte ori a fost comandat un anumit produs şi “price” care reprezintӑ preţul total pentru produsul comandat.



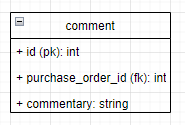
*Figura 14 – schema tabelei “purchase\_order\_line”*

### VII. Comment

Reprezintӑ tabela în care sunt pӑstrate comentariile clientului. Acestea se identificӑ unic prin cheia primarӑ “id”.

Conţine o cheie strӑinӑ numitӑ “purchase\_order\_id”. Astfel se va şti la ce comandӑ sunt fӑcute comentariile.

În coloana “commentary” sunt pӑstrate comentariile clientului.



*Figura 15 – schema tabelei “comment”*

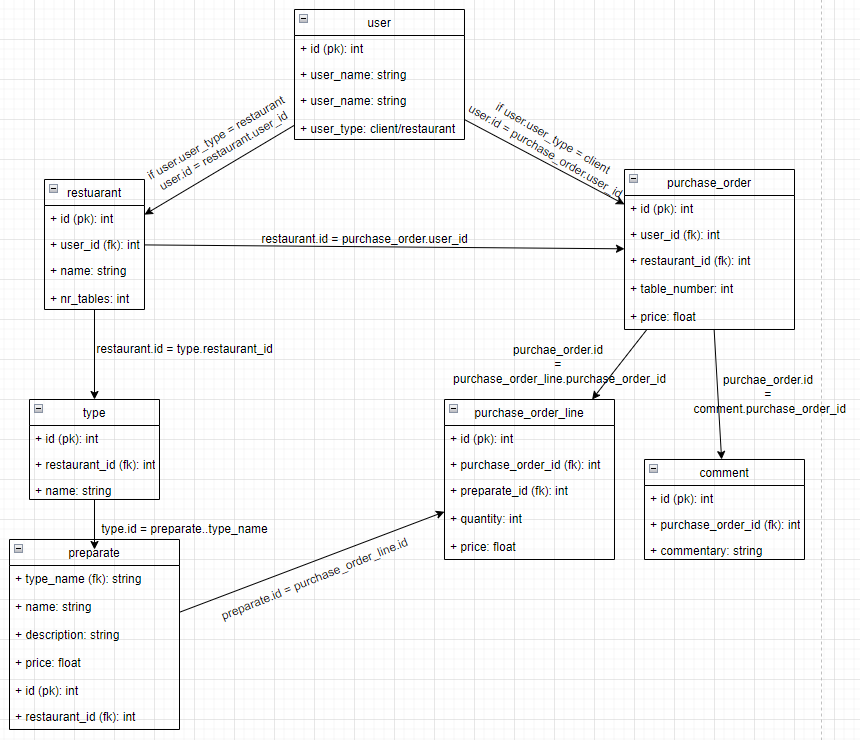
## Modul de utilizare:

Aplicaţia este destinatӑ atât persoanelor care deţin un restaurant, cât şi clienţilor unui restaurant înregistrat în platformӑ.

Modul de utilizare este foarte intuitiv, aplicaţia fiind foarte simplu de folosit.

Fiecare etapӑ a aplicaţiei este ori explicatӑ ori sugestivӑ.

Aplicaţia a fost conceputӑ şi dezvoltatӑ astfel încât experienţa utilizatorilor sӑ fie una cât mai plӑcutӑ.



*Figura 16 – diagrama aplicaţiei*

Aplicaţia poate fi utilizatӑ atât pe un smartphone (când utilizatorul este un client) cât şi pe un calculator (când utilizatorul este un restaurant).

Când accesӑm aplicaţia, prima paginӑ va fi cea în care trebuie sӑ menţionӑm dacӑ suntem un client sau un restaurant.



*Figura 17 – client sau restaurant*

Odatӑ selectat un rӑspuns, vom continua pe una dintre cele doua cӑi (calea “Restaurant” sau calea “Client”).

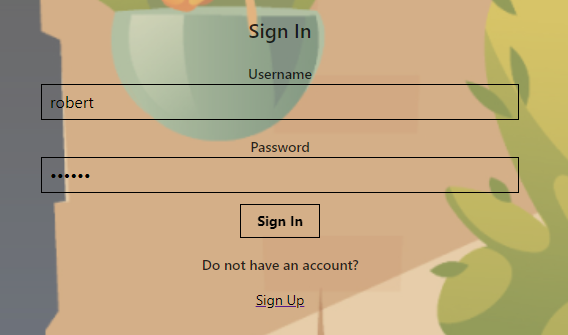
### I. Restaurant:

#### i. Logare şi înregistrare utilizator de tipul “restaurant”

În cazul în care selectӑm “Restaurant”, vom fi direcţionati cӑtre pagina de logare.

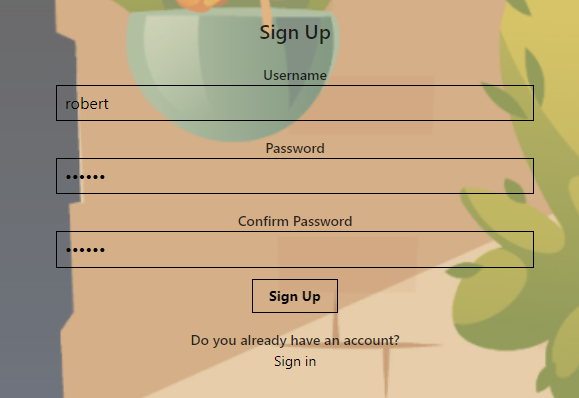
Pentru a ne loga trebuie sӑ ne introducem numele de utilizator şi parola contului deja creat.

Odatӑ introduse aceste date, se apasӑ butonul “Sign In”, iar dacӑ datele sunt corecte, vom ajunge pe pagina de administrare al restaurantului.



*Figura 18 – logare utilizator de tip restaurant*

În cazul în care utilizatorul nu are cont, dar acesta doreşte sӑ îşi creeze unul, poate sӑ apese pe butonul “Sign Up” unde va fi direcţionat pe pagina de înregistrare restaurant, ca mai apoi sӑ işi dea detaliile restaurantului propriu.

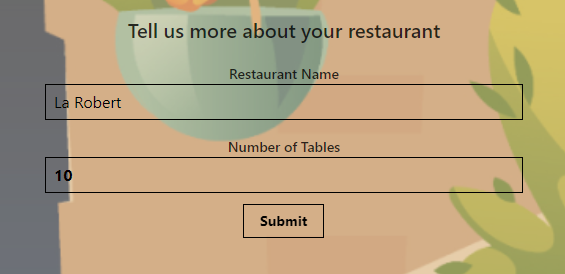
****

*Figura 19 – înregistrare utilizator de tip restaurant*

#### ii. Menţionare nume restaurant şi numӑr de mese

Odatӑ înregistrat utilizatorul de tip restaurant, vom fi direcţionaţi cӑtre pagina în care trebuie sӑ menţionӑm numele restaurantului şi de câte mese dispune restaurantul pe care dorim sӑ îl înregistrӑm.

Odatӑ înregistrate aceste detalii, vom fi direcţionaţi cӑtre pagina în care va trebui sӑ menţionӑm tipurile de preparate pe care restaurantul le vinde.

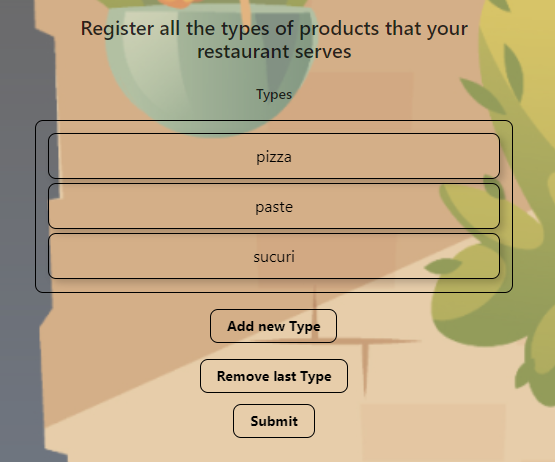


*Figura 20 – înregistrare nume restaurant şi numӑrul de mese al acestuia*

#### iii. Menţionare tipuri de produse vândute de restaurant

Aceastӑ paginӑ are ca scop menţionarea tipurilor de produse pe care restaurantul doreşte sӑ le vândӑ.

Avem opţiunea de a adӑuga tipuri de produse şi de a şterge tipuri de produse.



*Figura 21 – înregistrare tipuri de produse pe care restaurantul le vinde*

#### iv. Menţionare produse pe care restaurantul le vinde

Aceastӑ paginӑ are ca scop înregistrarea produselor pe care restaurantul le va vinde.

Avem opţiunea de adӑugare si stergere produse.

Pentru a putea adӑuga un produs suntem obligaţi sӑ înregistrӑm tipul din care face parte, numele acestuia şi preţul cu care va fi comercializat. Existӑ şi posibilitatea de a adӑuga o descriere la produse dar acest câmp nu este obligatoriu.



*Figura 22 - înregistrare produse pe care restaurantul le vinde*

Odatӑ completate produsele ce vor fi vândute de restaurant, prin apӑsarea butonului “Submit”, vom fi direcţionaţi cӑtre pagina de administrare a restaurantului şi se vor genera şi descӑrca coduri QR unice pentru fiecare masӑ. Dacӑ restaurantul a menţionat cӑ deţine 10 mese, se vor genera și descӑrca 10 coduri QR unice.

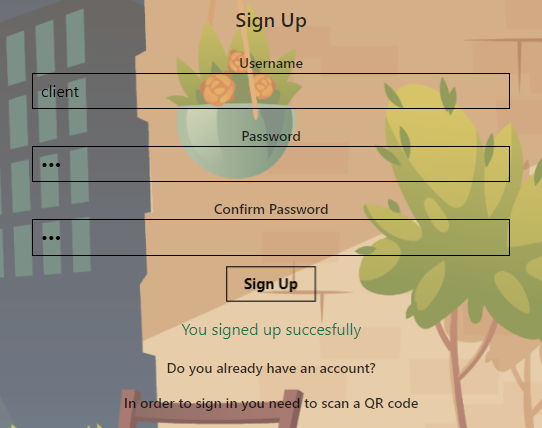
Aceste coduri QR vor fi printate, ca mai apoi fiecare masӑ sa aibӑ cate un cod propriu.

### II. Client:

#### i. Înregistrare utilizator de tipul “client”

Înregistrarea unui utilizator de tip ”client” este asemӑnӑtor cu cel de tip “resturant”.

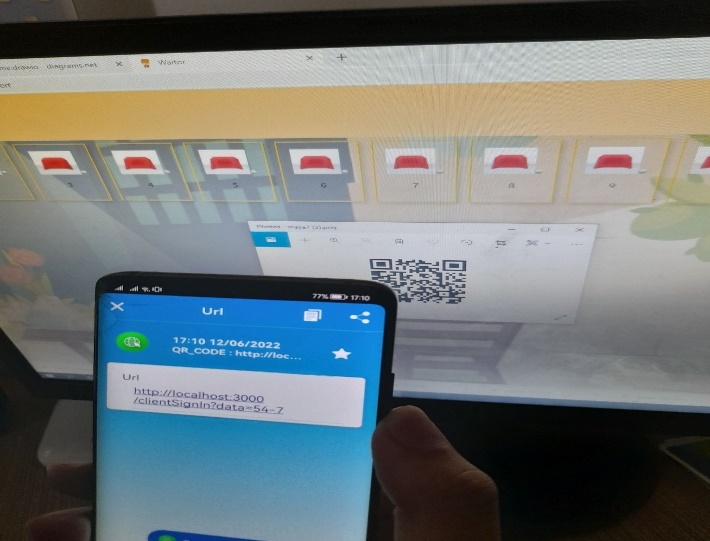
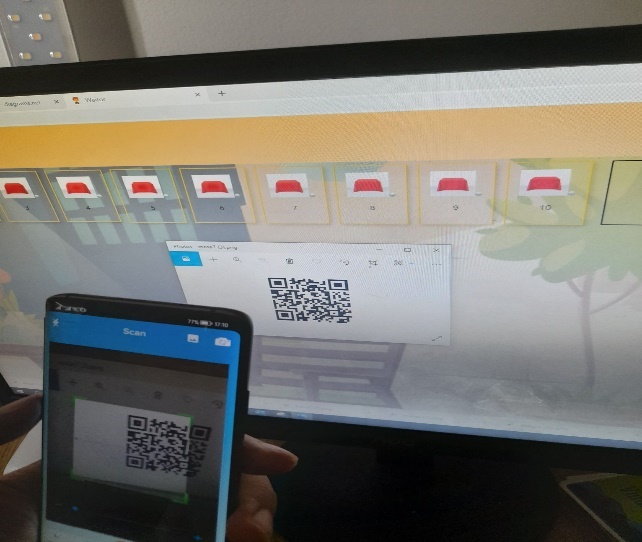
Odatӑ apӑsat butonul “Client” de la pagina evidenţiatӑ în Figura 17, vom fi direcţionaţi cӑtre pagina de înregistrare, logarea unui client fiind posibilӑ doar prin scanarea unui cod QR.



*Figura 23 – înregistrare utilizator de tip “client”*

#### ii. Logarea utilizatorului de tipul “client”

Pentru a accesa pagina de logare pentru utilizatorul de tip “client”, suntem nevoiţi sӑ scanӑm un cod QR.



*Figura 24 – Scanare cod QR şi redirecţionare spre pagina de logare*

Modul de logare al unui utilizator de tip “client” este asemӑnӑtor cu cel al utilizatorului de tip “restaurant”.

Odatӑ introduse datele utilizatorului, se apasӑ butonul “Sign In”, iar dacӑ datele sunt corecte, se va ajunge pe pagina cu meniul restaurantului la care utilizatorul se aflӑ.

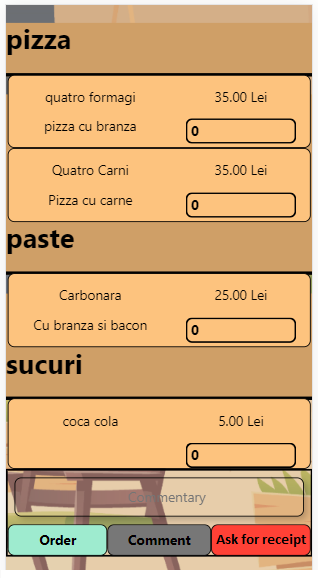
### III. Comunicarea dintre client şi restaurant:

În momentul înregistrӑrii sau logӑrii cu success a utilizatorilor, cei de tip “restaurant” vor fi direcţionati cӑtre pagina de *management* a restaurantului, iar cei de tip “client” vor fi direcţionaţi cӑtre pagina de unde pot face comenzi la masa la care se aflӑ.

#### i. Pagina pentru creare de comenzi

Aceastӑ paginӑ este accesatӑ de utilizatorii de tip “client” dupӑ înregistrarea sau logarea cu success pe platformӑ.

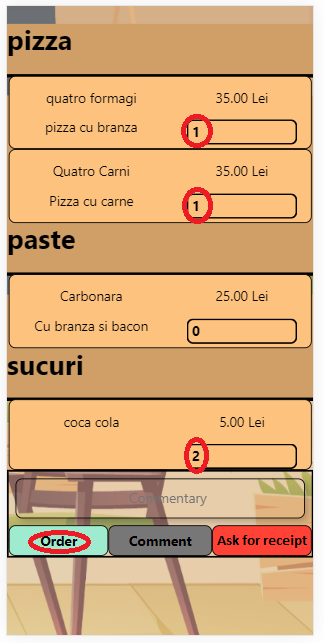
Pe aceastӑ paginӑ, utilizatorul de tip “client” are posibilitatea de a comanda produse vândute de restaurantul la care se aflӑ, de a trimite comentarii şi de a cere nota.



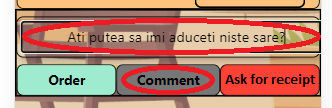
*Figura 25 – Pagina de comandӑ*

Modul de utilizare este urmӑtorul:

* Utilizatorul de tip “client” selecteazӑ produsele şi cantitatea pe care doreşte sӑ o comande.
* Dupӑ ce utilizatorul apasӑ pe butonul “Order”, comanda va fi trimisӑ cӑtre pagina de administrare a restaurantului.
* Prin apӑsarea butonului “Comment”, textul pe care utilizatorul de tip “Client” l-a introdus în cӑsuţa de text “Commentary” va fi trimis cӑtre restaurant.
* Prin apӑsarea butonului “Ask for receipt”, utilizatorul de tip client transmite restaurantului faptul cӑ acesta doreşte sӑ primeascӑ nota de platӑ.



*Figura 26 – Utilizatorul de tip “client” trimite comanda*

**

*Figura 27 – Utilizatorul de tip “client” trimite un comentariu*

#### ii. Pagina de administrare a restaurantului



Aceastӑ paginӑ este accesatӑ de utilizatorii de tip “restaurant” dupӑ înregistrarea sau logarea cu success pe platformӑ.



*Figura 28 – Pagina de manageriere a restaurantului*

Pe aceastӑ paginӑ, chelnerii au posibilitatea de a vedea ce comenzi sau cereri au fost fӑcute la toate mesele din incinta restaurantului.

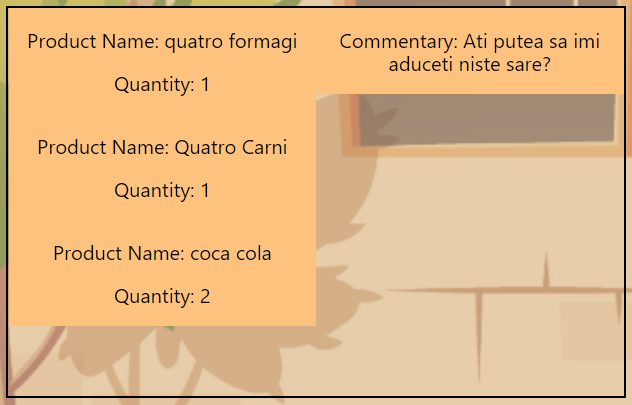
Aceastӑ paginӑ conţine câte un buton pentru fiecare masӑ pe care restaurantul o deţine.

Fiecare buton este dotat cu un “sistem de notificare”, astfel încât, când un client are orice fel de cerere (a comandat ceva sau a lӑsat un comentariu sau a cerut nota de platӑ) la o anumitӑ masӑ, bulina gri din dreptul mesei se va transforma în una roşie.

*Figura 29 – Sistemul de notificare*

Odatӑ apӑsat pe butonul unei mese, detaliile de comandӑ, sau comentariile vor fi afişate.



*Figura 30 – Comanda şi comentariul fӑcut de utilizatoriul de tip “client”*

În momentul în care utilizatorul de tip “client” apasӑ pe butonul “Ask for note”, bulina din dreptul mesei se va face roşie, astfel sistemul de notificare se activeazӑ, dar, odatӑ apӑsat butonul, nu vor mai fi afişate nici produsele comandate, nici comentariile trimise în trecut, astfel, în acest mod, restaurantul este informat cӑ utilizatorul de tip “client” doreşte nota de platӑ.

### IV. Cursul aplicaţiei:

1. Utilizatorul de tip “restaurant” se inregistreazӑ sau se conecteazӑ cu success la platformӑ.
2. Utilizatorul de tip “client” se inregistreazӑ şi se conecteazӑ, sau se conecteazӑ cu succes la platformӑ.
3. Utilizatorul de tip “client” face o comandӑ cu produsele pe care le doreşte.
4. Restaurantul este notificat.
5. Chelnerul apasa pe butonul care corespunde cu masa de unde utilizatorul de tip “client” a facut comanda şi observӑ detaliile comenzii.
6. \* Utilizatorul de tip “client” trimite un comentariu la restaurant.
7. \* Restaurantul este notificat.
8. \* Chelnerul apasӑ pe butonul care corespunde cu masa de unde utilizatorul de tip “client” a facut comentariul şi îl observӑ.
9. Utilizatorul de tip “client” trimite cererea pentru nota de platӑ.
10. Restaurantul este notificat.
11. Chelnerul apasӑ pe butonul care corespunde cu masa de unde utilizatorul de tip “client” a cerut nota, si observӑ faptul cӑ detaliile de la masa respectivӑ lipsesc, astfel va şti cӑ acesta trebuie sӑ ducӑ nota de platӑ la utilizatorul de tip “client”.

Notaţia “\*” evidenţiazӑ paşi opţionali

# Concluzie

## Posibile funcţionalitӑţi noi

Proiectul dezvoltat de mine ar putea fi imbunӑtӑţit, putând fi incluse funcţionalitӑţi noi.

În momentul de faţӑ, observ douӑ funcţionalitӑţi noi ce ar putea fi aduse pentru utilizatorii de tip “restaurant” şi una pentru utilizatorii de tip “client”.

Astfel, pentru utilizatorii de tip “restaurant”, consider cӑ o funcţionalitate ce permite restaurantului sӑ işi modifice meniul şi dupӑ înregistrarea acestuia în platformӑ, ar fi foarte utilӑ pentru buna funcţionare.

O altӑ functionalitate ar putea fi posibilitatea de a afişa istoricul vânzӑrilor fӑcute prin intermediul aplicaţiei şi astfel efectuarea unui calcul de stocuri necesare pe viitor in funcţie de perioada ce va avea sӑ urmeze. Aceste date pot fi influenţate de diferite evenimente şi perioade, precum: dacӑ este weekend, dacӑ condiţiile meteorologice sunt plӑcute, dacӑ se va difuza un eveniment sportiv, celebrarea anumitor sӑrbӑtori publice sau desfӑşurarea de spectacole în incinta restaurantului.

O funcţionalitate posibilӑ pentru utilizatorii de tip “client” care ar îmbunӑtӑţi şi modul de lucru al chelnerilor, ar fi posibilitatea de platӑ prin intermediul aplicaţiei.

## Concluzie finalӑ

Consider cӑ acest tip de aplicaţie îşi va putea gӑsi rostul în societate. Aplicaţia creatӑ de mine poate îmbunӑtӑţi enorm condiţiile de muncӑ a chelnerilor şi în același timp sӑ facӑ şi experienţa clienţilor în restaurant una cât mai plӑcutӑ.

Deoarece aplicaţia are ca rezultat diminuarea contactului chelner – client, acest lucru poate duce la posibilitӑţi noi de angajare în acest domeniu a persoanelor cu dizabilitaţi în ceea ce priveşte auzul. O altӑ îmbunӑtӑţire adusӑ de aplicaţie se poate observa într-un context pandemic sau atunci când bolile sezoniere îşi fac apariţia, deorece un contact om – om redus scade şansa transmiterii bolilor.

# Bibliografie

[1] anuntul.ro. (2022). *Angajare Livrator, curier Glovo, Tazz - VnjmXL*. <https://www.anuntul.ro/anunt-angajare-livrator-foodpanda-VnjmXL>

[2] Node.js (2022). *About | Node.js*. <https://nodejs.org/en/about/>

[3] Engineering Education (EngEd) Program | Section. (2020). *The History of Node.js*. <https://www.section.io/engineering-education/history-of-nodejs/>

[4] Wikipedia Contributors (2022). *Node.js* <https://en.wikipedia.org/wiki/Node.js#/media/File:Node.js_logo.svg>

[5] Expressjs.com. (2017). *Express - Node.js web application framework*. <https://expressjs.com>

[6] Wikipedia Contributors (2022). *Express.js*. Wikipedia. <https://en.wikipedia.org/wiki/Express.js#/media/File:Expressjs.png>

[7] Reactjs.org. (2021). *React – A JavaScript library for building user interfaces*. <https://reactjs.org/>

[8] Educationecosystem.com. (2022). Available at: <https://educationecosystem.com/guides/programming/react-js/history>

[9] mrmrs (2022). *TACHYONS - Css Toolkit*. Tachyons.io. Available at: <https://tachyons.io/#principles>

[10] Page, D. (2022). *FAQ*. Pgadmin.org. <https://www.pgadmin.org/faq/>

[11] www.pgadmin.org. (n.d.). *Screenshots*. <https://www.pgadmin.org/screenshots/#1>

[12] Postgresql.org. (2022). *PostgreSQL: About*. <https://www.postgresql.org/about/>

[13] Group, D. (2022). *PostgreSQL*: <https://www.postgresql.org/>

[14] Knexjs.org. (2022). *SQL Query Builder for Javascript | Knex.js*. [ht tp://knexjs.org/](http://knexjs.org/)

[15] Goqr.me. (2022). *QR Code Generator – create QR codes for free (Logo, T-Shirt, vCard, EPS)*. <https://goqr.me/>

[16] Postman.com. (2022). *Postman*: <https://www.postman.com/>

[17] Postman API Platform. (2022). *About Postman*. <https://www.postman.com/company/about-postman/>

[18] Keff (2020). *Download Any File from Blob*. DEV Community. <https://dev.to/nombrekeff/download-file-from-blob-21ho>

[19] free (2022). *Flowchart Maker & Online Diagram Software*. Diagrams.net. <https://app.diagrams.net/>

[20] https://www.facebook.com/flaticon (2022). *Waiter Icon - 843259*. Flaticon. (imagine folositӑ ca iconiţӑ) <https://www.flaticon.com/free-icon/waiter_843259>

[21] Freepik. (2020). *Summer terrace, night outdoor city cafe, coffeehouse with wooden table, chairs, illumination and potted plants, chalkboard menu on cityscape view. modern street cafeteria, cartoon illustration Free Vector (imagine folositӑ pentru fundalul aplicaţiei)*. <https://www.freepik.com/free-vector/summer-terrace-night-outdoor-city-cafe-coffeehouse-with-wooden-table-chairs-illumination-potted-plants-chalkboard-menu-cityscape-view-modern-street-cafeteria-cartoon-illustration_10308146.htm#page=1&query=restaurant&position=20&from_view=search>

[22] Freepik. (2020). *Vector realistic red tablecloth on tables Free Vector*. (imagine folositӑ pentru reprezentarea meselor) <https://www.freepik.com/free-vector/vector-realistic-red-tablecloth-tables_9395684.htm#page=1&query=table&position=47&from_view=search>

[23] KindPNG.com. (2019). *Bubble Grey Clipart - Gray Dot Icon Png, Transparent Png – kindpng (imagine folositӑ pentru “sistemul de notificare” valoarea inactiv)*: <https://www.kindpng.com/imgv/omTTRb_bubble-grey-clipart-gray-dot-icon-png-transparent/>

[24] TopPNG. (2019). *red dot icon png - bowling ball transparent background PNG image with transparent background png - Free PNG Images (imagine folositӑ pentru “sistemul de notificare” valoarea activ)*: <https://toppng.com/red-dot-icon-png-bowling-ball-transparent-background-PNG-free-PNG-Images_181168>

‌[25] in (2022). *The Complete Web Developer in 2022: Zero to Mastery*. Udemy. <https://www.udemy.com/course/the-complete-web-developer-zero-to-mastery/learn/lecture/8757614?start=0#overview> ‌