**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**PROJEKTUM**

**Web programozás tantárgyból**

|  |  |
| --- | --- |
| **HALLGATÓ(K)**  **Sőregi Viktor 26222073**  **Balog Bálint 26222072** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SZABADKA, 2024.** | |

Tartalom

[1. A feladatkör meghatározása 3](#_Toc22259)

[2. Kidolgozás 4](#_Toc21592)

[3. Az adatbázis struktúrája (MySQL Workbench) 17](#_Toc9998)

[4. Projekt működésének leírása 18](#_Toc30185)

[5. Használt irodalmak: 23](#_Toc32081)

# A feladatkör meghatározása

Projekt webcíme:https://lite.stud.vts.su.ac.rs/

Téma körünk egy fogorvosi rendelő oldalának kivitelezése volt.

A weboldalon három fajta hozzá férhetőség van:

-Első a látogató aki csak az orvosok adatait,az árakat lássák illetve bírnak regisztrálni és bejelentkezni.

-Második szint a páciens ők már rendelkeznek profillal és bejelentkezés után több mindent megtudnak nézni például időpont foglalást tudnak leadni, a foglalásaikat megbírják nézni vagy a kartonukat.

-Harmadik szint az orvos ők mindenhez hozzá férnek írharnaka a páciensek kartonjaikba vagy adhatnak hozzá új orvost a rendszerhez vagy új szolgáltatást.

# Kidolgozás

db-config file:

<?php  
$dsn = 'mysql:host=localhost;dbname=dentist';  
$username = 'root';  
$password = '';  
  
try {  
 $pdo = new PDO($dsn, $username, $password);  
 $pdo->setAttribute(PDO::*ATTR\_ERRMODE*, PDO::*ERRMODE\_EXCEPTION*);  
} catch (PDOException $e) {  
 echo 'Connection failed: ' . $e->getMessage();  
 exit;  
}

Nagyon sok helyt megjelenik az adatok betöltése és frissítése az adatbázisban:

<?php  
include 'db-config.php';  
  
global $pdo;  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $operation = $\_POST["operation"];  
 $doctorID = isset($\_POST["doctorID"]) ? $\_POST["doctorID"] : null;  
 $firstName = isset($\_POST["firstName"]) ? $\_POST["firstName"] : null;  
 $lastName = isset($\_POST["lastName"]) ? $\_POST["lastName"] : null;  
 $password = isset($\_POST["password"]) ? password\_hash($\_POST["password"], *PASSWORD\_BCRYPT*) : null;  
 $phoneNumber = isset($\_POST["phoneNumber"]) ? $\_POST["phoneNumber"] : null;  
 $email = isset($\_POST["email"]) ? $\_POST["email"] : null;  
 $worktime = isset($\_POST["worktime"]) ? $\_POST["worktime"] : null;  
 $specialisation = isset($\_POST["specialisation"]) ? $\_POST["specialisation"] : null;  
 $forget = isset($\_POST["forget"]) ? $\_POST["forget"] : null;  
 $remember = isset($\_POST["remember"]) ? $\_POST["remember"] : null;  
  
 if ($operation == "insert" || $operation == "update") {  
 // Check if email already exists  
 $checkEmailSql = "SELECT doctorID FROM Doctor WHERE email = :email";  
 $stmt = $pdo->prepare($checkEmailSql);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->execute();  
 $existingDoctor = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 $existingDoctorID = $existingDoctor ? $existingDoctor['doctorID'] : null;  
 }  
  
 if ($operation == "insert") {  
 if ($existingDoctorID === null) {  
 $sql = "INSERT INTO Doctor (firstName, lastName, password, phoneNumber, email, worktime, specialisation, forget, remember)  
 VALUES (:firstName, :lastName, :password, :phoneNumber, :email, :worktime, :specialisation, :forget, :remember)";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':firstName', $firstName, PDO::PARAM\_STR);  
 $stmt->bindParam(':lastName', $lastName, PDO::PARAM\_STR);  
 $stmt->bindParam(':password', $password, PDO::PARAM\_STR);  
 $stmt->bindParam(':phoneNumber', $phoneNumber, PDO::PARAM\_STR);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->bindParam(':worktime', $worktime, PDO::PARAM\_STR);  
 $stmt->bindParam(':specialisation', $specialisation, PDO::PARAM\_STR);  
 $stmt->bindParam(':forget', $forget, PDO::PARAM\_STR);  
 $stmt->bindParam(':remember', $remember, PDO::PARAM\_STR);  
  
 if ($stmt->execute()) {  
 echo "Bevitel sikeres volt";  
 } else {  
 echo "Error: " . $stmt->errorInfo()[2];  
 }  
 } else {  
 echo "Email már létezik";  
 }  
 } elseif ($operation == "update") {  
 if (!empty($doctorID)) {  
 if ($existingDoctorID === null || $existingDoctorID == $doctorID) {  
 $sql = "UPDATE Doctor  
 SET firstName = :firstName, lastName = :lastName, password = :password, phoneNumber = :phoneNumber, email = :email, worktime = :worktime, specialisation = :specialisation  
 WHERE doctorID = :doctorID";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':firstName', $firstName, PDO::PARAM\_STR);  
 $stmt->bindParam(':lastName', $lastName, PDO::PARAM\_STR);  
 $stmt->bindParam(':password', $password, PDO::PARAM\_STR);  
 $stmt->bindParam(':phoneNumber', $phoneNumber, PDO::PARAM\_STR);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->bindParam(':worktime', $worktime, PDO::PARAM\_STR);  
 $stmt->bindParam(':specialisation', $specialisation, PDO::PARAM\_STR);  
 $stmt->bindParam(':doctorID', $doctorID, PDO::*PARAM\_INT*);  
  
 if ($stmt->execute()) {  
 echo "Módosítás sikeres.<br>";  
 } else {  
 echo "Error: " . $stmt->errorInfo()[2] . "<br>";  
 }  
 } else {  
 echo "Email már létezik<br>";  
 }  
 } else {  
 echo "Doctor ID szűkséges a módosításhoz.<br>";  
 }  
 }  
}  
  
header("Location: ../admin.php");  
exit();  
?>

Az adatbázisból való törlés is sűrűn feltűnik a kódban:

<?php  
include 'db-config.php';  
  
global $pdo;  
  
$doctorID = isset($\_POST['doctorID']) ? intval($\_POST['doctorID']) : null;  
  
if (!empty($doctorID)) {  
 try {  
 $sql = $pdo->prepare("DELETE FROM Doctor WHERE doctorID = :doctorID");  
 $sql->bindParam(':doctorID', $doctorID, PDO::*PARAM\_INT*);  
  
 if ($sql->execute()) {  
 echo "Sikeres törlés<br>";  
 } else {  
 echo "Error.<br>";  
 }  
 } catch (PDOException $e) {  
 echo "Error: " . $e->getMessage() . "<br>";  
 }  
} else {  
 echo "Doctor ID szükségés a törléshez<br>";  
}  
  
$pdo = null;  
?>

Megmutassa a lefoglalt időpontokat:

<?php  
require 'db-config.php';  
  
global $pdo;  
  
if (isset($\_GET['doctor\_id']) && isset($\_GET['date'])) {  
 $doctor\_id = intval($\_GET['doctor\_id']);  
 $date = $\_GET['date'];  
  
 try {  
 $stmt = $pdo->prepare("SELECT TIME(schedule) AS time FROM Appointment WHERE doctorID = :doctor\_id AND DATE(schedule) = :date");  
 $stmt->bindParam(':doctor\_id', $doctor\_id, PDO::*PARAM\_INT*);  
 $stmt->bindParam(':date', $date, PDO::PARAM\_STR);  
 $stmt->execute();  
  
 $bookedTimes = [];  
 while ($row = $stmt->fetch(PDO::*FETCH\_ASSOC*)) {  
 $bookedTimes[] = $row['time'];  
 }  
  
 echo json\_encode($bookedTimes);  
 } catch (PDOException $e) {  
 echo json\_encode(['error' => $e->getMessage()]);  
 }  
} else {  
 echo json\_encode([]);  
}

Doktorok munkaideje:

<?php  
require 'db-config.php';  
  
global $pdo;  
  
if (isset($\_GET['doctor\_id'])) {  
 $doctor\_id = intval($\_GET['doctor\_id']);  
  
 try {  
 $stmt = $pdo->prepare("SELECT worktime FROM Doctor WHERE doctorID = :doctor\_id");  
 $stmt->bindParam(':doctor\_id', $doctor\_id, PDO::*PARAM\_INT*);  
 $stmt->execute();  
 $doctor = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 // Fetch schedule from DoctorSchedule table  
 $stmt2 = $pdo->prepare("SELECT Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday FROM DoctorSchedule WHERE doctorID = :doctor\_id");  
 $stmt2->bindParam(':doctor\_id', $doctor\_id, PDO::*PARAM\_INT*);  
 $stmt2->execute();  
 $schedule = $stmt2->fetch(PDO::*FETCH\_ASSOC*);  
  
 if ($doctor && $schedule) {  
 $worktime = explode('-', $doctor['worktime']);  
 $start\_time = $worktime[0];  
 $end\_time = $worktime[1];  
  
 echo json\_encode([  
 'start' => $start\_time,  
 'end' => $end\_time,  
 'days' => $schedule  
 ]);  
 } else {  
 echo json\_encode(['error' => 'Doctor not found.']);  
 }  
 } catch (PDOException $e) {  
 echo json\_encode(['error' => $e->getMessage()]);  
 }  
} else {  
 echo json\_encode(['error' => 'Invalid request.']);  
}  
?>

Páciens fogainak egészsége:

<?php  
require 'db-config.php';  
global $pdo;  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['patientID'])) {  
 $patientID = intval($\_POST['patientID']);  
  
 try {  
 $stmt = $pdo->prepare("SELECT procedureDate, healthRating FROM PatientRecords WHERE patientID = ? ORDER BY procedureDate");  
 $stmt->bindParam(1, $patientID, PDO::*PARAM\_INT*);  
 $stmt->execute();  
  
 $result = $stmt->fetchAll(PDO::*FETCH\_ASSOC*);  
  
 if ($result) {  
 $labels = [];  
 $scores = [];  
  
 foreach ($result as $row) {  
 $labels[] = $row['procedureDate'];  
 $scores[] = $row['healthRating'];  
 }  
  
 echo json\_encode(["success" => true, "labels" => $labels, "scores" => $scores]);  
 } else {  
 echo json\_encode(["success" => false, "message" => "Nincs adat a megadott pácienshez."]);  
 }  
 } catch (PDOException $e) {  
 echo json\_encode(["success" => false, "message" => 'Error: ' . $e->getMessage()]);  
 }  
} else {  
 echo json\_encode(["success" => false, "message" => "Hibás kérés."]);  
}

Doktor log:

<?php  
session\_start();  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 require 'db-config.php';  
 global $pdo;  
  
 $email = $\_POST['email'];  
 $password = $\_POST['password'];  
 $remember\_me = isset($\_POST['remember\_me']);  
  
 $sql = "SELECT \* FROM Doctor WHERE email = :email";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->execute();  
 $row = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 if ($row) {  
 if (password\_verify($password, $row['password'])) {  
 $\_SESSION['doctorID'] = $row['doctorID'];  
 $\_SESSION['firstName'] = $row['firstName'];  
 $\_SESSION['lastName'] = $row['lastName'];  
 $\_SESSION['phoneNumber'] = $row['phoneNumber'];  
 $\_SESSION['email'] = $row['email'];  
  
 if ($remember\_me) {  
 $token = bin2hex(random\_bytes(32));  
 setcookie('remember\_me\_cookie', $token, time() + (86400 \* 30), "/"); // 30 days cookie  
 $sql = "UPDATE Doctor SET remember = :token WHERE doctorID = :doctorID";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':token', $token, PDO::PARAM\_STR);  
 $stmt->bindParam(':doctorID', $row['doctorID'], PDO::*PARAM\_INT*);  
 $stmt->execute();  
 }  
  
 header("Location: ../index.php");  
 exit();  
 } else {  
 header("Location: ../login.php?message=" . urlencode("Helytelen jelszó vagy email cím.") . "&type=alert");  
 }  
 } else {  
 header("Location: ../login.php?message=" . urlencode("Helytelen jelszó vagy email cím.") . "&type=alert");  
 }  
}  
?>

Bejelentkezés:

<?php  
session\_start();  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 require 'db-config.php';  
 global $pdo;  
  
 $email = $\_POST['email'];  
 $password = $\_POST['password'];  
 $remember\_me = isset($\_POST['remember\_me']) ? true : false;  
  
 $sql = "SELECT \* FROM Patient WHERE email = :email";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->execute();  
 $row = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 if ($row) {  
 if (password\_verify($password, $row['password'])) {  
 $\_SESSION['patientID'] = $row['patientID'];  
 $\_SESSION['firstName'] = $row['firstName'];  
 $\_SESSION['lastName'] = $row['lastName'];  
 $\_SESSION['phoneNumber'] = $row['phoneNumber'];  
 $\_SESSION['email'] = $row['email'];  
  
 if ($remember\_me) {  
 $token = bin2hex(random\_bytes(32));  
 setcookie('remember\_me\_cookie', $token, time() + (86400 \* 30), "/"); // 30 days cookie  
 $sql = "UPDATE Patient SET remember = :token WHERE patientID = :patientID";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':token', $token, PDO::PARAM\_STR);  
 $stmt->bindParam(':patientID', $row['patientID'], PDO::*PARAM\_INT*);  
 $stmt->execute();  
 }  
  
 header("Location: ../index.php");  
 exit();  
 } else {  
 $error\_message = "Helytelen jelszó vagy email cím.";  
 }  
 } else {  
 $error\_message = "Helytelen jelszó vagy email cím.";  
 }  
  
 // Redirect back to login page with error message  
 header("Location: ../login.php?error=" . urlencode($error\_message));  
 exit();  
} else {  
 header("Location: ../login.php");  
 exit();  
}  
?>

Kijelentkezés:

<?php  
session\_start();  
require 'db-config.php';  
global $pdo;  
  
// Delete cookie  
if (isset($\_SESSION['patientID'])) {  
 if (isset($\_COOKIE['remember\_me\_cookie'])) {  
 unset($\_COOKIE['remember\_me\_cookie']);  
 setcookie('remember\_me\_cookie', '', ['expires' => time() - 3600, 'path' => '/']);  
 $sql = "UPDATE Patient SET remember = NULL WHERE patientID = :patientID";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':patientID', $\_SESSION['patientID'], PDO::*PARAM\_INT*);  
 $stmt->execute();  
 }  
}  
  
if (isset($\_SESSION['doctorID'])) {  
 if (isset($\_COOKIE['remember\_me\_cookie'])) {  
 unset($\_COOKIE['remember\_me\_cookie']);  
 setcookie('remember\_me\_cookie', '', ['expires' => time() - 3600, 'path' => '/']);  
 $sql = "UPDATE Doctor SET remember = NULL WHERE doctorID = :doctorID";  
 $stmt = $pdo->prepare($sql);  
 $stmt->bindParam(':doctorID', $\_SESSION['doctorID'], PDO::*PARAM\_INT*);  
 $stmt->execute();  
 }  
}  
  
// Destroy session  
$\_SESSION = array();  
session\_destroy();  
  
header("Location: ../index.php");  
exit();  
?>

Regisztráció:

<?php  
if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {  
 require 'db-config.php';  
 require 'send\_verification\_email.php';  
  
 global $pdo;  
 $firstName = $\_POST["firstName"];  
 $lastName = $\_POST["lastName"];  
 $phoneNumber = $\_POST["phoneNumber"];  
 $password = $\_POST["password"];  
 $passwordCon = $\_POST["passwordConfirm"];  
 $email = $\_POST["email"];  
 $username = strstr($email, '@', true);  
  
 if ($password !== $passwordCon) {  
 header("Location: ../register.php?message=" . urlencode("Jelszó nem egyezik.") . "&type=alert");  
 exit();  
 }  
  
 $hashed\_password = password\_hash($password, *PASSWORD\_BCRYPT*);  
 $token = bin2hex(random\_bytes(16));  
  
 try {  
 $stmt = $pdo->prepare("INSERT INTO Patient (userName, firstName, lastName, phoneNumber, email, password, auth) VALUES (:username, :firstName, :lastName, :phoneNumber, :email, :password, :auth)");  
 $stmt->bindParam(':username', $username, PDO::PARAM\_STR);  
 $stmt->bindParam(':firstName', $firstName, PDO::PARAM\_STR);  
 $stmt->bindParam(':lastName', $lastName, PDO::PARAM\_STR);  
 $stmt->bindParam(':phoneNumber', $phoneNumber, PDO::PARAM\_STR);  
 $stmt->bindParam(':email', $email, PDO::PARAM\_STR);  
 $stmt->bindParam(':password', $hashed\_password, PDO::PARAM\_STR);  
 $stmt->bindParam(':auth', $token, PDO::PARAM\_STR);  
  
 if ($stmt->execute()) {  
 if (send\_verification\_email($email, $token)) {  
 header("Location: ../index.php?message=" . urlencode("Regisztráció sikeres volt. Ellenőrizze az email címét a megerősítéshez.") . "&type=success");  
 } else {  
 header("Location: ../index.php?message=" . urlencode("Regisztráció sikeres volt, de a megerősítő email küldése nem sikerült.") . "&type=alert");  
 }  
 exit();  
 } else {  
 header("Location: ../index.php?message=" . urlencode("Regisztráció sikertelen volt.") . "&type=alert");  
 exit();  
 }  
 } catch (PDOException $e) {  
 header("Location: ../index.php?message=" . urlencode("Regisztráció sikertelen volt: " . $e->getMessage()) . "&type=alert");  
 exit();  
 }  
} else {  
 header("Location: ../register.php");  
 exit();  
}  
?>

Regisztráció java script:

function validateForm() {  
 var emailInput = *document*.getElementById("email");  
 var emailHelpBlock = *document*.getElementById("emailHelpBlock");  
 var passwordInput = *document*.getElementById("password");  
 var passwordConfirmInput = *document*.getElementById("passwordConfirm");  
 var passwordHelpBlock = *document*.getElementById("passwordHelpBlock");  
 var passwordConfirmHelpBlock = *document*.getElementById("passwordConfirmHelpBlock");  
 var registerBtn = *document*.getElementById("registerBtn");  
  
 var email = emailInput.value;  
 var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
 var password = passwordInput.value;  
 var numberRegex = /\d/;  
 var uppercaseRegex = /[A-Z]/;  
 var specialCharRegex = /[!@#$%^&\*()\_+\-=\[\]{};':"\\|,.<>\/?]/;  
 var passwordConfirm = passwordConfirmInput.value;  
  
 var validEmail = emailRegex.test(email);  
 var validPassword = numberRegex.test(password) && uppercaseRegex.test(password) && specialCharRegex.test(password);  
 var passwordMatch = password === passwordConfirm;  
  
 // Show or hide email validation message  
 if (emailInput.value.trim() !== "") {  
 emailHelpBlock.style.display = validEmail ? "none" : "block";  
 }  
  
 if (passwordInput.value.trim() !== "") {  
 passwordHelpBlock.style.display = validPassword ? "none" : "block";  
 }  
  
 if (passwordConfirmInput.value.trim() !== "") {  
 passwordConfirmHelpBlock.style.display = passwordMatch ? "none" : "block";  
 }  
  
 registerBtn.disabled = !(validEmail && validPassword && passwordMatch);  
  
 return validEmail && validPassword && passwordMatch;  
}  
  
*window*.addEventListener("load", function () {  
 var emailInput = *document*.getElementById("email");  
 var passwordInput = *document*.getElementById("password");  
 var passwordConfirmInput = *document*.getElementById("passwordConfirm");  
  
 emailInput.addEventListener("input", validateForm);  
 passwordInput.addEventListener("input", validateForm);  
 passwordConfirmInput.addEventListener("input", validateForm);  
  
 validateForm();  
});

Jelszó módosítása:

<?php  
require 'db-config.php';  
global $pdo;  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $token = trim($\_POST['token']);  
 $password = trim($\_POST['password']);  
 $confirm\_password = trim($\_POST['confirm\_password']);  
  
 if ($password !== $confirm\_password) {  
 header("Location: ../reset\_password.php?token=" . urlencode($token) . "&message=" . urlencode("A jelszavak nem egyeznek.") . "&type=alert");  
 exit();  
 }  
  
 $hashed\_password = password\_hash($password, *PASSWORD\_BCRYPT*);  
  
 try {  
 $stmt = $pdo->prepare("SELECT patientID FROM Patient WHERE forgot = ?");  
 $stmt->execute([$token]);  
 $patient = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 $stmt2 = $pdo->prepare("SELECT doctorID FROM Doctor WHERE forgot = ?");  
 $stmt2->execute([$token]);  
 $doctor = $stmt2->fetch(PDO::*FETCH\_ASSOC*);  
  
 if ($patient) {  
 $stmt = $pdo->prepare("UPDATE Patient SET password = ?, forgot = NULL WHERE forgot = ?");  
 $stmt->execute([$hashed\_password, $token]);  
 } elseif ($doctor) {  
 $stmt2 = $pdo->prepare("UPDATE Doctor SET password = ?, forgot = NULL WHERE forgot = ?");  
 $stmt2->execute([$hashed\_password, $token]);  
 } else {  
 header("Location: ../reset\_password.php?token=" . urlencode($token) . "&message=" . urlencode("Érvénytelen vagy lejárt token.") . "&type=alert");  
 exit();  
 }  
  
 header("Location: ../login.php?message=" . urlencode("A jelszó sikeresen visszaállítva.") . "&type=success");  
 exit();  
 } catch (PDOException $e) {  
 header("Location: ../reset\_password.php?token=" . urlencode($token) . "&message=" . urlencode("Hiba történt a jelszó visszaállítása közben: " . $e->getMessage()) . "&type=alert");  
 exit();  
 }  
} else {  
 header("Location: ../index.php?message=" . urlencode("Hibás kérés.") . "&type=alert");  
 exit();  
}

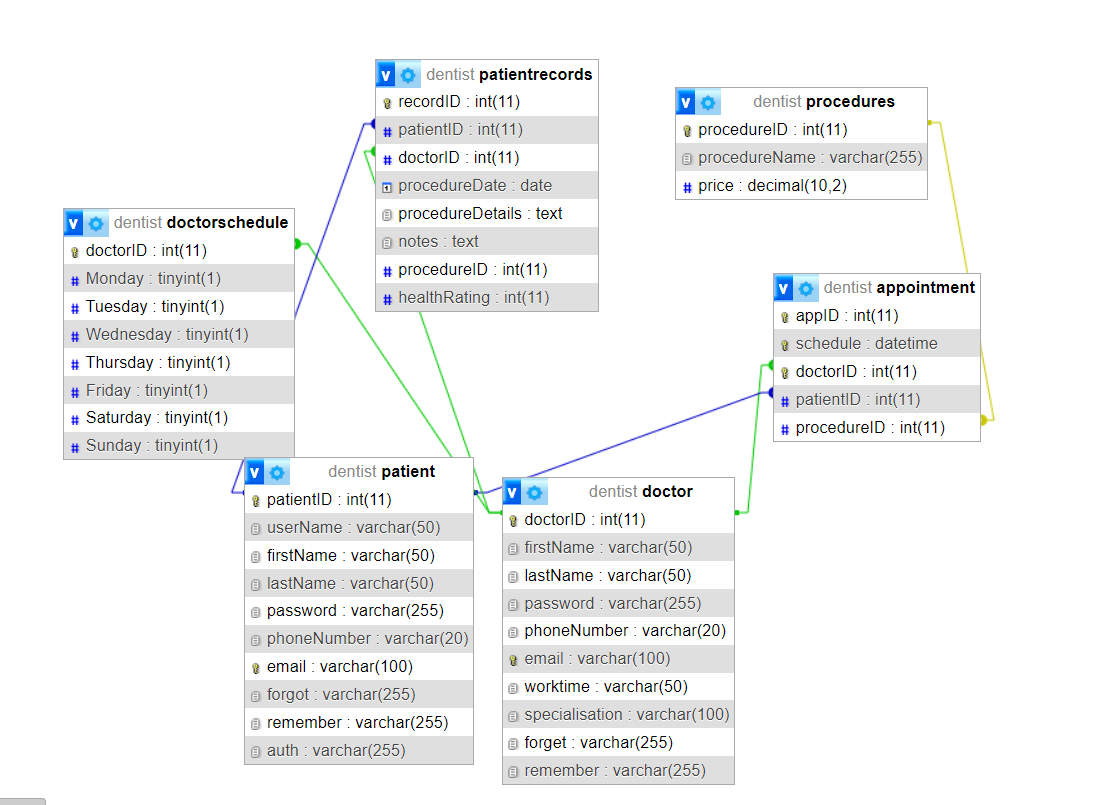
Jelszó módosító e-mail:

<?php  
require '../vendor/autoload.php';  
require 'db-config.php';  
global $pdo;  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $email = trim($\_POST['email']);  
  
 $stmt = $pdo->prepare("SELECT patientID, email FROM Patient WHERE email = ?");  
 $stmt->execute([$email]);  
 $patient = $stmt->fetch(PDO::*FETCH\_ASSOC*);  
  
 $stmt2 = $pdo->prepare("SELECT doctorID, email FROM Doctor WHERE email = ?");  
 $stmt2->execute([$email]);  
 $doctor = $stmt2->fetch(PDO::*FETCH\_ASSOC*);  
  
 if ($patient || $doctor) {  
 $token = bin2hex(random\_bytes(16));  
  
 if ($patient) {  
 $stmt = $pdo->prepare("UPDATE Patient SET forgot = ? WHERE email = ?");  
 $stmt->execute([$token, $email]);  
 } elseif ($doctor) {  
 $stmt2 = $pdo->prepare("UPDATE Doctor SET forgot = ? WHERE email = ?");  
 $stmt2->execute([$token, $email]);  
 }  
  
 $mail = new PHPMailer\PHPMailer\PHPMailer();  
 $mail->isSMTP();  
 $mail->Host = 'in-v3.mailjet.com';  
 $mail->SMTPAuth = true;  
 $mail->Username = '81ea24ce778b6e7ecc44af9aaaca1da3';  
 $mail->Password = '418bade66c7e26bbc9fb672efadd6512';  
 $mail->SMTPSecure = PHPMailer\PHPMailer\PHPMailer::*ENCRYPTION\_STARTTLS*;  
 $mail->Port = 587;  
 $mail->setFrom('balogbalesz1234@gmail.com', 'Fogorvosi rendelő');  
 $mail->addAddress($email);  
 $mail->isHTML(true);  
 $mail->SMTPOptions = array(  
 'ssl' => array(  
 'verify\_peer' => false,  
 'verify\_peer\_name' => false,  
 'allow\_self\_signed' => true  
 )  
 );  
 $mail->Subject = 'Jelszó visszaállítás';  
 $mail->Body = "<p>Kedves felhasználó!</p>  
 <p>Kattintson az alábbi linkre a jelszó visszaállításához:</p>  
 <p><a href='https://lite.stud.vts.su.ac.rs/reset\_password.php?token=$token'>Jelszó visszaállítása</a></p>  
 <p>Ha nem Ön kérte a jelszó visszaállítását, kérjük, hagyja figyelmen kívül ezt az üzenetet.</p>";  
  
 if (!$mail->send()) {  
 error\_log("Mailer Error: " . $mail->ErrorInfo);  
 header("Location: ../request\_reset.php?message=" . urlencode($mail->ErrorInfo) . "&type=alert");  
 } else {  
 header("Location: ../request\_reset.php?message=" . urlencode("Ellenőrizze az email fiókját a jelszó visszaállításához.") . "&type=success");  
 }  
 } else {  
 header("Location: ../request\_reset.php?message=" . urlencode("Az email cím nem található.") . "&type=alert");  
 }  
 exit();  
} else {  
 header("Location: ../index.php?message=" . urlencode("Hibás kérés.") . "&type=alert");  
 exit();  
}

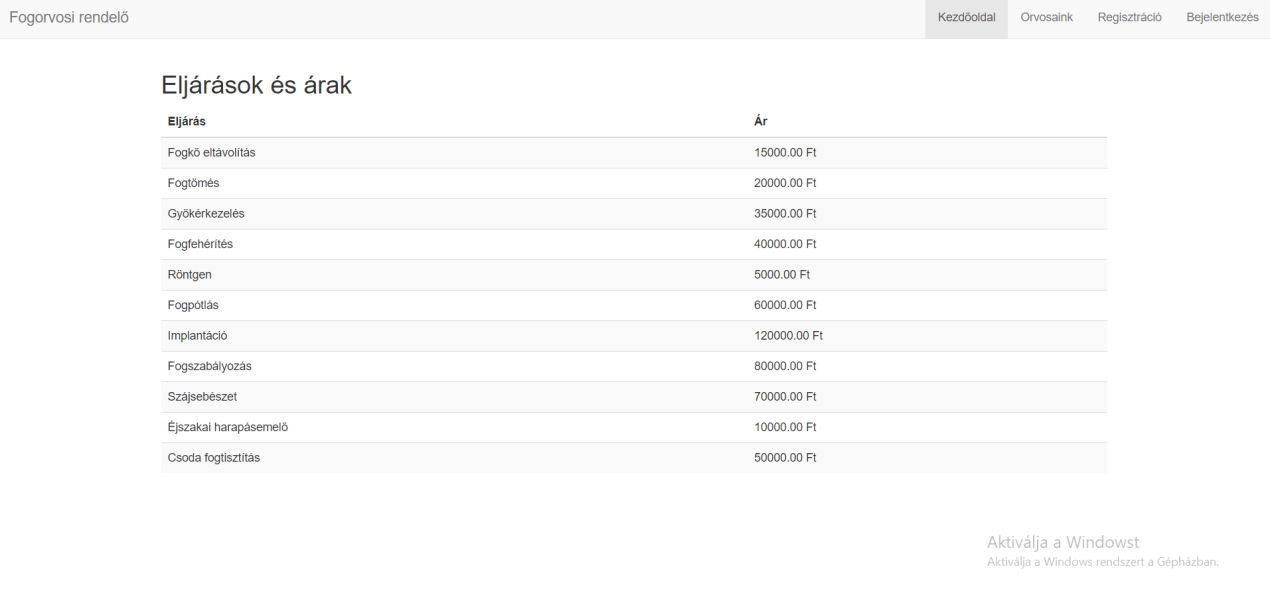
Megerősítő e-mail:

<?php  
  
use PHPMailer\PHPMailer\Exception;  
use PHPMailer\PHPMailer\PHPMailer;  
  
require '../vendor/autoload.php';  
  
function send\_verification\_email($email, $token)  
{  
 $mail = new PHPMailer(true);  
  
 try {  
 // Server settings  
 $mail->isSMTP();  
 $mail->Host = 'in-v3.mailjet.com';  
 $mail->SMTPAuth = true;  
 $mail->Username = '81ea24ce778b6e7ecc44af9aaaca1da3';  
 $mail->Password = '418bade66c7e26bbc9fb672efadd6512';  
 $mail->SMTPSecure = PHPMailer::*ENCRYPTION\_STARTTLS*;  
 $mail->Port = 587;  
 $mail->CharSet = 'UTF-8';  
  
 $mail->SMTPOptions = array(  
 'ssl' => array(  
 'verify\_peer' => false,  
 'verify\_peer\_name' => false,  
 'allow\_self\_signed' => true  
 )  
 );  
  
 $mail->setFrom('balogbalesz1234@gmail.com', 'Balint');  
 $mail->addAddress($email);  
  
 // Content  
 $mail->isHTML(true);  
 $mail->Subject = 'Email megerősítés';  
 $verificationLink = "https://lite.stud.vts.su.ac.rs/verify.php?token=$token";  
 $mail->Body = "Kérjük, erősítse meg az email címét az alábbi linkre kattintva: <a href='$verificationLink'>Email megerősítése</a>";  
  
 $mail->send();  
 return true;  
 } catch (Exception $e) {  
 error\_log("Mailer Error: " . $mail->ErrorInfo);  
 return false;  
 }  
}  
  
?>

# Az adatbázis struktúrája (MySQL Workbench)

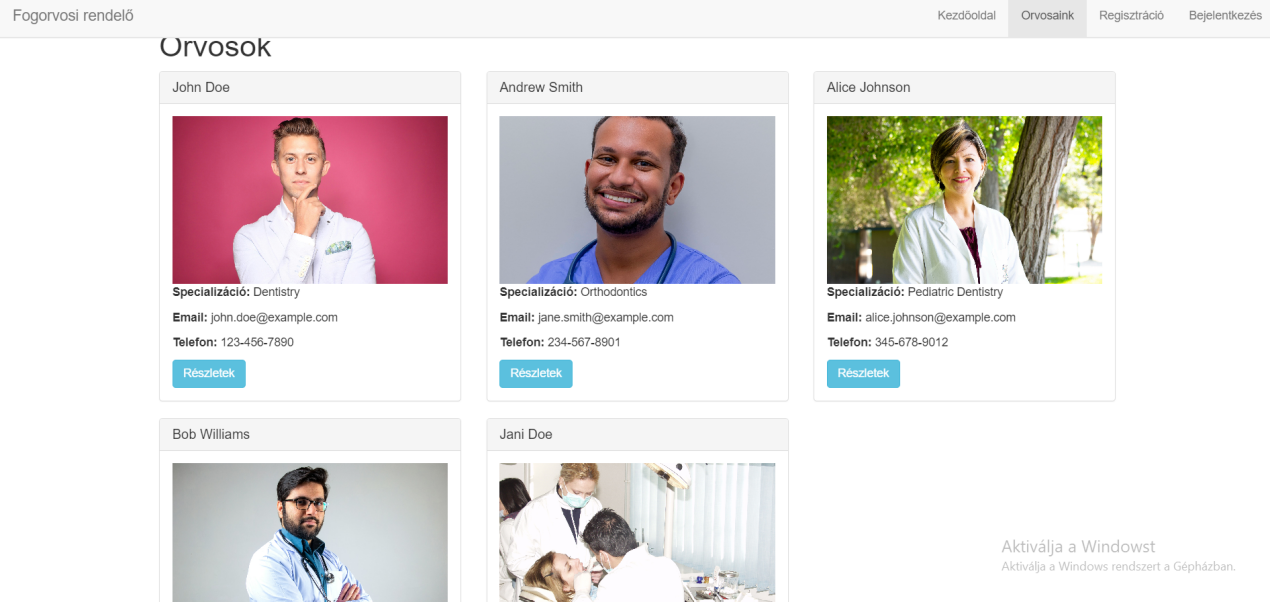


# Projekt működésének leírása

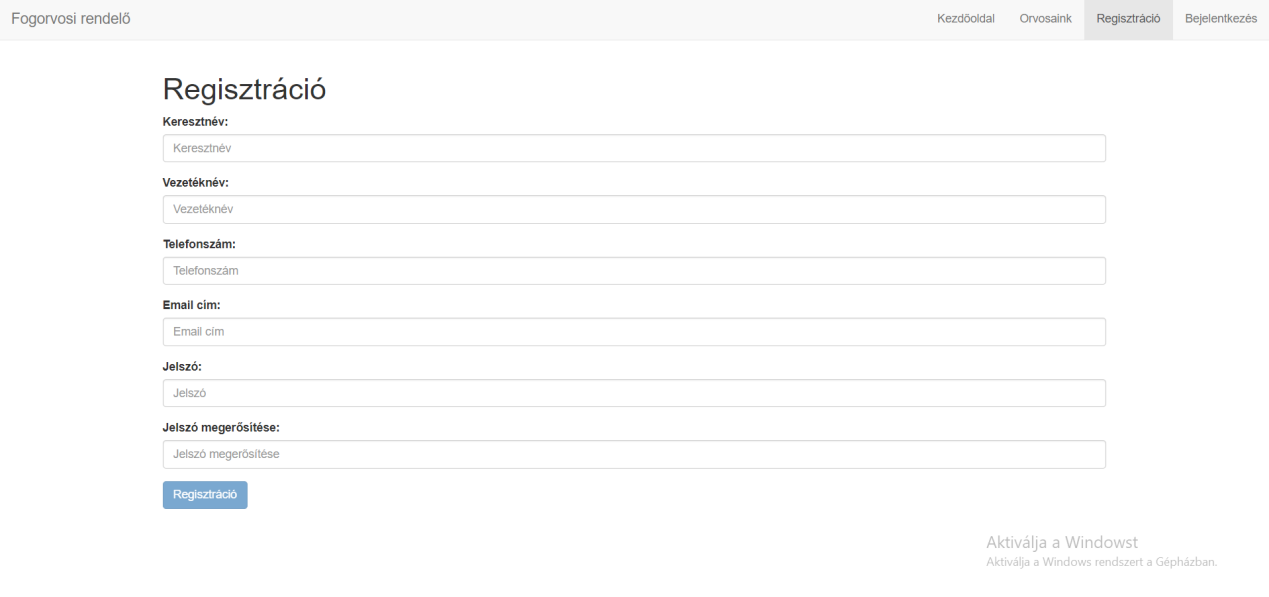


Amikor az oldalra fellép egy látogató ez várja megbírja nézni az orvosokat az árakat illetve bebír jelentkezni vagy bír regisztrálni.

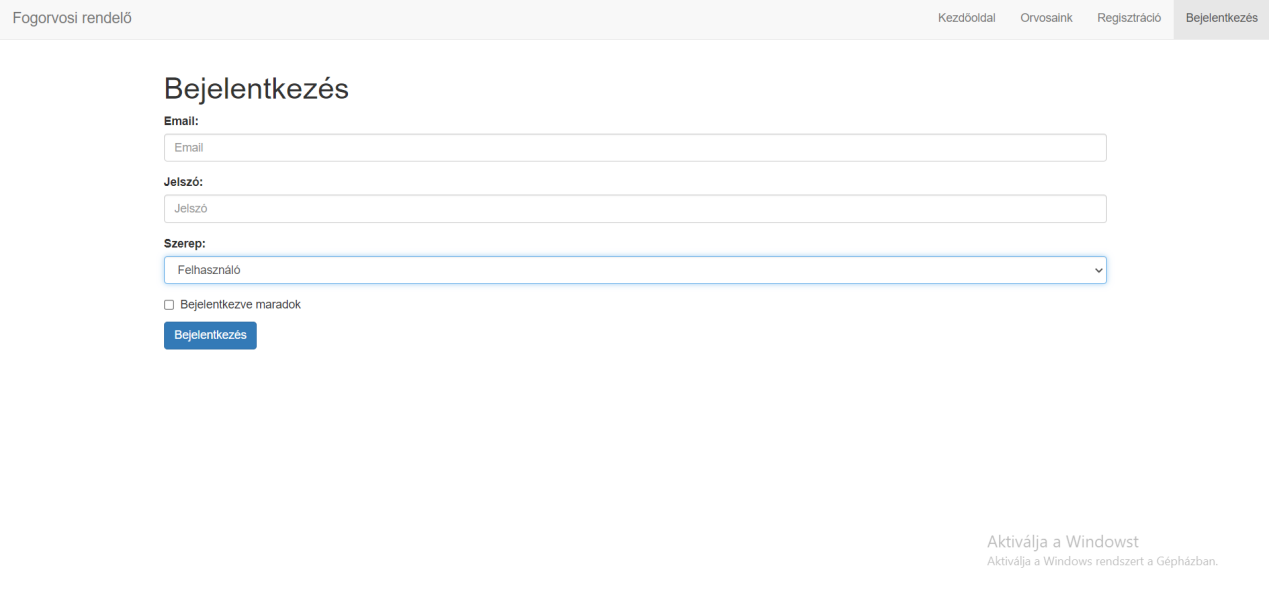
Orvosaink oldal:



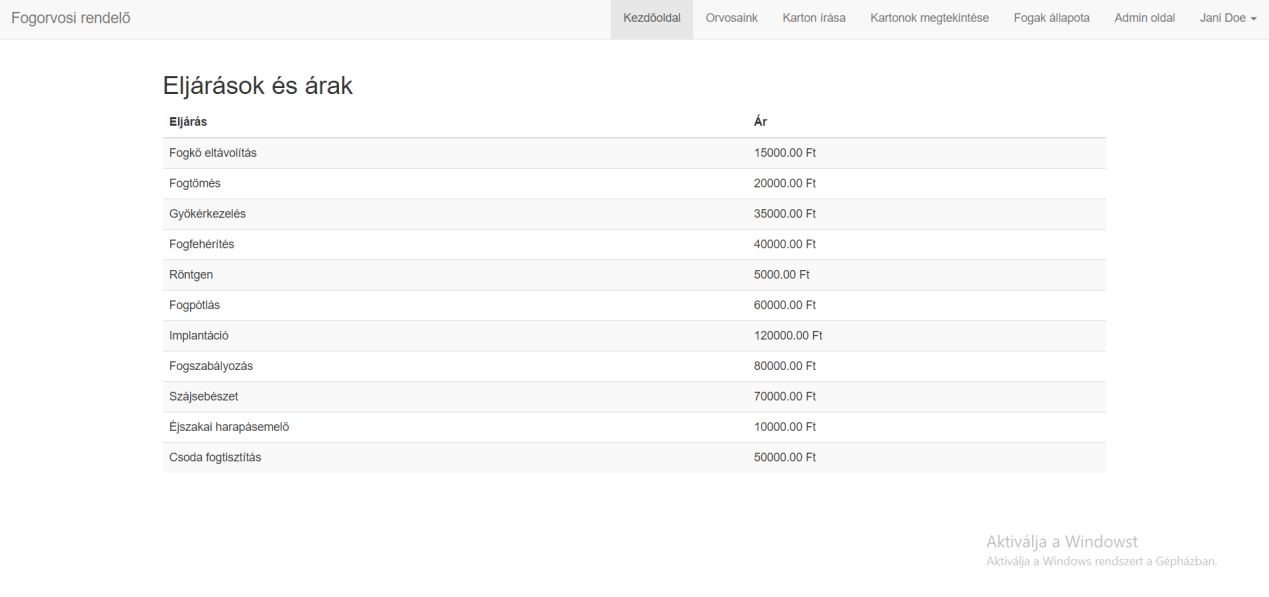
Regisztráció oldal:



Bejelentkezés oldal, ahol kibírjuk választani hogy páciens vagy orvos ként szeretnénk bejelentkezni:

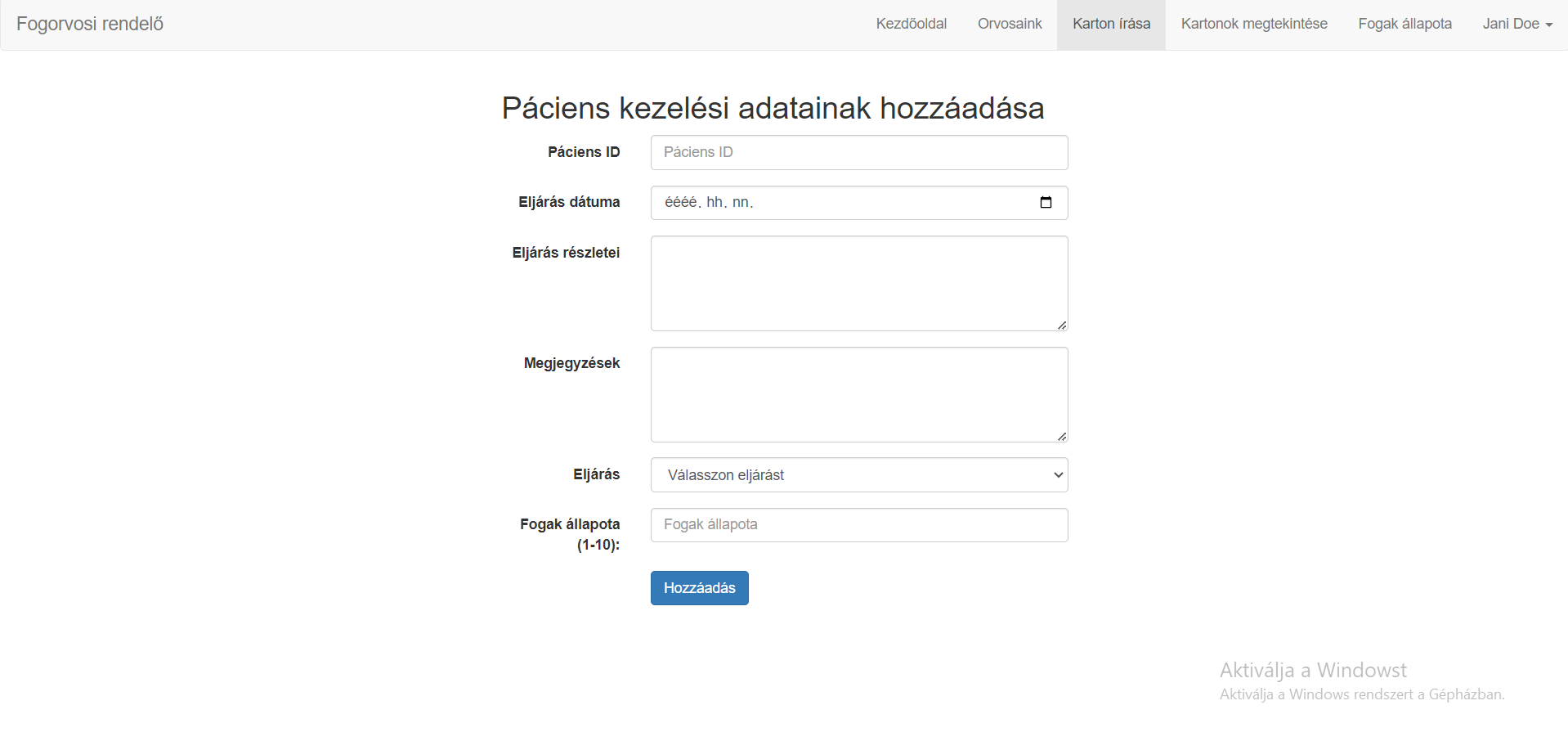


Amennyiben orvos ként jelentkezünk be ezeket lássuk:

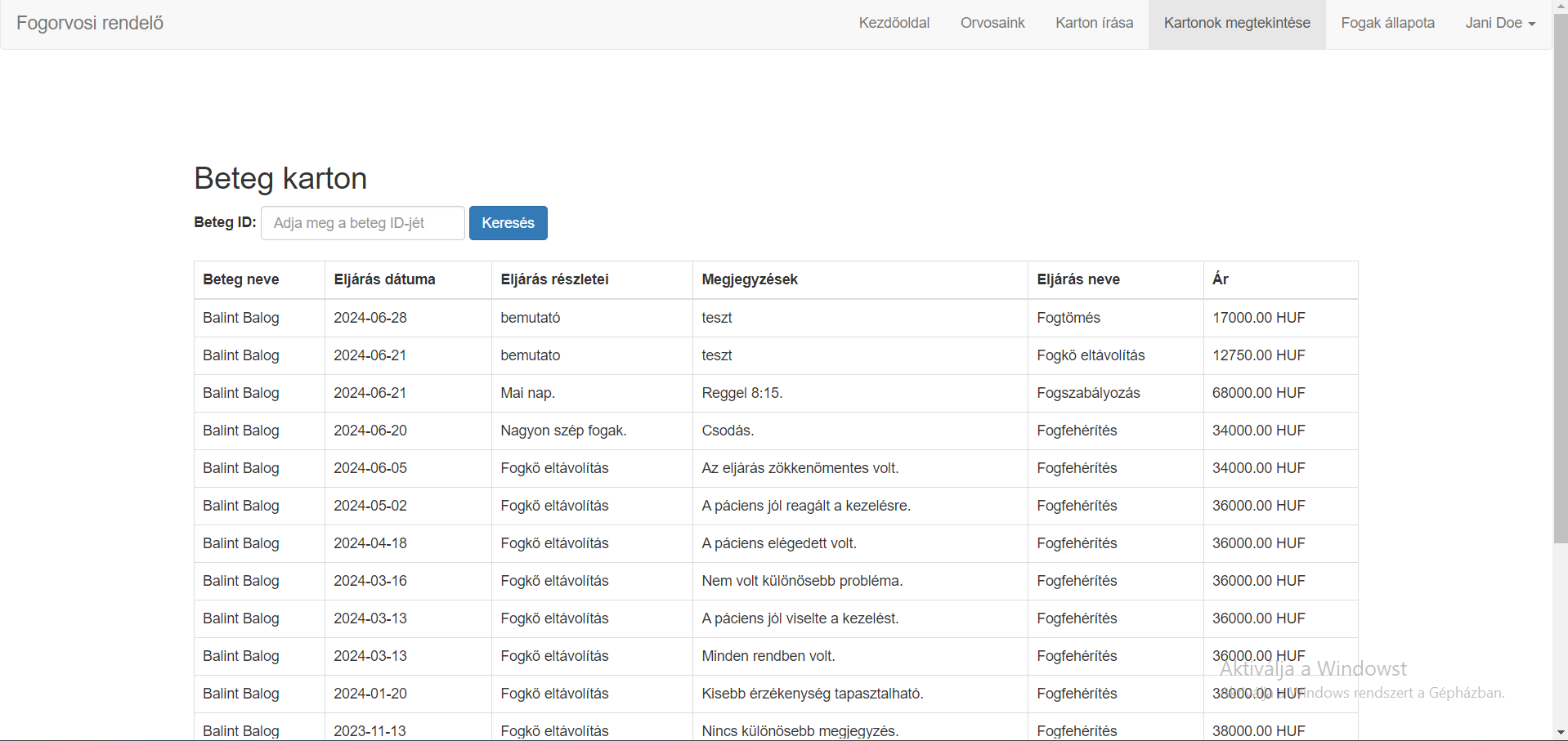


Úgyszint lássuk az orvosaink és az kezdő oldalt, de ezek mellé van még 4 új gomb.

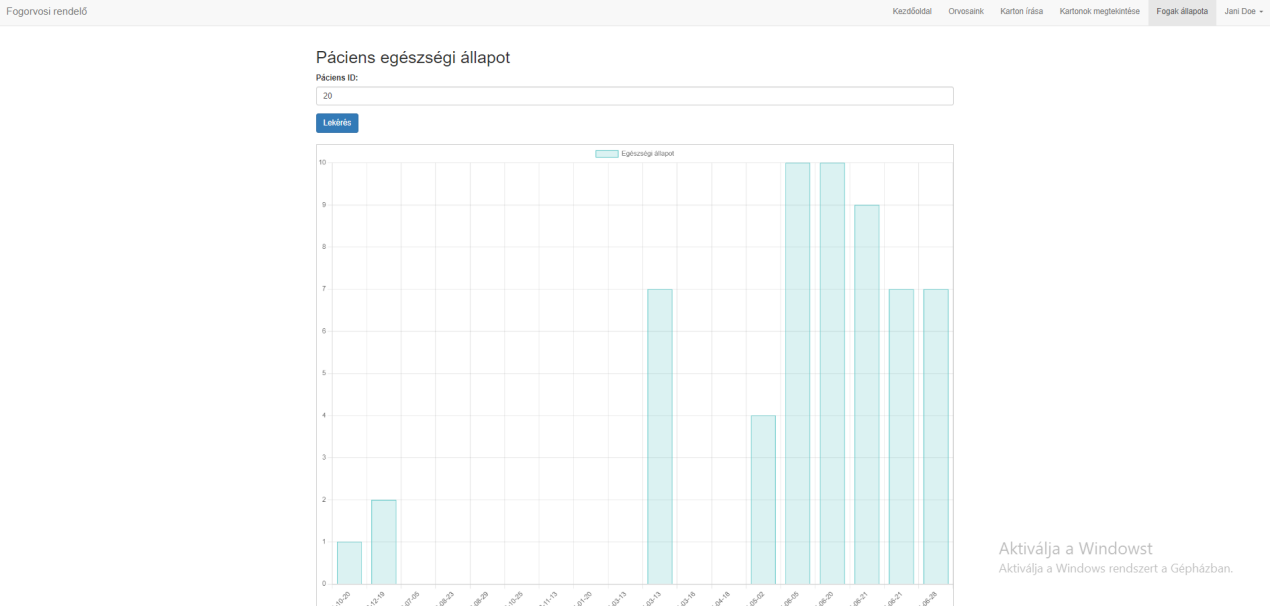
Karton írása a páciensnek írhatunk be adatokat:



Kartonok megtekintése:

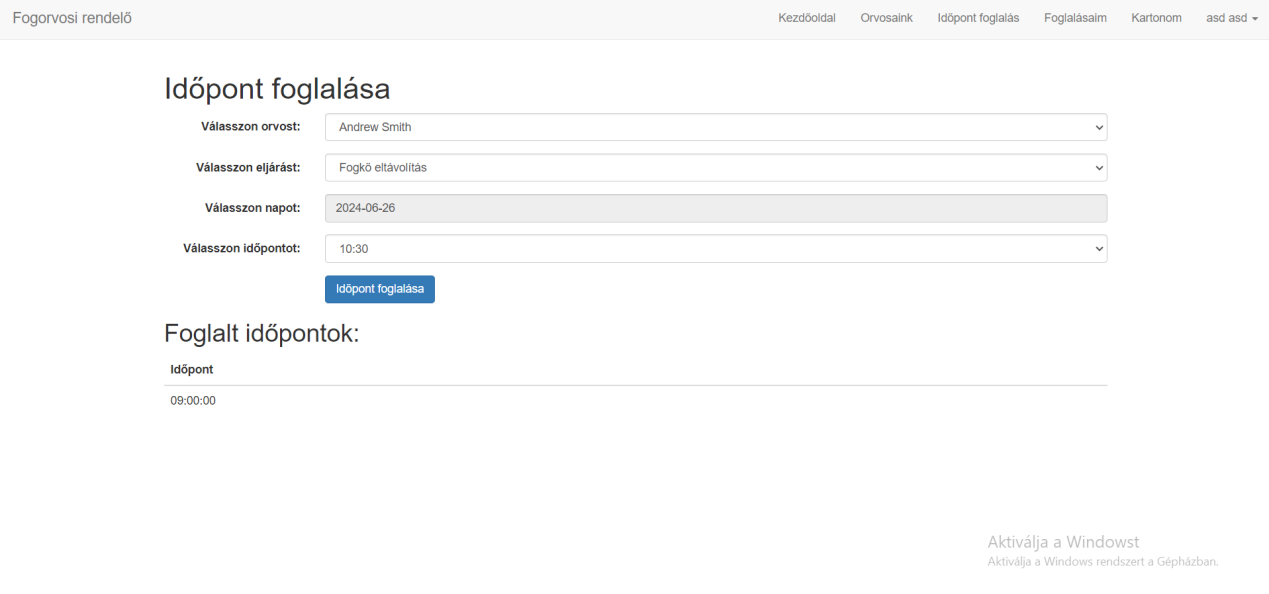


Fogak állapota itt egy charton megmutassa milyenek a fogak állapota az idő telésével:

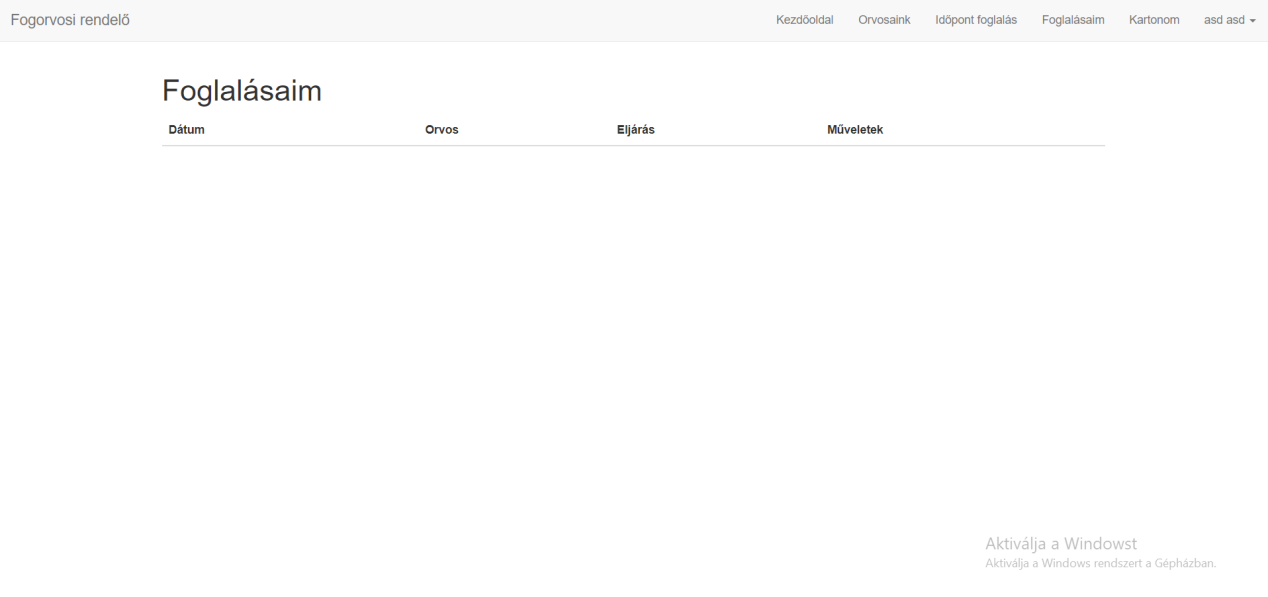


Ha páciens ként jelentkezünk be úgyszint lássuk az orvosaink fület és az árakat csak van még három új fül:

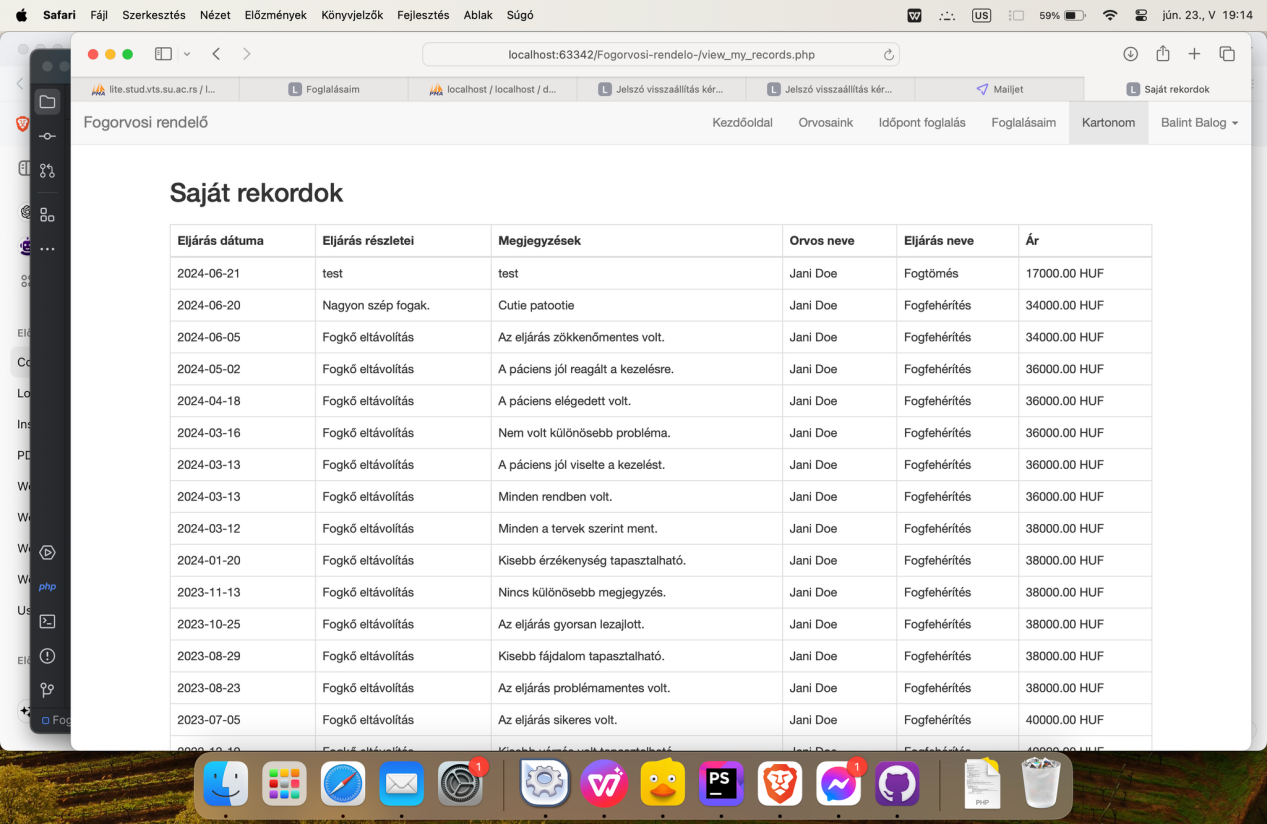
Időpont foglalás:



Foglalásaim, itt láthatóak a már lefoglalt időpontok:



Kartonom:

****

# Használt irodalmak:

1. https://www.php.net/
2. https://www.tutorialspoint.com/javascript/javascript\_comments.htm
3. https://www.w3schools.com/