**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**PROJEKTUM**

**Web programozás tantárgyból**

|  |  |
| --- | --- |
| **HALLGATÓ(K)**  **Hupkó Dominik**  **Varró Róbert** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SZABADKA, 2024.** | |

Tartalom

[1. A feladatkör meghatározása 3](#_Toc23709)

[2. Kidolgozás 4](#_Toc9523)

[3. Az adatbázis struktúrája (MySQL Workbench) 11](#_Toc20429)

[4. A projekt működésének leírása 12](#_Toc26398)

[5. Felhasznált irodalom 20](#_Toc12096)

# A feladatkör meghatározása

https://humanz.stud.vts.su.ac.rs/Restaurant/

A projekt célja egy étterem weboldalának elkészítése, amely kezeli az asztalok foglalását. A felhasználók regisztrálhatnak e-mail verifikációval, majd asztalt foglalhatnak legalább egy órára, maximum hat órára. A felhasználók lemondhatják a foglalásukat, amennyiben az nem az elkövetkező négy órában van. Az adminisztrátorok munkásokat vehetnek fel, hozzáadhatnak és törölhetnek asztalokat, valamint láthatják minden felhasználó adatait és felfüggeszthetik a tagságukat. A munkások asztalokat adhatnak hozzá, megtekinthetik a felhasználókat és követhetik a foglalásokat.

A weboldal funkciói közé tartozik a felhasználói regisztráció és bejelentkezés, ahol az új felhasználói fiókok e-mail verifikációval jönnek létre. A felhasználók legalább egy órára, de maximum hat órára foglalhatnak asztalt, és lehetőségük van módosítani vagy lemondani a foglalást, amennyiben az legalább négy órával később kezdődik. Az adminisztrátorok feladatai közé tartozik a munkások felvétele és kezelése, asztalok hozzáadása és törlése, valamint a felhasználói adatok megtekintése és szükség esetén a tagságok felfüggesztése. A munkások az asztalok kezelésével, a felhasználók megtekintésével és a foglalások követésével foglalkoznak. Ez a struktúra biztosítja, hogy az étterem weboldala könnyen kezelhető és hatékony legyen mind a felhasználók, mind az adminisztrátorok számára.

# Kidolgozás

Az oldal funkcionalitását egy functions.php nevű fájlba irányítjuk, ahol egy Functions nevű osztályban tároljuk a függvényeket és ezeket egy Switch-el hívjuk meg az előző oldalak általa bejött adatok alapján.

$functions = new Functions();

class Functions

{

public function \_\_construct()

{

$this->run();

}

public function run(){

if (isset($\_POST['action'])) {

switch ($\_POST['action']) {

case 'registration':

$this->registration();

break;

case 'Log in':

$this->login();

break;

case 'kijelentkezes':

$this->logOut();

break;

case 'picture':

$this->picture($\_SESSION['backPic']);

break;

case 'ResetPass':

$this->resetPassword();

break;

case 'ModifyUser':

$this->modifyUser();

break;

case 'Send':

$this->mailAddAndPasswordChange();

break;

case 'AddWorker':

$this->addWorker();

break;

case 'ModifyTable':

$this->modifyTable();

break;

case 'AddMenu':

$this->addMenu();

break;

case 'ModifyDish':

$this->modifyMenu();

break;

case 'AddCoupon':

$this->addCoupon();

break;

case 'BanPerson':

case 'UnBanPerson':

$this->ban();

break;

case 'AddTable':

$this->addTable();

break;

default:

$\_SESSION['message'] = "Something went wrong in switch";

header('Location:index.php');

exit();

}

} elseif (isset($\_SESSION['action'])) {

switch ($\_SESSION['action']) {

case 'kijelentkezes':

$this->logOut();

break;

}

}

}

Ez a kod beleilleszti a hibákat az errorlog táblába ahol meg van hívva

public function errorLogInsert($mail, $errorText, $logType, $logMessage) {

global $conn;

sleep(2);

$time = time();

$currentTime = date("Y-m-d H:i:s", $time);

$sql = "INSERT INTO errorlog( errorType, errorMail, errorText, errorTime) VALUES (?,?,?,?)";

$stmt = $conn->prepare($sql);

if (!$stmt) {

$\_SESSION['message'] = 'Error in SQL query: ' . $conn->error;

}

$stmt->bind\_param("ssss", $logType, $mail, $errorText, $currentTime);

if ($stmt->execute()) {

$\_SESSION['message'] = $logMessage;

} else {

$\_SESSION['message'] = "NUAH.";

}

$stmt->close();

}

Ez a kod a bejelentkezésre szolgál:

public function login() {

global $conn;

if (isset($\_POST['mail'], $\_POST['pass'])) {

$mail = $\_POST['mail'];

$password = $\_POST['pass'];

$sql = "SELECT \* FROM user WHERE userMail = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("s", $mail);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

$row = $result->fetch\_assoc();

sleep(2);

if (empty($row['userPassword'])) {

$\_SESSION['message'] = "The worker did not set up a password!";

} else {

if (password\_verify($password, $row['userPassword'])) {

if ($row['banned']) {

$\_SESSION['message'] = "You have been banned from our website!";

} else {

$\_SESSION['email'] = $row['userMail'];

$\_SESSION['name'] = $row['firstName'] . " " . $row['lastName'];

$\_SESSION['profilePic'] = $row['profilePic'];

$\_SESSION['userID'] = $row['userId'];

header('Location: index.php');

exit();

}

} else {

$logType = "Log in";

$logText = "The password was not valid!";

$logMessage = "Wrong password!";

$this->errorLogInsert($mail, $logText, $logType, $logMessage);

$\_SESSION['message'] = "Wrong password!";

}

}

} else {

$\_SESSION['message'] = "Something went wrong, maybe the mail is not registered!";

$logType = "Log in";

$logText = "The E-mal is not in our database";

$logMessage = $\_SESSION['message'];

$this->errorLogInsert($mail, $logText, $logType, $logMessage);

}

} else {

$\_SESSION['message'] = "Email or password not set!";

}

header('Location: logIn.php');

exit();

}

Ez a kód kiírassa a felhasználókat az adatbázisból, e-mail keresés alapján, illetve az összes felhasználót, ha nem keresünk felhasználót

class User

{

public function \_\_construct($rank,$mail,$previousPage)

{

$this->previousPage = $previousPage;

$this->mail = $mail;

$this->rank=$rank;

}

public function userString()

{

global $conn;

if($this->mail){

$sql = mysqli\_prepare($conn, "SELECT \* FROM user WHERE privilage = ? AND userMail LIKE ?");

$mailLike = "%" . $this->mail . "%"; // Prepare the mail variable with wildcards

$sql->bind\_param("ss", $this->rank, $mailLike);

}

else{

$sql = mysqli\_prepare($conn, "SELECT \* FROM user where privilage =?");

$sql->bind\_param("s", $this->rank);

}

Ez a kód a kuponok kiírására szolgál:

<?php

if (isset($\_SESSION['email']) && isset($\_SESSION['name']) && isset($\_SESSION['profilePic'])) {

if (isset($\_POST['searchAction']) && !empty($\_POST['searchDate'])) {

$searchDate = $\_POST['searchDate'];

$sql = mysqli\_prepare($conn, "SELECT \* FROM coupon WHERE discount = ? and discountValidate=1 ORDER BY discount DESC");

$sql->bind\_param("s", $searchDate);

} else {

$sql = mysqli\_prepare($conn, "SELECT \* FROM coupon WHERE discountValidate=1 ORDER BY discount DESC");

}

$sql->execute();

$result = $sql->get\_result();

// Check if there are results to display

if ($result->num\_rows > 0) {

$\_SESSION['previousPage'] = "coupon.php";

echo '<div class="container">

<div class="row justify-content-around">';

while ($row = $result->fetch\_assoc()) {

echo '<div class="col-xl-4 p-3 border bg-dark" style="margin: auto; margin-top:100px; margin-bottom: 50px; width: fit-content;">';

echo '<label>Coupon discount: ' . htmlspecialchars($row['discount']) . '%</label><br>';

echo '<label>Coupon code: ' . htmlspecialchars($row['discountCode']) . '</label><br>';

echo '</div>';

}

echo '</div></div>';

}

}

?>

Ez a kód a kuponok Cookie-k számlálására szolgál, ami akkor generálódik ha valaki meglátogatja az oldalt, és a Cookie 24 óráig tart:

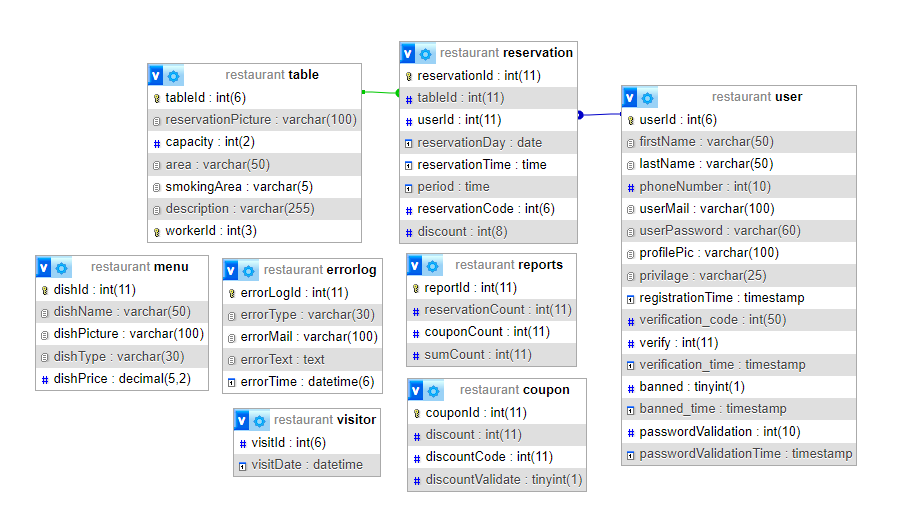
<?php

global $conn;

$query = mysqli\_query($conn,"SELECT count(\*) from visitor");

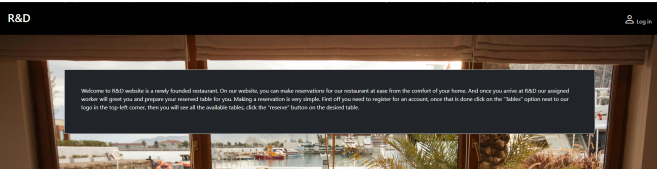
echo '<a style="color: #ffffff">Visit Count: ' . $query->fetch\_array(MYSQLI\_NUM)[0] . '</a>';

# Az adatbázis struktúrája (MySQL Workbench)

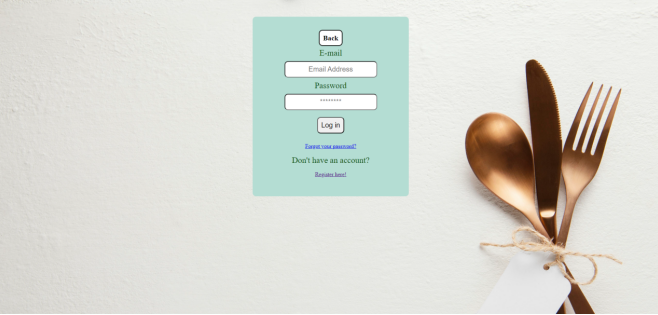


# A projekt működésének leírása

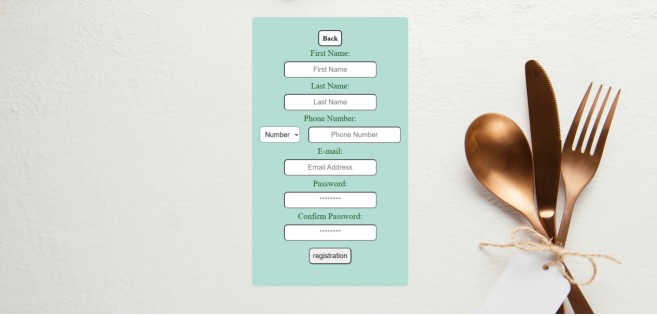
Ez a főoldal amikor először nyitjuk meg a böngészőben

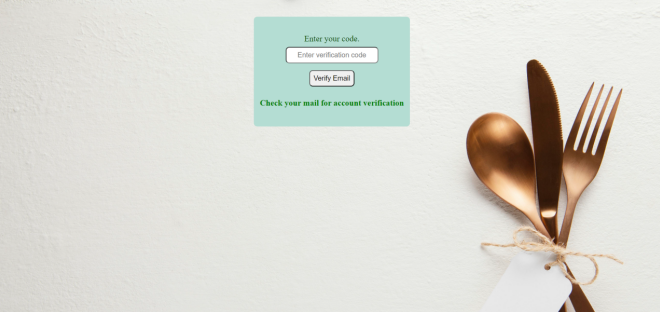


Itten bejelentkezünk az oldalra:

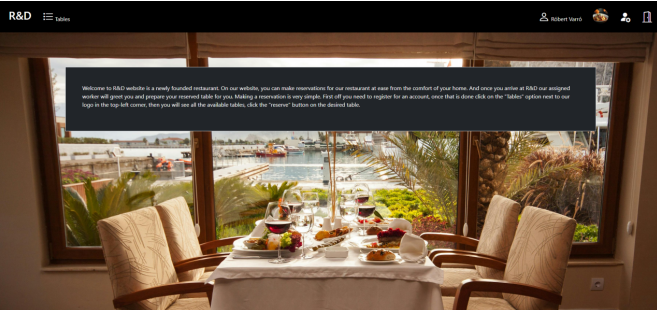


Hogyha nincsen profilunk akkor regisztrálunk és a validációs kód segítségével érvényesítjük a profilunkat majd vissza bejelentkezünk.

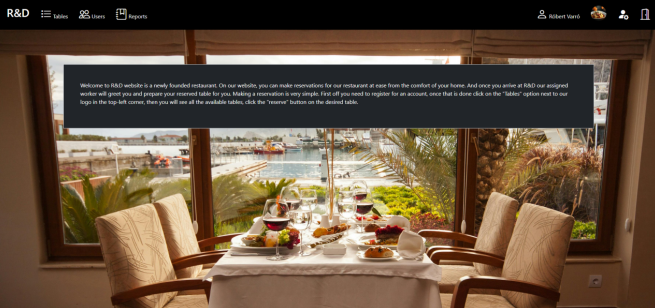




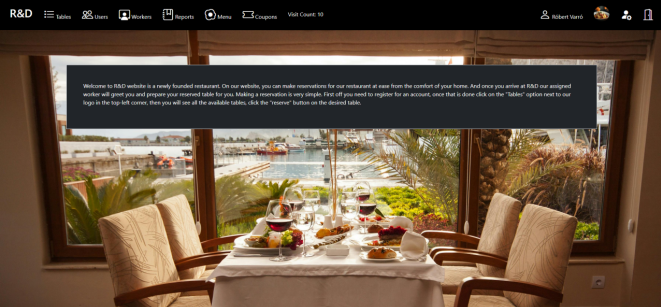
Miután be vagyunk jelentkezve, az oldal így fog kinézni a sima felhasználónak:



Így néz ki az oldal egy munkásnál:



És így néz ki az rendszergazdánál az oldal. Itten az összes funkció látható, illetve hogy hányan látogatták meg az oldalt.



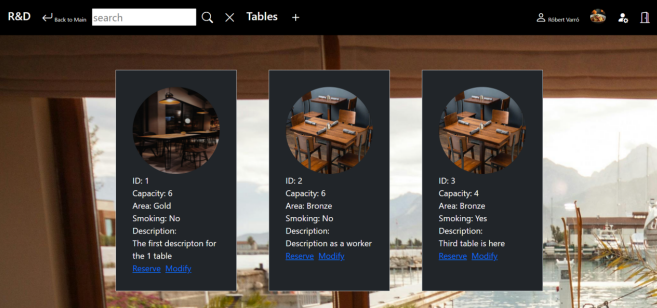
A felhasználó innentől kezdve tudja módosítani az adatit illetve profilképet cserélni. A kép csere úgy megy hogy rákattint a profilképre és kiválasztja a képet és ha az oldal megfelelőnek találja, akkor be is illeszti azt.

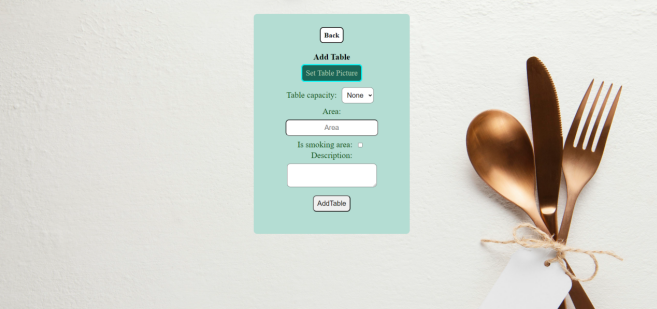


Illetve így néz ki az oldal ahol módosítani tuja a felhasználó az adatait, és nem kötelező mindent kitöltenie.

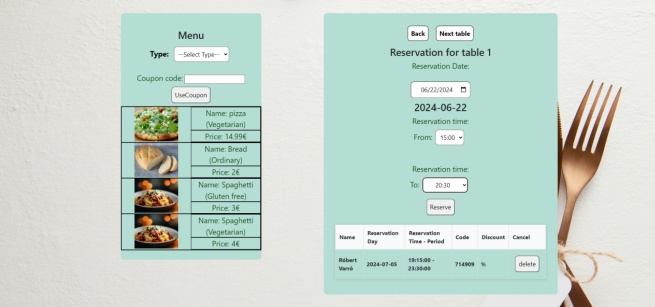


Az első oldal az tables, ami az összes asztalt mutassa az admin szemszögéből. Itten kiválasztunk egy asztalt, amit ki is tudunk keresni. Admin illetve dolgozóként bírjuk csak látni a hozzáadás illetve a modifikálni az asztalt.

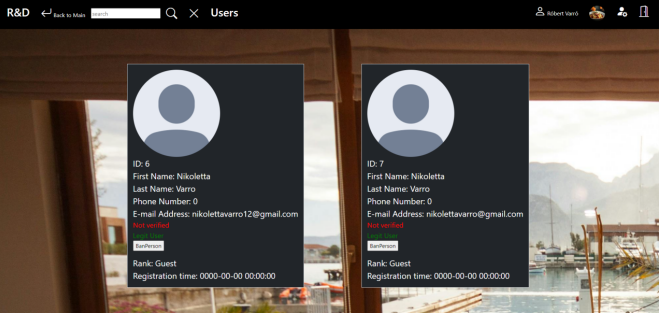




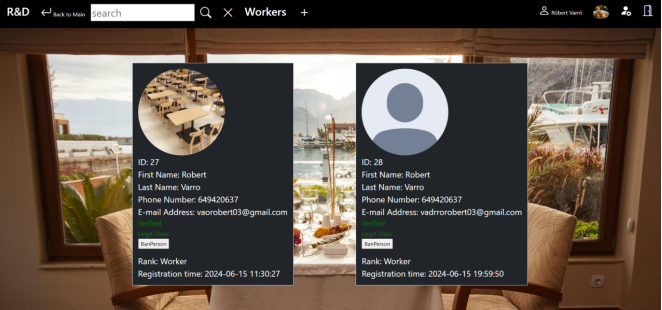
Itten le tudjuk foglalni az asztalt, és megtudjuk nézni az étlapot a kuponnal. Az étlapon kategóriák szerint lehet keresni az ételt és a kupont felhasználjuk amikor lefoglaltuk az asztalt. Hogyha a dátum még nem járt le és az asztal nincsen lefoglalva, akkor a felhasználó letudja foglalni az asztal. Hogyha valamilyen oknál fogva nem bírja lefoglalni az asztalt akkor azt kijelzi az oldal. Például a felhasználó nem tudja lefoglalni az asztalt hogyha azon a napon már egyszer lefoglalta, több mint öt foglalása van a jövőben, illetve az időpont foglalt más által. Miután ez megvan a felhasználó láthassa az összes foglalását az összes foglalás mellet. Ez után kap egy e-mail a foglalás sikerességéről és arról hogy mikor fog ez bekövetkezni. Ameddig a foglalás nem négy óra múlva fog bekövetkezni akkor le tudja mondani a foglalást, más esetben az oldal hibaüzenet küld hogy nem lehet lemondani a foglalást.



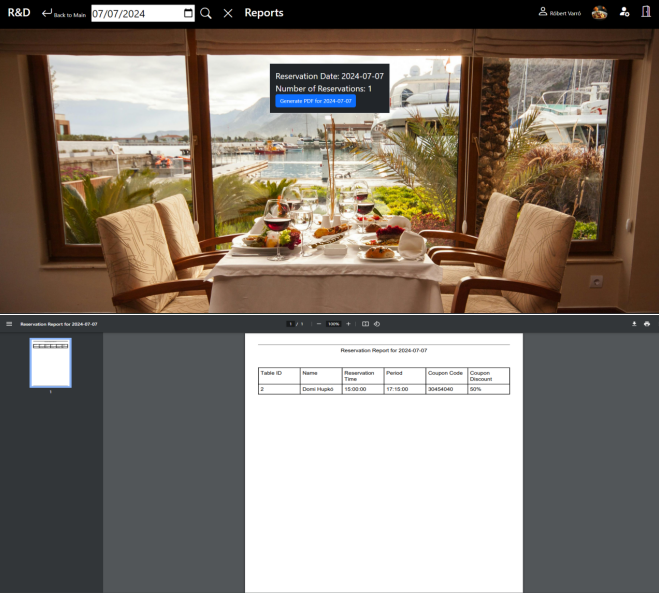
Ezen az oldalon láthassuk az összes felhasználót. A felhasználókat tudjuk keresni e-mail alapján illetve le tudjuk tiltani őket az oldalról szükség van rá és vissza is tudjuk vonni. Ezt az oldalt csak a munkások és a rendszergazda láthassák, hogyha valaki más próbál hozzáférni az oldalhoz, linken keresztül, akkor a fő oldalra lesz átirányítva. Minden oldal amihez csak a munkások és a rendszergazda férhetnek hozzá rendelkeznek ezzel a védelemmel. User class segítségével írjuk ki a felhasználókat.



Ugyanez a logika a dolgozók oldalon is, és a felhasználókat a User class segítségével írjuk ki. Az oldalhoz egyedül csak a rendszergazda férhet hozzá.

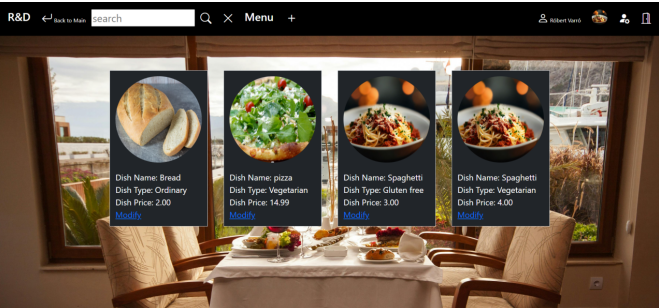


A következő oldalhoz az admin illetve a munkás is hozzáférhet. Itten jelentéseket tudunk generálni PDF formátumban. Az utolsó hét napot mutassa ki alapból, és ha valamelyik PDF generálás kódra kattintunk akkor egy másik oldalon le generálja nekünk az adott napon lévő foglalásokat. Hogyha meg nincsenek foglalások aznap akkor egyszerűen a PDF-be legenerálódik, viszont egy üzenet lesz benne, miszerint nincsenek foglalások aznapra.

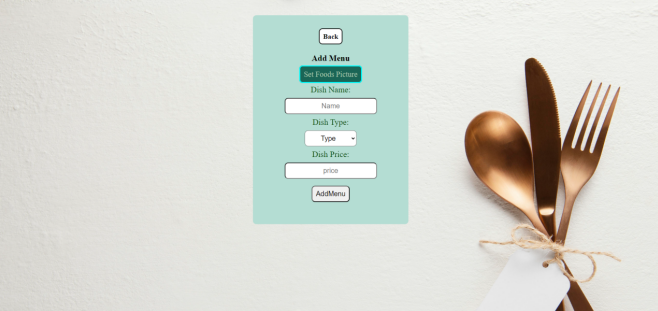


Az itteni keresőben egy kalendárium segítségével konkrét napot tudunk kikeresni, aminek a jelentését legenerálhatjuk, ha még nem érkezett el a nap akkor is megtudjuk tenni ezt a jelenlegi foglalásokkal. Mivel az adatbázisban nem törlődnek az adatok, ezért mindig hozzá lehet férni itten a nap foglalásaihoz, amit a felhasználók is látnak az oldalon.

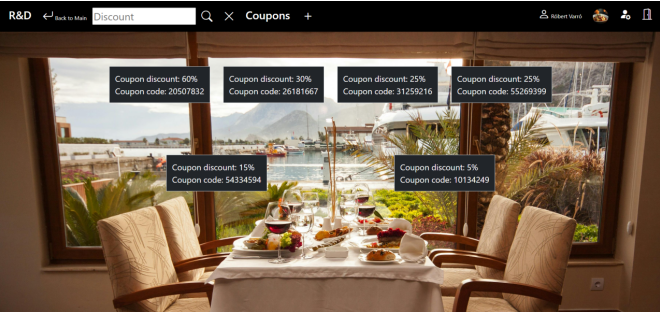
A következő oldal az étlapon szereplő ételeket lehet látni, amihez a rendszergazda fér hozzá. Itten tudja az ételeket hozzáadni illetve módosítani az oldalon.

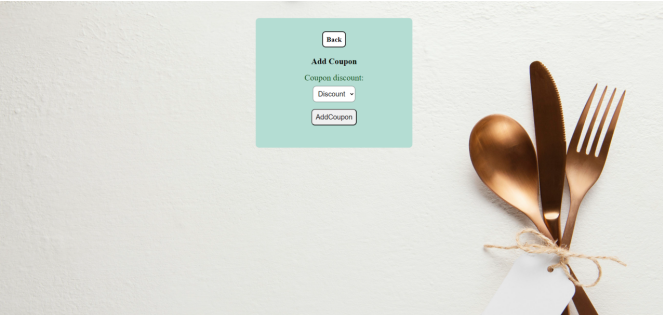


Ez az oldal felel azért hogy új ételeket rakjunk az adatbázisba. Hasonló a kinézett a az oldanak ahol ezt módosítani lehet. Illetve az asztalok hozzáadásánál és módosításánál is ugyanaz a logika és nagyon hasonló a kinézett.



A következő oldal ami szintén csak a rendszergazda férhet hozzá, a coupon oldal ahol az érvényes kuponokat lehet látni illetve újakat generálni.





# Felhasznált irodalom

W1. https://stackoverflow.com/

W2. <https://github.com/>

W3. https://www.php.net/