COLEGIUL NAȚIONAL “ELENA CUZA”

**PROIECT PENTRU ATESTAT LA INFORMATICĂ**

**SITE WEB - PRĂVĂLIA CU MOBILĂ**

**(HTML, CSS, PHP, SQL)**

ELEV PROF. COORDONATOR

PETCU MIHNEA RAVICI-TĂTĂRANU GEORGIANA

MAI, 2024

**CUPRINS**

1. **Alegerea temei****3**
2. **Aplicații și software utilizat 4**
   1. **Supabase, Vercel si Github 4**
   2. **Limbajul HTML, CSS și PHP 5**
3. **Prezentarea proiectului 9**
4. **Bibliografie 25**

# 

# 

**1. Alegerea temei**

Tema proiectului meu de atestat la informatica, intitulat "Prăvălia cu Mobilă", a fost selectată cu grijă, având în vedere mai multe aspecte importante.

Am ales să dezvolt un site web pentru un magazin de mobilă având în vedere următoarele motive:

Oportunitate de inovare în domeniu: Observând că domeniul mobilierului nu este atât de dezvoltat online precum alte domenii, am văzut o oportunitate de a aduce inovație și de a îmbunătăți experiența utilizatorilor prin introducerea unor funcționalități noi și utile.

Accesibilitate pentru clienți: Am dorit să creez un mediu online accesibil și ușor de utilizat pentru potențialii cumpărători de mobilă, oferindu-le posibilitatea de a explora și achiziționa produse într-un mod convenabil și plăcut.

Diversificarea serviciilor: Prin introducerea unor funcționalități precum restaurarea mobilierului, consultanță în design interior și designul de mobilier personalizat, am dorit să ofer clienților o gamă mai largă de opțiuni și servicii valoroase.

Alegerea temei a fost determinată de dorința de a aduce valoare într-un domeniu în care consider că există potențial pentru îmbunătățire și inovare. Site-ul web "Prăvălia cu Mobilă" își propune să ofere o experiență de cumpărare plăcută și eficientă pentru clienții săi, fiind axat pe calitate, accesibilitate și diversitate.

**2. Aplicații și Software Utilizat**

**2.1 Supabase, Vercel si Github**

Pentru dezvoltarea și implementarea proiectului "Prăvălia cu Mobila", am utilizat mai multe servicii și platforme online. Trei dintre acestea sunt Supabase, Vercel și Github.

Supabase.com: Am ales Supabase pentru a gestiona baza de date a proiectului. Supabase este o platformă open-source care oferă servicii de bază de date și autentificare, bazate pe PostgreSQL. Acest serviciu mi-a permis să creez și să gestionez ușor baza de date a magazinului meu online, inclusiv funcționalitățile de autentificare pentru utilizatori.

Vercel.com: Pentru hosting-ul și implementarea site-ului web, am ales Vercel. Vercel este o platformă de hosting pentru aplicații web și site-uri statice, care oferă o gamă largă de instrumente și servicii pentru dezvoltarea și implementarea proiectelor web. Am ales Vercel datorită facilității sale de utilizare, performanței ridicate și scalabilității.

Github.com: GitHub este o platformă esențială pentru dezvoltatorii de proiecte software, oferind o gamă largă de servicii și instrumente care facilitează colaborarea, gestionarea codului și urmărirea progresului în cadrul echipelor de dezvoltare. Am folosit GitHub dintr-o varietate de motive, iar unul dintre cele mai relevante pentru mine este capacitatea sa de a găzdui proiecte web. De exemplu, am folosit GitHub în legătură cu platforma Vercel pentru a hosta un site web. Această integrare între GitHub și Vercel permite o gestionare ușoară a codului și un deployment rapid și fiabil al proiectelor web. Astfel, GitHub devine un pilon solid în ecosistemul meu de dezvoltare, oferindu-mi uneltele de care am nevoie pentru a construi și a livra aplicații web de calitate.

Aceste platforme au fost fundamentale pentru dezvoltarea și implementarea proiectului meu, asigurându-se că site-ul web este bine hostat și că baza de date este gestionată eficient.

**2.2 Limbajul HTML, CSS și PHP**

HTML (HyperText Markup Language): HTML este limbajul fundamental al web-ului, folosit pentru a structura și formata conținutul paginilor web. În esență, HTML definește scheletul unei pagini web, indicând browserului cum să afișeze diverse elemente, cum ar fi textul, imaginile, link-urile și altele. Cu ajutorul tag-urilor HTML, putem crea structura logică a unei pagini web, inclusiv capul documentului (<head>), corpul (<body>), secțiuni (<div>), titluri (<h1>, <h2>, etc.) și multe altele. De asemenea, HTML oferă suport pentru formulare (<form>), tabele (<table>), listele (<ul>, <ol>) și alte elemente esențiale pentru interacțiunea și prezentarea conținutului.

Markup Language (Limbaj de marcări):

HTML este un limbaj de marcări, ceea ce înseamnă că este compus dintr-un set de elemente și tag-uri care sunt utilizate pentru a marca și structura conținutul paginii web. Aceste tag-uri sunt folosite pentru a defini diferite elemente precum titluri, paragrafe, imagini, link-uri etc.

Structura de bază:

O pagină HTML începe cu un tag de început <html> și se încheie cu un tag de sfârșit </html>. În interiorul acestui element, se găsesc două secțiuni principale: <head> și <body>.

În secțiunea <head> sunt definite informații despre pagină, cum ar fi titlul, legături către fișiere CSS sau JavaScript, metadata etc.

Secțiunea <body> conține conținutul efectiv al paginii, cum ar fi text, imagini, link-uri, formulare etc.

Elemente și tag-uri:

HTML este construit dintr-un set de elemente și tag-uri. Fiecare element este definit prin intermediul unui tag de început și unul de sfârșit, care delimitează conținutul elementului. De exemplu:

<p>Acesta este un paragraf.</p>

Tag-ul <p> este tag-ul de început, iar </p> este tag-ul de sfârșit al elementului paragraf.

Atribute:

Multe tag-uri HTML pot avea și atribut, care oferă informații suplimentare despre acel element. De exemplu, tag-ul pentru o imagine (<img>) poate include un atribut src care specifică sursa imaginii. Exemplu:

<img src="image.jpg" alt="Imagine">

În acest exemplu, src este atributul care indică sursa imaginii, iar alt oferă un text alternativ care va fi afișat în cazul în care imaginea nu poate fi încărcată.

Structurare și semantica:

HTML oferă o gamă largă de tag-uri pentru a structura conținutul într-un mod semnificativ. De exemplu, există tag-uri pentru titluri (<h1> până la <h6>), paragrafe (<p>), liste (<ul>, <ol>, <li>), tabele (<table>, <tr>, <td>), formulare (<form>, <input>, <button>) etc. Utilizarea adecvată a acestor tag-uri ajută la îmbunătățirea accesibilității și SEO-ului paginii.

Compatibilitate cu browserele:

HTML este un standard recunoscut și acceptat de către majoritatea browserele web moderne. Cu toate acestea, este important să se scrie cod HTML într-un mod care să fie compatibil cu diferitele versiuni de browsere și să respecte standardele web pentru a asigura o afișare corectă și consistentă a paginilor web pe diverse platforme și dispozitive.

CSS (Cascading Style Sheets): CSS este un limbaj de stilizare folosit pentru a adăuga aspect și estetică paginilor web create cu HTML. CSS (Cascading Style Sheets) este un limbaj de stilizare folosit pentru a controla aspectul și prezentarea elementelor HTML pe o pagină web. Folosind CSS, putem defini stilurile vizuale pentru diferite elemente HTML, cum ar fi culorile, fonturile, marginile, fundalurile și poziționarea lor pe pagină. Principiul de bază al CSS-ului este cascada, care permite suprascrierea și moștenirea stilurilor, facilitând astfel crearea unor design-uri coerente și flexibile pentru paginile web.

Stilizarea elementelor:

CSS permite aplicarea stilurilor (cum ar fi culori, fonturi, dimensiuni, margini etc.) la elementele HTML. Stilurile sunt definite folosind selecții și declarații. De exemplu:

p {

color: blue;

font-size: 16px;

}

Această regulă CSS va face ca toate paragrafele (<p>) să aibă textul albastru și dimensiunea fontului de 16 pixeli.

Selecții:

Selecțiile CSS sunt utilizate pentru a specifica la ce elemente HTML se aplică un anumit set de stiluri. Acestea pot fi selecții de tip, clasă, ID sau alte atribute. De exemplu:

p - selectează toate elementele <p> (paragrafe).

.clasa - selectează toate elementele care au clasa specificată în atributul class.

#id - selectează elementul cu ID-ul specificat în atributul id.

Cascadarea și moștenirea:

CSS-ul este numit "Cascading Style Sheets" deoarece stilurile pot fi aplicate într-o ierarhie, iar conflictele sunt rezolvate prin reguli de moștenire și prioritizare. De exemplu, un stil definit la nivel global poate fi suprascris de unul specific unei clase sau unui ID.

Unități de măsură:

CSS oferă o varietate de unități de măsură pentru dimensiuni și distanțe, cum ar fi pixeli, procente, em-uri etc. Acestea permit scalabilitate și flexibilitate în designul paginilor web.

Media Queries:

CSS permite definirea unor reguli specifice pentru diferite tipuri de dispozitive sau dimensiuni de ecran folosind media queries. Acestea permit adaptarea aspectului paginii web în funcție de caracteristicile dispozitivului pe care este afișată.

Compatibilitate cu browserele:

Similar cu HTML-ul, CSS-ul este un standard recunoscut și acceptat de majoritatea browserelelor moderne. Este important să se scrie cod CSS într-un mod care să fie compatibil cu diferitele versiuni de browsere pentru a asigura o afișare corectă și uniformă a paginilor web.

PHP (Hypertext Preprocessor): PHP este un limbaj de programare server-side utilizat pentru dezvoltarea aplicațiilor web dinamice și interacțiunea cu bazele de date. PHP permite generarea de conținut dinamic în paginile web, procesarea datelor introduse de utilizatori prin formulare, gestionarea sesiunilor și multe altele. Oferind o integrare strânsă cu HTML-ul, PHP permite crearea unor pagini web personalizate și interactive, adaptate nevoilor specifice ale utilizatorilor.

Server-Side Scripting:

PHP este un limbaj de programare server-side, ceea ce înseamnă că codul este executat pe server și rezultatele sunt trimise către browserul utilizatorului. Acest lucru permite generarea dinamică a conținutului web în funcție de diverse condiții și date.

Interacțiunea cu Bazele de Date:

PHP este adesea utilizat pentru a interacționa cu baze de date, permițând stocarea și manipularea datelor. Prin intermediul funcțiilor native sau a unor biblioteci precum MySQLi sau PDO, PHP poate executa interogări către bazele de date și poate manipula datele returnate.

Integrare cu HTML:

Codul PHP este încorporat în documentele HTML folosind delimitatoare speciale, de obicei <?php și ?>. Acest lucru permite dezvoltatorilor să creeze pagini web dinamice, în care conținutul este generat în funcție de variabile, condiții și alte logici de programare.

Variabile și Structuri de Control:

PHP suportă variabile și structuri de control similare altor limbaje de programare, cum ar fi if-else, for, while etc. Acestea permit dezvoltatorilor să controleze fluxul programului și să efectueze operații în funcție de condiții.

Funcții și Biblioteci:

PHP include o gamă largă de funcții și biblioteci încorporate, care facilitează sarcini comune, cum ar fi manipularea șirurilor de caractere, lucrul cu fișiere, trimiterea de e-mailuri etc. De asemenea, este posibilă încorporarea și utilizarea de biblioteci externe pentru funcționalități extinse.

Gestionarea Sesiunilor și a Cookie-urilor:

PHP oferă suport pentru gestionarea sesiunilor și a cookie-urilor, permițând dezvoltatorilor să mențină starea utilizatorilor între diferite cereri HTTP și să personalizeze experiența utilizatorului în funcție de acțiunile acestuia.

Securitate:

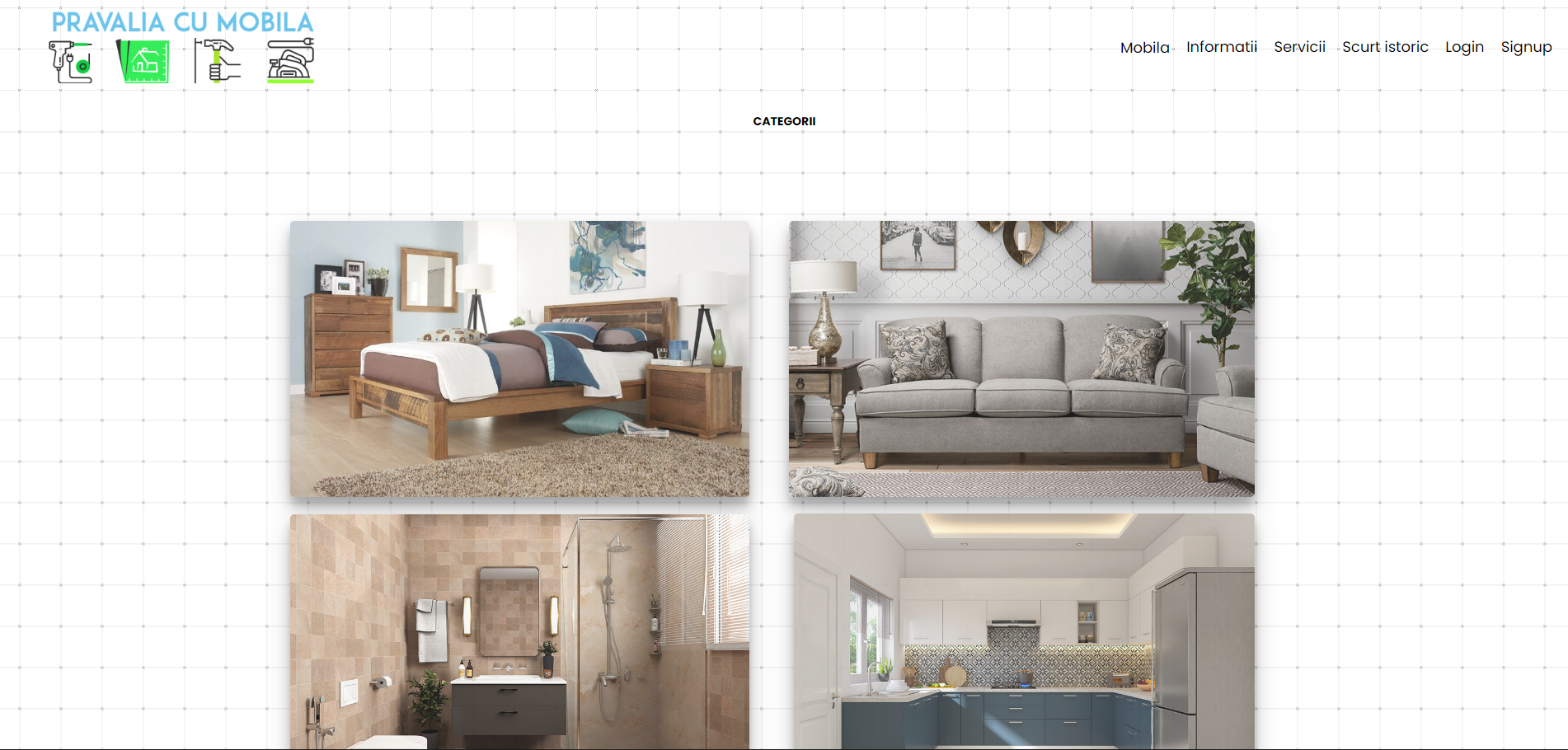
PHP include funcționalități pentru a ajuta la securizarea aplicațiilor web, cum ar fi filtrarea datelor de intrare pentru a preveni injecțiile SQL sau Cross-Site Scripting (XSS), și oferă funcții pentru criptare și gestionarea parolelor.

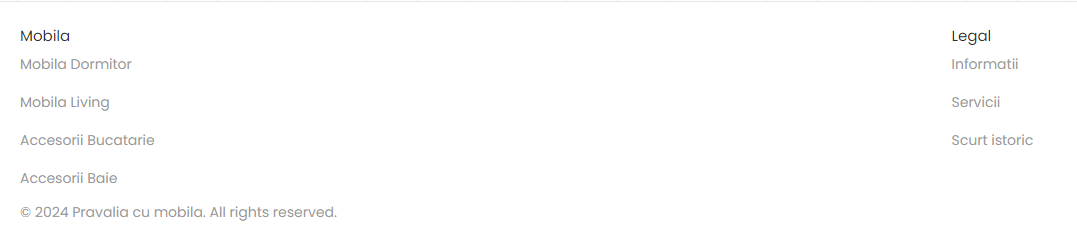
Aceste limbaje și tehnologii lucrează împreună pentru a crea un site web complet și funcțional, oferind o experiență plăcută și eficientă utilizatorilor. Prin combinarea puterii lor, putem dezvolta aplicații web complexe și atractive, satisfăcând nevoile și așteptările din ce în ce mai ridicate ale utilizatorilor online.

**3.Prezentarea proiectului**

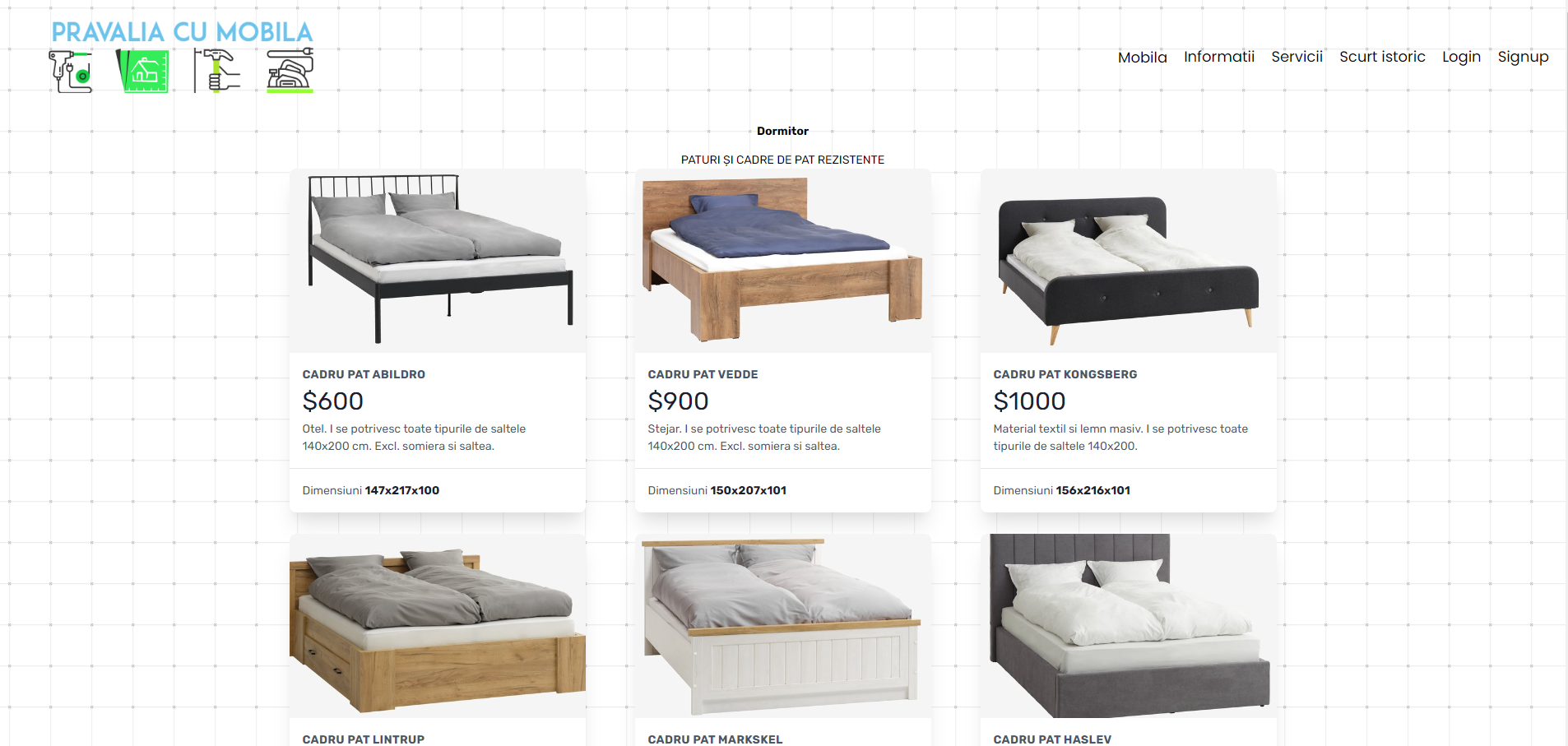
Proiectul "Prăvălia cu Mobilă" este un site web dedicat comercializării de mobilier, care oferă o gamă largă de produse și servicii clienților săi. Iată o prezentare generală a funcționalităților și caracteristicilor principale ale site-ului:

Pagina Principală: Pagina principală include un header și un footer unde se regăsesc toate paginile site-ului, oferind navigare rapidă către paginile adiacente și paginile cu informații despre magazin. În corpul paginii principale sunt afișate patru poze care direcționează utilizatorii către paginile dedicate produselor specifice (mobilier dormitor, living, bucătărie și baie).

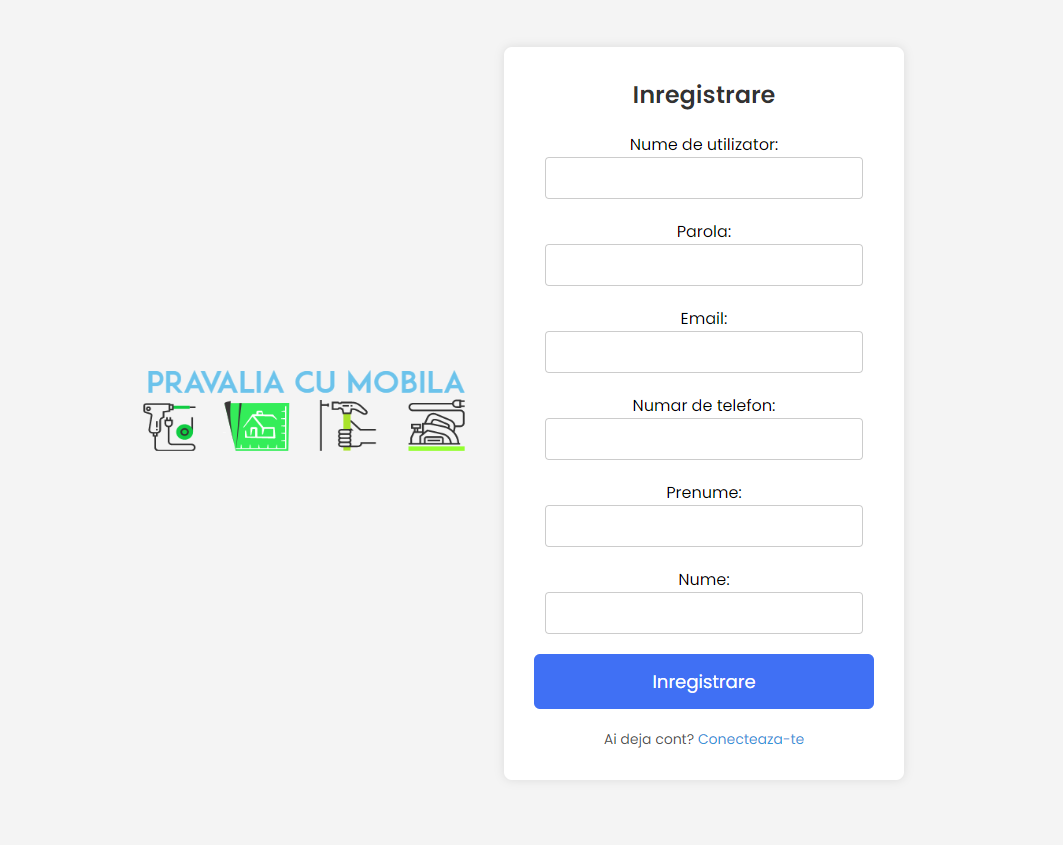




Paginile dedicate produselor: Aceste pagini conțin câte un card pentru fiecare produs, care includ o imagine cu produsul respectiv, o scurtă descriere, preț și dimensiunile produsului. Informațiile sunt preluate din baza de date a site-ului, unde sunt stocate detalii despre fiecare produs.



Paginile legate de contul de utilizator (Login și Sign up): Există două pagini pentru gestionarea contului de utilizator. Pe pagina de Sign up, utilizatorii pot completa șase câmpuri cu datele necesare pentru crearea unui cont. Aceste date sunt apoi trimise în baza de date, iar utilizatorii pot alege să se conecteze sau să se întoarcă la pagina principală. Pe pagina de Login, utilizatorii trebuie să introducă username-ul și parola. Procesul de login presupune interogarea bazei de date pentru verificarea datelor, iar în cazul în care acestea sunt corecte, se creează o sesiune în baza de date pentru utilizator.



**Crearea și gestionarea sesiunilor utilizatorilor**

Pentru a gestiona autentificarea utilizatorilor și pentru a le oferi o experiență personalizată în cadrul aplicației noastre, am implementat un sistem robust de sesiuni.

Identificarea sesiunilor

Pentru a identifica un utilizator și a-i gestiona sesiunea, folosim adresa IP a clientului. Aceasta este obținută prin funcția get\_client\_ip() și este apoi procesată pentru a elimina punctele și a obține un identificator unic al sesiunii. Această abordare ne permite să identificăm fiecare sesiune în mod unic și să gestionăm accesul utilizatorilor în aplicație.

**Crearea sesiunilor**

Atunci când un utilizator se autentifică cu succes, o înregistrare nouă este inserată în tabela sessions a bazei de date. Această înregistrare conține ID-ul utilizatorului și identificatorul unic al sesiunii, indicând faptul că utilizatorul este autentificat și sesiunea este activă.

if (pg\_num\_rows($result) > 0) {

// Authentication successful

$sessionId = ipToConcatenatedString(get\_client\_ip());

// Retrieve user ID from the PostgreSQL table

$getUserSql = "SELECT id FROM users WHERE username = '$newUsername'";

$userResult = pg\_query($dbconn, $getUserSql);

$userData = pg\_fetch\_assoc($userResult);

$userId = $userData['id'];

// Update the PostgreSQL row in the sessions table

$updateSql = "INSERT INTO sessions (session\_id, user\_id, is\_active) VALUES ('$sessionId', '$userId', true)";

$updateResult = pg\_query($dbconn, $updateSql);

}

1. Verificarea rezultatelor: Linia if (pg\_num\_rows($result) > 0) verifică dacă interogarea (query) către baza de date a returnat cel puțin o înregistrare. Aceasta este o verificare a succesului autentificării.
2. Generarea identificatorului de sesiune: Pentru a crea un identificator unic al sesiunii, se folosește funcția ipToConcatenatedString(get\_client\_ip()). Aceasta presupune că fiecare sesiune are un identificator format din adresa IP a utilizatorului și alte informații relevante, concatenându-le și generând astfel un identificator unic.
3. Obținerea ID-ului utilizatorului autentificat: Prin intermediul unei interogări către tabela "users" din baza de date, se caută ID-ul utilizatorului autentificat, bazându-se pe numele de utilizator ($newUsername). Rezultatul este stocat în variabila $userId.
4. Actualizarea tabelei de sesiuni: Se construiește o interogare SQL pentru a introduce o nouă înregistrare în tabela "sessions" a bazei de date. Această înregistrare conține identificatorul unic al sesiunii ($sessionId), ID-ul utilizatorului ($userId) și un indicator că sesiunea este activă (true).

**Gestionarea sesiunilor**

Sesiunile utilizatorilor sunt gestionate în mod corespunzător pentru a asigura securitatea și confidențialitatea datelor lor. Aceasta include actualizarea stării sesiunilor, de exemplu, atunci când utilizatorii se autentifică sau se deconectează.

Prin implementarea acestui sistem de sesiuni, asigurăm o gestionare eficientă și sigură a autentificării utilizatorilor în aplicația noastră, garantând în același timp confidențialitatea și securitatea datelor lor.

Aceste sesiuni contribuie la menținerea utilizatorului conectat în cadrul aplicației tale. Atunci când un utilizator se autentifică cu succes, o nouă înregistrare este adăugată în tabela de sesiuni a bazei de date, indicând faptul că utilizatorul este autentificat și că sesiunea este activă

Aceasta permite aplicației să țină evidența utilizatorilor autentificați și să le ofere acces continuu la funcționalitățile sistemului fără a fi necesară reautentificarea la fiecare accesare.

Pentru a oferi utilizatorului posibilitatea de a se deloga în mod controlat, am implementat un buton dedicat de "logout". Atunci când utilizatorul dorește să se delogheze, acțiunea asociată butonului de logout va declanșa o solicitare către server pentru a șterge sesiunea corespunzătoare din baza de date.

Acest lucru se realizează printr-o altă interogare SQL care șterge înregistrarea corespunzătoare din tabela de sesiuni bazată pe identificatorul unic al sesiunii.

<?php

session\_start();

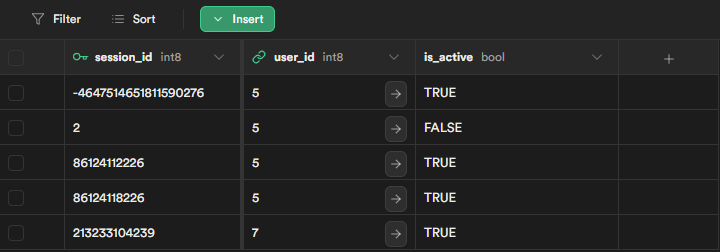
session\_unset();

session\_destroy();

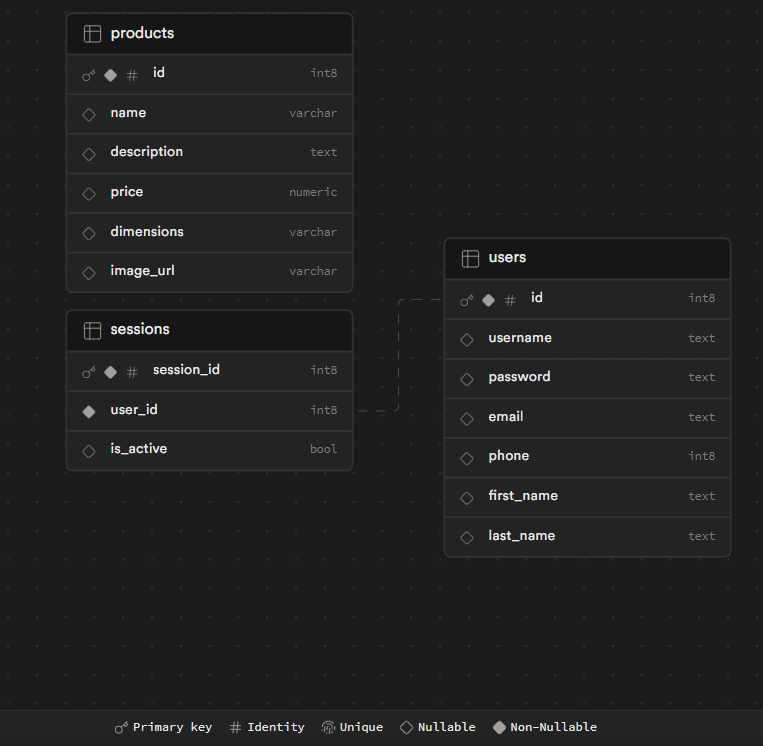
header("Location: index.php");

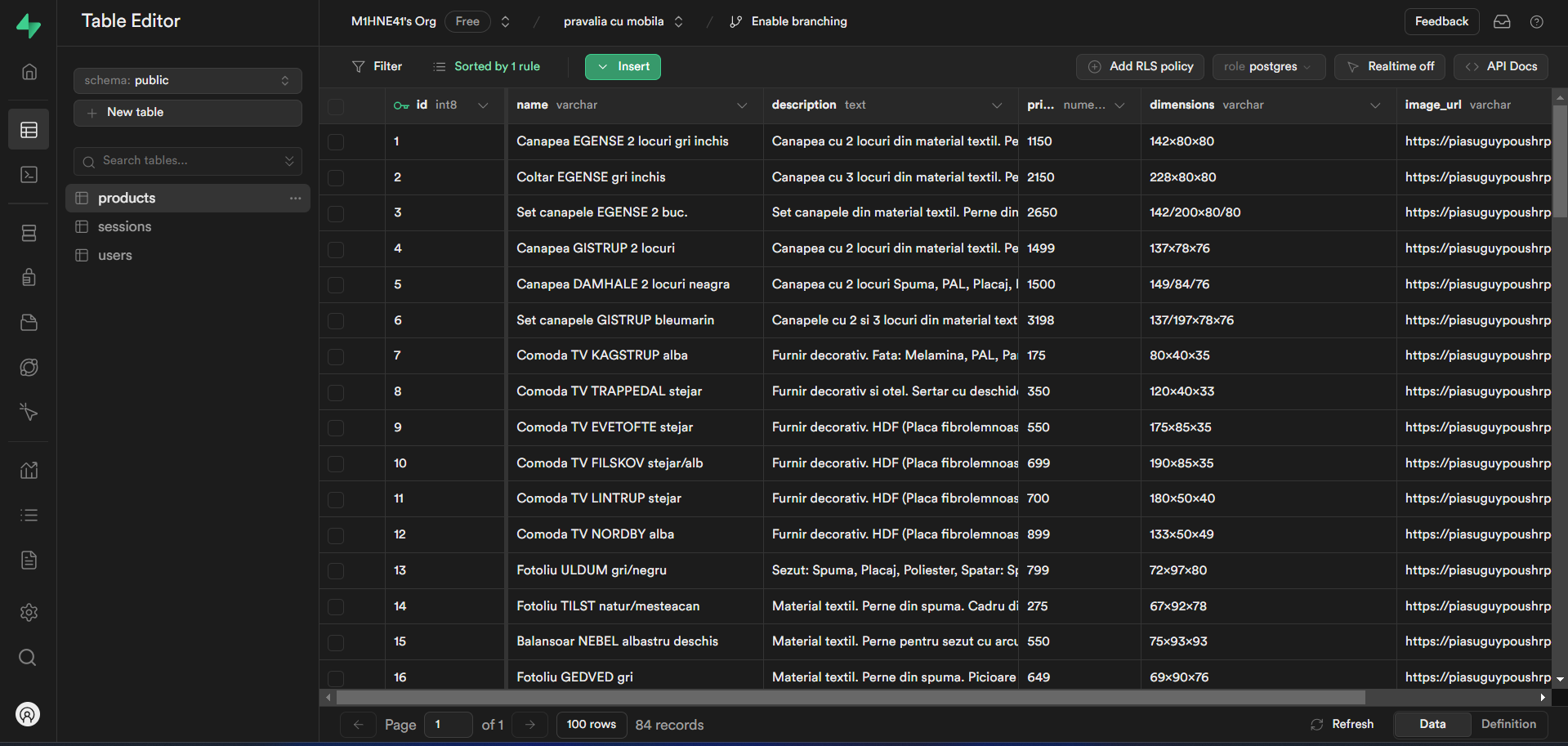
exit();

?>



Baza de date: Schema bazei de date include trei tabele: user, products și sessions. Tabela user conține informații despre utilizatori, tabela products stochează informații despre fiecare produs, iar tabela sessions gestionează sesiunile de utilizator.





Conexiunea între site și server este crucială pentru funcționarea corectă a aplicației mele. În acest caz, site-ul meu este hostat pe Vercel și baza de date este pe Supabase.

**Definirea conexiunii la baza de date:**

**$dbconn = pg\_connect("host=... port=... dbname=... user=... password=...")**

**or die('Could not connect: ' . pg\_last\_error());**

Linia de cod stabilește o conexiune la baza de date folosind PostgreSQL. pg\_connect este o funcție PHP care se conectează la un server PostgreSQL. Parametrii din interiorul funcției sunt host, port, db name, user, password.

**Procesarea datelor trimise de utilizator pentru înregistrare (signup)**:

if(isset($\_POST['save']))

{

$newUsername = $\_POST['new\_username'];

$newPassword = $\_POST['new\_password'];

$newEmail = $\_POST['email'];

$newPhoneNumber = $\_POST['phone'];

$newFirstName = $\_POST['first\_name'];

$newLastName = $\_POST['last\_name'];

$sql\_query = "INSERT INTO users (username, password, email, phone, first\_name, last\_name)

VALUES ('$newUsername', '$newPassword', '$newEmail', '$newPhoneNumber', '$newFirstName', '$newLastName')";

$result = pg\_query($dbconn, $sql\_query);

if ($result)

{

include 'signup-popup.html';

}

else

{

echo "Error: " . pg\_last\_error($dbconn);

}

pg\_close($dbconn);

}

1. Verificăm dacă s-au trimis datele pentru înregistrare (isset($\_POST['save'])).
2. Dacă sunt date trimise, le preluăm din formular (username, password, email, telefon, prenume, nume).
3. Construim o interogare SQL ($sql\_query) pentru a insera aceste date în baza de date.
4. Folosim funcția pg\_query pentru a executa interogarea pe conexiunea la baza de date ($dbconn).
5. Verificăm rezultatul executării interogării:
6. Dacă este un succes ($result), includem un fișier HTML pentru a afișa un mesaj de succes sau un popup de înregistrare.
7. Dacă apare o eroare, o afișăm

(echo "Error: " . pg\_last\_error($dbconn)).

**Login**

Pentru a asigura securitatea și integritatea datelor, am implementat următoarele măsuri de securitate în cadrul codului meu:

Utilizarea interogărilor pregătite: Pentru a preveni injecția SQL și pentru a asigura că datele introduse de utilizatori sunt tratate în mod corespunzător, am utilizat interogări pregătite. Acestea permit separarea datelor de intrare de comanda SQL și furnizarea parametrilor în mod sigur.Fără utilizarea interogărilor pregătite, datele introduse de utilizatori, cum ar fi numele de utilizator și parola, ar fi fost concatenate direct în interogările SQL. Aceasta ar fi expus aplicația la atacuri de injecție SQL.

Un atacator ar fi putut introduce date special construite în câmpurile de autentificare pentru a modifica comportamentul interogării SQL și pentru a obține acces neautorizat la datele din baza de date. În absența interogărilor pregătite, datele introduse de utilizatori ar fi fost tratate ca parte a comenzilor SQL, fără nicio verificare sau prelucrare specială. Aceasta ar fi făcut aplicația vulnerabilă la injecția SQL, care ar fi putut fi exploatată pentru a compromite baza de date și pentru a obține acces neautorizat la datele utilizatorilor.

În cadrul dezvoltării aplicației mele, am efectuat o serie de teste pentru a evalua securitatea și robustețea acesteia. Unul dintre aspectele cheie pe care le-am investigat a fost vulnerabilitatea la atacuri de injecție SQL, o problemă comună care poate expune datele sensibile și poate afecta integritatea aplicației noastre.

Pentru a testa această vulnerabilitate, am simulat un scenariu de injecție SQL în care am introdus manual comanda ' OR '1'='1' -- în câmpurile de autentificare (username și parolă) ale formularului de conectare.

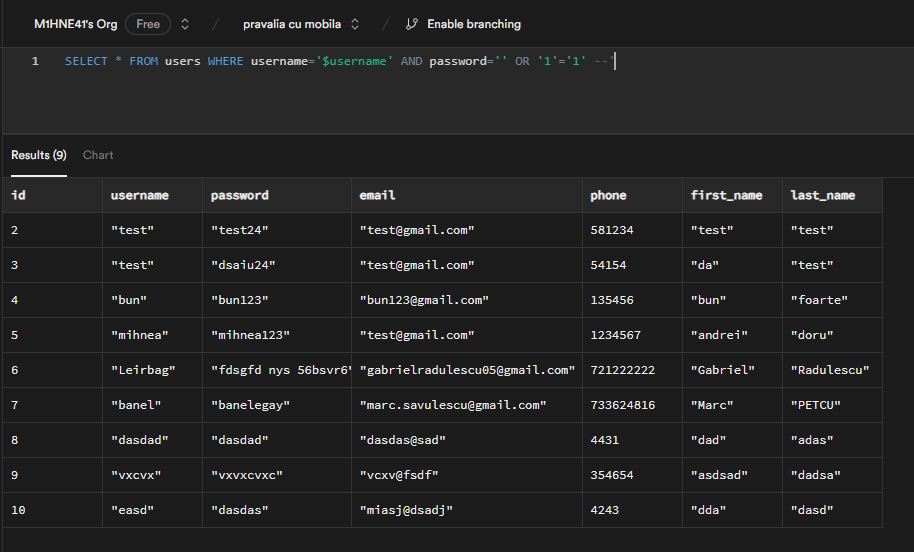
SELECT \* FROM users WHERE username='$username' AND password='' OR '1'='1' --'

În această situație, clauza OR '1'='1' va fi întotdeauna adevărată, ignorând parola și permițând accesul neautorizat la conturi de utilizator.

Această comandă este cunoscută pentru a forța întotdeauna condiția să fie adevărată, permițând accesul neautorizat la datele din baza de date.

Rezultatul acestui test a fost că am putut accesa cu succes un cont din baza de date, fără a fi necesară introducerea unei parole valide. Acest lucru demonstrează că aplicația noastră a fost vulnerabilă la injecția SQL și că datele introduse de utilizator nu au fost corect izolate de comanda SQL.

Prin implementarea acestor măsuri de securitate, mă asigur că aplicația mea este protejată împotriva atacurilor de injecție SQL și că datele utilizatorilor sunt tratate în mod sigur și corect.

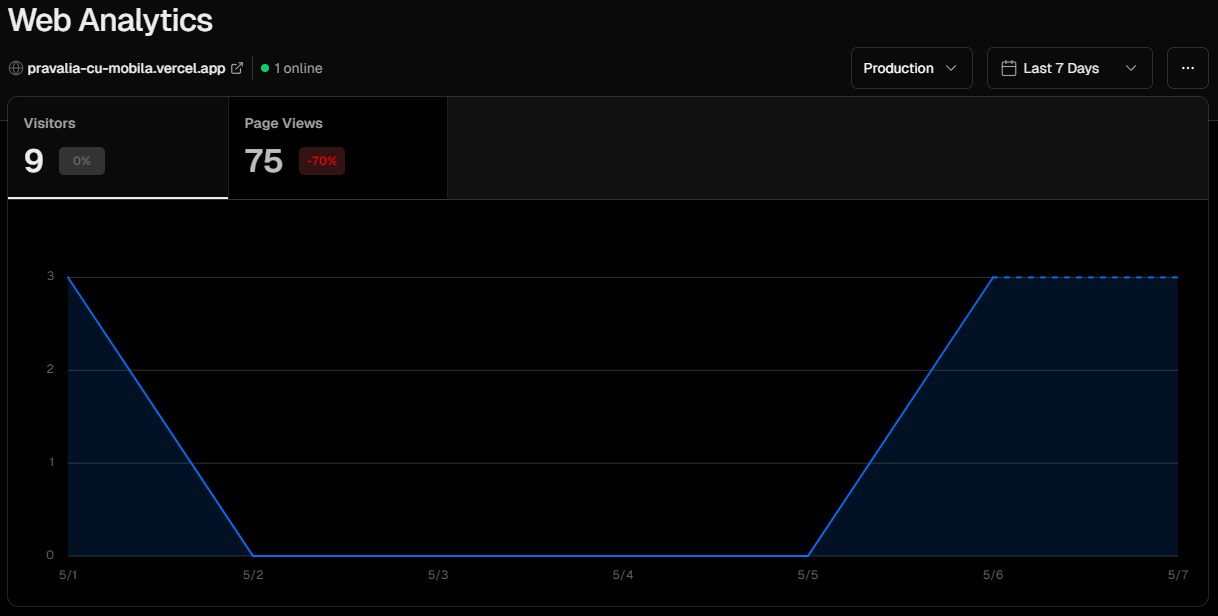


**Sistemul de Date Analitice Vercel:**

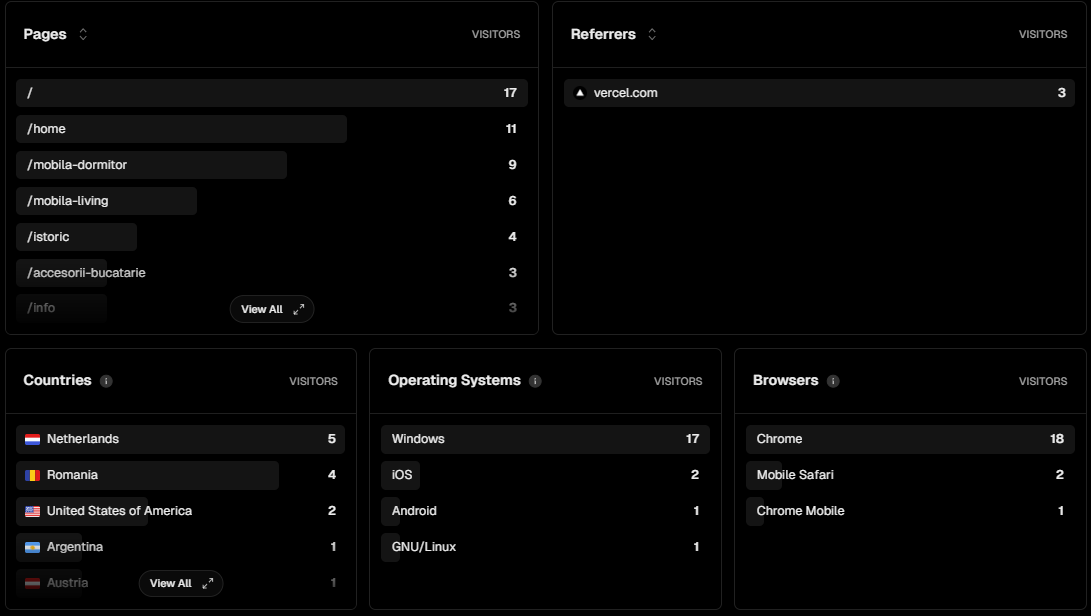
Sistemul de date analitice oferit de Vercel este o componentă esențială pentru înțelegerea și optimizarea performanței site-ului meu web. Această platformă furnizează o gamă diversă de instrumente și metrici pentru a monitoriza și analiza comportamentul utilizatorilor, performanța paginilor și alte aspecte relevante pentru experiența utilizatorului.

Principalele caracteristici ale sistemului de date analitice Vercel includ:

Monitorizarea traficului: Vercel oferă informații detaliate despre traficul site-ului, inclusiv numărul de vizitatori, pagini vizitate, ședințe și alte aspecte relevante.



Analiza performanței: Prin intermediul sistemului de date analitice, putem evalua performanța site-ului în timp real sau în perioade prestabilite. Acest lucru include metrici precum timpul de încărcare al paginilor, timpul de răspuns al serverului și altele.



Segmentarea utilizatorilor: Putem segmenta utilizatorii în funcție de diferite criterii, precum locația geografică, dispozitivul folosit sau alte atribute relevante. Această funcționalitate permite personalizarea și îmbunătățirea experienței utilizatorului în funcție de nevoile și preferințele acestora.

Integrare cu alte instrumente: Sistemul de date analitice Cercel poate fi integrat cu alte instrumente și platforme, precum Google Analytics, pentru a oferi o imagine completă și detaliată asupra performanței și comportamentului utilizatorilor.

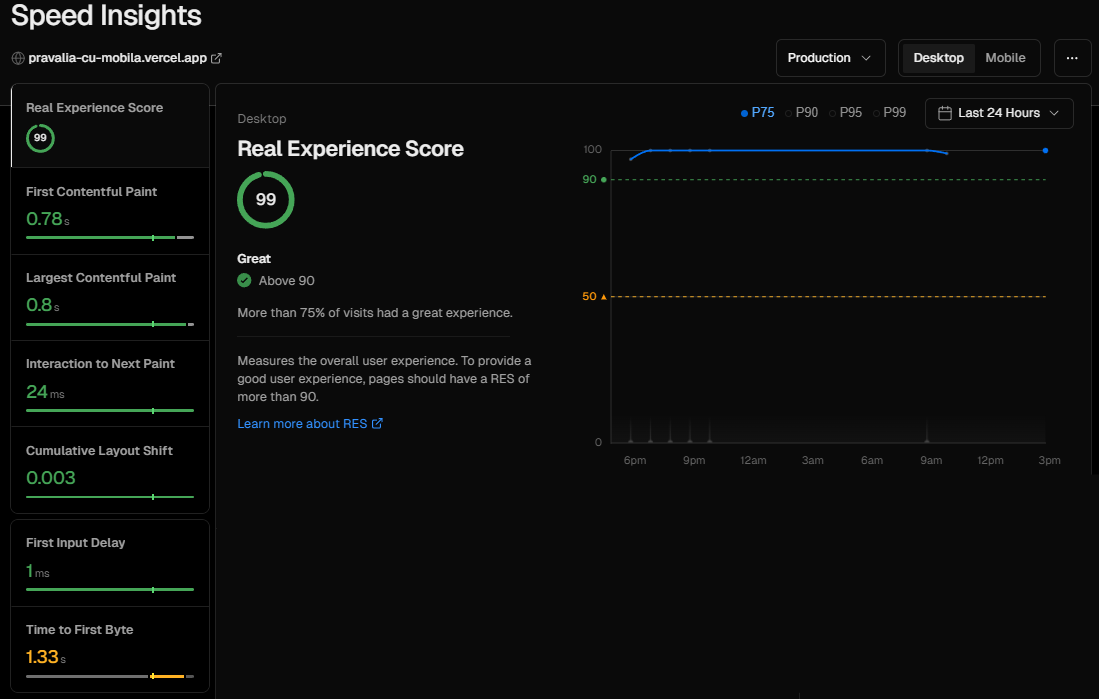
**Speed Insights:**

Categoria Speed Insights este o măsură a performanței site-ului în ceea ce privește viteza de încărcare și timpul de răspuns. Această metrică este esențială pentru a asigura o experiență plăcută și eficientă utilizatorilor și pentru a optimiza clasamentul site-ului în motoarele de căutare.

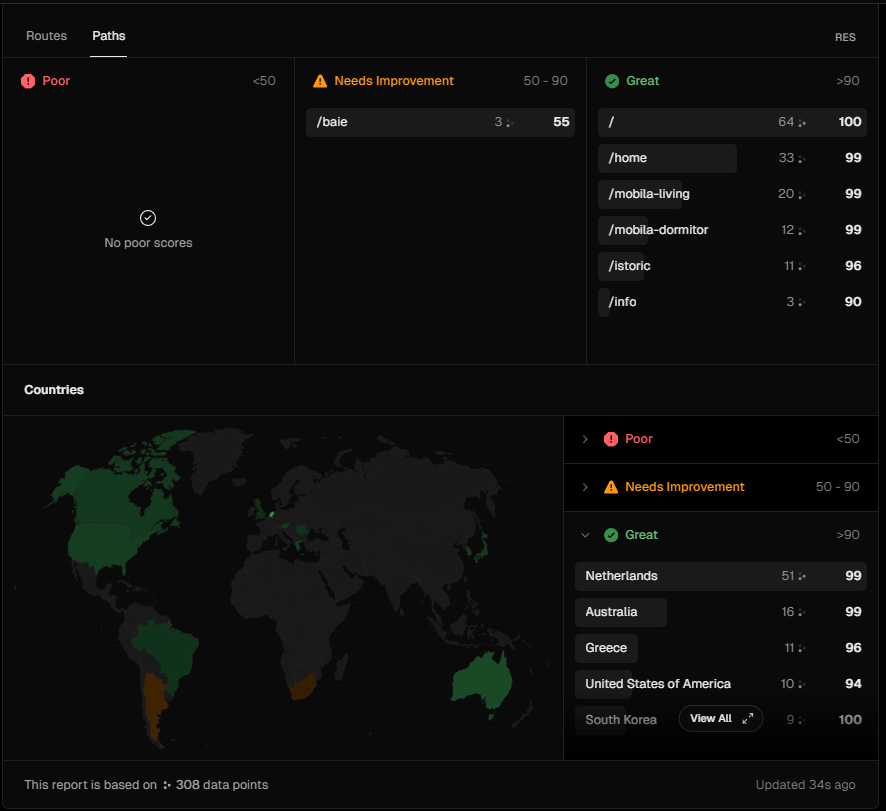
Principalele aspecte ale categoriei Speed Insights includ:

Viteza de încărcare a paginilor:

Speed Insights evaluează timpul necesar pentru încărcarea completă a paginilor site-ului meu web. Viteza optimă de încărcare este crucială pentru menținerea angajamentului utilizatorilor și reducerea ratei de abandon a paginilor.

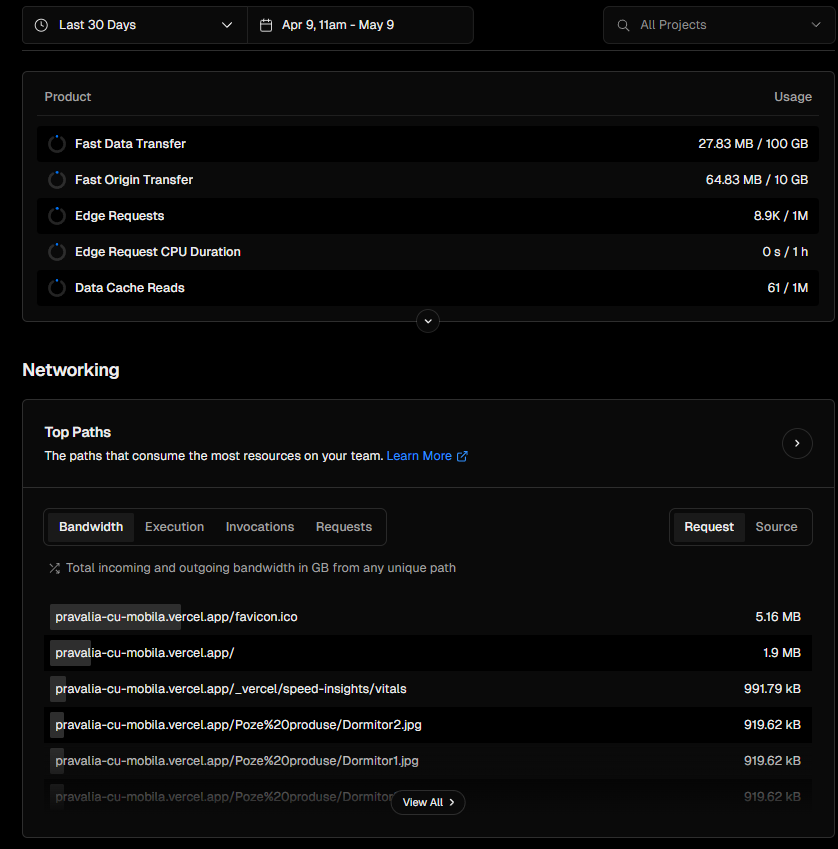


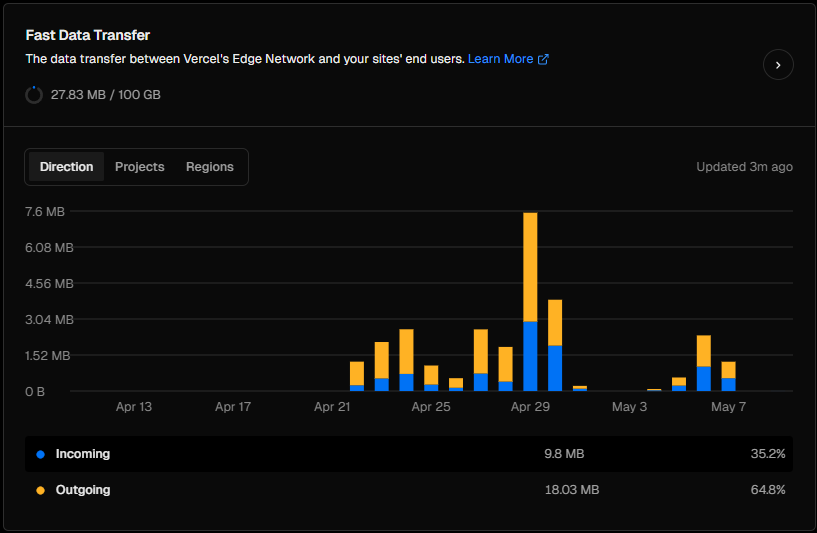
Optimizarea elementelor media: Analiza Speed Insights include evaluarea și recomandări pentru optimizarea imaginilor, videoclipurilor și altor elemente media pentru a reduce timpul de încărcare al paginilor.



Compatibilitate cu dispozitivele mobile: Evaluarea performanței site-ului pe dispozitive mobile este un aspect crucial al categoriei Speed Insights, având în vedere creșterea utilizării mobile în rândul utilizatorilor. Un site web optimizat pentru dispozitivele mobile are un timp de încărcare rapid și o experiență fluidă de navigare.

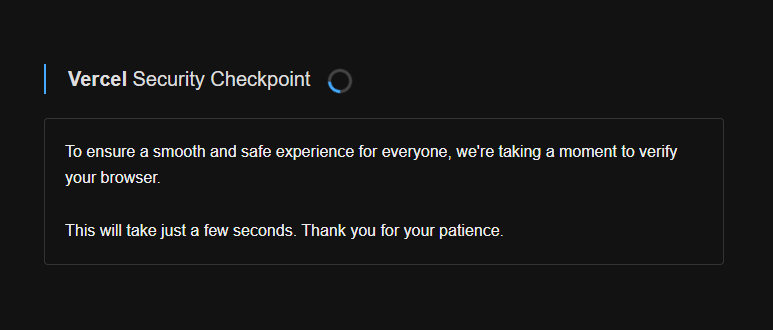
Caching și comprimare: Categoria Cache oferă sfaturi și indicații pentru implementarea eficientă a tehnologiilor de caching și comprimare a fișierelor pentru a accelera încărcarea paginilor și a reduce timpul de răspuns al serverului.





Funcții de securitate anti-DDoS

Attack Challenge Mode reprezintă o modalitate de a oferi mai mult control în cazul atacurilor DDoS. Atunci când este activat, toți vizitatorii site-ului vor vedea un test de verificare înainte de a li se permite accesul. Firewall-ul automat al Vercel oprește automat atacurile DDoS a asigura că tot traficul către site este legitim.



**Bibliografie**

[**https://www.php.net/docs.php**](https://www.php.net/docs.php)

[**https://developer.mozilla.org/en-US/docs/Web/CSS**](https://developer.mozilla.org/en-US/docs/Web/CSS)

[**https://www.w3schools.com/css/**](https://www.w3schools.com/css/)

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

[**https://vercel.com/docs**](https://vercel.com/docs)

[**https://supabase.com/docs**](https://supabase.com/docs)

[**https://docs.github.com/en**](https://docs.github.com/en)

[**https://www.winterwind.com/tutorials/css/15**](https://www.winterwind.com/tutorials/css/15)

[**https://codepen.io/TheMOZZARELLA/pen/oNpMxyy**](https://codepen.io/TheMOZZARELLA/pen/oNpMxyy)

[**https://fonts.google.com/selection/embed**](https://fonts.google.com/selection/embed)