**Ingineria programarii**

**CUPRINS:**

**1. Prezentare proiect**

1.1. Ideea site-ului (scopul si obiective).........................................................1

**2**. **Detalii tehnice**

2.1.Limbajul si mediul de dezvoltare.............................................................1

2.2.Resurse software necesare......................................................................2

2.3. Resurse hardware necesare....................................................................2

2.4.Estimari asupra performantei..................................................................2

2.5.Dezvoltare ulterioara a aplicatiei.............................................................3

**3. Diagrame UML**

3.1**.** Modelarea cazurilor de utilizare..............................................................5

3.2. Modelarea conceptuală. Diagrama de clase............................................6

3.3. Diagrame de activităţi...............................................................................7

**4.** **Interfata aplicatiei**..................................................................................................9

**5. Codul aplicatiei**........................................................................................................15

**1. Prezentarea proiectului:**

Proiectul constă într-o aplicaţie WEB destinata iubitorilor de calatorii. Este vorba de o agentie de turism unde sunt expuse mai multe calatorii pe care utilizatorii le pot cumpara. Deocamdata pe site apar numai 4 calatorii: la Berlin, Stockholm, Bruxelles si Palermo. Site-ul este creat in scop comercial. Incearca sa se diferentieze de restul agentiilor de turism, prin oferta mai avantajoasa si mai accesibila.

**1.1 Ideea site-ului (scopul si obiective)**

Mondializarea din ultimii 20 de ani a dus la aparitia unei societati care isi doreste si isi permite sa intreprinda calatorii. Pentru ca acestia sa gaseasca avion si cazare trebuie, momentan, sa caute pe mai multe site-uri: Booking, Vola, WizzAir, RyanAir, Airbnb etc.

Ideea site-ului T@world este de a oferi pachete ieftine avion/cazare/transfer de la aeroport pentru a oferi clientilor sai un mod mai usor de a-si planfica calatoria – fara batai de cap si fara timp pierdut pentru cautarea celei mai bune oferte.

Acest site ofera confert si flexibilitate pentru ca utilizatorii sa fie multumiti de servicii si sa se reintoarca.

## 2. Detalii tehnice:

**2.1. Limbajul si mediu de dezvoltare**

Pentru crearea acestei pagini web sau folosit mai multe limbaje de programare. Cea mai mare parte a proiectul a fost facuta in **HTML**, acesta fiind ,,imbunatatit” prin intermediul limbajului Cascaded Style Sheet (**CSS)**, limbaj care ii da aspectul placut inca de la prima pagina. Partea de design a aplicaţiei a fost realizată, de asemenea, si cu programul Adobe Photoshop CS5, unde pozele au fost editate.

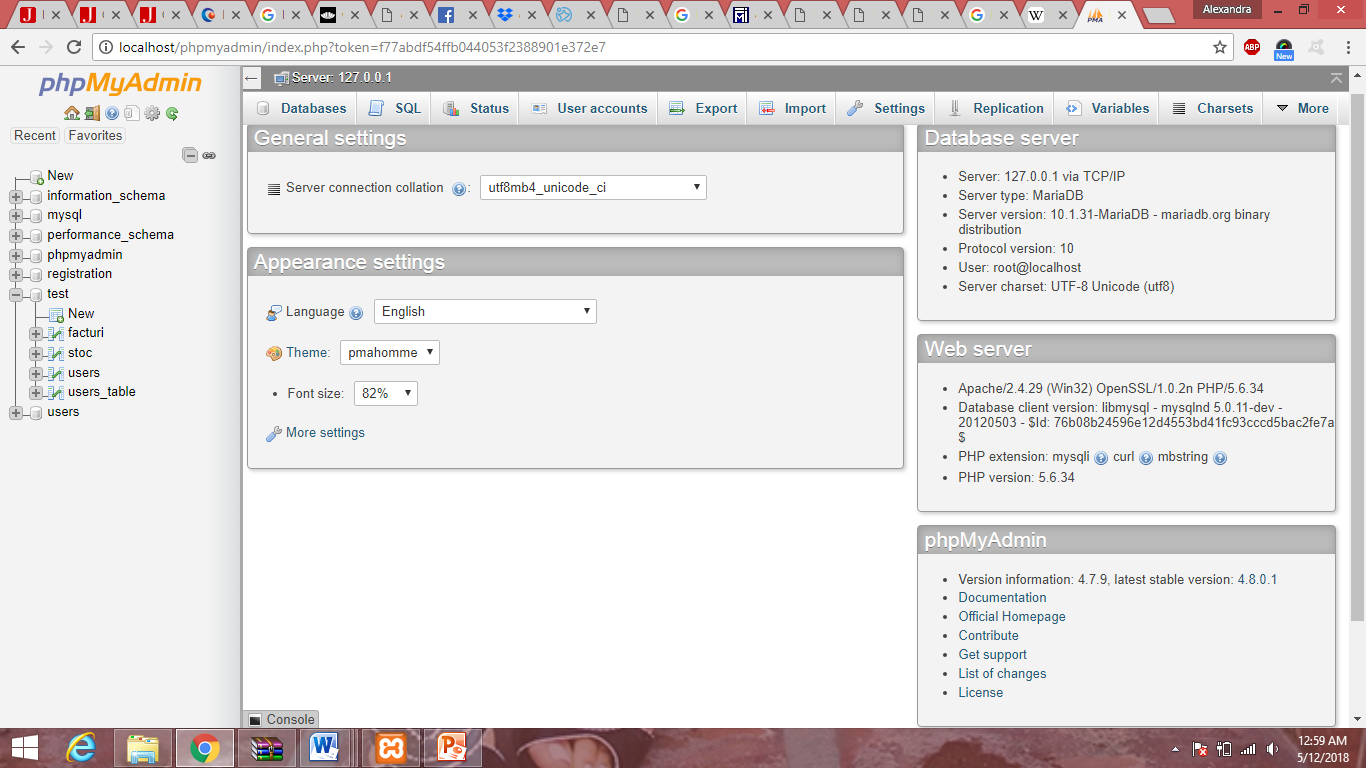
Cantitativ, urmeaza **PHP**-ul cu ajutorul caruia s-au creat aceste pagini dinamice. Site-ul este unul dinamic, nu static. PHP-ul a fost folosit in special pentru pagina de inregistrare si pentru cea de autentificare.

O importanta contributie o are si **JavaScriptul** care a fost utilizat in verificarea erorilor de inregistrare/autentificare. Daca campurile nu sunt complete, utilizatorul nu poate trece la pagina urmatoare, iar un mesaj de eroare speific apare:”Please enter an email”, “The two passwords don’t match” etc.

De asemenea, pentru gestionarea bazei de date s-a folosit limbajul de programare **MySQL**. Diferite instrumente s-au utilizat pentru administrarea acestei baze de date, cum ar fi **PHPMyAdmin**.

Acest proiect a fost creat in mediul de dezvoltare **NetBeans IDE 8.2** pentru ca este un mediu usor de folosit pentru orice programator si, de asemenea, prezinta versatilitatea de a lucra cu mai multe limbaje (cum ar fi cele mentionate mai sus).

**2.2.Resurse Software necesare:**



* Sistemul de operare: Windows 8;
* Internet Explorer sau
* Mozila Firefox sau
* Google Chrome;
* XAMPP Control Panel v3.2.2;
* Apache v 2.4.29;
* MySQL v 5.0.11;
* Limbaj de scripting – PHP v. 5.6.34;
* Aplicaţie management baze de date - PHPMyAdmin v 4.7.9;

**2.3.Resurse Hardware necesare:**

Elementele de mai jos reprezinta elementele pe care a fost testat proiectul in momentul in care a fost conceput:

* Procesor : Intel(R) Core(TM) i7-5500U CPU @ 2.40 GHz 2.40GHz
* Memorie RAM: 4.00 GB
* Sistem: 64-bit Operating System

2.4.**Estimari asupra performantelor:**

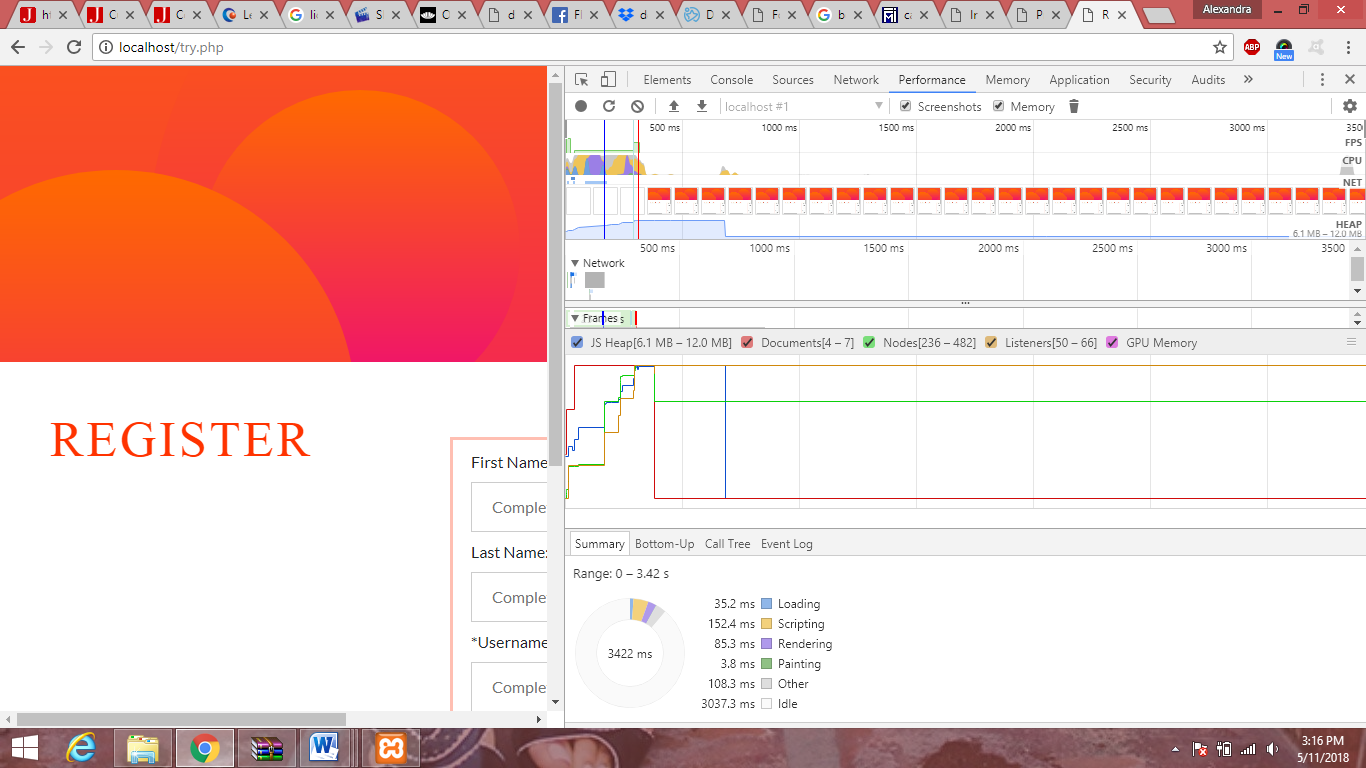
Performantele depind foarte mult de viteza internetului fiecarui utilizator. Ele se pot deschide intre 1 si 3 secunde, insa datorita faptului ca site-ul nu are foarte multe informatii, imagini, etc. acesta se deschide destul de repede conform graficului.

De asemenea, performantele mai depind si de resursele hardware-ul asociate serverului ce susţine aplicaţia. De obicei, pentru o conexiune se utilizază aproximativ 0,1 % din memoria maşinii, astfel că, în cazul în care se folosesc 2 GB de memorie RAM se estimează aproximativ 2000 de clienti ce navighează simultan în aplicaţie.

Aplicatia este detinata pentru a fi utilizata pe un desktop obisnuit, iar totul este stocat si accesat online nu este necesar spatiu pe disc.

Alte detalii:

* Se pot deschide mai multe pagini simultan si utilizatorii se pot conecta de la calculatoarele proprii oricand.
* Asta inseamna ca utilizatorii pot accesa simultan aplicatia fara ca acesta sa se blocheze.



## 2.5. Dezvoltare ulterioară a aplicaţiei:

Ca orice alta aplicatie web aceasta poate fi dezvoltata si imbunatatita pentru a facilita accesul la produse. Numeroase schimbari ii pot fi aduse, insa am optat doar pentru prezentarea a sase dintre ele:

* Aplicatia va avea statistici si rapoarte pentru a vedea traficul de persoane care au accesat site-ul.
* Se vor adauga mai multe calatorii pe masura ce numarul cererilor se mareste.
* Va exista o inregistrare in doi pasi. Mai intai iti introduci datele, apoi pe mail o sa-ti vina un cont de activare si cu acesta vai putea sa-ti faci cont.
* Vor exista mai multe categorii de utilizatori. Vor exista si utilizatori fideli care vor primi promotii.
* La aparitia unei noi calatorii, tuturor utilizatorilor li se va trimite un mail de prezentare a ofertei.
* Va exista o metoda de plata cu cardul

**3. Diagrame UML**

Folosirea de astfel de modele face mai usoara intelegerea programului, iar impactul se observa mai ales atunci cand avem de-a face cu probleme complexe greu de urmarit cu ochiul liber.

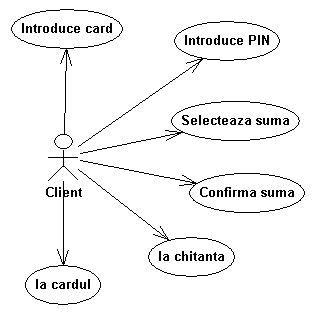
Pentru a realiza toate aceste lucruri vom folosi Limbajul Unificat de modelare (UML). Acesta este un limbaj care te ajuta sa vizualizezi mai bine, sa construiesti si sa vezi specificitatea elementelor sistemelor software.

Acest model a fost deja folosit cu succes pentru probleme mari si complexe, astfel demonstrandu-si pragmatismul si flexibilitatea.

**3.1. Modelarea cazurilor de utilizare**

Ne propunem in aceasta parte sa reprezentam toate interactiuniile posibile dintre utilizator si sistem. Initiatorii unei actiuni se cheama actori. In cazul nostru, actorul va fi clientul – persoana care va comanda calatoria.

Pentru sistemul nostru, un singur actor poate interactiona cu mai multe cazuri de utilizare (isi poate face un cont, se poate autentifica, poate sa-si rezerve o calatorie, poate sa trimita un mesaj – feedback etc.)



Creeaza cont

Trimite mesaj

Client (Actor)

Rezerva calatorie

2. **3.2. Modelarea conceptuală. Diagrama de clase**

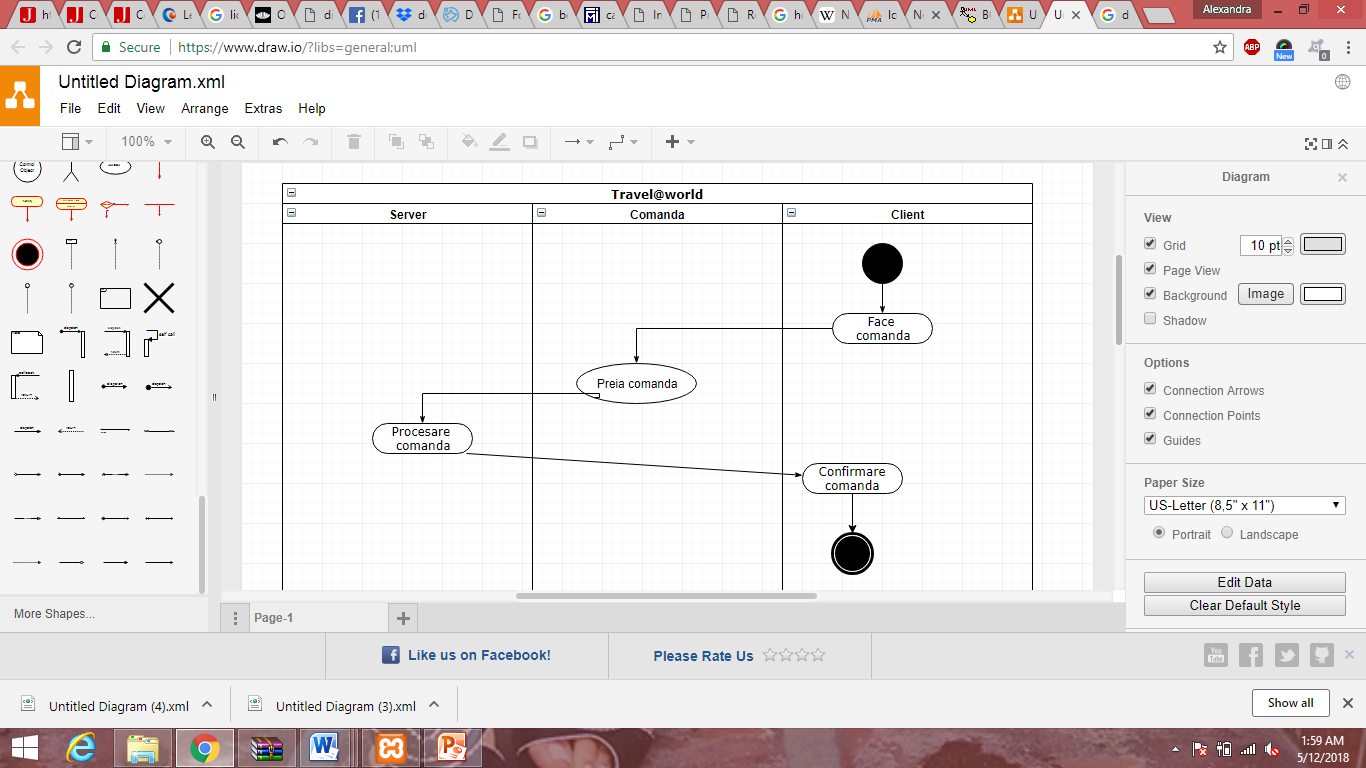
## Modelarea conceptuala este activitatea de identificare a conceptelor importante pentru sistem. Astfel diagrama claselor furnizeaza structura codului care va fi scris.

O clasa se reprezinta printr-o casuta impartita in trei. In partea de sus este notat numele clasei, in partea din mijloc atributele, iar in partea de jos operatiile sale.

## 

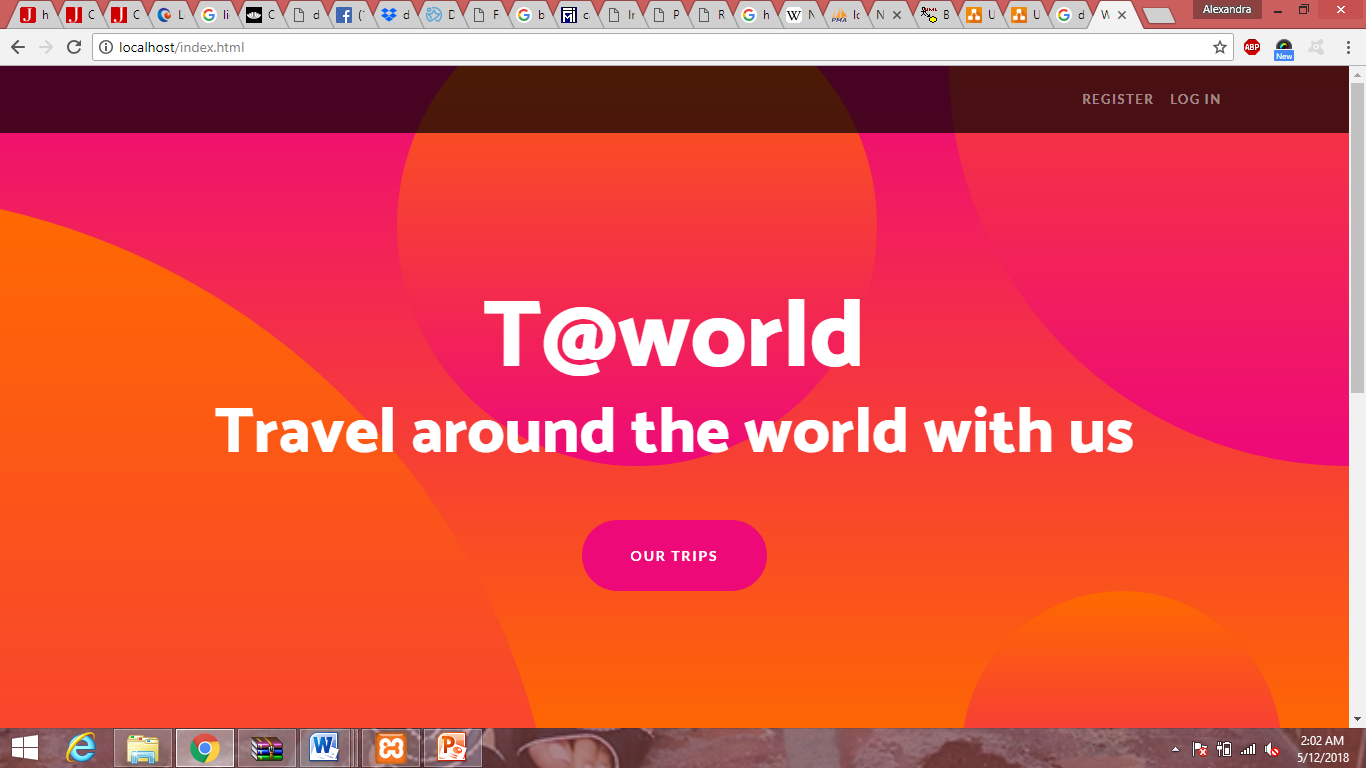
## 3.3. Diagrame de activitati

Sunt folosite pentru modelarea proceselor sau a algoritmilor din spatele unui anumit caz de utilizare.

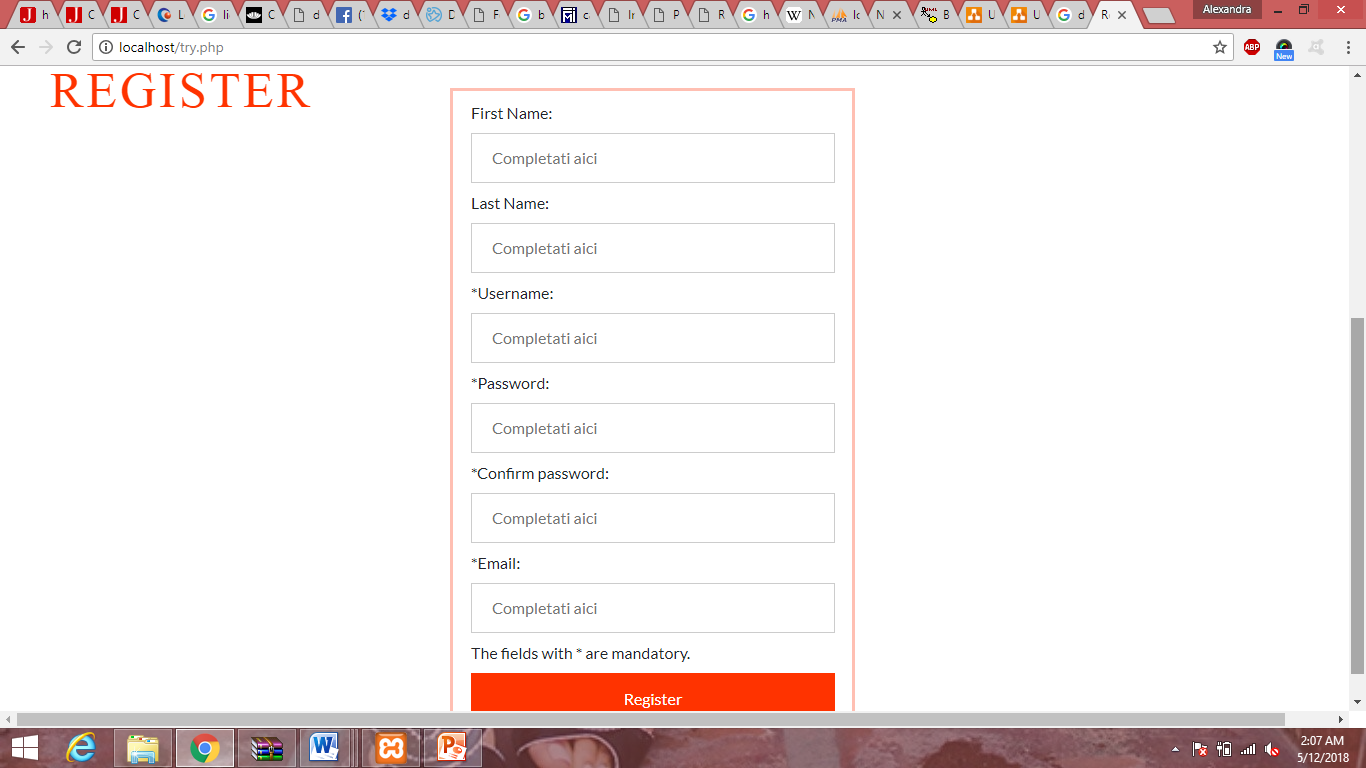


**4. Interfata aplicatiei**

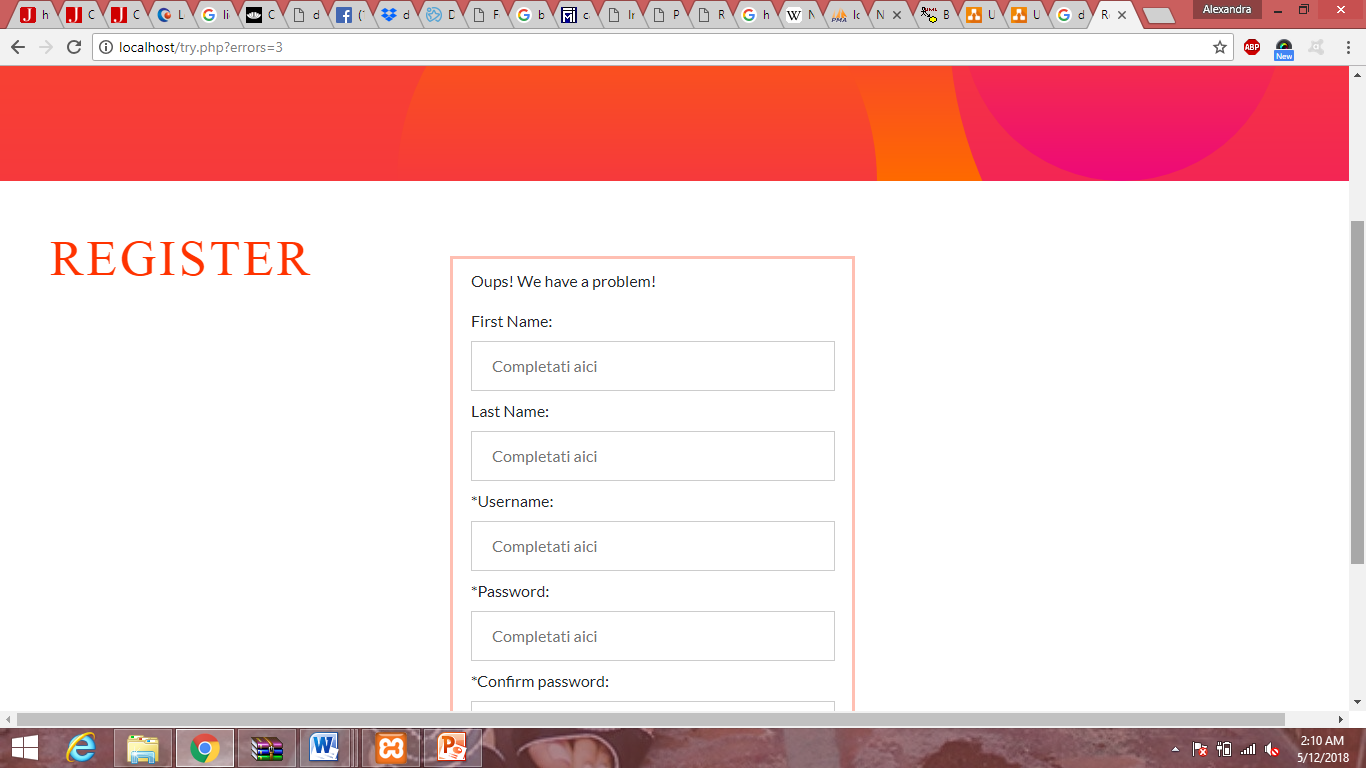
Pagina principala de unde poti ajunge si in pagina de inregistrare si in pagina de autentificare. De asemenea poti vizualiza pafina de calatorii, dar nu poti face comanda inainte de a-ti face un cont.



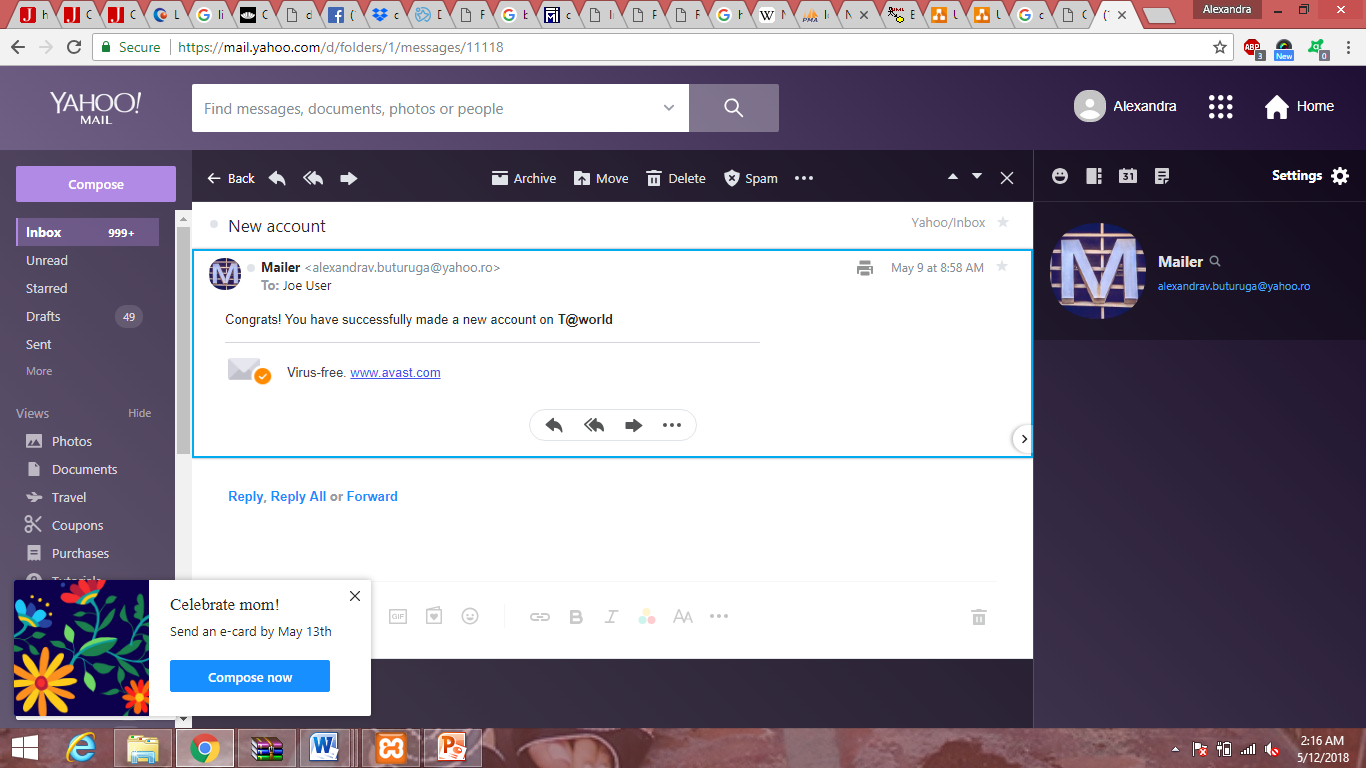
Clientul poate sa acceseze paginile site-ului, dar este recomandat sa intre pe pagina ,,Register” si sa-si faca un cont. Trebuie sa introduca un nume, un prenume, un nume de utilizator si o parola. I se cere sa confirme parola prin inserarea ei inca o data in ,,Cofirm password”. Ultima informatie necesara pentru a termina de inregistrat este introducerea unei adrese valide de email.



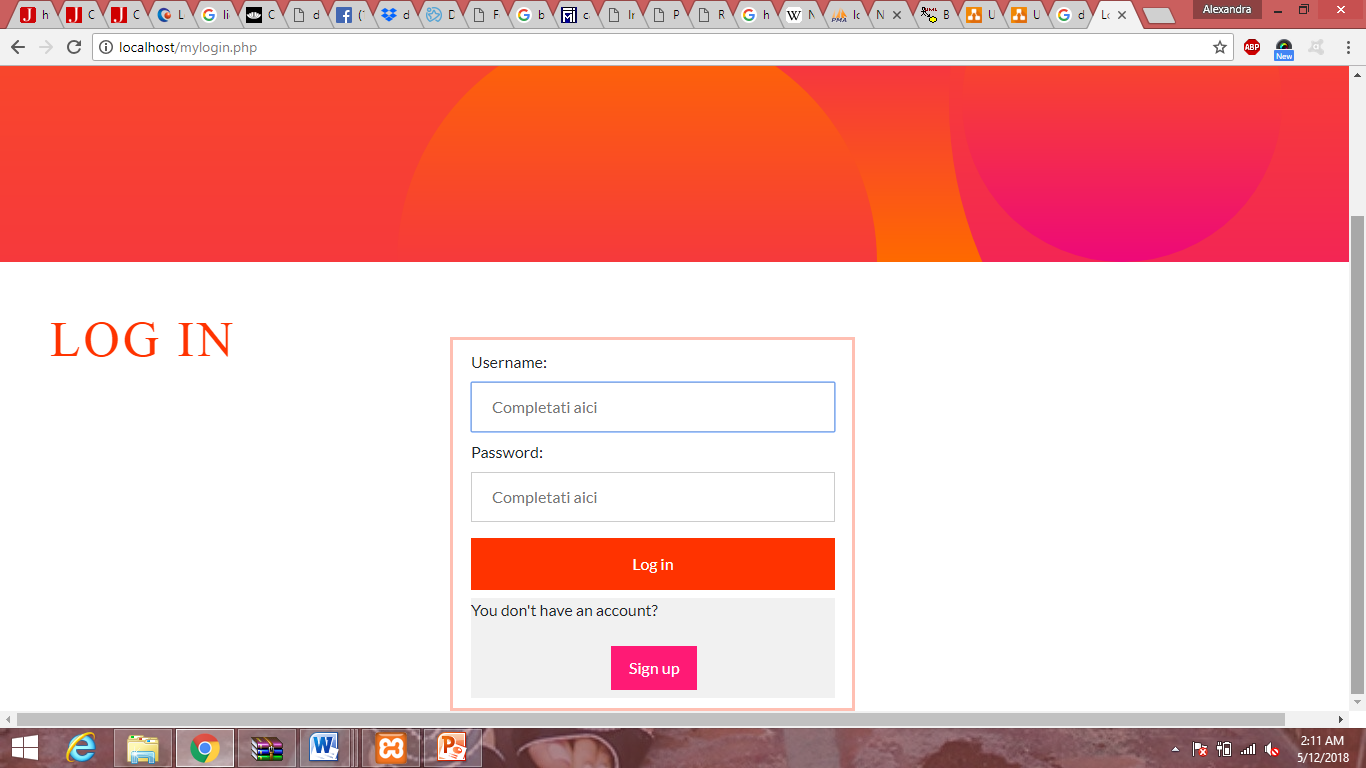
Daca cumva utilizatorul nu introduce datele corect atunci ii va aparea un mesaj de eroare. In PHP se verifica daca exista text in casutele de username, email si parola, dar si daca cele doua parole sunt identice.



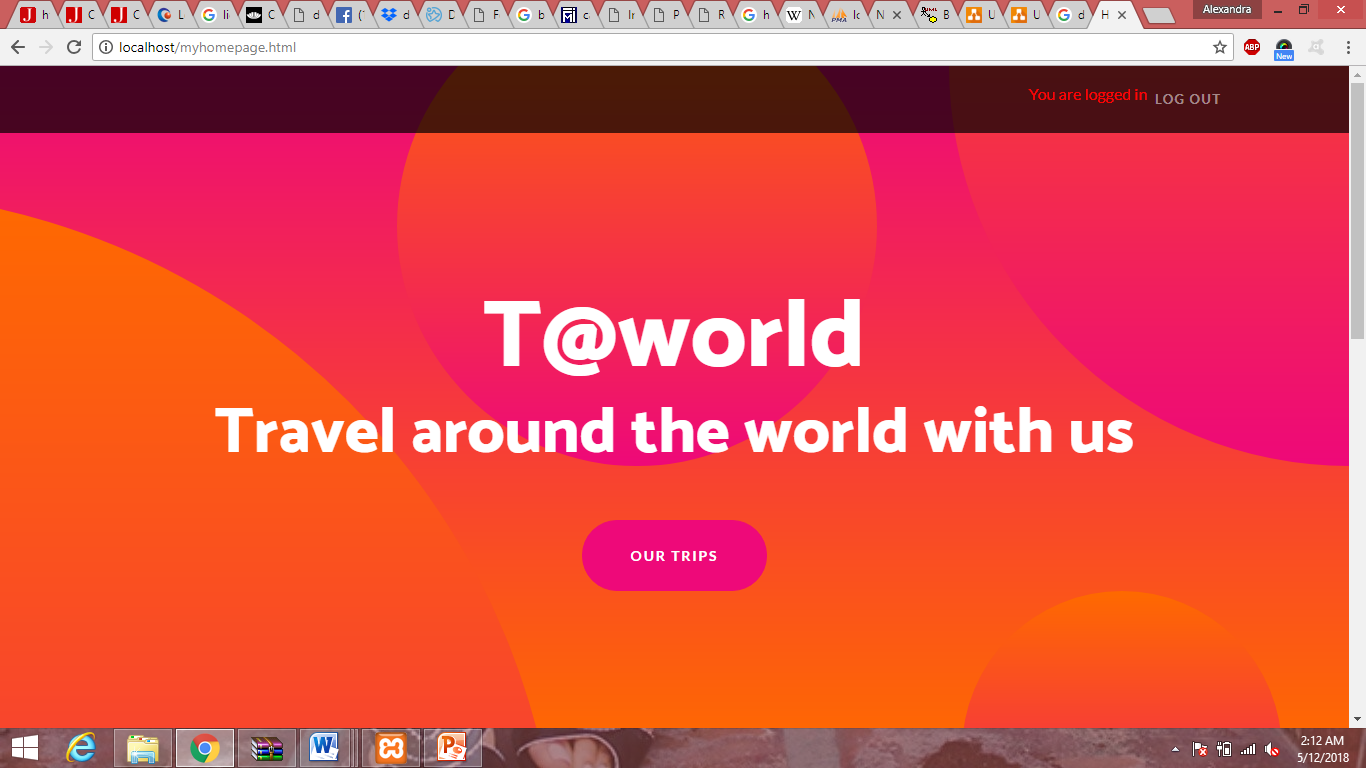
De asemenea, utilizatorul v-a primi pe mail un mail de confirmare. Acest serviciu a fost implementat cu PHPMailer.



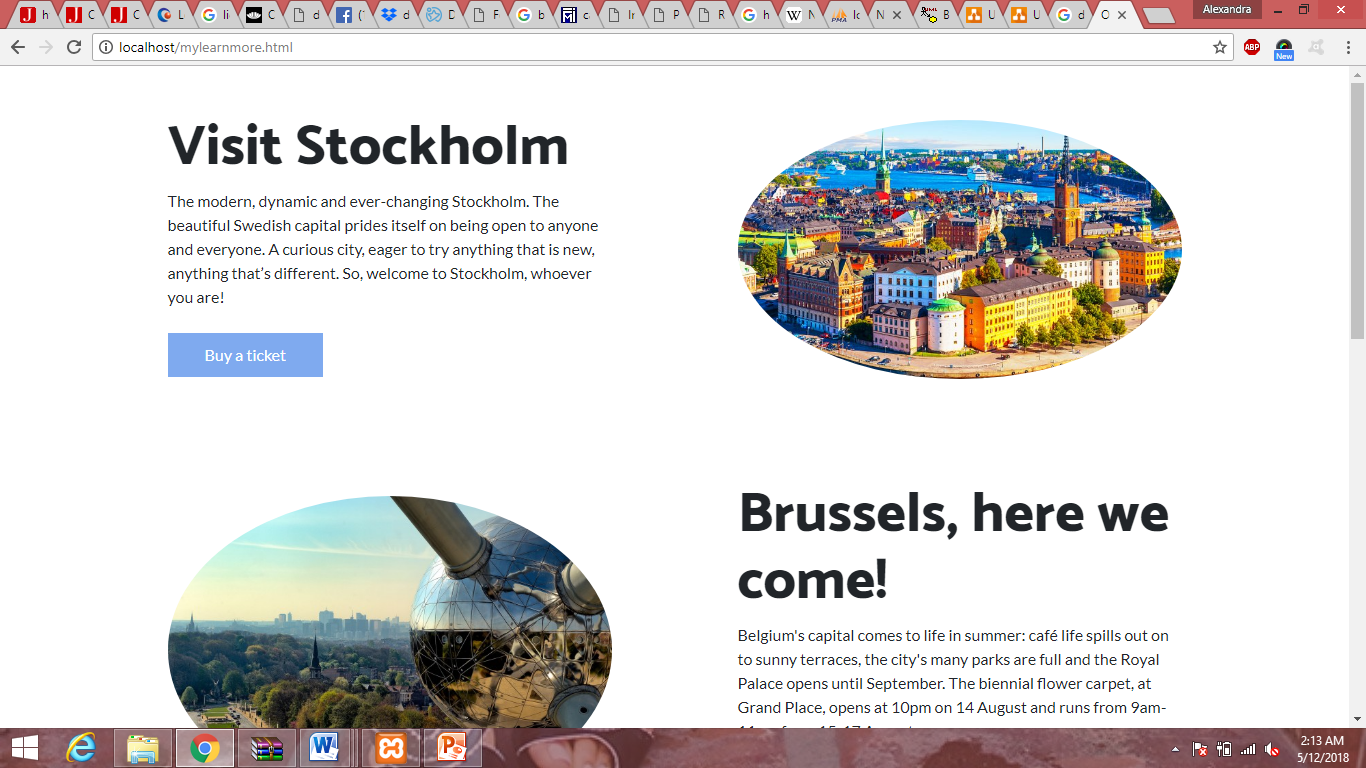
Exista si o pagina speciala pentru autentificare. In cazul in care un utilizator a intrat din greseala, el poate trece direct in pagina de Inregistrare prin apasarea butonului ,,Sign up”.



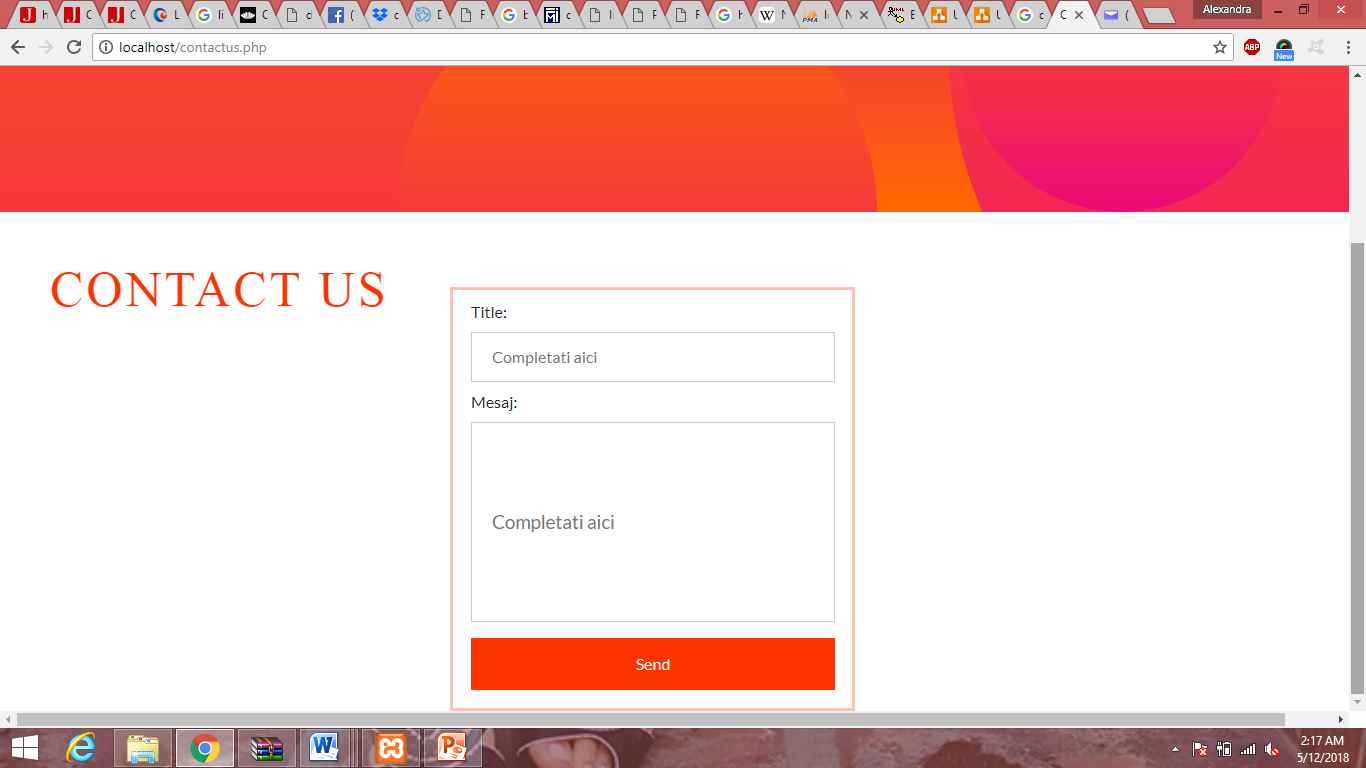
Odata ce este logat, utilizatorului ii va aparea un mesaj in partea din dreapta sus.



Acum poti rezerva o calatorie.



Dar poate sa trimita si un mesaj administratorului daca vrea sa-si dea feedbackul.



**5. Codul aplicatiei**

**Fisierul server.php**

<?php

include("mailer.php");

session\_start();

//var\_dump($\_POST);

$username = "";

$password\_1 = "";

$password\_2 = "";

$errors = array();

//connect to the database

$db = mysqli\_connect('localhost', 'root', '', 'registration');

//when you click on the register button

if(isset($\_POST['register'])){

//our variables

$fname = $\_POST['fname'];

$lname = $\_POST['lname'];

$username = $\_POST['username'];

$password\_1 = mysql\_real\_escape\_string($\_POST['password\_1']);

$password\_2 = mysql\_real\_escape\_string($\_POST['password\_2']);

$email = mysql\_real\_escape\_string($\_POST['email']);

$error\_code = 0;

//checking if all the fields are filled

if(empty($username)){

array\_push($errors, "You must enter a username");

$error\_code = 1;

}

if(empty($password\_1)){

array\_push($errors, "You must enter a password");

$error\_code = 2;

}

if(empty($email)){

array\_push($errors, "Please enter a valid email adress");

$error\_code = 3;

}

if($password\_1 != $password\_2){

array\_push($errors, "The two passwords don't match");

$error\_code = 4;

}

//echo "isset register";

//if there are no errors, save user to database

if(count($errors) == 0){

$password = md5($password\_1);//encrypt password before storing in database

$sql = "INSERT INTO users(prenume, nume, username, password, email) VALUES ('$fname' , '$lname' , '$username' , '$password' , '$email')";

mysqli\_query($db, $sql);

$\_SESSION['username'] = $username;

$\_SESSION['succes'] = "You are now logged in";

//send mail

send\_mail\_to\_user($email);

header('location: myhomepage.html');//redirect to home page of logged user

}else{

header('location: try.php?errors='. $error\_code);

}

//log user in from login page

}else if(isset($\_POST['login34'])){

$username = mysql\_real\_escape\_string($\_POST['username']);

$password = mysql\_real\_escape\_string($\_POST['password']);

//checking if all the fields are filled

//if(empty($username)){

// array\_push($errors, "You must enter a username");

//}

//if(empty($password)){

// array\_push($errors, "You must enter a password");

//}

if(count($errors) == 0){

$password = md5($password);//encrypt password before storing in database

$sql = "SELECT \* FROM users WHERE username ='$username' AND password = '$password'";

mysqli\_query($db, $sql);

$\_SESSION['username'] = $username;

$\_SESSION['succes'] = "You are now logged in";

header('location: myhomepage.html');//redirect to home page of logged user

}else{

array\_push($errors,"Wrong username or password");

header('location: mylogin.php');

}

//logout

}else if(isset($\_GET['logout'])){

session\_destroy();

unset($\_SESSION['username']);

header('location: index.html');

//default case

} //default case

else{

header('location: index.html');

}

**Fisierul mailer.php**

<?php

// Import PHPMailer classes into the global namespace

// These must be at the top of your script, not inside a function

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader

require 'phpMailer/src/Exception.php';

require 'C:/xampp/php/PEAR/PHPMailer/src/PHPMailer.php';

require 'PHPMailer/src/SMTP.php';

function send\_mail\_to\_user ($email){

$mail = new PHPMailer(true); // Passing `true` enables exceptions

//Server settings

$mail->SMTPDebug = 2; // Enable verbose debug output

$mail->isSMTP(); // Set mailer to use SMTP

$mail->Host = 'smtp.mail.yahoo.com'; // Specify main and backup SMTP servers

$mail->SMTPAuth = true; // Enable SMTP authentication

$mail->Username = 'alexandrav.buturuga@yahoo.ro'; // SMTP username

$mail->Password = 'cursdephp223'; // SMTP password

$mail->SMTPSecure = 'tls'; // Enable TLS encryption, `ssl` also accepted

$mail->Port = 587; // TCP port to connect to

$mail->SMTPOptions = array(

'ssl' => array(

'verify\_peer' => false,

'verify\_peer\_name' => false,

'allow\_self\_signed' => true

)

);

//Recipients

$mail->setFrom('alexandrav.buturuga@yahoo.ro', 'Mailer');

$mail->addAddress($email, 'Joe User'); // Add a recipient

//$mail->addAddress('ellen@example.com'); // Name is optional

//$mail->addReplyTo('info@example.com', 'Information');

//$mail->addCC('cc@example.com');

//$mail->addBCC('bcc@example.com');

//Attachments

//$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments

//$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

//Content

$mail->isHTML(true); // Set email format to HTML

$mail->Subject = 'New account';

$mail->Body = 'Congrats! You have successfully made a new account on <b> T@world</b>';

$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

try {

$mail->send();

//header('location: myhomepage.html');

}

catch (phpmailerException $e) {

echo $e->errorMessage(); //error messages from PHPMailer

} catch (Exception $e) {

echo $e->getMessage();

}

}

Send message:

<?php include('mailer.php');

$title = $\_POST['title'];

$mesaj = $\_POST['mesaj'];

//echo"heeei";die;

//send mail to me

send\_mail\_to\_user\_feedback($mesaj, $title);

header('location: myhomepage.html');

Va multumesc pentru atentie!