**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Informatică și Ingineria Sistemelor**

**RAPORT**

Lucrare de laborator nr. 3

la cursul „***Baze de date***”

A efectuat: **st. gr. MI-221, Murafa Dan**

A verificat: **conf. univ., dr. M. Perebinos**

**Chișinău 2023**

**Cuprins**

SARCINA LUCRĂRII............................................................................................................3

FABULA MODIFICATĂ.......................................................................................................4

PROIECTAREA INPUT -> OUTPUT....................................................................................7

MODELUL LOGIC AL FABULEI MODIFICATE...............................................................9

SCHEMA ER A BAZEI DE DATE.......................................................................................11

REALIZAREA SARCINII DE LUCRU................................................................................12

CONCLUZII...........................................................................................................................36

**Sarcina lucrării**

1. **Lucrarea de laborator se va realiza conform următorilor pași:**
2. Textul fabulei modificat ( a prezenta textul modificat utilizând culoarea albastră);
3. Modelul conceptual al fabulei modificate utilizând abordarea I-O;
4. Modelul logic al fabulei modificate;
5. Schema ER a BD din Designer (PhpMyAdmin ori echivalent lui);
6. Bloc-schema aplicației;
7. Prezentarea elementelor CRUD utilizate și realizate în aplicația WEB, în conformitate cu schema din laborator, pentru fiecare componentă;
8. Descrierea pentru fiecare componentă a tuturor funcțiilor utilizate, variabilelor de sistem, elementelor de control și structurilor repetitive în PHP, vectori GET și POST ( dacă se va considera necesar, se va prezenta sintaxa și mici exemple pentru confirmare);
9. Codul aplicației WEB împreună cu BD exportată.

**Fabulă modificată**

Au fost odată doi fraţi, un băiat şi o fetiţă, pe nume Hănsel şi Gretel, care locuiau într-o căscioară, la marginea unui codru nesfârşit. Tatăl lor era tăietor de lemne, iar mama vitregă nu-i iubea deloc.  
           Într-o zi, se treziră cu toţii că nu mai era nimic de mâncare prin casă şi nici bani. Maştera se gândi că nu mai avea încotro şi trebuia să scape degrabă de copii, două guri în plus la masă.  
          Aşa stând lucrurile, femeia îi duse pe Hănsel şi pe Gretel în adâncul pădurii.  
Le lăsă două coltuce de pâine şi le spuse că va veni mai târziu după ei, ca să-i ia acasă.

Pe tot parcursul drumului copii aruncau farîmături pentru a putea gasi drumul inapoi , însă niste pasarele ciguleau firimiturile dupa căruță.

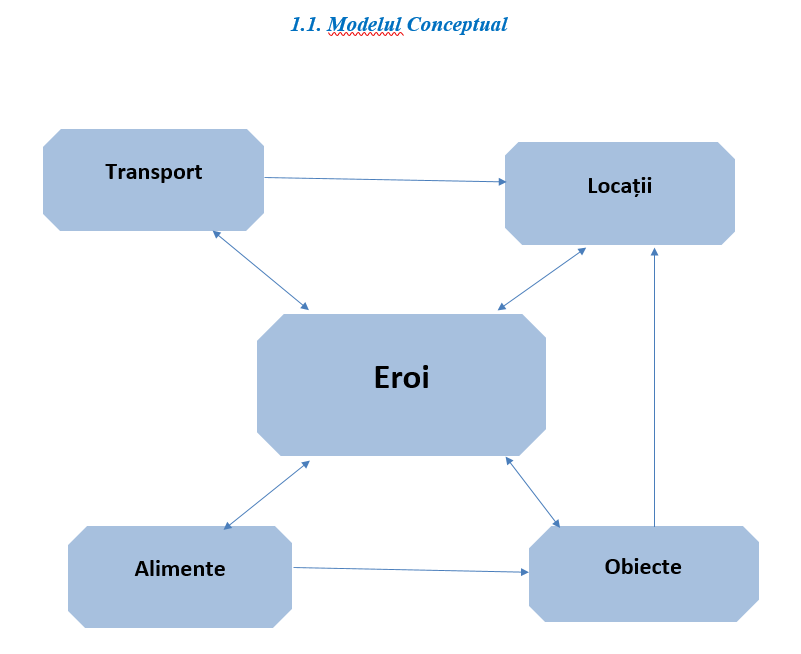
Umbrele înserării se lăsară peste pădure şi copiii aşteptară în zadar, căci nici vorbă ca maştera să se întoarcă să-i ia. Neştiind ce să facă, cei doi fraţi dormiră în noaptea aceea sub un copac.  
          De cum răsări soarele, copilaşii o luară la picior şi căutară drumul spre casă. Dar, oricât încercară, nu izbutiră să-l afle. Ba mai mult, cu cât mergeau, cu atât se afundau şi mai adânc în pădurea întunecoasă.

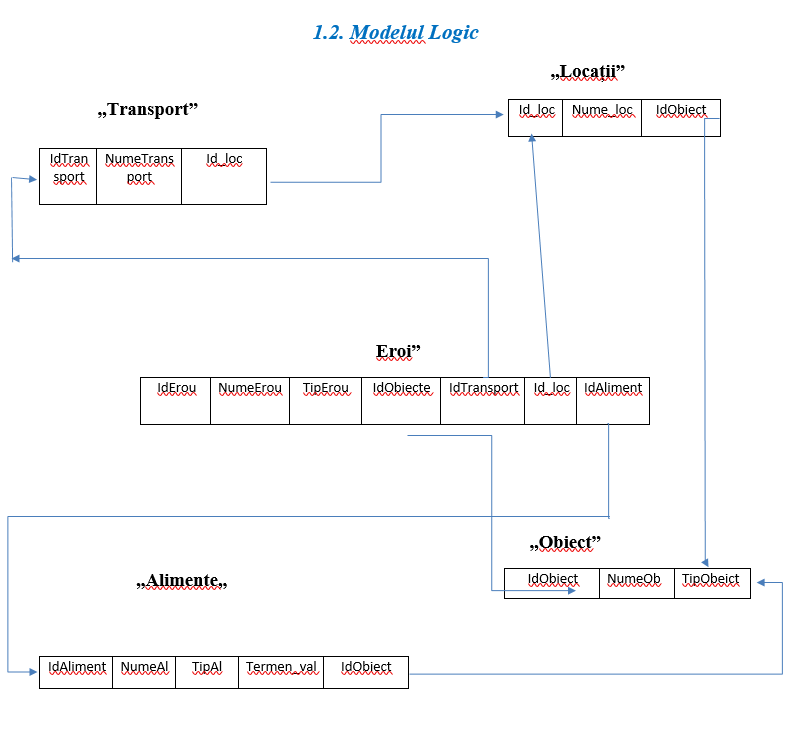
Deodata pasarelele de ziua trecută drept mulțumire copiilor pentru firmituri le-au adus un telefon mobil. La inceput copiii nu stiau ce sa faca cu el deoarece tatal nu avea telefon mobil si nu puteau sa vorbească cu el. După ce umblară toată ziua, Hnsel şi Gretel simţiră că îi lasă puterile. Dar tocmai atunci, într-o poiană, dădură peste o căsuţă cum nu mai văzuseră vreodată: era făcută din turtă dulce, biscuiţi şi bombonele, iar acoperişul, din felii mari de cozonac şi halviţă alaturi se afla un râuleț din lapte dulce topit!  
          Cu ultimele puteri, copiii se repeziră într-acolo şi se puseră pe înfulecat, rupând bucăţi mari şi gustoase din acoperiş , hogeac , ușă pereți. Nu trecu mult însă şi, din căsuţă, se auzi o voce hodorogită:  
          — Cine ronţăie din căsuţa mea?  
Speriaţi, copiii încercară să scape cu fuga,vazand ca au telefon repede au sunat la 112 dar stăpâna căsuţei, care era o vrăjitoare, îi prinse din urmă.  
          Îl închise pe Hansel într-o odaie și-l ameninţă că o să-l gătească şi apoi o să-l mănânce. Tocmai când vrăjitoarea pregătea focul, se aude afara o masina de poliție . 2 polițai au stricat ușa și au intrat in casa vazand totul ce sa intamplat , vrajitoarea a fost impuscata si omorata de pistol. După ce găsiră şi comoara vrăjitoarei, copiii au fost premiati drept recomompensă cu o casă nouă in satul central . Pe drum spre sediul politiei îl întâlniră pe tatăl lor, care plecase de ceva vreme să-i caute. Tatăl îi îmbrăţişă bucuros şi îi duse la casa noua. Supărat foc pentru ceea ce le făcuse copiilor, tăietorul de lemne o izgoni pe maşteră.  
          Şi de-atunci au trăit toţi trei fericiti, fără să mai fi suferit vreodată de foame în casa nouă

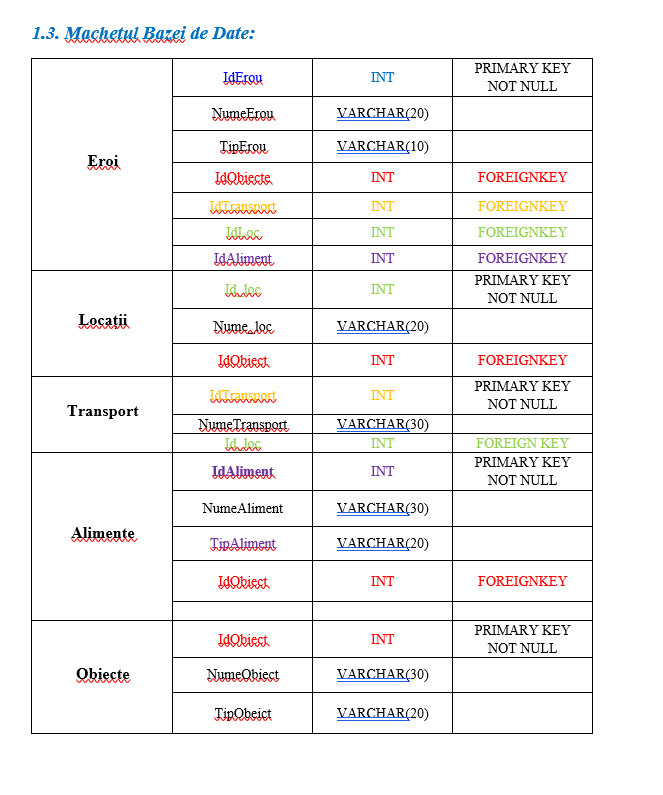
**Proiectarea Input -> Output**

1. **Eroi:** „Hensel ”, „Grettel”, „Tatal”, „Mama Vitrega”, „Vrajitoarea”, „Politia”,”Pasarile”;
2. **Locatii:** „cascioara langa padure ”, „casuta din turta dulce”, „drumul rpin padure ”;, „sediul de politie”;
3. **Transport:** „Caruta”, „Masina de politie”
4. **Obiecte:** „Telefon”, „Pistol”, „usa” „hogeac”, „pereti” „acoperis”,,;
5. **Alimente:** „2 coltuce de paine”, „turta dulce”, „biscuiti”, „cozonac”, „halvita” „lapte topit”,;

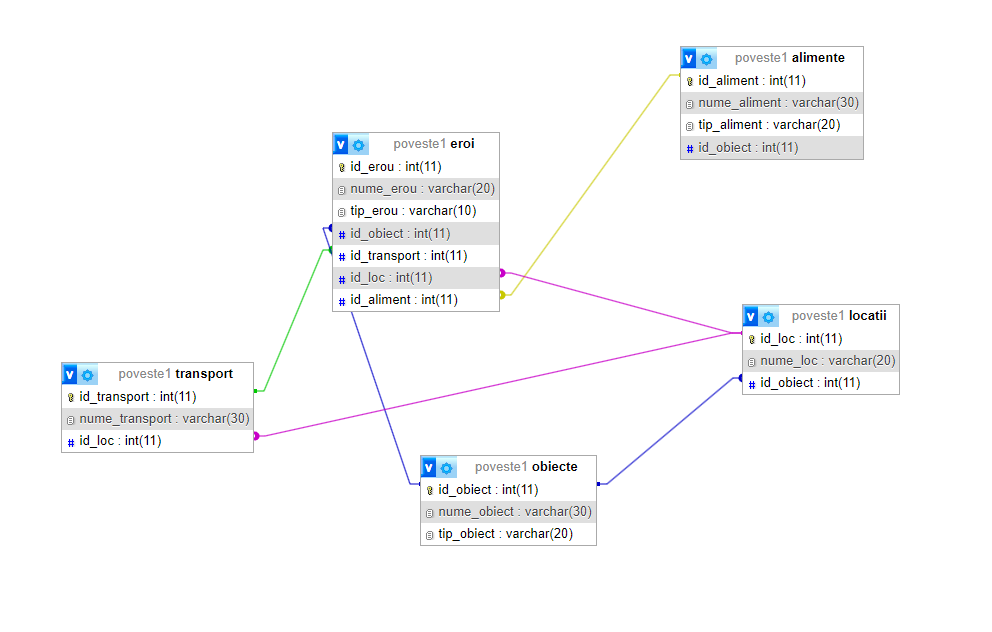
**Modelul conceptual:**





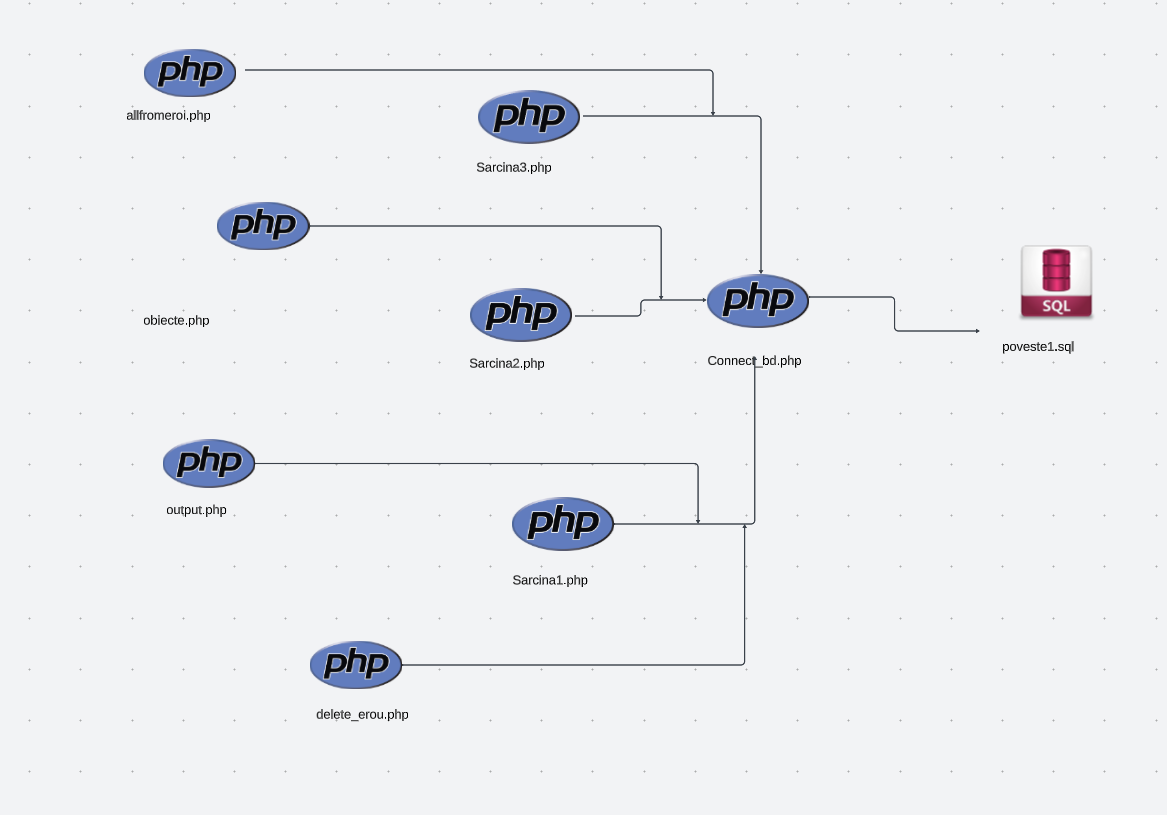


**ER:**



**modificare\_personaje.php**

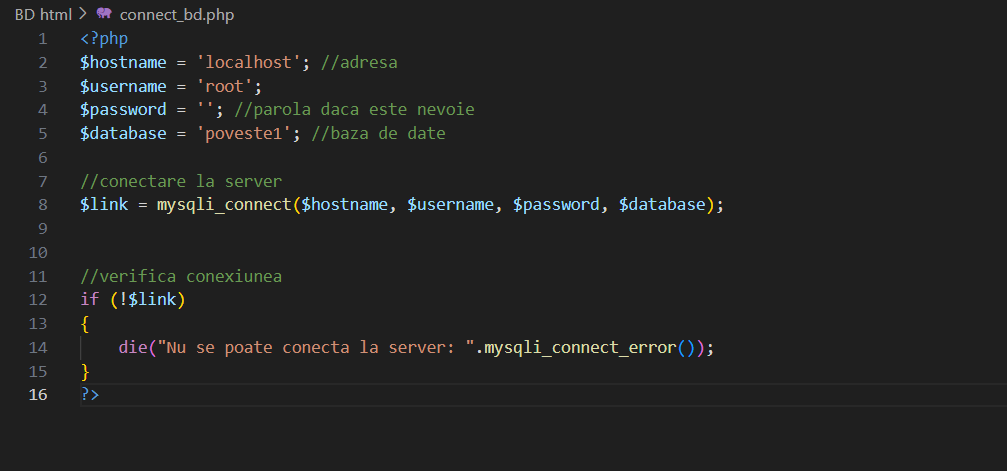
**Schema bloc a scriptului de lucru:**



**Descrierea funcțiilor utilizate**

**Scriptul de conectare la baza de date:**

connect\_db.php

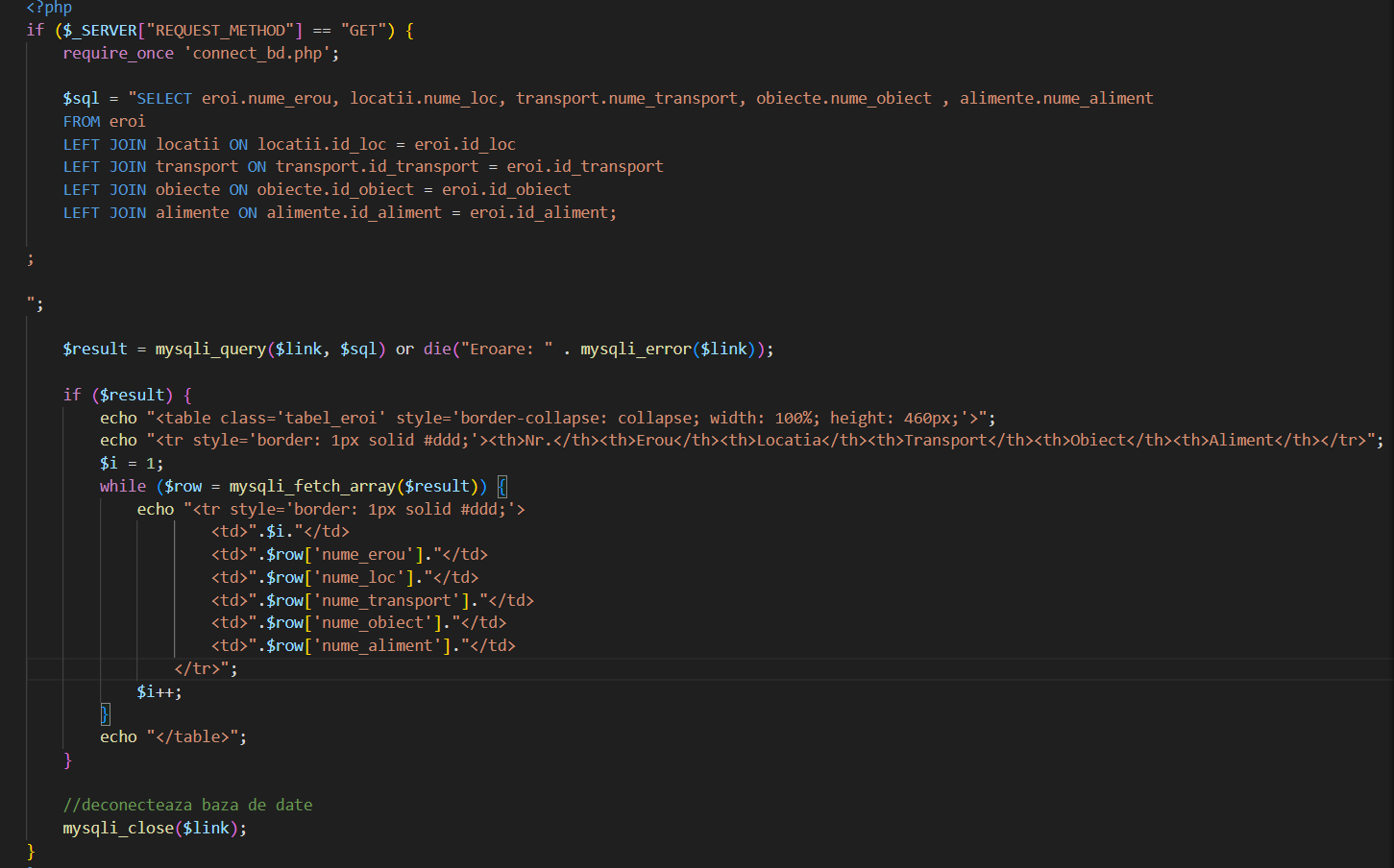


Funcțiile utilizate:

* $hostname, $username, $password, $database – variabile care conțin informațiile necesare pentru conectarea la baza de date;
* mysqli\_connect – funcție de stabilire a unei conexiuni cu serverul MySQL;
* mysqli\_connect\_error – funcție de afișare a erorii.

**Scriptul de afișare a personajelor:**

allfromeroi.php



Codul începe prin includerea fișierului 'connect\_bd.php', care presupun că conține funcția de conectare la baza de date.

Interogarea SQL selectează date din mai multe tabele folosind operații LEFT JOIN.

mysqli\_query($link, $sql): Execută interogarea SQL folosind conexiunea la baza de date stabilită ($link).

Funcția mysqli\_query returnează un set de rezultate, care este apoi procesat într-o buclă folosind mysqli\_fetch\_array.

Datele recuperate sunt afișate pentru a crea o tabelă HTML care prezintă rezultatele.

În final, mysqli\_close($link) este utilizat pentru a închide conexiunea la baza de date.

Sarcina1.php



Codul începe prin includerea fișierului 'connect\_bd.php', care presupun că conține funcția de conectare la baza de date.

Interogarea SQL selectează date din mai multe tabele folosind operații LEFT JOIN.

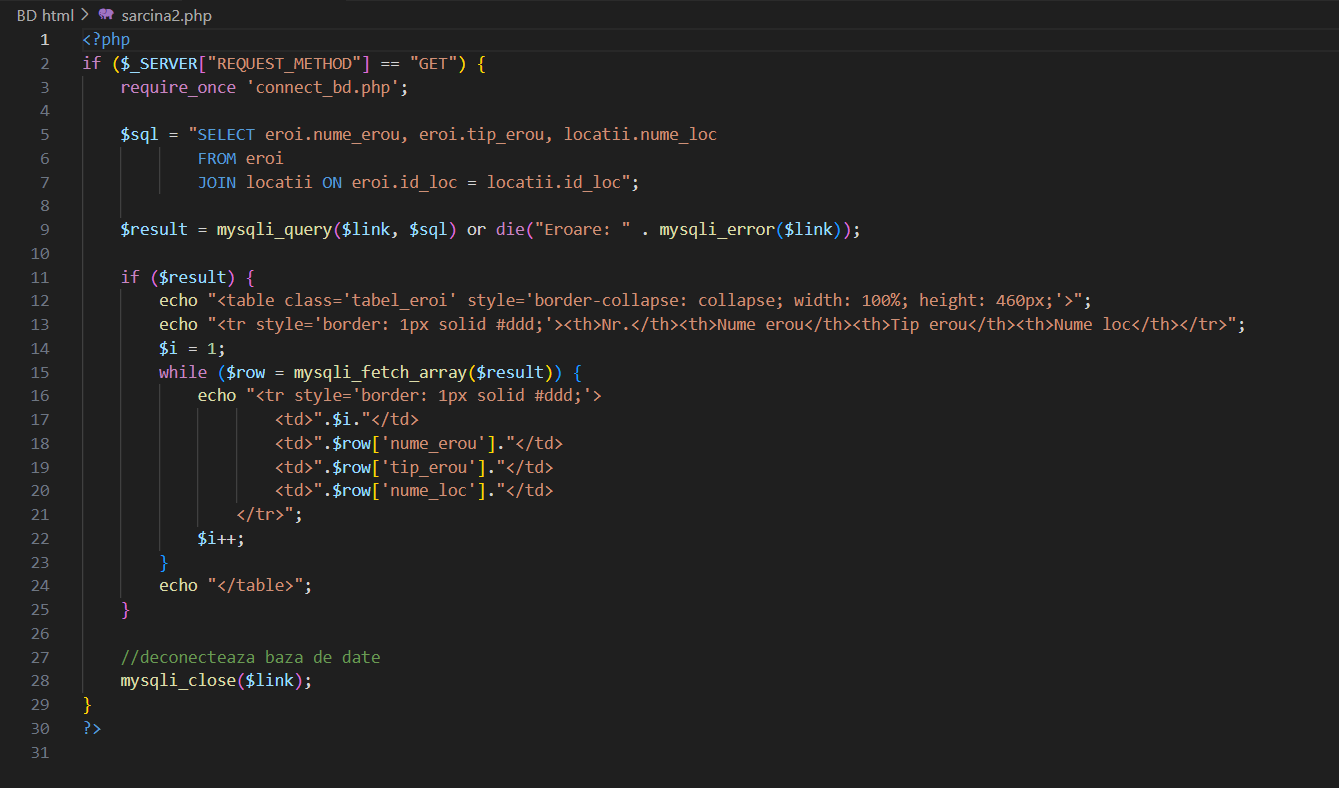
mysqli\_query($link, $sql): Execută interogarea SQL folosind conexiunea la baza de date stabilită ($link).

Funcția mysqli\_query returnează un set de rezultate, care este apoi procesat într-o buclă folosind mysqli\_fetch\_array.

Datele recuperate sunt afișate pentru a crea o tabelă HTML care prezintă rezultatele.

În final, mysqli\_close($link) este utilizat pentru a închide conexiunea la baza de date.

Sarcina2.php



Codul începe prin includerea fișierului 'connect\_bd.php', care presupun că conține funcția de conectare la baza de date.

Interogarea SQL selectează date din mai multe tabele folosind operații LEFT JOIN.

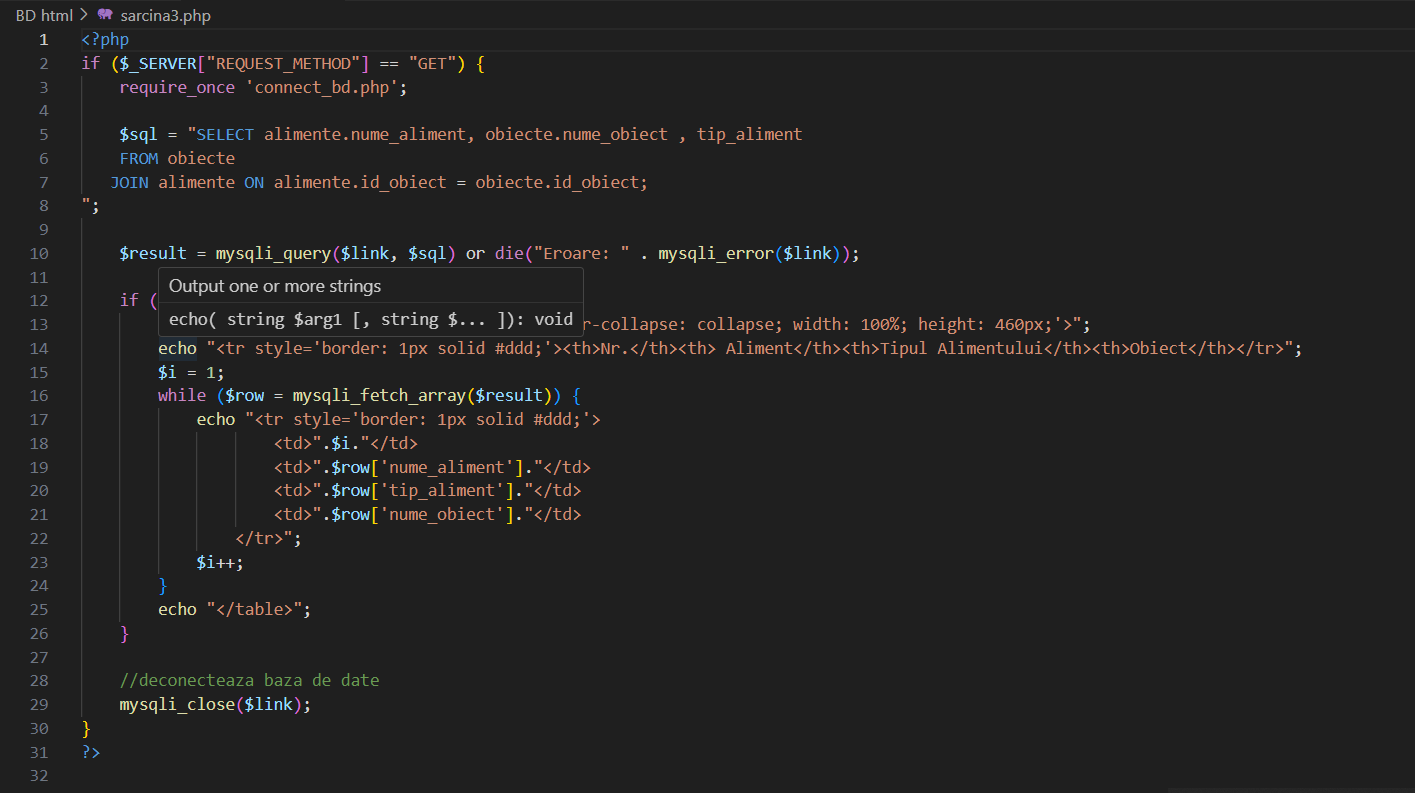
mysqli\_query($link, $sql): Execută interogarea SQL folosind conexiunea la baza de date stabilită ($link).

Funcția mysqli\_query returnează un set de rezultate, care este apoi procesat într-o buclă folosind mysqli\_fetch\_array.

Datele recuperate sunt afișate pentru a crea o tabelă HTML care prezintă rezultatele.

În final, mysqli\_close($link) este utilizat pentru a închide conexiunea la baza de date.

Sarcina3.php



Codul începe prin includerea fișierului 'connect\_bd.php', care presupun că conține funcția de conectare la baza de date.

Interogarea SQL selectează date din mai multe tabele folosind operații LEFT JOIN.

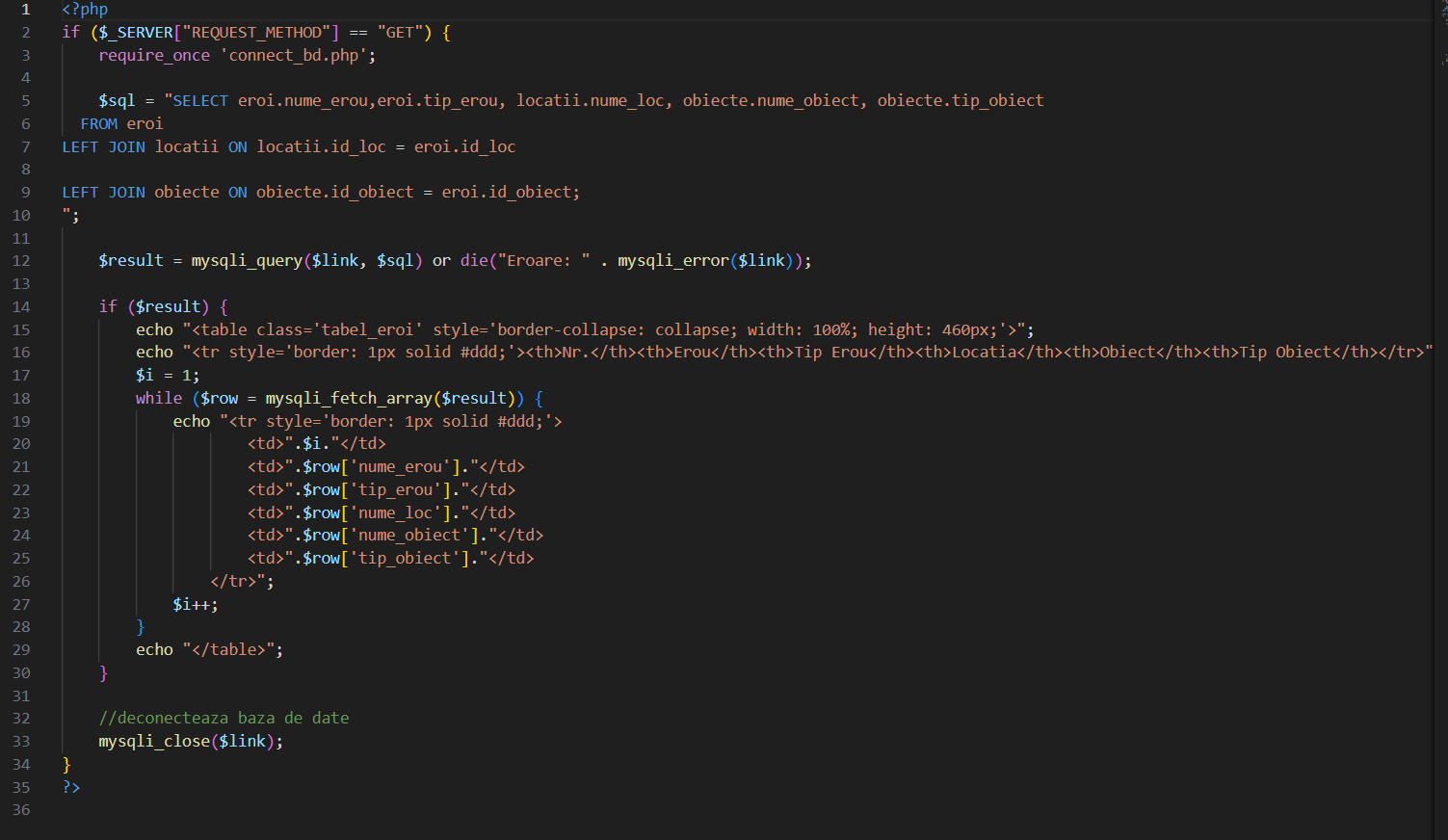
mysqli\_query($link, $sql): Execută interogarea SQL folosind conexiunea la baza de date stabilită ($link).

Funcția mysqli\_query returnează un set de rezultate, care este apoi procesat într-o buclă folosind mysqli\_fetch\_array.

Datele recuperate sunt afișate pentru a crea o tabelă HTML care prezintă rezultatele.

În final, mysqli\_close($link) este utilizat pentru a închide conexiunea la baza de date.

Obiecte.php



Construiește o interogare SQL care selectează nume, tipuri și alte informații despre eroi, locații și obiecte.

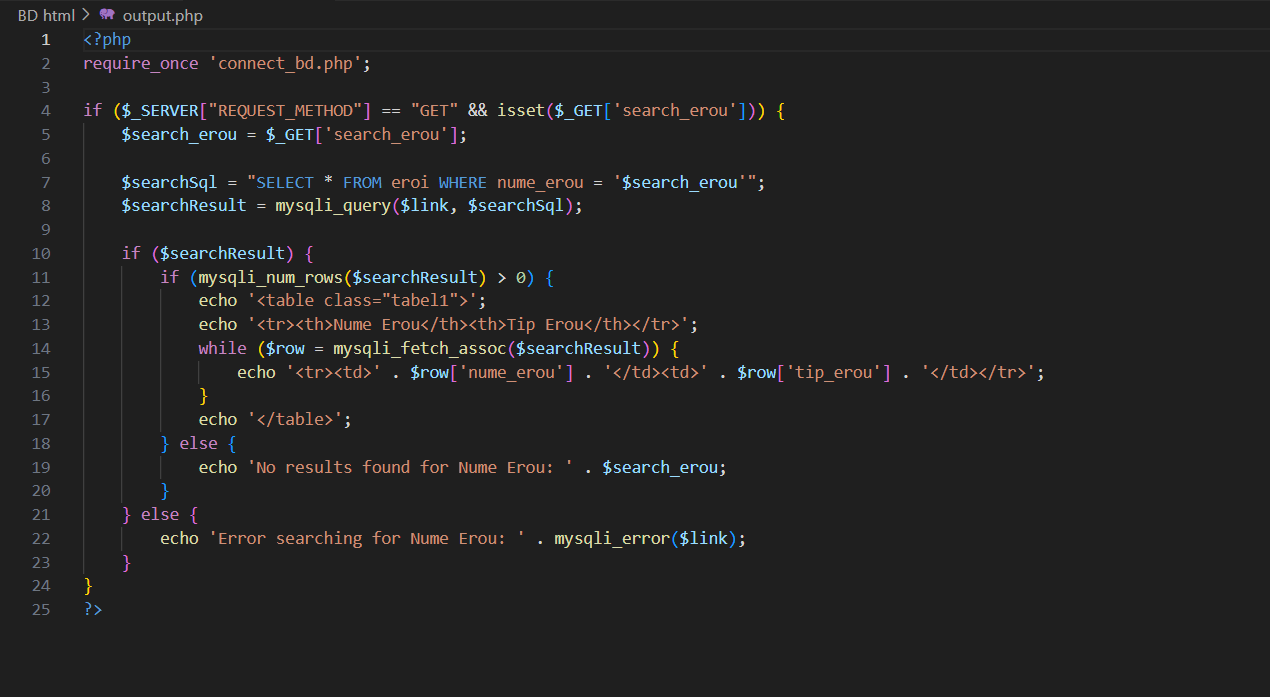
Executarea interogării și afișarea rezultatelor:

$result = mysqli\_query($link, $sql) or die("Eroare: " . mysqli\_error($link));

Execută interogarea folosind funcția mysqli\_query.

Dacă există erori în timpul execuției interogării, se afișează un mesaj de eroare și se oprește execuția scriptului.

Output.php



Construiește o interogare SQL pentru a căuta eroi cu numele specificat.

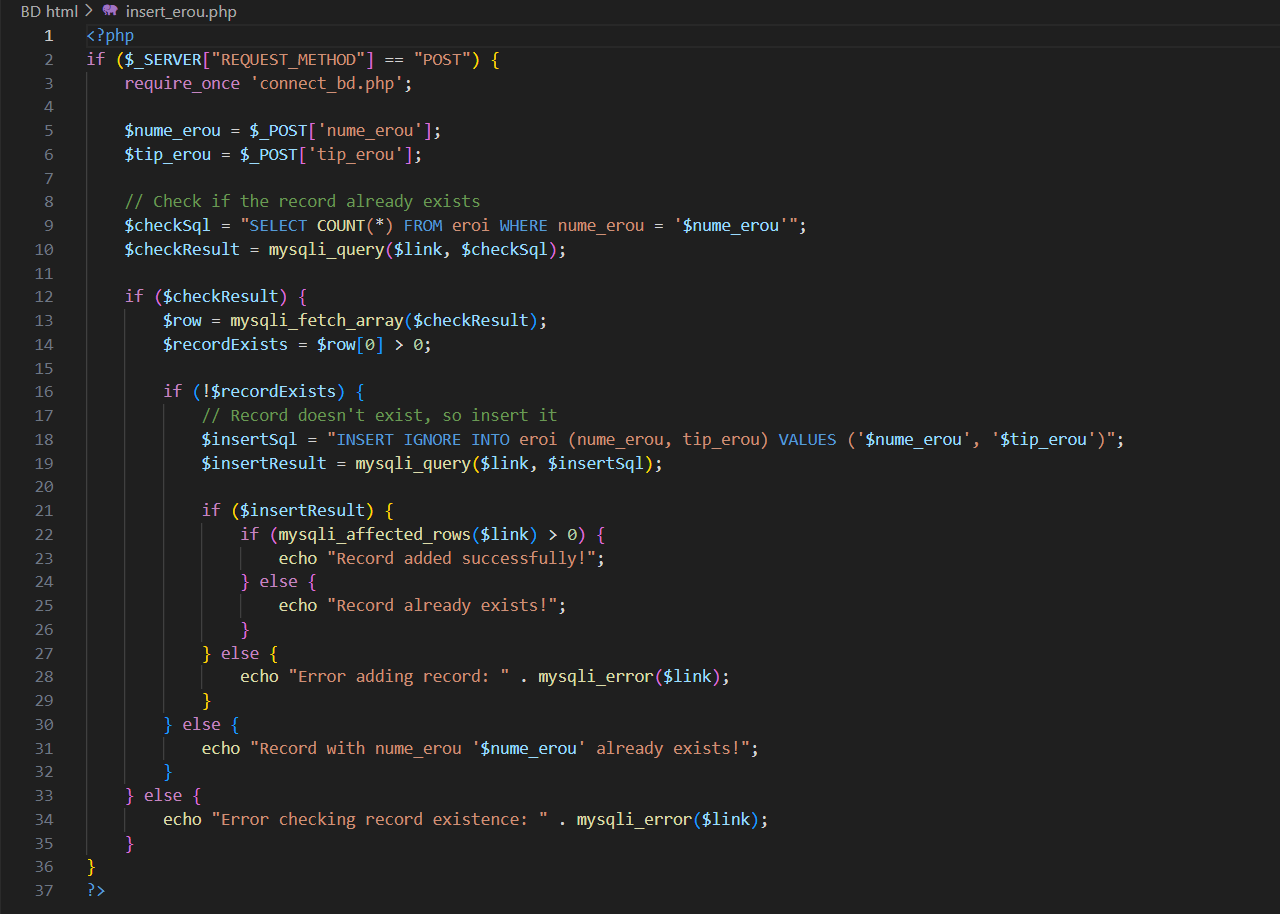
Execută interogarea folosind funcția mysqli\_query și obține rezultatul în variabila $searchResult.

Verifică dacă rezultatul căutării este valid.

Afișează rezultatele într-o tabelă HTML sau mesajul că nu s-au găsit rezultate.

În caz de eroare, afișează un mesaj de eroare.

Insert\_erou.php



Includerea fișierului de conexiune: require\_once 'connect\_bd.php';

Obținerea datelor din formular: $nume\_erou = $\_POST['nume\_erou'];

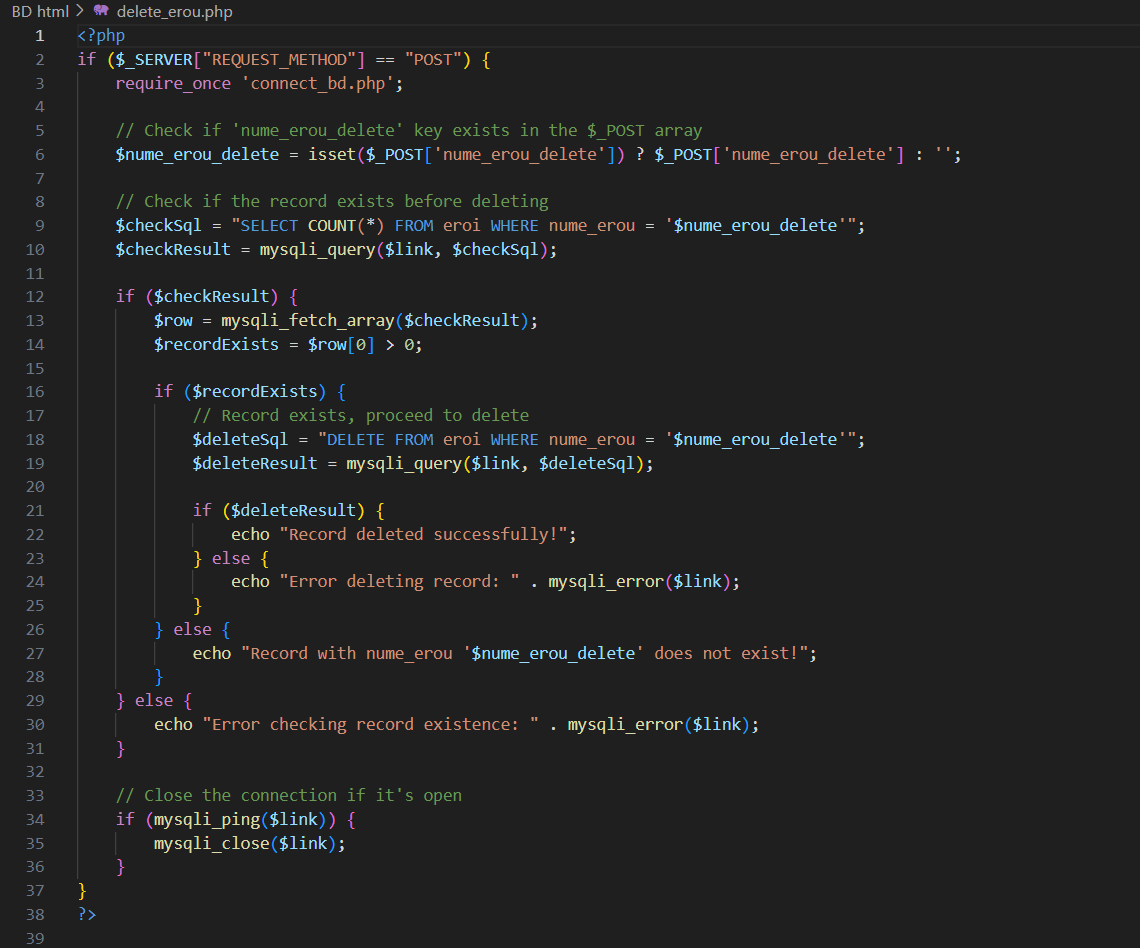
$tip\_erou = $\_POST['tip\_erou'];

Verificarea existenței înregistrării

Inserarea înregistrării Dacă înregistrarea nu există, se efectuează o instrucțiune de inserare folosind INSERT IGNORE pentru a evita duplicarea înregistrării

Afișarea mesajului de eroare pentru verificarea existenței înregistrării:

Delete\_erou.php



Funcțiile / variabilele utilizate:

* require\_once ‘connect\_bd.php’ – include fișierul de conectare la baza de date
* mysqli\_query($link, $sql) – execută o interogare MySQL pe baza de date folosind conexiunea stabilită anterior;
* mysqli\_fetch\_array($result) – preia rândurile rezultate din interogare ca un array asociativ;
* mysqli\_close($link) – închide conexiunea cu baza de date;
* die() – oprește scriptul;
* mysqli\_error() – obține descrierea textului erorii asociate;
* $result – primește rezultatul interogării bazei de date;
* $row – primește fiecare rând de date din tabel;
* echo – are aceeași funcție ca și print, de a afișa ceva la ecran (în lucrarea dată a fost folosit pentru a crea tabele și pentru a afișa mesajele de eroare).

JavaScript:

    <script>

      window.onload = function () {

          var form = document.getElementById('tableForm');

          form.addEventListener('submit', function (event) {

              event.preventDefault();

              var selectedTable = form.querySelector('input[name="tabel"]:checked').value;

              var xhr = new XMLHttpRequest();

              xhr.onreadystatechange = function () {

                  if (xhr.readyState === 4 && xhr.status === 200) {

                      document.getElementById('eroi').innerHTML = xhr.responseText;

                  }

              };

              // Adjust the URL based on the selected table

              xhr.open('GET', selectedTable + '.php', true);

              xhr.send();

          });

      var deleteForm = document.getElementById('deleteForm');

deleteForm.addEventListener('submit', function (event) {

  event.preventDefault();

  var xhr = new XMLHttpRequest();

  xhr.onreadystatechange = function () {

    if (xhr.readyState === 4) {

      if (xhr.status === 200) {

        document.getElementById('eroi').innerHTML = xhr.responseText;

      } else {

        console.error('Error deleting record:', xhr.status, xhr.statusText);

      }

    }

  };

  // Adjust the URL based on the selected table

  xhr.open('POST', 'delete\_erou.php', true);

  // Serialize form data

  var formData = new FormData(deleteForm);

  // Send the form data using POST

  xhr.send(formData);

});

var insertForm = document.getElementById('insertForm');

    insertForm.addEventListener('submit', function (event) {

      event.preventDefault();

      var xhr = new XMLHttpRequest();

      xhr.onreadystatechange = function () {

        if (xhr.readyState === 4) {

          if (xhr.status === 200) {

            document.getElementById('eroi').innerHTML = xhr.responseText;

          } else {

            console.error('Error adding record:', xhr.status, xhr.statusText);

          }

        }

      };

      // Adjust the URL based on the selected table

      xhr.open('POST', 'insert\_erou.php', true);

      // Serialize form data

      var formData = new FormData(insertForm);

      // Send the form data using POST

      xhr.send(formData);

    });

  };

    document.getElementById('outputForm').addEventListener('submit', function (event) {

        event.preventDefault();

        var search\_erou = document.getElementById('search\_erou').value;

        var xhr = new XMLHttpRequest();

        xhr.onreadystatechange = function () {

            if (xhr.readyState === 4 && xhr.status === 200) {

                document.getElementById('outputResult').innerHTML = xhr.responseText;

            }

        };

        xhr.open('GET', 'output.php?search\_erou=' + search\_erou, true);

        xhr.send();

    });

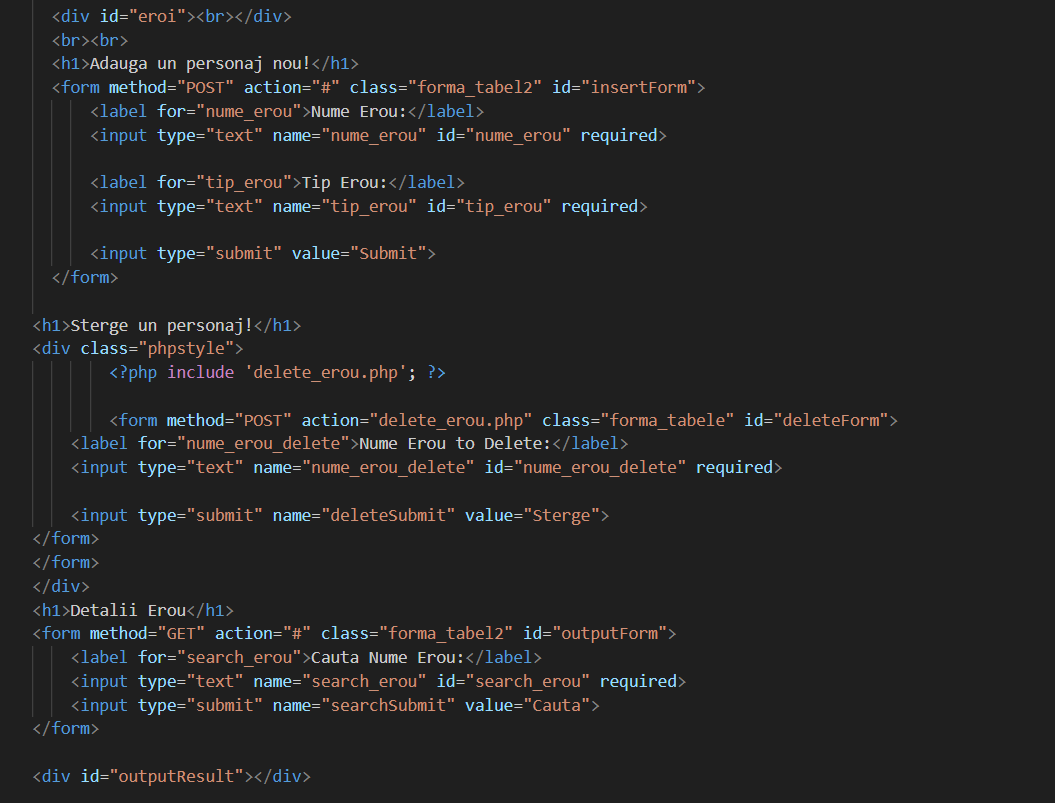
  </script>

Funcțiile utilizate:

* window.onload – asigură ca codul din interior să fie executat atunci când toată pagina s-a încărcat;
* new XMLHttpRequest() – creează un obiect XMLHttpRequest pentru a face cereri HTTP către server fără a reîmprospăta întreaga pagină;
* xhr.onreadystatechange – definește o funcție care va fi apelată de fiecare dată când starea obiectului se schimbă;
* xhr.open – specifică tipul de cerere și adresa către care se face cererea;
* xhr.send() – trimite cererea la server.

HTML:



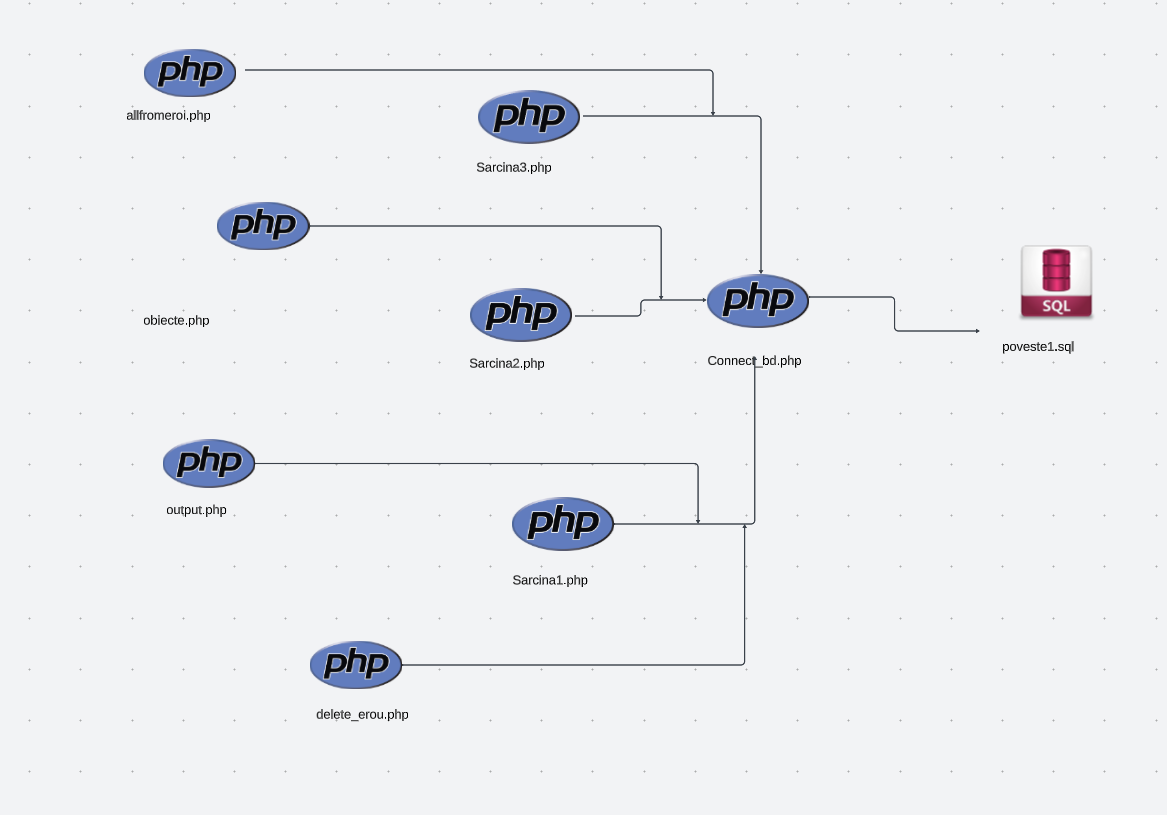


Elementele utilizate:

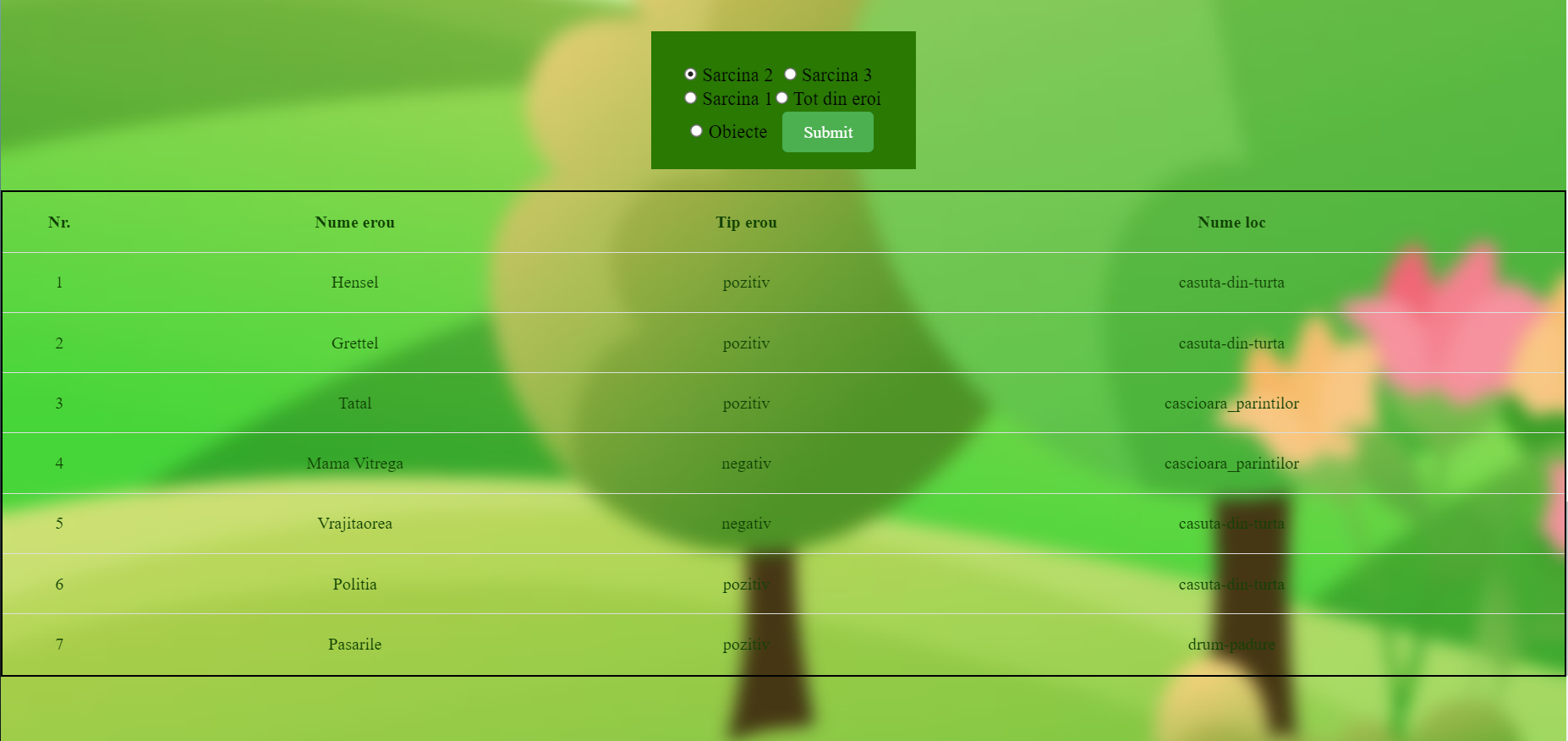
* <form> - formă pentru input-ul utilizatorului;
* <label> - label pentru elementul de input;
* <button> - buton de tip click;
* <input type=”text”> - căsuță de input pentru introducerea datelor de la utilizator;
* <input type=”button”> - input de tip buton.
* <table> - inserarea unui tabel

Când pagina se încarcă, codul JavaScript face o cerere către afisare\_personaje.php și actualizează conținutul div-ului cu id-ul table-personaje.

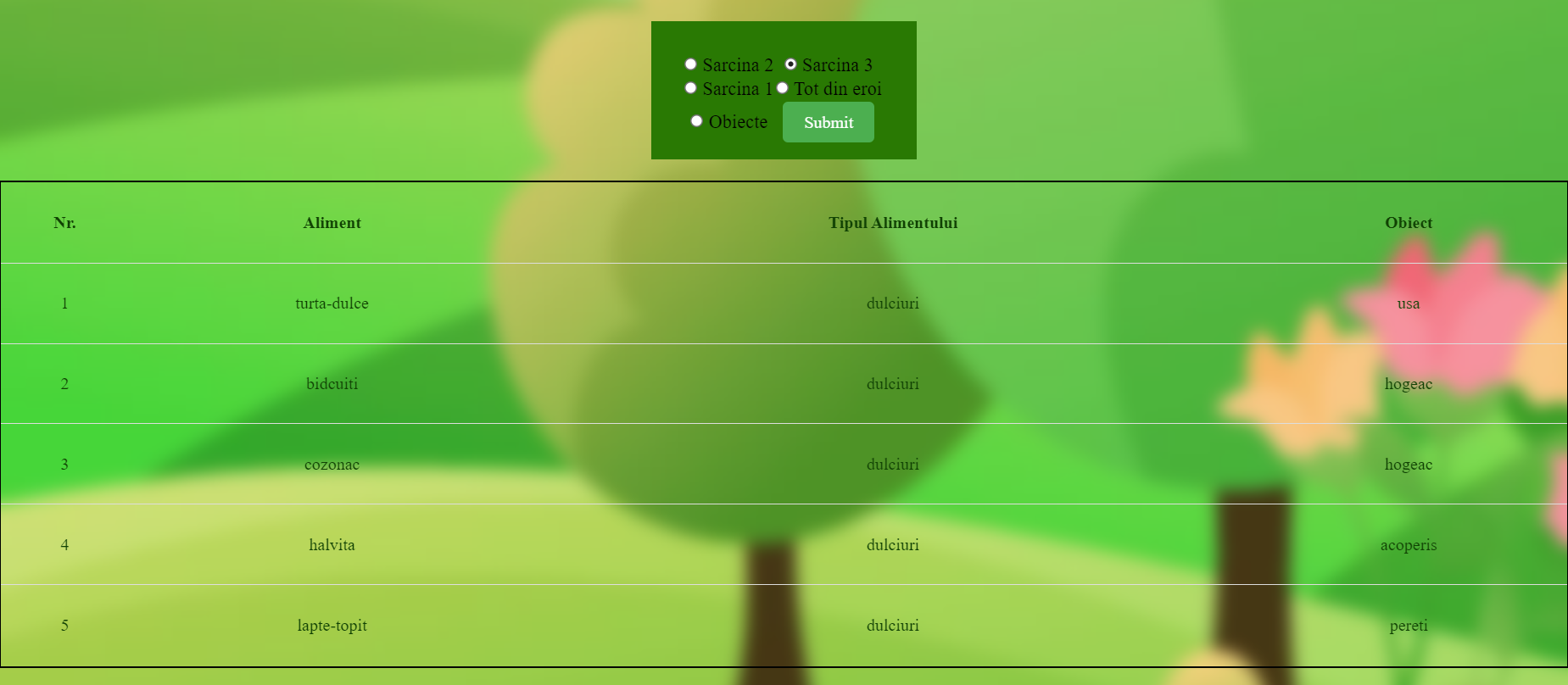
**Schema bloc a scriptului de lucru:**



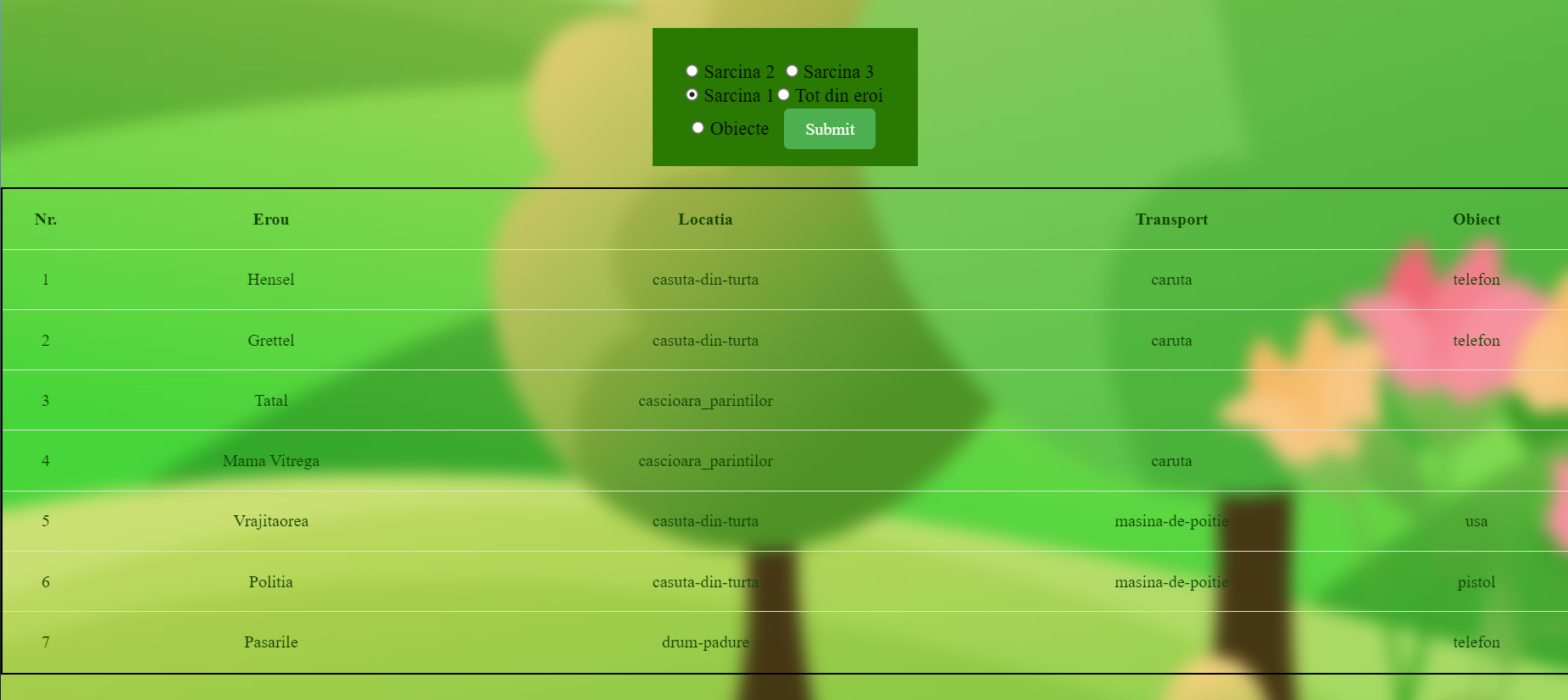
**Rezultatele de lucru a scriptului**



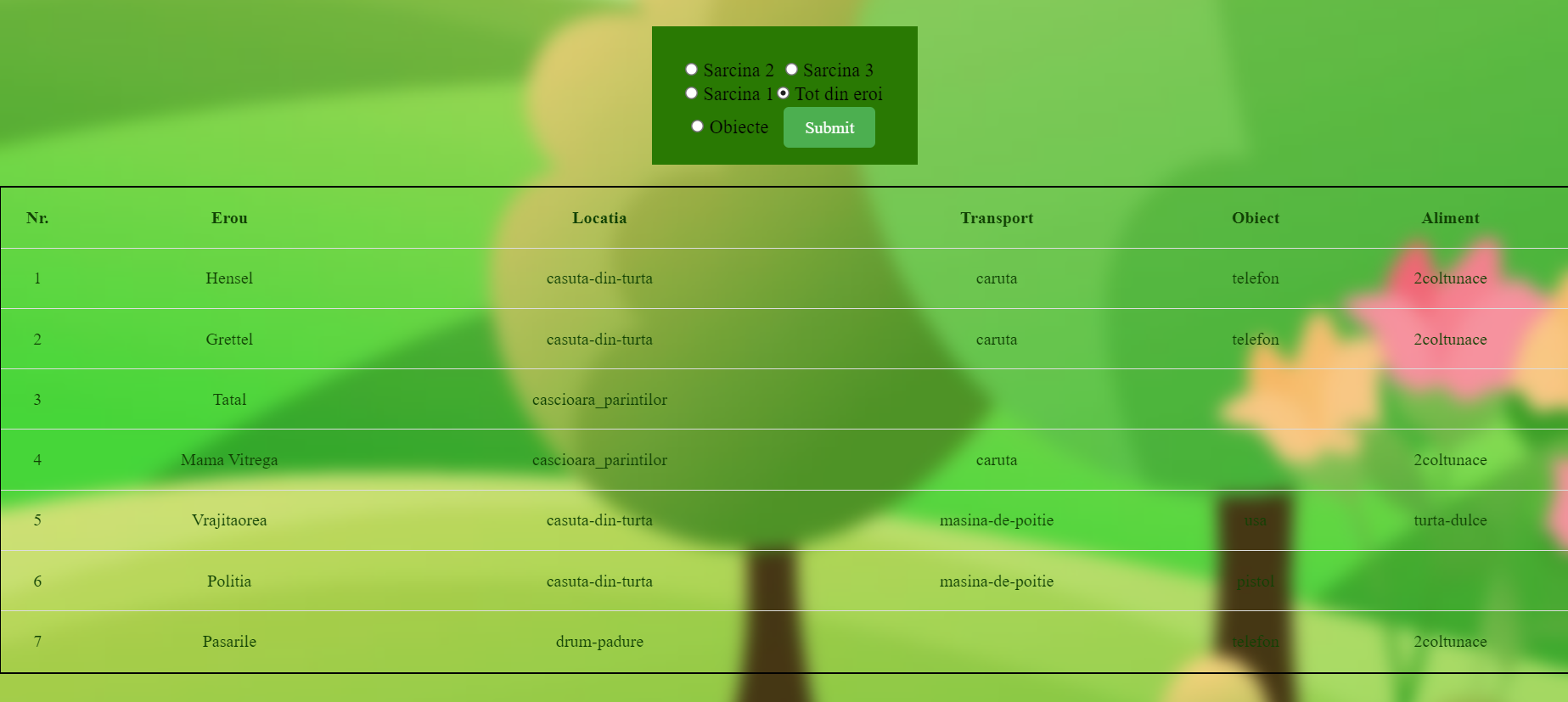
**Sarcina2.php**



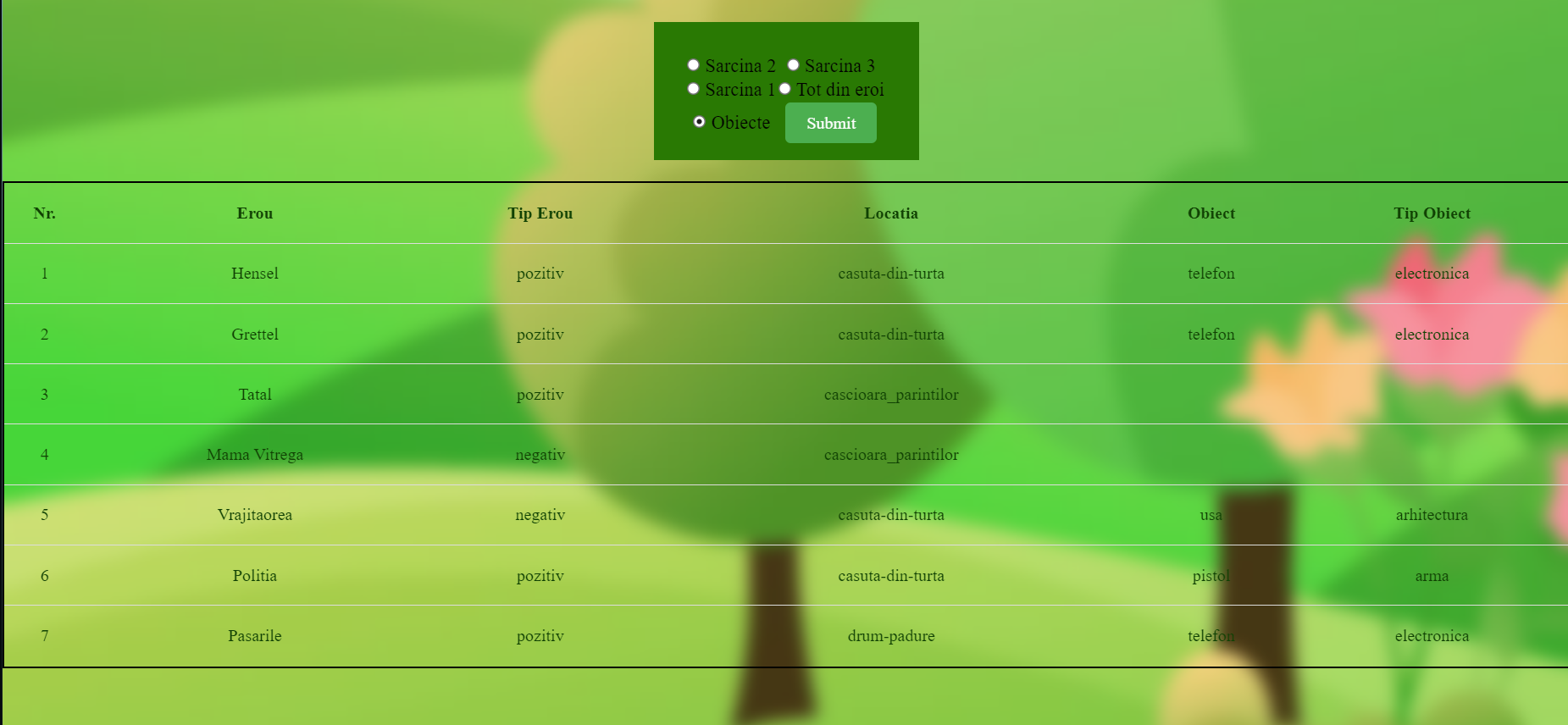
**Sarcina3.php**



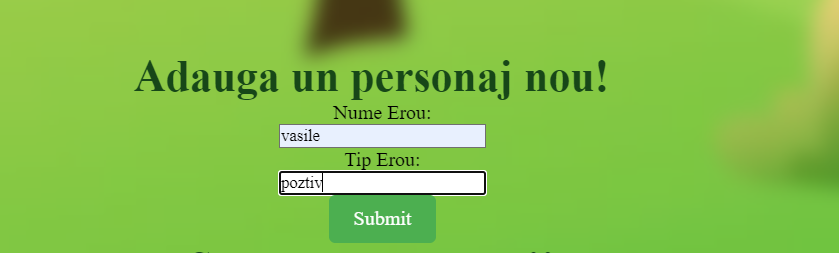
**Sarcina1.php**

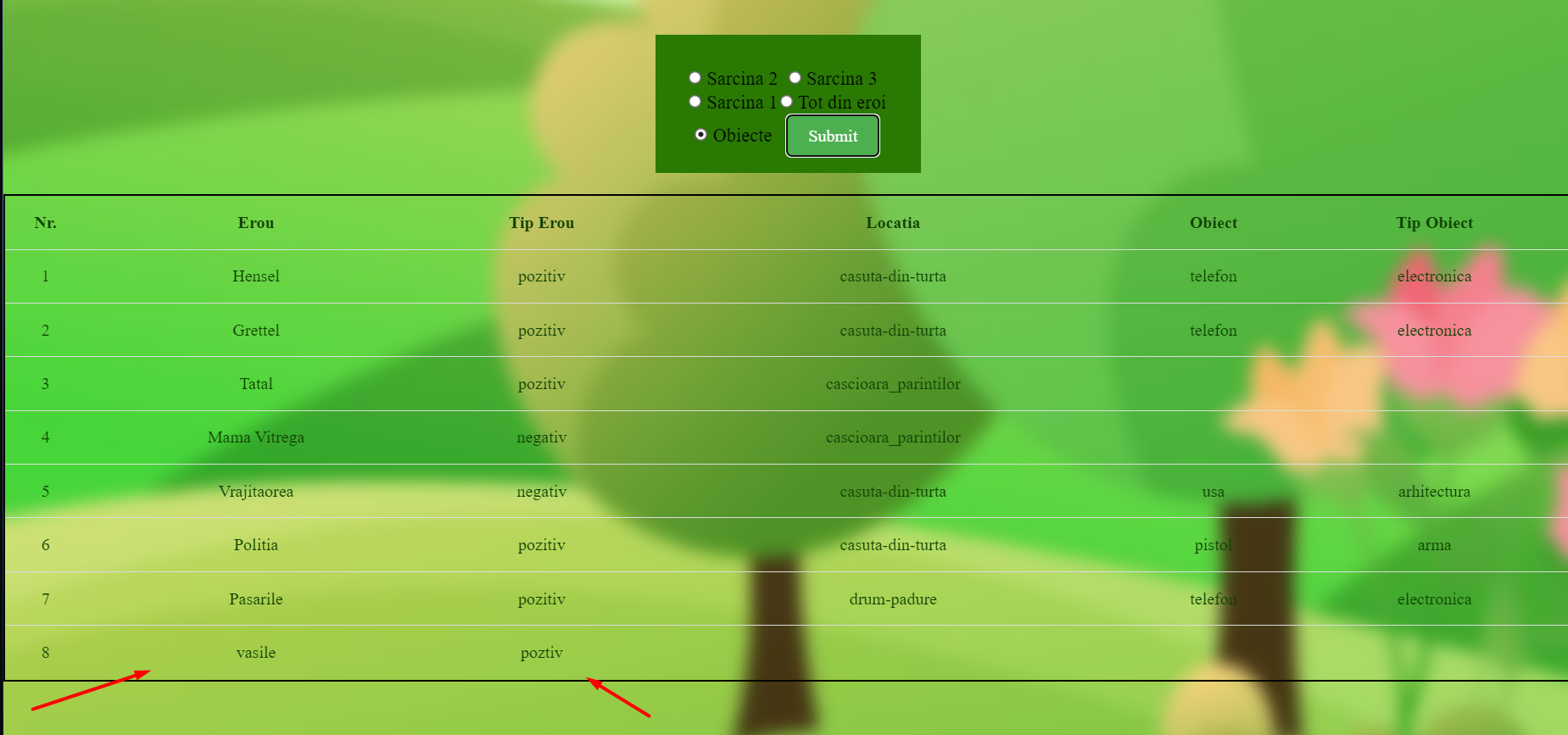


**Totdineroi.php**



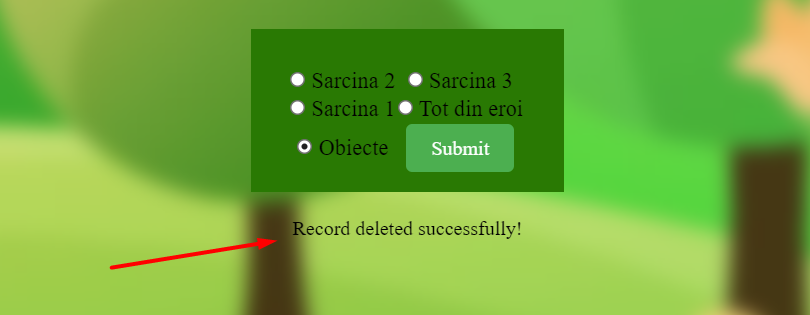
**Obeicte.php**



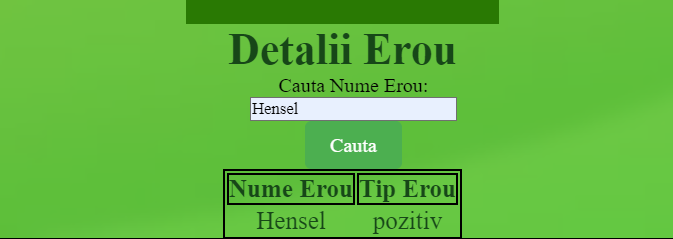


**Input.php**





**Delete\_erou.php**



**Rezultatele de lucru a scriptului**

Toate scripturile fac conexiune cu baza de date, fiecare are rolul sau de afisare si de corectare CRUD.

**Concluzie**

În cadrul acestei lucrări de laborator, s-a realizat o integrare eficientă între un site web și o bază de date bazată pe povestea Soldățelul de plumb. Pentru a putea realiza această lucrare, am beneficiat de suportul oferit de Wampserver, o soluție care integrează serviciile web Apache, baza de date MySQL și limbajul de scriptare PHP, facilitând astfel dezvoltarea și testarea aplicațiilor web. Implementarea s-a făcut folosind tehnologii precum JavaScript, PHP și editorul de cod Visual Studio Code. Un punct important în acest laborator, este faptul că s-au utilizat fișiere externe, deoarece prin divizarea codului în fișiere separate, ajută la organizarea și structura lucrării. Aceasta face codul mai ușor de gestionat și înțeles, totodată facilitând reutilizarea codului, pentru a evita duplicarea și erorile. Prin intermediul acestui laborator, s-au învățat concepte esențiale legate de gestionarea bazelor de date într-un mediu web interactiv. S-a obținut o înțelegere practică a modalităților de căutare, afișare, modificare, inserare și ștergere a datelor într-o bază de date, oferind o experiență completă de interacțiune utilizator-bază de date. De asemenea, s-au dezvoltat abilități în lucrul cu limbajele JavaScript și PHP pentru a facilita comunicarea și manipularea datelor între site și baza de date. În final, această lucrare a ilustrat importanța și beneficiile utilizării tehnologiilor web într-un context practic, evidențiind modul în care acestea pot fi integrate pentru a crea aplicații interactive și funcționale.