**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 5

la cursul ***„Tehnologii web”***

***Tema: Creare de servicii API/REST/altele pentru aplicația Web***

**A efectuat :**   **St. gr. CR-221FR Serba Cristina**

**A verificat: Rusu Viorel**

**Chișinău 2024**

# Sarcina lucrării:

# a) Adaptati careva pagini a site-ului creat in laboratoarele precedente astfle incit in loc sa fie generate pe partea server, sa utilizeze Ajax pentru transmiterea datelor prin API-uri, iar cu JS incarcati continutul in locul necesar din pagina;

# b) Creati API-uri REST cel putin pentru metodele GET/POST/PUT/DELETE si utilizati-le pentru sarcina descrisa in a).

# c) Puneti in raport secvente de cod Ajax/JS si exemple de API-uri create.

# Mersul lucrării:

Funcția care utilizează metoda GET

|  |
| --- |
| function fetchUserData() {          fetch('api/getUserData.php')          .then(response => response.json())          .then(data => {              // Daca exista user logat, arata continut corespunzator              if (data.username) {                  document.getElementById('welcome-message').textContent = "Bine ai venit, " + data.username;                  document.getElementById('welcome-text').textContent = "Descoperă oportunitățile de a face parte din comunitatea noastră și de a te implica în noi proiecte.";                  document.getElementById('welcome-cta').textContent = "";                  document.getElementById('registration-button').style.display = "none"; // ascunde butonul de inregistrare                    // Necesar pentru API DELETE; daca utilizatorul este logat, logic va fi sa existe buton deleteAccount                  var deleteAccountBtn = document.getElementById('deleteAccountBtn');                  if (deleteAccountBtn) { //adauga listener pentru cand va fi apasat                      deleteAccountBtn.addEventListener('click', function() {                          if (confirm('Are you sure you want to delete your account? This action cannot be undone.')) {                              deleteAccount();                          }                      });              }              } else { // cand nu este utilizator logat                  document.getElementById('welcome-message').textContent = "Bine ați venit pe Skill";                  document.getElementById('welcome-text').textContent = "Găsește freelanceri talentați sau ia pasul să începi un nou proiect.";                  document.getElementById('welcome-cta').textContent = "Înscrie-te acum pentru a accesa cele mai bune servicii de dezvoltare web și design grafic.";                  document.getElementById('registration-button').style.display = "block"; // Arata butonul de inregistrare              }          })          .catch(error => {              console.error(Eroare in timpul primirii datelor  :', error);          });      } |

Respectiv răspunsul la metoda GET:

|  |
| --- |
| <?php  // Fallback cand nu este sesiune  if (session\_status() === PHP\_SESSION\_NONE) {      session\_start();  }  //permite doar daca este cineva logat  if (!isset($\_SESSION['username'])) {      http\_response\_code(401); // Unauthorized      exit("Utilizator nelogat");  }  if ($\_SERVER['REQUEST\_METHOD'] !== 'PUT') {      http\_response\_code(405); // Method Not Allowed      exit("Doar metoda PUT");  }  $inputJSON = file\_get\_contents('php://input');  $inputData = json\_decode($inputJSON, true);  //din json parseaza informatia necesara  $name = $inputData['name'] ?? '';  $email = $inputData['email'] ?? '';  $message = $inputData['message'] ?? '';  // Se va conecta la database  $conn = new mysqli("localhost", "root", "", "skill\_database");  if ($conn->connect\_error) {      http\_response\_code(500);      exit("Eroare la conectarea la baza de date: " . $conn->connect\_error);  }  $updateQuery = "UPDATE contacts SET name=?, email=? WHERE id=?";  $stmt = $conn->prepare($updateQuery);  if (!$stmt) {      http\_response\_code(500);      exit("Parametrizare esuata: " . $conn->error);  }  $stmt->bind\_param("ssi", $name, $email, $id);  $id = 1;  if (!$stmt->execute()) {      http\_response\_code(500); // Internal Server Error      exit("Executia parametrizarii esuata: " . $stmt->error);  }  http\_response\_code(200); // Fallback  exit("Mesajul a fost trimis");  ?> |

Funcția pentru metoda POST este din laboratorul trecut (login si register process)   
Request-ul vine direct de pe pagina login.php

|  |
| --- |
| <h2>Înregistrare</h2>          <form action="register\_process.php" method="post">            <div class="form-group">              <label for="registerEmail">Email</label>              <input type="email" class="form-control" id="registerEmail" name="registerEmail" required>            </div>            <div class="form-group">              <label for="username">Nume de utilizator</label>              <input type="text" class="form-control" id="username" name="username" required>            </div>            <div class="form-group">              <label for="registerPassword">Parolă</label>              <input type="password" class="form-control" id="registerPassword" name="registerPassword" required>            </div>            <button type="submit" class="btn btn-primary">Înregistrare</button>          </form> |

Iar răspunsul e primit din register\_process.php

|  |
| --- |
| <?php  if (session\_status() === PHP\_SESSION\_NONE)  {    session\_start();  }  if ($\_SERVER["REQUEST\_METHOD"] == "POST")  {      // Conectare la baza de date      $conn = new mysqli("localhost", "root", "", "skill\_database");      if ($conn->connect\_error)      {          // ------------- LOG INFO -------------          $log\_entry = date('Y-m-d H:i:s') . " - " . "Conexiune nereusita\n";          file\_put\_contents('log.txt', $log\_entry, FILE\_APPEND | LOCK\_EX);          $\_SESSION["registration\_message"] = "Error: Conexiune nereusita";          die("Connection failed: " . $conn->connect\_error);      }        // Prelucrare date din formular      $username = $\_POST["username"];      $email = $\_POST["registerEmail"];      $password = $\_POST["registerPassword"];      // Verificare dacă utilizatorul există deja      $check\_query = "SELECT \* FROM users WHERE username='$username' OR email='$email'";      $result = $conn->query($check\_query);      if ($result->num\_rows > 0)      {          // ------------- LOG INFO -------------          $error\_message = "Username sau email deja folosit!";          $log\_entry = date('Y-m-d H:i:s') . " - " . $error\_message . "\n";          file\_put\_contents('log.txt', $log\_entry, FILE\_APPEND | LOCK\_EX);          $\_SESSION["registration\_message"] = "Error: " . $error\_message;      }      else      {          // Adăugare utilizator nou în baza de date          $insert\_query = "INSERT INTO users (username, email, password) VALUES ('$username', '$email', '$password')";            if ($conn->query($insert\_query) === TRUE)          {              // Acordare drepturi utilizatorului nou creat              $grant\_query = "GRANT SELECT, INSERT, UPDATE, DELETE ON skill\_database.\* TO '$username'@'localhost'";              if ($conn->query($grant\_query) === TRUE)              {                  // ------------- LOG INFO -------------                  $success\_message = "Utilizator creat cu succes. Urmați procedura de logare.";                  $log\_entry = date('Y-m-d H:i:s') . " - " .  $success\_message . "\n";                  file\_put\_contents('log.txt', $log\_entry, FILE\_APPEND | LOCK\_EX);                  $\_SESSION["registration\_message"] = $success\_message;              } else              {                  // ------------- LOG INFO -------------                  $error\_message = "Eroare la acordarea drepturilor utilizatorului: " . $conn->error;                  $log\_entry = date('Y-m-d H:i:s') . " - " . $error\_message . "\n";                  file\_put\_contents('log.txt', $log\_entry, FILE\_APPEND | LOCK\_EX);                  $\_SESSION["registration\_message"] = "Error: " . $error\_message;              }          } else          {              // ------------- LOG INFO -------------              $error\_message = "Eroare la crearea utilizatorului: " . $conn->error;              $log\_entry = date('Y-m-d H:i:s') . " - " . $error\_message . "\n";              file\_put\_contents('log.txt', $log\_entry, FILE\_APPEND | LOCK\_EX);              $\_SESSION["registration\_message"] = "Error: " . $error\_message;          }      }      header("Location: login.php"); // Urmatorul pas va fi logarea      $conn->close();  }  ?> |

Funcție pentru metoda PUT:

|  |
| --- |
| function updateContactInfo() {      var updatedContactInfo = {          name: document.getElementById('name').value,          email: document.getElementById('email').value,          message: document.getElementById('message').value      };      fetch('api/updateContactInfo.php', {          method: 'PUT',          headers: {              'Content-Type': 'application/json'          },          body: JSON.stringify(updatedContactInfo)      })      .then(response => {          if (response.ok) {              alert('Mesajul a fost trimis cu succes');          } else {              alert('Mesajul nu a fost trimis');          }      })      .catch(error => console.error('Eroare in timpul trimiterii mesajului:', error));  } |

Respectiv răspunsul

|  |
| --- |
| <?php  // Fallback cand nu este sesiune  if (session\_status() === PHP\_SESSION\_NONE) {      session\_start();  }  //permite doar daca este cineva logat  if (!isset($\_SESSION['username'])) {      http\_response\_code(401); // Unauthorized      exit("Utilizator nelogat");  }  if ($\_SERVER['REQUEST\_METHOD'] !== 'PUT') {      http\_response\_code(405); // Method Not Allowed      exit("Doar metoda PUT");  }  $inputJSON = file\_get\_contents('php://input');  $inputData = json\_decode($inputJSON, true);  //din json parseaza informatia necesara  $name = $inputData['name'] ?? '';  $email = $inputData['email'] ?? '';  $message = $inputData['message'] ?? '';  // Se va conecta la database  $conn = new mysqli("localhost", "root", "", "skill\_database");  if ($conn->connect\_error) {      http\_response\_code(500);      exit("Eroare la conectarea la baza de date: " . $conn->connect\_error);  }  $updateQuery = "UPDATE contacts SET name=?, email=? WHERE id=?";  $stmt = $conn->prepare($updateQuery);  if (!$stmt) {      http\_response\_code(500);      exit("Parametrizare esuata: " . $conn->error);  }  $stmt->bind\_param("ssi", $name, $email, $id);  $id = 1;  if (!$stmt->execute()) {      http\_response\_code(500); // Internal Server Error      exit("Executia parametrizarii esuata: " . $stmt->error);  }  http\_response\_code(200); // Fallback  exit("Mesajul a fost trimis");  ?> |

Funcție pentru metoda DELETE:

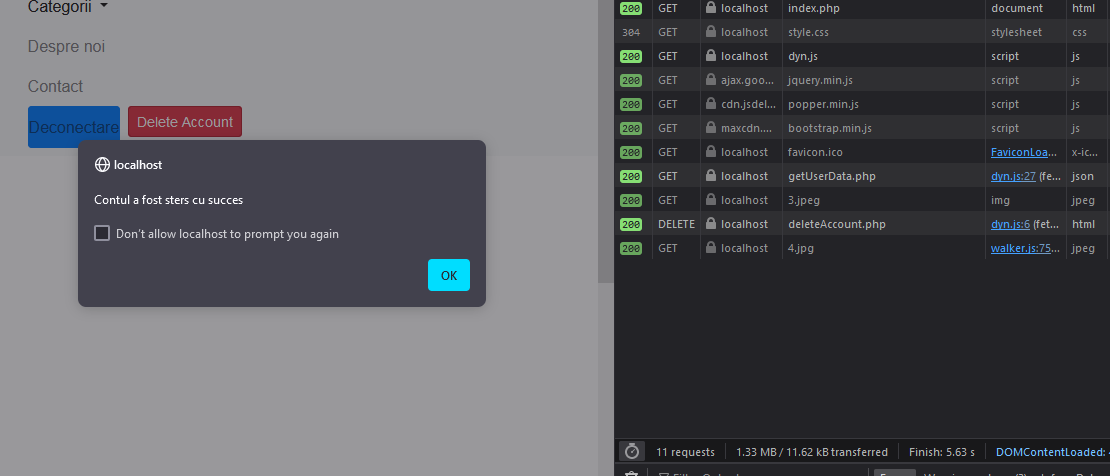
|  |
| --- |
| function deleteAccount() {          fetch('api/deleteAccount.php', {              method: 'DELETE',          })          .then(response => {              if (response.ok) {                  alert(Contul a fost sters cu succes');                  window.location.href = 'logout.php';              } else {                  alert('Nu s-a putut sterge contul');              }          })          .catch(error => {              console.error('Eroare la stergerea contului:', error);              alert('Nu s-a putut sterge contul');          });      } |

Răspunsul:

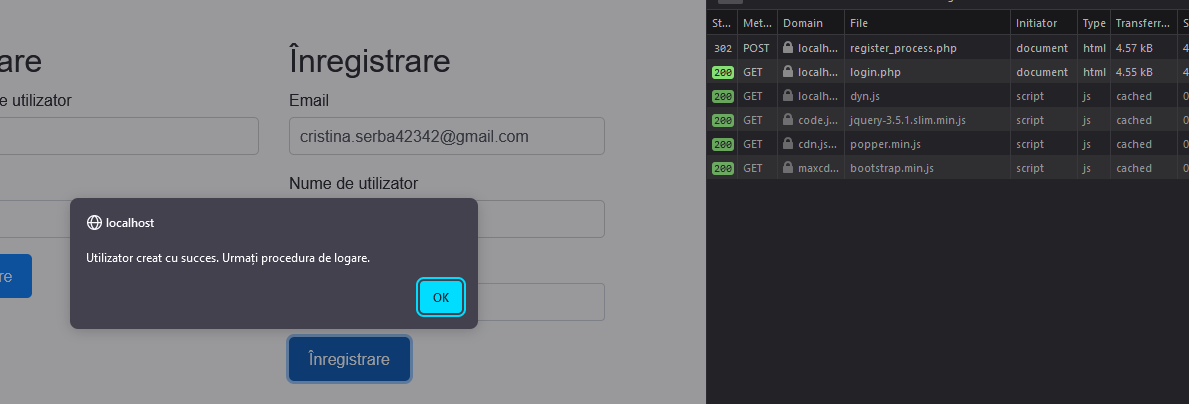
|  |
| --- |
| <?php  //Fallback daca nu este sesiune  if (session\_status() === PHP\_SESSION\_NONE) {      session\_start();  }  // Poate sterge accountul doar daca este cineva logat in primul rand  if (!isset($\_SESSION['username'])) {      http\_response\_code(401); // Unauthorized      exit("Utilizator nelogat");  }  //Creaza sesiune noua  $conn = new mysqli("localhost", "root", "", "skill\_database");  if ($conn->connect\_error) {      http\_response\_code(500); // Internal Server Error      exit("Conexiune esuata: " . $conn->connect\_error);  }  $username = $\_SESSION['username'];  $sql = "DELETE FROM users WHERE username = '$username'";  if ($conn->query($sql) === TRUE) {      http\_response\_code(200); // OK      exit("Contul a fost sters cu success");  } else {      http\_response\_code(500); // Internal Server Error      exit("Eroare in stergerea contului: " . $conn->error);  }  $conn->close();  ?> |

***Exemplu de pagini:***

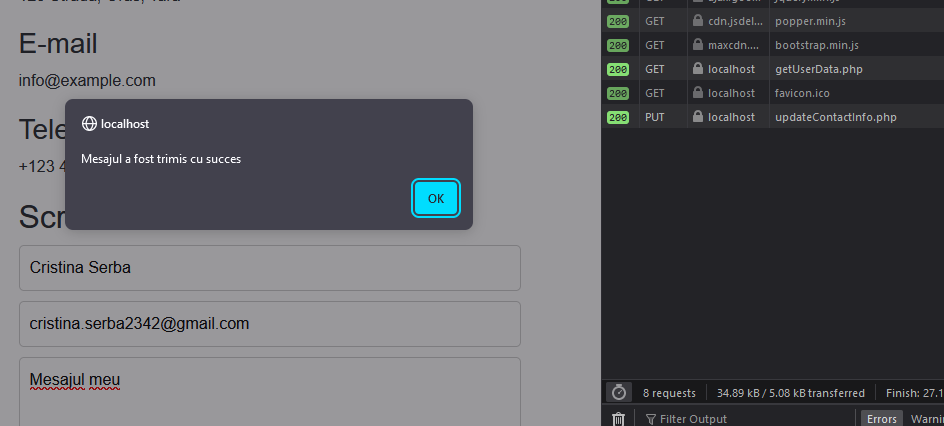
Metoda DELETE:



Metoda GET si POST



Metoda PUT:



# Concluzii:

În concluzie, utilizarea metodelor HTTP precum GET, POST, PUT și DELETE a facilitat procesul de comunicare dintre website și server. Fiecare metodă servește un scop distinct și oferă avantaje unice, contribuind la funcționalitatea generală și la experiența utilizatorului.

De exemplu, metoda GET a fost esențială în preluarea datelor esențiale, cum ar fi informațiile utilizatorului sesiunii curente, în mod dinamic. Acest lucru permite actualizări în timp real și prezentare personalizată a conținutului.

În mod similar, metoda POST a permis transmiterea în siguranță a datelor utilizatorului, în cazul dat trimiterea formularelor de înregistrare, asigurând integritatea și confidențialitatea informațiilor sensibile.

Metoda PUT a fost utilizată pentru a trimite informația de contact în cazul în care utilizatorul va da feedback.

În plus, metoda DELETE oferă capacitatea de a șterge, la dorință, contul de utilizator. Această funcționalitate promovează autonomia utilizatorului și eficientizează procesele de gestionare a datelor.

În final, aceste metode au înbunătațit dinamic website-ul dezvoltat, contribuind la funcționalitatea generală a acestuia.