Permise și înmatriculări – Website

Realizat de Batalan Vlad, clasa a XII-a D, Colegiul Național “A.T.Laurian”

1. **Secțiunea:**

- Web

1. **Autor:**

- Batalan Vlad, clasa a XII-a D, Colegiul Național “A.T.Laurian”

1. **Limbaje de programare web folosite:**

- PHP (server Apache, folosind programul XAMPP v3.2.2)

- HTML

- CSS

- BOOTSTRAP (front-end Framework)

- JavaScript

- jQuery

- MySqli database, folosind PhpMyAdmin pentru gestionarea bazei de date

- Sql

1. **Descriere site:**

a) Descriere generală

- Serviciul de Permise și înmatriculări a devenit din ce în ce mai îngrămădit și mai utilizat de către cetățeni, iar sistemul bazat pe bilete de ordine este învechit, nepractic și generează multe conflicte între cei care doresc să-și termine de făcut actele cât mai repede, ori abia așteaptă momentul când vor obține permisul după 18 ani de așteptare (de la naștere până în prezent).

- Personalul fiind puțin, programările pentru examenul, atât teoretic, cât și practic, pentru permis, dar și simplele înmatriculări, radieri, etc. au ajuns să fie decalate chiar și cu patru luni.

- Realizat după cerințele serviciului de Permise și înmatriculări Botoșani, website-ul îndeplinește cu success sarcina de a gestiona programările din partea utilizatorilor și de a face cât mai eficient și user-frindly acest process, oferind totodată ușurință angajaților de la Permise și înmatriculări să vizualizeze, sa listeze programările zilnice și să schimbe fără efort unele procese de pe site.

b) Grupul țintă

- Website-ul este realizat spre a fi folosit atât de masele de cetățeni care au nevoie de anumite procese în cadrul serviciului Permise și înmatriculări și doresc să-și facă programare, cât și de angajații acestei instituții pentru a vizualiza și printa programările zilnice, dar și pentru a schimba parametrii unor procese de pe platformă.

c) Funcționalitate

- Utilizatorii pot vizualiza calendarul și-și pot alege ziua în care vor să-și facă programarea.

- Administratorul poate gestiona aceste programări, le poate afișa sub diferite criterii și le poate printa pentru a cunoaște ordinea persoanelor care trebuie sa apară la ghișeu în ziua respectivă.

d) Alte facilități

- Website-ul este realizat folosind Framework-ul Bootstrap, făcându-l responsive.

- Platforma este ergonomică, în sensul în care se poate ajunge pe orice pagină dintr-un număr minim de click-uri

- Permise și înmatriculări se află în momentul de față într-o variantă minimalistă, dar care poate fi îmbunătățită în foarte multe direcții precum: posibilitatea administratorului de a schimba datele de sărbătoare, programul zilnic specific pentru diferite proceduri, un sistem de înregistrare pentru administratori dotat cu diferite nivele de accesibilitate, posibilitatea utilizatorului de a specifica ora exactă la care dorește să vină la programare, adaptabilitatea site-ului si pentru alte județe și multe altele.

1. **Porțiuni de cod care au dat dovadă de dificultate mai mare**

a) Realizarea calendarului unde utilizatorul poate bifa data programării, dar poate afla și ora

<div class="tab-content">

<!-- Tabel in care putem alege direct data in care vrem sa facem rezervare -->

<?php

//vom genera datele disponibile pentru urmatoarele 3 luni

//setam punctul de start pe care il ducem in prima zi a lunii curente

$MonthStart = date("Y-m-d", strtotime($today));

// setam o data pe care o ducem cu un while pana in prima zi

$StartDate = date("Y-m-d", strtotime($MonthStart));

while(Get\_Month($StartDate) === Get\_Month($MonthStart))

{

$StartDate = Add\_k\_Days($StartDate, "Y-m-d", -1);

}

$MonthStart = Add\_k\_Days($StartDate, "Y-m-d", 1);

//fixam data cand sa se termine

$MonthEnd = Add\_k\_Months($MonthStart, "Y-m-d", 5);

$MonthEnd = Add\_k\_Days($MonthEnd, "Y-m-d", -1);

// petru fiecare luna din acest interval, vom afisa calendar

for($CurrentMonth = date("Y-m-d", strtotime($MonthStart)), $MonthIndex = 1; $CurrentMonth < $MonthEnd; $CurrentMonth = Add\_k\_Months($CurrentMonth,"Y-m-d", 1), $MonthIndex++)

{

// $MonthIndex o folosesc pentru a imi arata a cata luna sunt, pentru a denumi meniurile

// afisam numele lunii

echo "

<div id='menu$MonthIndex' class='tab-pane fade'>

<div class='table-responsive'>

<table class='table table-bordered date-table'>

<tr>

<th colspan=7> ".($MonthNames[intval(date("m", strtotime($CurrentMonth)))-1])."

</th>

</tr>";

echo "

<tr>

<th>Du</th>

<th>Lu</th>

<th>Ma</th>

<th>Mi</th>

<th>Jo</th>

<th>Vi</th>

<th>Sa</th>

</tr>

"

;

//generam zilele din calendar in functie de prima zi unde se afla

$CurrentWeekDay = date("w", strtotime($CurrentMonth));

$CurrentIndexDate = date("Y-m-d", strtotime($CurrentMonth));

$IndexDay = 0; //index day tine pasul si pune <tr> si </tr> cand se termina un rand

for($i = 0; $i < $CurrentWeekDay; $i++) // mergem pana la prima aparitie a zilei de 1 din luna current month

{

echo "<td></td>";

$IndexDay ++;

}

// cat timp inca sunt in aceeasi luna, scrie date

while(Get\_Month($CurrentIndexDate) == Get\_Month($CurrentMonth))

{

if($IndexDay == 7) //daca am ajuns la capatul saptamanii

{

$IndexDay = 0;

echo "</tr><tr>";

}

$avanable\_state = ""; // memoreaza daca trebuie pus clasa not-avanable-date

$de\_la\_ora = 0; //notam de la ora cu o valoare default

// verificam daca exista ziua respectivă in program, in caz contrar o dezactivam

$sql = "SELECT \* FROM program\_zilnic WHERE indice\_zi = '".$IndexDay."' AND tip\_procedura = '".$procedura\_value['tip\_procedura']."'";

$query = mysqli\_query($conn, $sql);

if(mysqli\_num\_rows($query) == 0) //nu am gasit ziua

$avanable\_state = "class='not-avanable-date'";

else

{

//am gasit, verific daca mai este timp de o programare aici

//initializez variabile necesare

$program\_astazi = mysqli\_fetch\_assoc($query);

$de\_la\_ora = $program\_astazi['de\_la\_ora'];

$pana\_la\_ora = $program\_astazi['pana\_la\_ora'];

//selectam programarea din data respectiva cu timpul de programare cel mai mare daca exista

$sql = "SELECT `ora\_programare`, `id\_procedura` FROM `programari` WHERE `data\_programare` = '".date("Y-m-d", strtotime($CurrentIndexDate))."' AND `tip\_procedura` = '".$procedura\_value['tip\_procedura']."'";

$query = mysqli\_query($conn, $sql);

//setez $max\_time ca fiind elementul gasit cu cel mai mare timp

$max\_time = array();

$max\_time['ora\_programare'] = $de\_la\_ora;

//caut manual elementul pt ca m am complicat cu MAX(ora\_programare) si mi a facut 2 ore fite, si asa ma grabesc ca am maine deadline si csf

while($row = mysqli\_fetch\_assoc($query))

{

if($row['ora\_programare'] >= $max\_time['ora\_programare'])

{

$max\_time['ora\_programare'] = $row['ora\_programare'];

$max\_time['id\_procedura'] = $row['id\_procedura'];

}

}

//daca exista programare

if(isset($max\_time['id\_procedura']))

// daca am gasit cel putin o programare ii voi cauta data de terminare

//adaug la ora maxima timpul terminarii procedurii

{

$sql = "SELECT timp\_procedura FROM proceduri WHERE id\_procedura = '".$max\_time['id\_procedura']."'";

$query = mysqli\_query($conn, $sql);

$timp\_max\_procedura = mysqli\_fetch\_assoc($query);

//echo "<script>alert('data: ".$CurrentIndexDate."; de la ora: ".$timp\_max\_procedura['timp\_procedura']."');</script>";

}

else{

$timp\_max\_procedura = array("timp\_procedura" => 0);

}

//upgradam $de\_la\_ora

$de\_la\_ora = $max\_time['ora\_programare'] + $timp\_max\_procedura['timp\_procedura'];

//calculam daca pote fi incadrata in interval timpul realizarii procedurii

if($de\_la\_ora + $procedura\_value['timp\_procedura'] > $pana\_la\_ora)

{

//echo "<script>alert('Gasit inregistrare! Data: $CurrentIndexDate');</script>";

//nu se poate incadra, deci anulam rezervarea

$avanable\_state = "class='not-avanable-date'";

}

}

//daca este o data mai devreme de ziua de astazi, nu luam in considerare

if($CurrentIndexDate < $today)

{

//nu se mai poate face rezervare

$avanable\_state = "class='not-avanable-date'";

}

//daca este o sarbatoare nu se vor face programari in ziua respectiva

$sql = "SELECT \* FROM sarbatori\_legale WHERE data\_sarbatoare = '".date("Y-m-d", strtotime($CurrentIndexDate))."'";

$query = mysqli\_query($conn, $sql);

//daca am gasit o data in baza de date, atunci nu se vor mai face programari

if(mysqli\_num\_rows($query) > 0)

$avanable\_state = "class='not-avanable-date'";

//valoarea returnata de catre radio va avea atat data programarii, cat si ora la care se face aceasta in formatul:

// [data programarii]#[ora programarii]

echo "

<td $avanable\_state>

".date("d", strtotime($CurrentIndexDate))."

<br>

<input type='radio' name='calendar' value='".date("Y-m-d", strtotime($CurrentIndexDate))."' onchange='Change\_Hour(\"$CurrentIndexDate\", \"".Mins\_to\_time($de\_la\_ora)."\");'>

</td>";

$CurrentIndexDate = Add\_k\_Days($CurrentIndexDate, "Y-m-d", 1);

$IndexDay++;

}

// umplem tabelul cu restul zilelor care lipsesc

while($IndexDay < 7)

{

echo "<td></td>";

$IndexDay ++;

}

echo "

</tr>

</div>

</table>

</div>

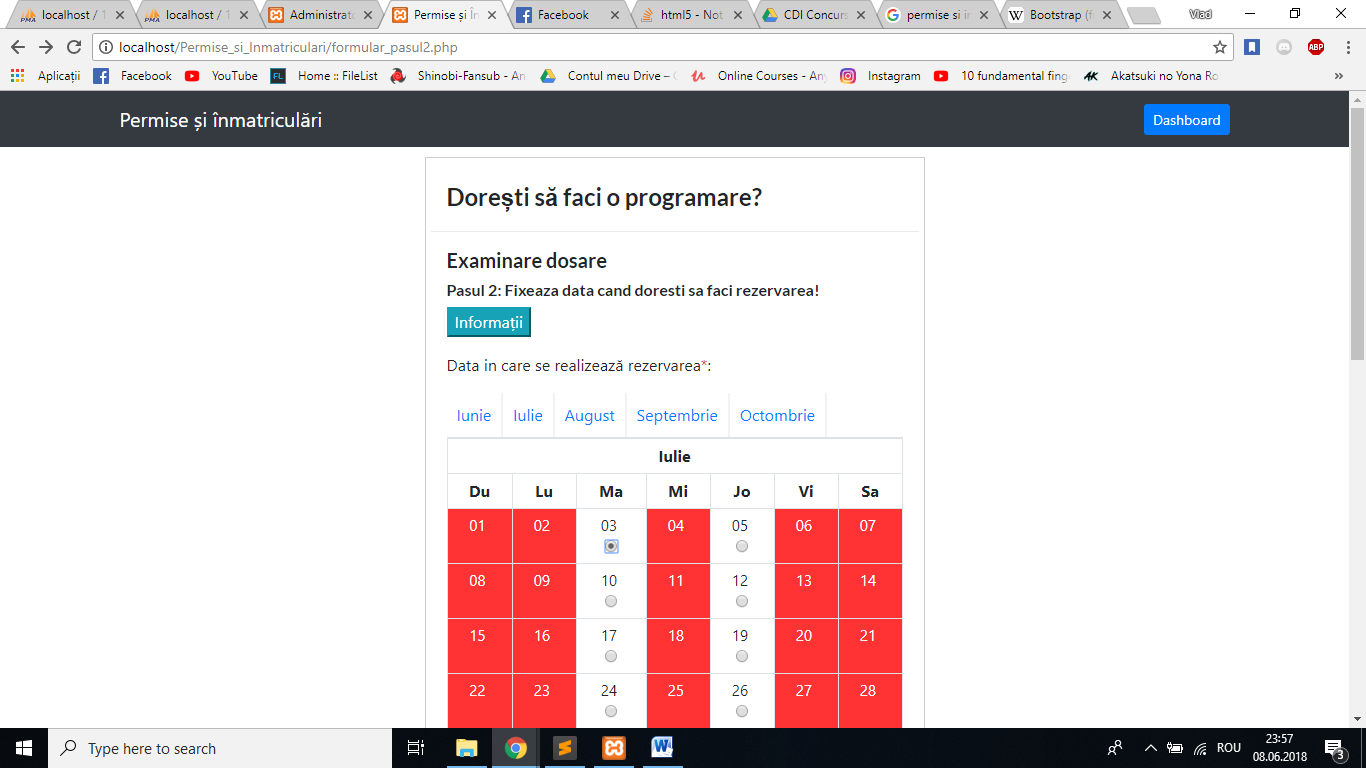
</div>

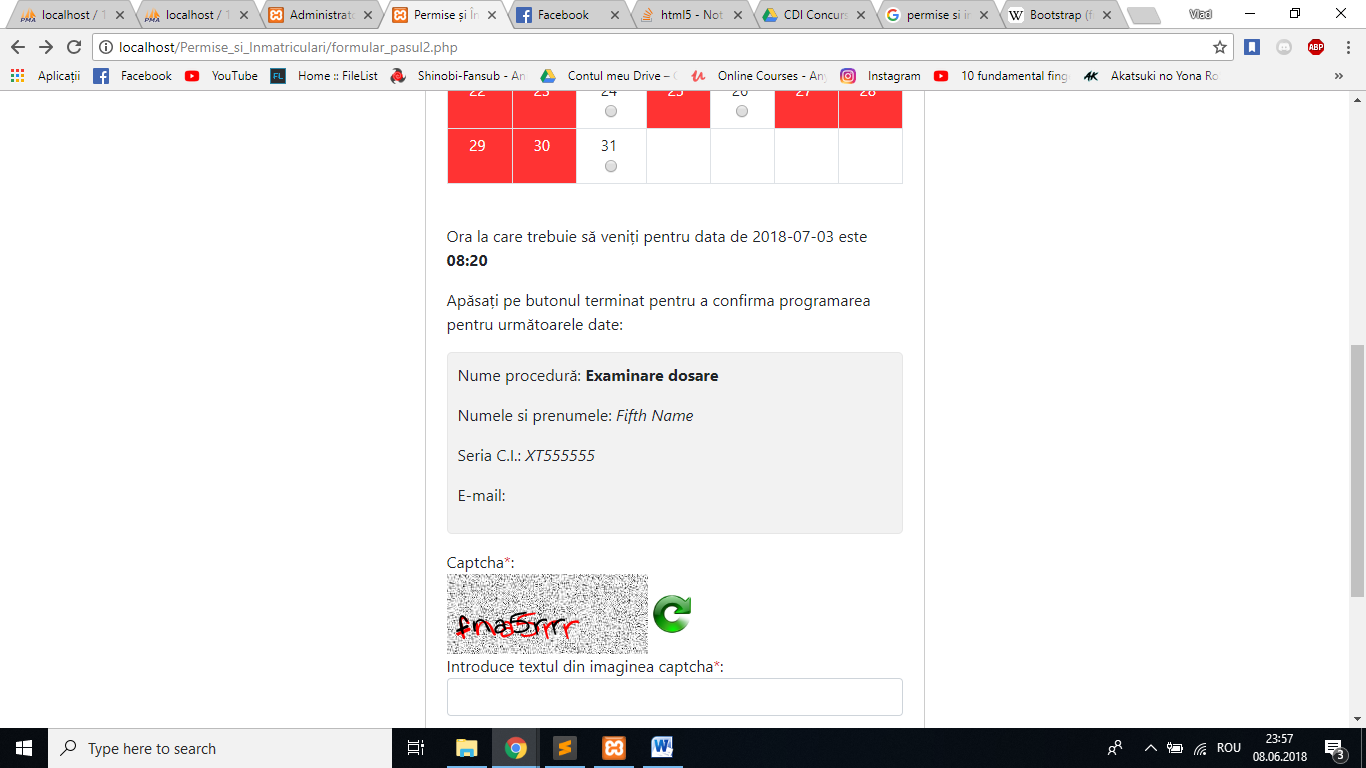
";

}

?>

</div>





b) Realizarea unei comenzi sql în urma primirii unor date prin metoda “get”

//vom prelua aici datede din search si vom compune un text sql care va cauta dupa toate criteriile

$sql\_command = "SELECT a.\*, b.descriere\_procedura FROM programari AS a, proceduri AS b WHERE a.id\_procedura LIKE b.id\_procedura"; // a. = programari, b. = proceduri

//in paralel construim un link prin care sa trimitem toate variabilele pentru a genera pe alta pagina cautarea noastra

$print\_link = "printeaza\_programari.php";

$indexlink = 0; // arata a cata componenta adaug

if(isset($\_GET['submit\_search'])) // a fost trimisa o cautare

{

// cautare dupa proceduri

if(isset($\_GET['proceduri\_search']) && !empty($\_GET['proceduri\_search']))

{

$proceduri\_search = validate\_string($\_GET['proceduri\_search']);

$sql\_command .= " AND a.id\_procedura LIKE '$proceduri\_search'";

$indexlink ++;

if($indexlink != 1)

$print\_link .= "&";

else

$print\_link .= "?";

$print\_link .= "proceduri\_search=$proceduri\_search";

}

//cautare dupa tip procedura

if(isset($\_GET['tip\_proceduri\_search']) && !empty($\_GET['tip\_proceduri\_search']))

{

$tip\_proceduri\_search = validate\_string($\_GET['tip\_proceduri\_search']);

$sql\_command .= " AND a.tip\_procedura LIKE '$tip\_proceduri\_search'";

$indexlink ++;

if($indexlink != 1)

$print\_link .= "&";

else

$print\_link .= "?";

$print\_link .= "tip\_proceduri\_search=$tip\_proceduri\_search";

}

// cautare dupa data

if(isset($\_GET['date\_search']) && !empty($\_GET['date\_search']))

{

$date\_search = date("Y-m-d", strtotime(validate\_string($\_GET['date\_search'])));

$sql\_command .= " AND a.data\_programare LIKE '$date\_search'";

$indexlink ++;

if($indexlink != 1)

$print\_link .= "&";

else

$print\_link .= "?";

$print\_link .= "date\_search=$date\_search";

}

// cautare dupa nume, cod unic, serie de sasiu

if(isset($\_GET['text\_search']) && !empty($\_GET['text\_search']))

{

$text\_search = validate\_string($\_GET['text\_search']);

$sql\_command .= " AND (a.nume\_realizator LIKE '%$text\_search%' OR a.cod\_unic LIKE '$text\_search' OR a.serie\_sasiu LIKE '$text\_search')";

$indexlink ++;

if($indexlink != 1)

$print\_link .= "&";

else

$print\_link .= "?";

$print\_link .= "text\_search=$text\_search";

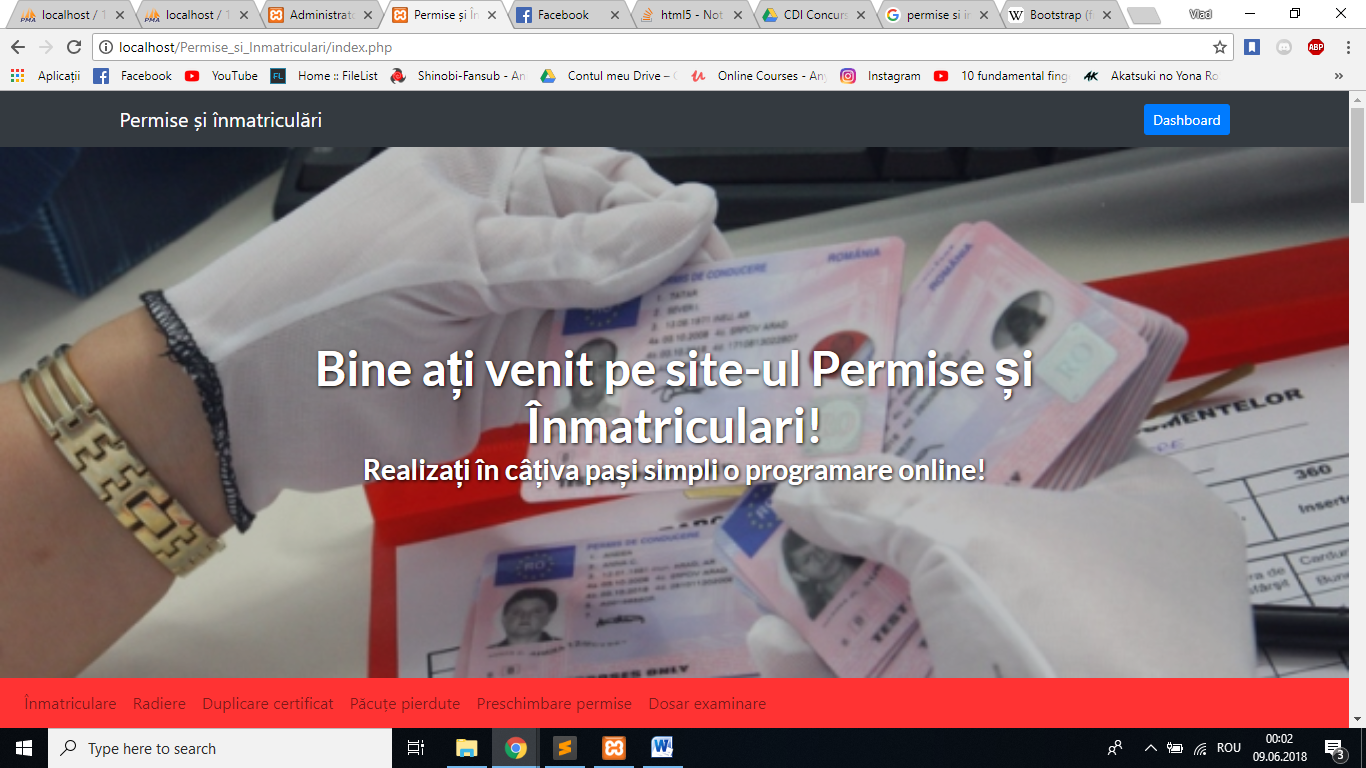
}

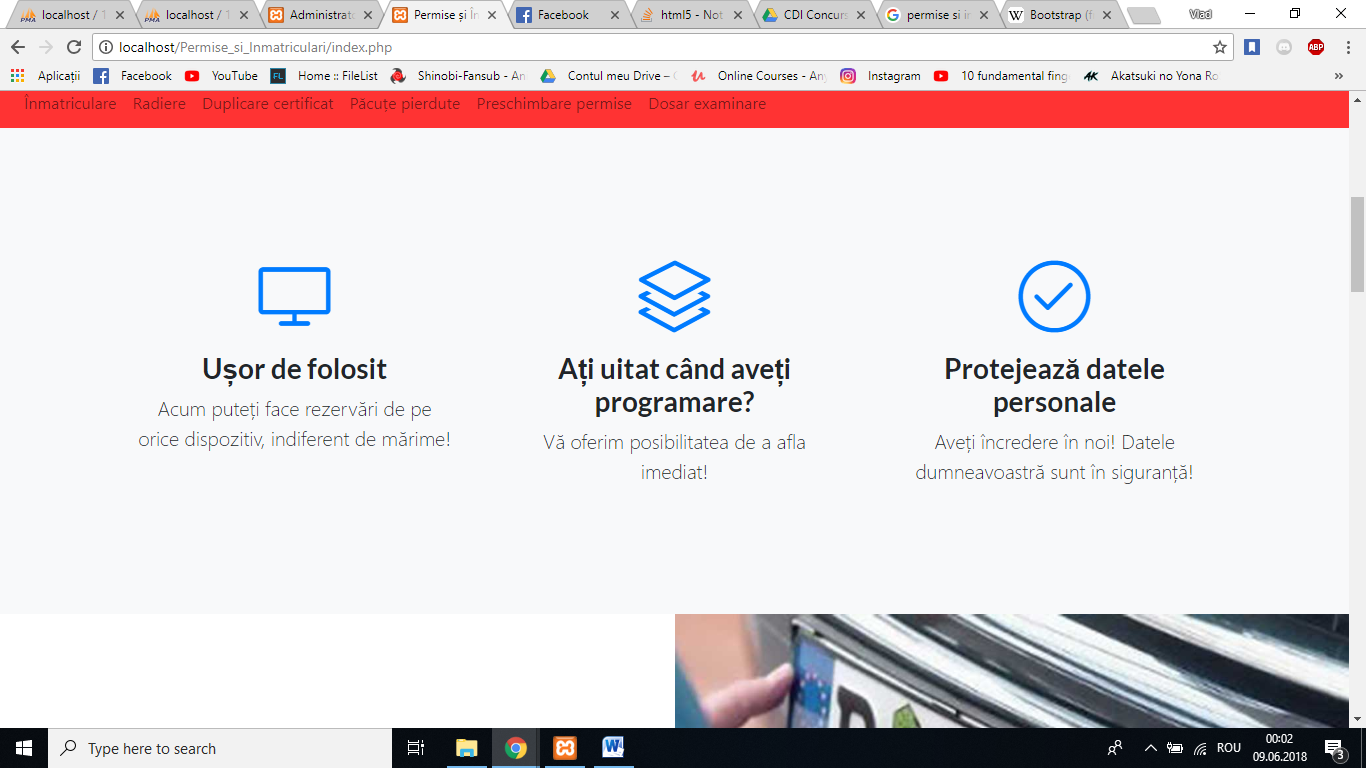
}

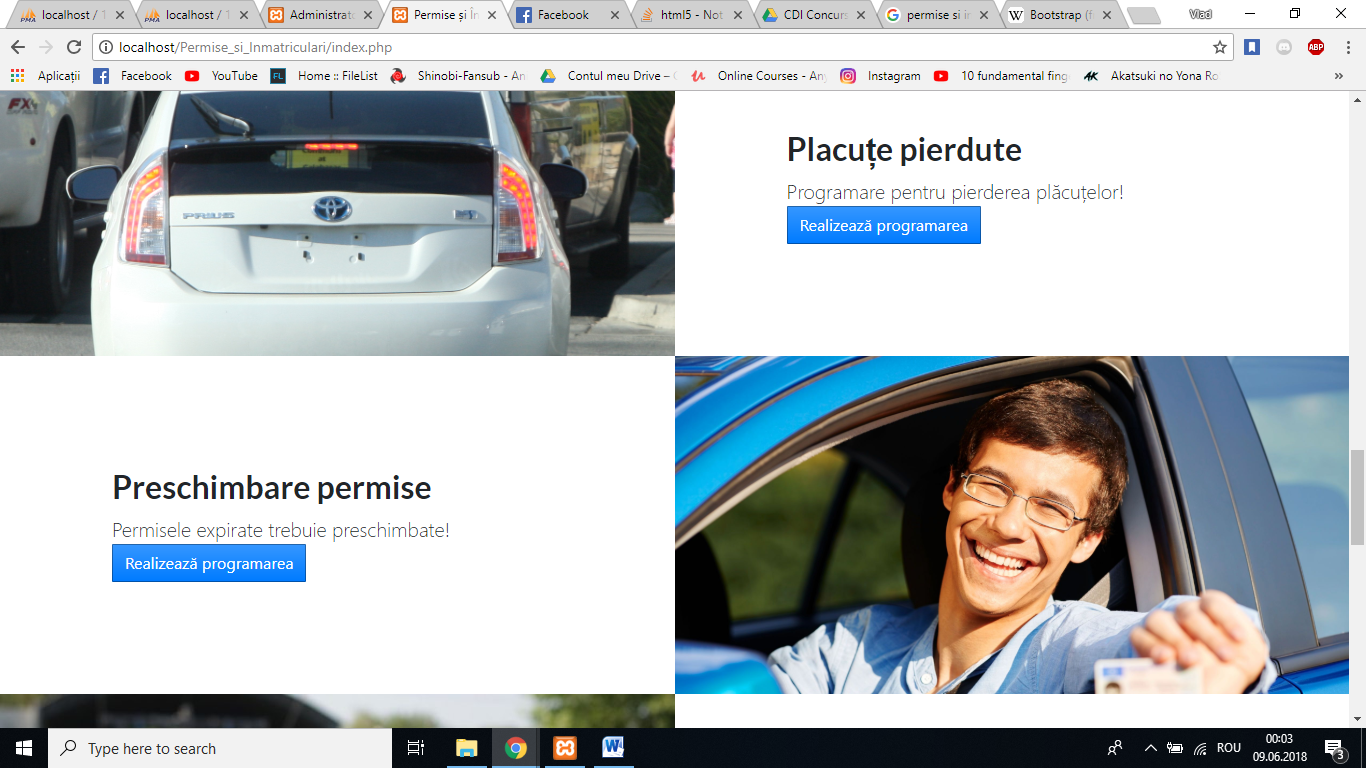
$sql\_command .= " ORDER BY data\_programare ASC, ora\_programare ASC, id\_procedura ASC, nume\_realizator ASC";

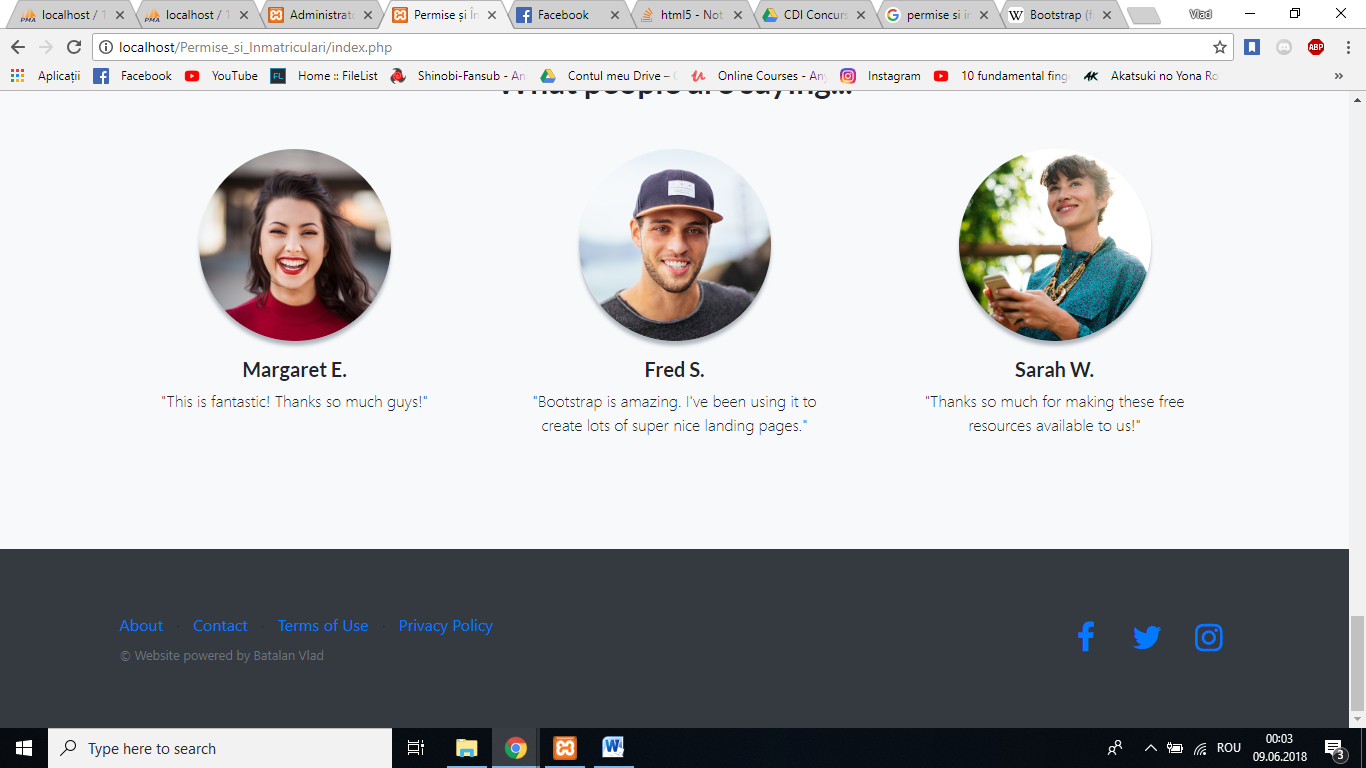
1. **Imaginea finală a site-ului**

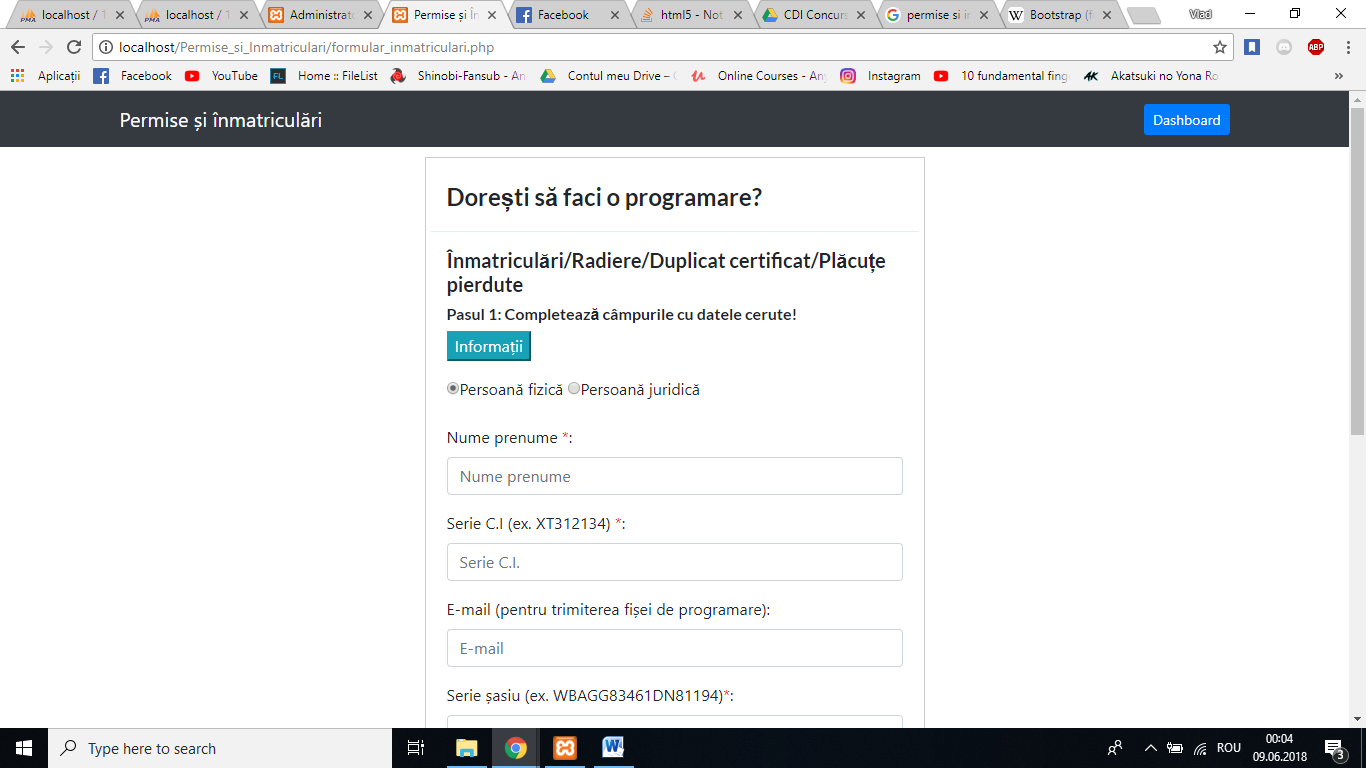
a) Interfața utilizatorului

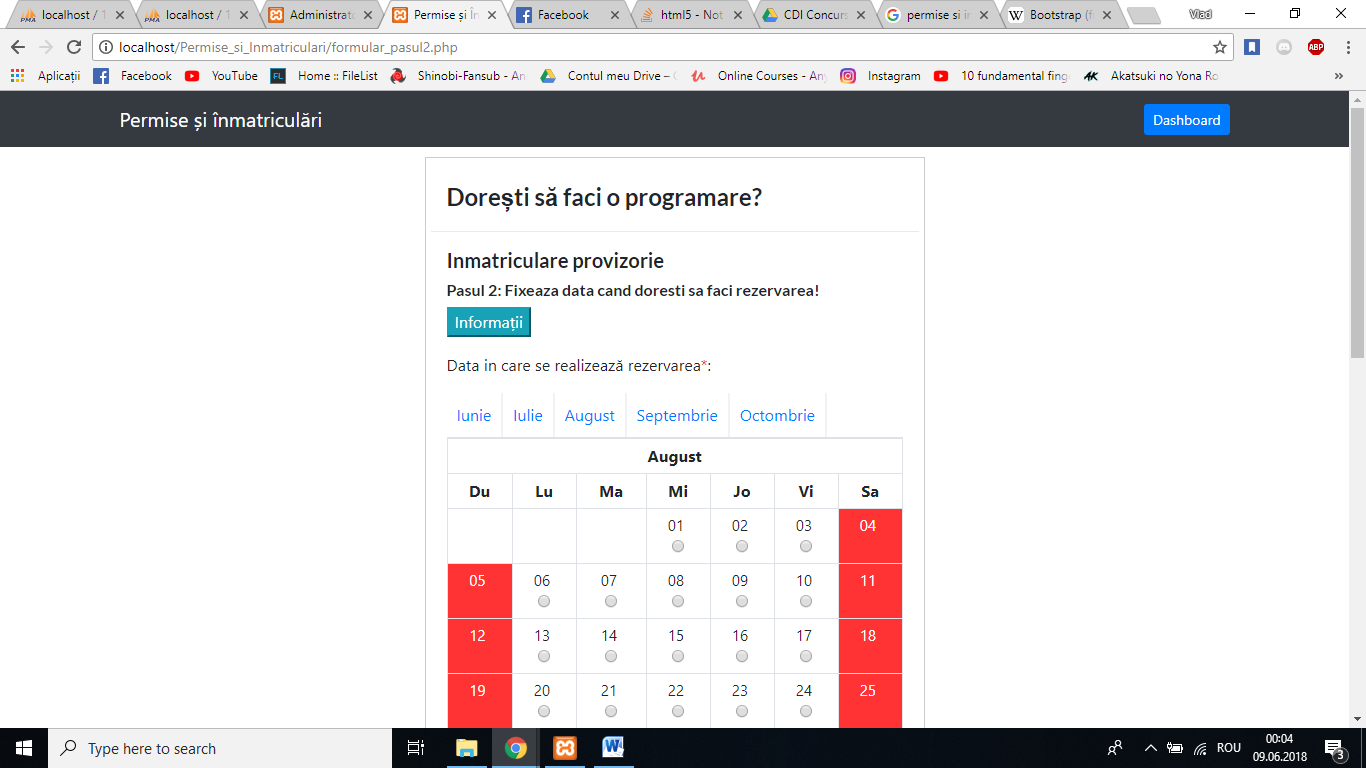












b) Interfața administratorului

