

Proiect ~Mini Facebook~

Proiectul mini facebook este un proiect ce simuleaza :

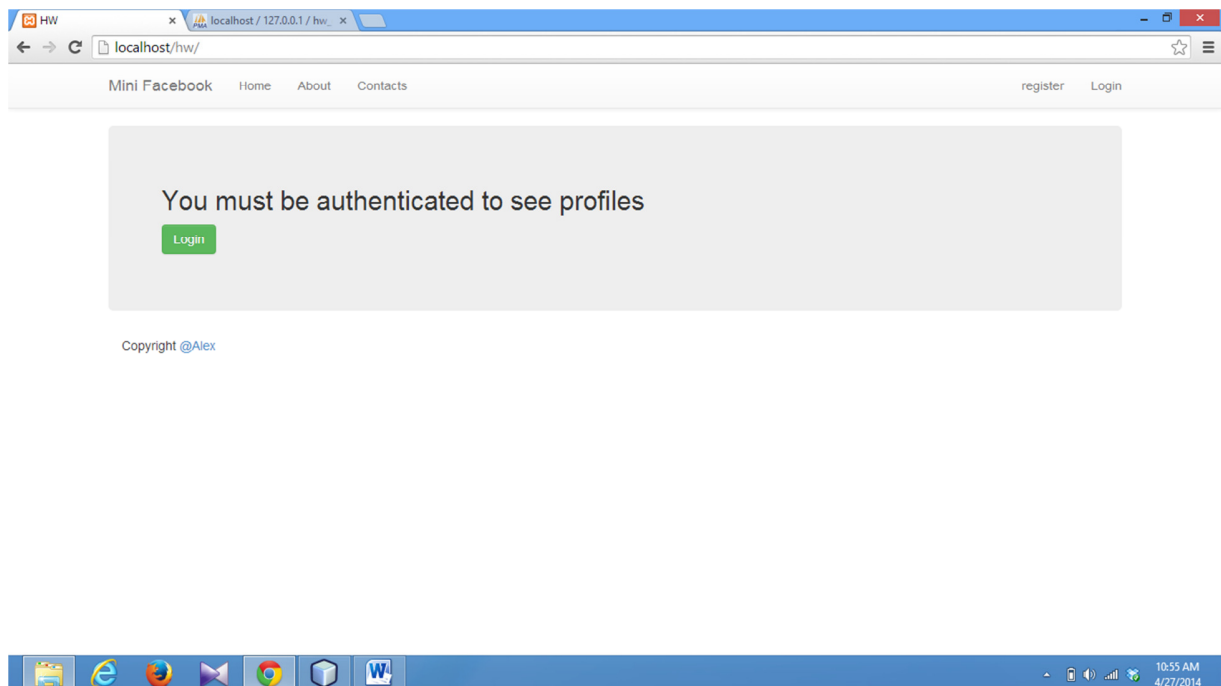
- Inregistrarea unui user
- Creare profil
- Upload de poze profil
- Vizualizare profilului altor useri
- Adaugare like , comentarii pozelor altor useri

Pentru realizarea acestui proiect m-am folosit de PHP pentru programarea server-side , frameworkul Twitter Bootstrap pentru a realiza designul site-ului , de o baza de date creata in MySql pentru salvarea datelor , de frameworkul jquery scris in javascript pentru a crea efecte si a manipula DOM-ul , dar si pentru a executa chemari de AJAX pentru functionalitate asincrona a site-ului.

Am optat pentru o structura OOP a implementarii PHP pentru a generaliza codul si a nu ma repeta (DRY).

Cum functioneaza acest proiect :

Pasul 1. –intrarea in site



- Autentificare - Daca nu sunteti inca un utilizator aveti posibilitatea de a va inregistra accesand optiunea de register :

The screenshot shows a web browser window with the address bar displaying 'localhost/hw/files/register.php'. The page has a navigation bar with links: 'Mini Facebook', 'Home', 'About', 'Contacts', 'register', and 'Login'. The main content area is titled 'Register here' and contains a registration form with the following fields: 'Name', 'Username', 'Email', 'Password', and 'Confirm password'. Each field has a placeholder text 'Enter ...'. Below the fields are two buttons: 'Submit' and 'Reset values'. At the bottom left of the form area, it says 'Copyright @Alex'. The Windows taskbar at the bottom shows various application icons and the system clock indicating 10:58 AM on 4/27/2014.

Daca aveti deja un cont alegeti optiunea login :

The screenshot shows a web browser window with the address bar displaying 'localhost/hw/files/login.php'. The page has a navigation bar with links: 'Mini Facebook', 'Home', 'About', 'Contacts', 'register', and 'Login'. The main content area is titled 'Login here' and contains a login form with the following fields: 'username' and 'Password'. Each field has a placeholder text 'Enter ...'. Below the fields are two buttons: 'Submit' and 'Reset values'. At the bottom left of the form area, it says 'Copyright @Alex'. The Windows taskbar at the bottom shows various application icons and the system clock indicating 10:59 AM on 4/27/2014.

Datele dumneavoastra personale sunt stocate in baza de date a proiectului in tabelul "users" . Pentru confidentialitatea parolei am folosit algoritmul de criptare md5

Asa arata tabelul userilor :

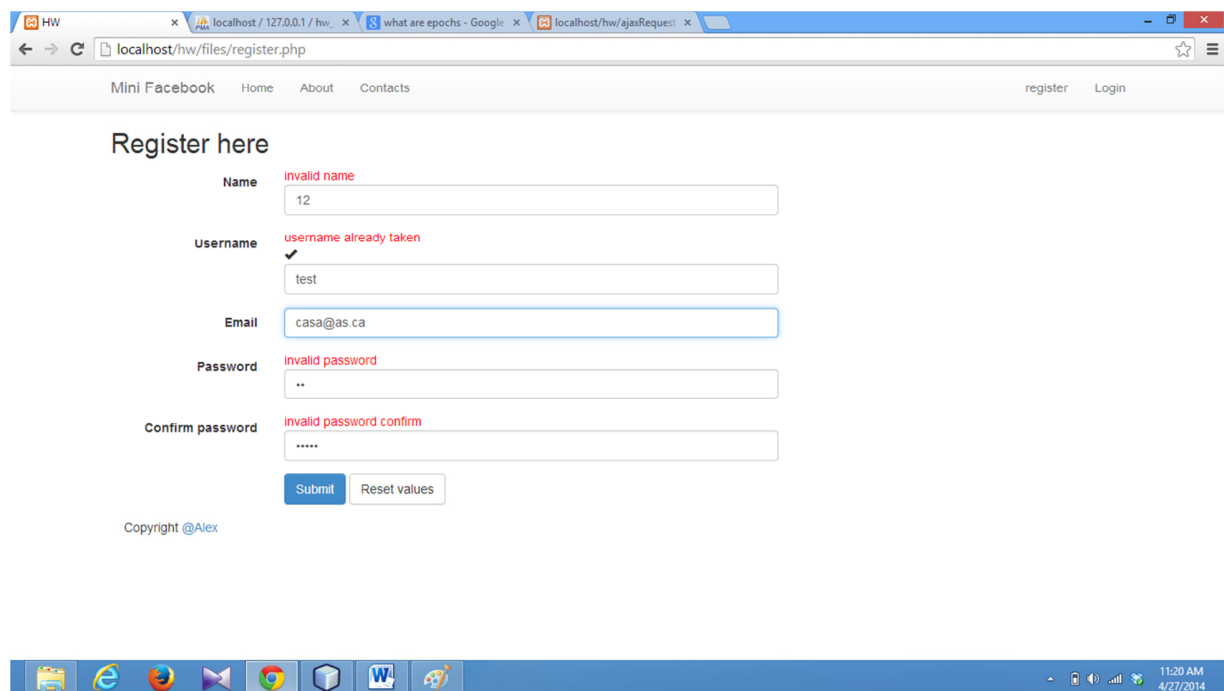
← T →				id	name	username	email	password	created_at
☐	 Edit	 Copy	 Delete	1	popica	alexandru	email@trest	81dc9bdb52d04dc20036dbd8313ed055	1395059582
☐	 Edit	 Copy	 Delete	2	test	test	test@test2.com	81dc9bdb52d04dc20036dbd8313ed055	1395072019

Vom crea un nou user cu datele :

- name – TestLive
- username – live
- email – testlive@minif.co
- password – 123456
- created reprezinta epoch timestamp generat in momentul realizarii contului

Fiecare dintre campurile de la inregistrare dispune de o validare realizata cu ajutorul Javascriptului si jquery .

Exemplu date scrise gresit :

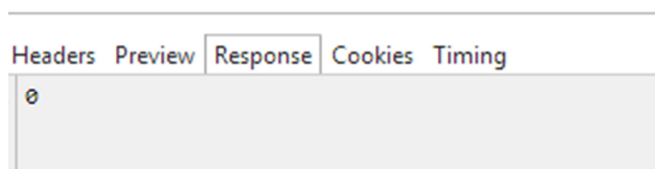
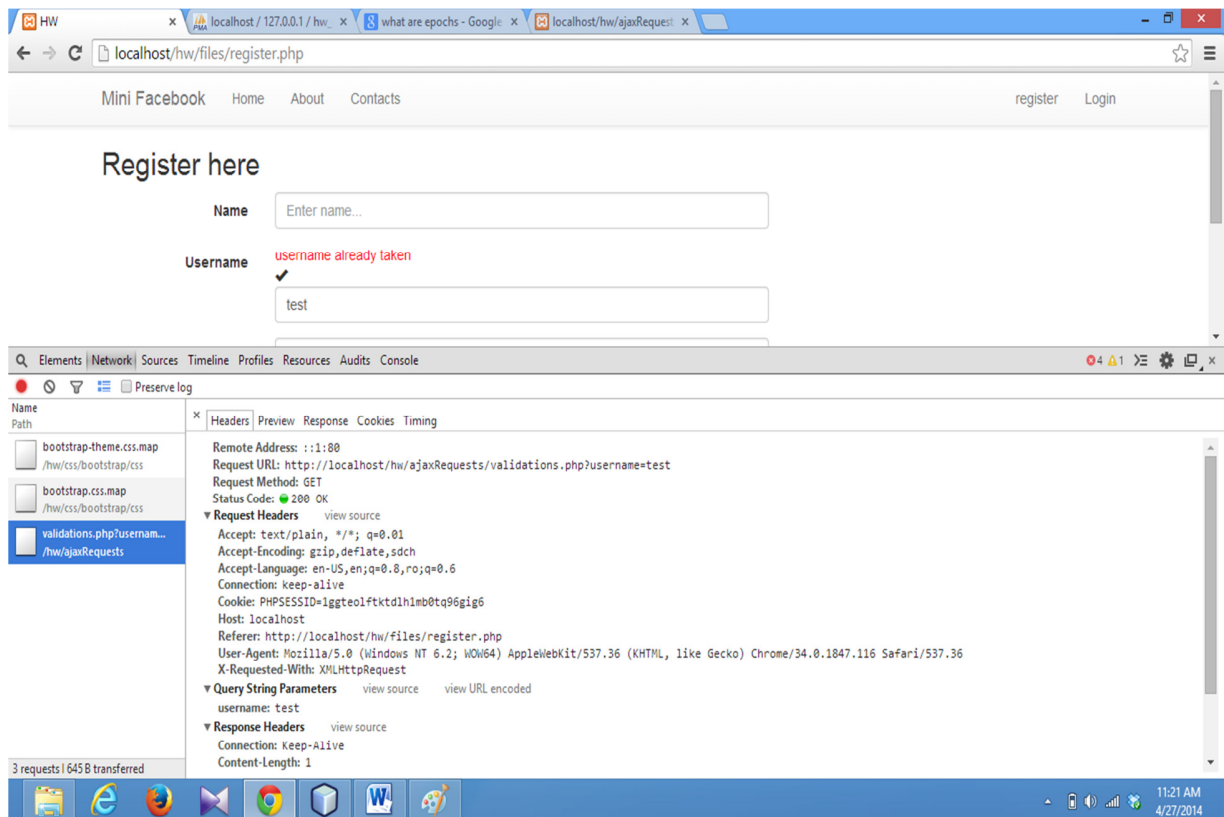


The screenshot shows a web browser window with the address bar displaying 'localhost/hw/files/register.php'. The page has a navigation bar with links: 'Mini Facebook', 'Home', 'About', 'Contacts', 'register', and 'Login'. The main content area is titled 'Register here' and contains a registration form with the following fields and validation messages:

- Name:** '12' (invalid name)
- Username:** 'test' (username already taken)
- Email:** 'casa@as.ca' (invalid email)
- Password:** '..' (invalid password)
- Confirm password:** '.....' (invalid password confirm)

At the bottom of the form are 'Submit' and 'Reset values' buttons. The footer of the page says 'Copyright @Alex'. The Windows taskbar at the bottom shows the time as 11:20 AM on 4/27/2014.

Pentru validarea usernamului am folosit un call de AJAX care verifica baza de date daca usernamul exista si returneaza 0 in cazul in care acesta este prezent in baza de date si 1 daca nu exista.



Dupa introducerea datelor mentionate mai sus user-ul este creat cu success iar datele lui sunt memorate in baza de date :

	id	name	username	email	password	created_at
☐ Edit ☐ Copy ☐ Delete	1	popica	alexandru	email@test	81dc9bdb52d04dc20036dbd8313ed055	1395059582
☐ Edit ☐ Copy ☐ Delete	2	test	test	test@test2.com	81dc9bdb52d04dc20036dbd8313ed055	1395072019
☐ Edit ☐ Copy ☐ Delete	3	TestLive	live	testlive@minif.co	e10adc3949ba59abbe56e057f20f883e	1398594452

Un user beneficiaza de toate serviciile site-ului pe cand un simplu vizitator este restrctionat la accesul anumitor pagini . Aceasta restrictionare este posibila datorita variabilelor de sesiune care se creaza la logarea userului.

In functie de necesitatea datelor unui user ,am ales sa stochez urmatoarele date in vectorul de sesiune:

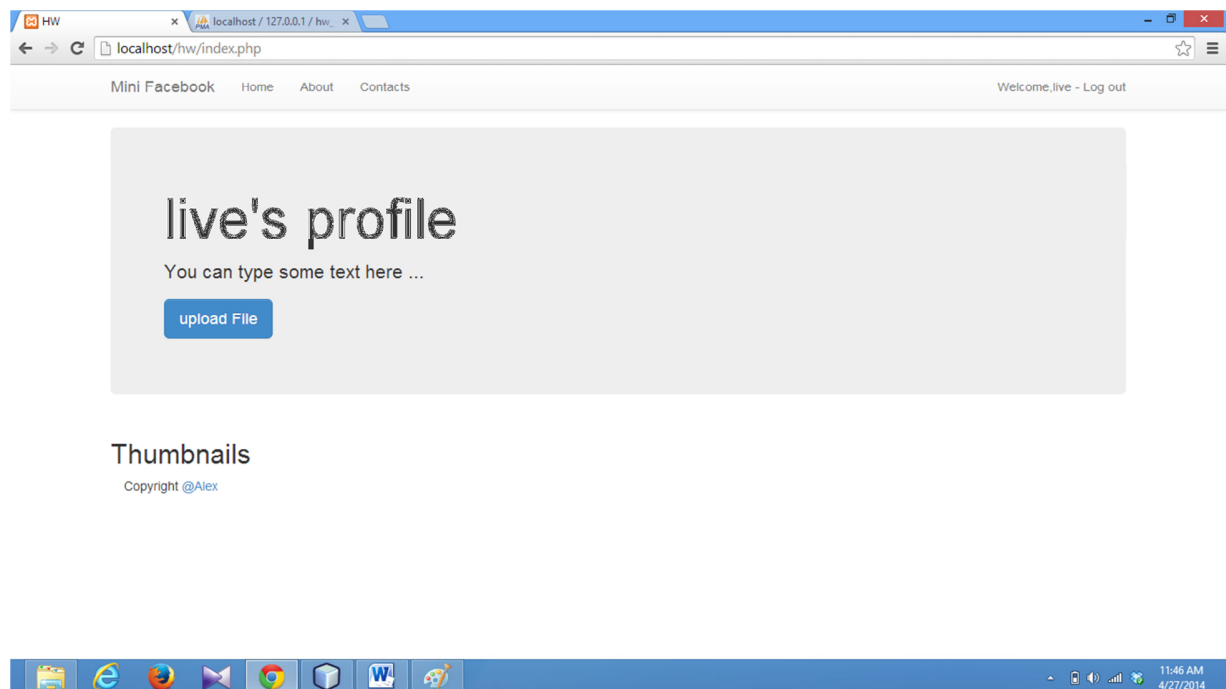
```
$loginClass->setState('user_id', $result[0]['id']);  
$loginClass->setState('username', $result[0]['username']);  
$loginClass->setState('Created', time());  
$loginClass->setState('expire', time() + 1800);
```

Unde setState este metoda clasei “LoginClass”

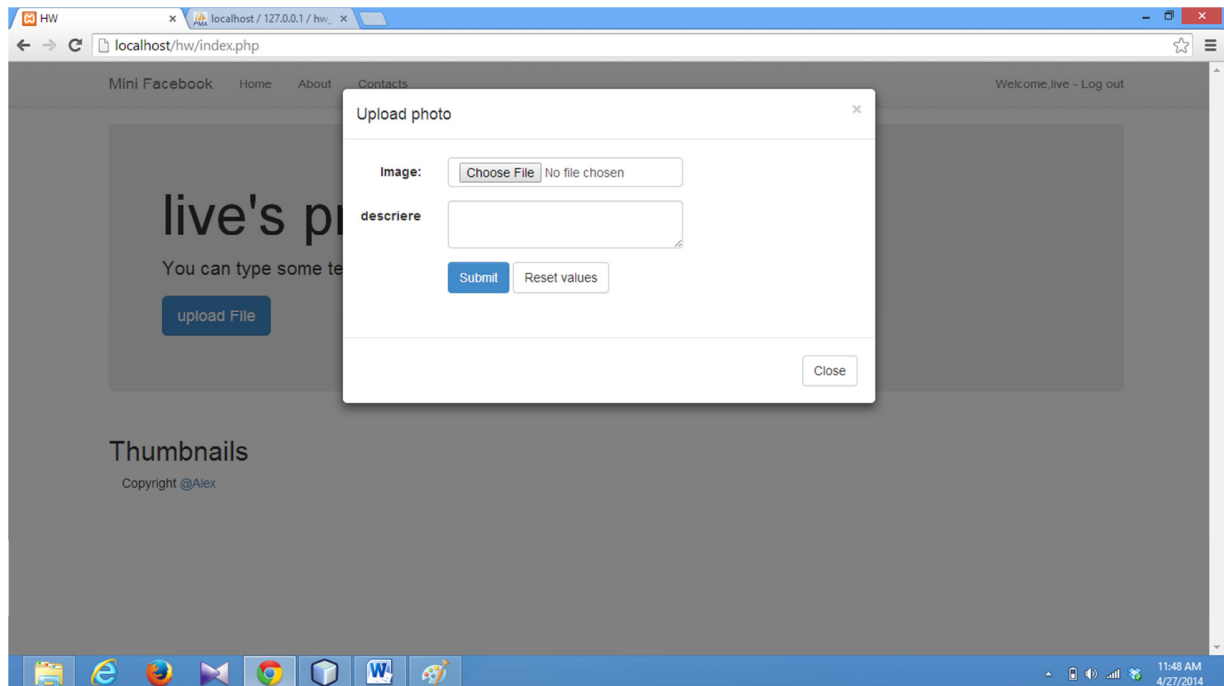
```
public function setState($name,$value)  
{  
    $_SESSION[$name]=$value;  
}
```

O sesiune este valabila pentru 30 de minute.

Folosind userul proaspat creat , dupa logare pagina de home se modifica :



Acum se poate folosi butonul de upload pentru adaugarea de poze :



Poza pe care doriti sa o alegeti va trebui sa treaca prin validari :

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_FILES) && !empty($_FILES)) {
        $upload = new Upload($_FILES, $_POST['descriere']);
        header('location: index.php');
    }
}
```

Actionarea butonului submit va crea o noua instant a clasei Upload . In constructorul acestei clase vom primi un array cu informatii despre fisierul introdus si descrierea adaugata pozei .

Se verifica daca fisierul introdus este intradevar o poza .

```
private function is_image() {
    $allowedExts = array("gif", "jpeg", "jpg", "png");
    $ext = pathinfo($this->file_name, PATHINFO_EXTENSION);
    if (in_array(strtolower($ext), $allowedExts)) {
```

```

    return true;

} else

    return false;

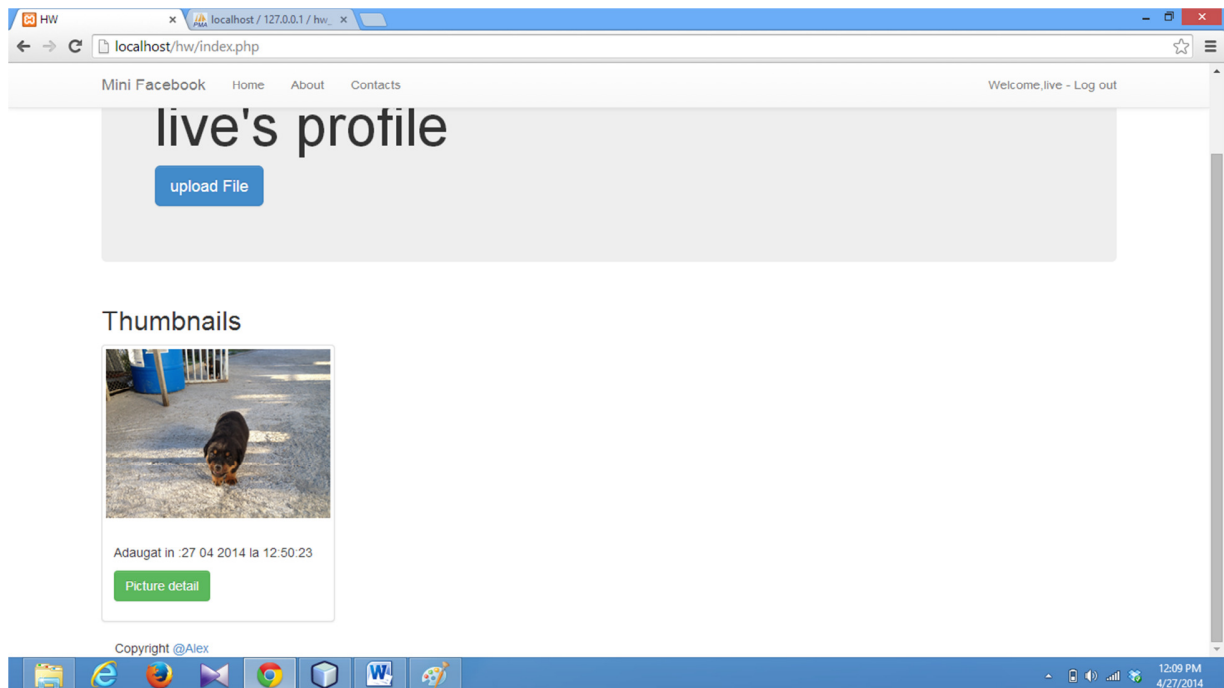
}

```

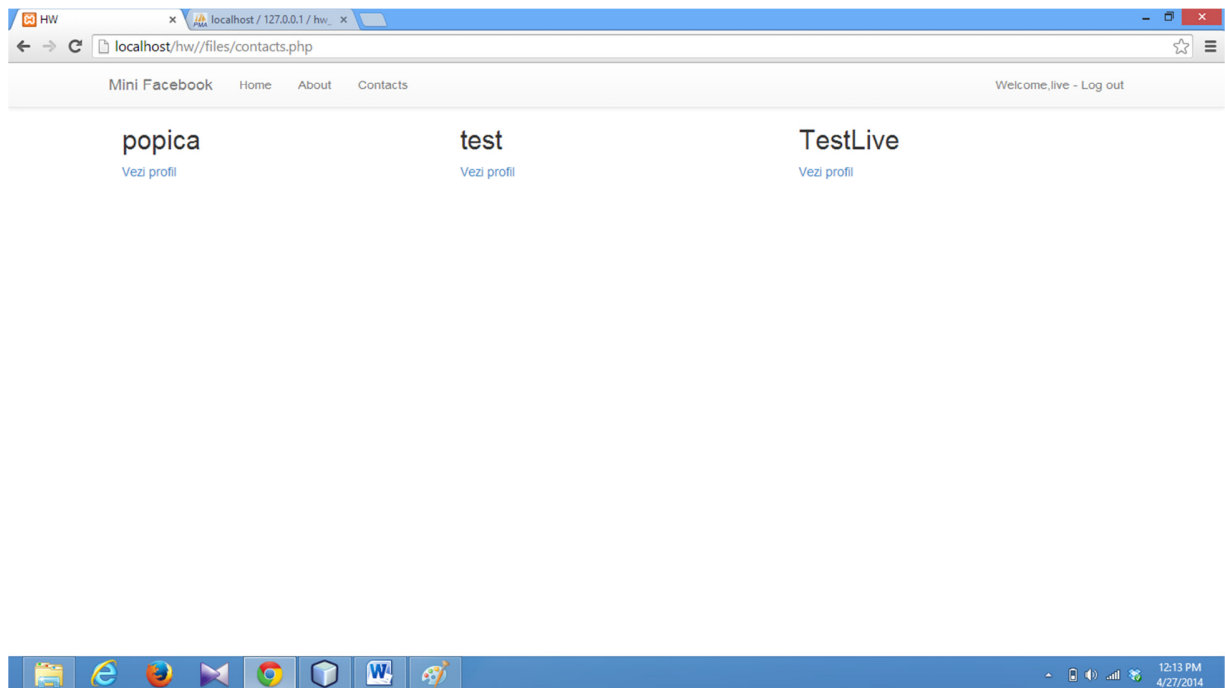
Salvarea presupune stocarea numelui fisierului in baza de date in tabelul "user_profile" si stocarea imaginii in folderul images din proiect.

						id	user_id	picture_name	added	description	likes	
☐		Edit		Copy		Delete	2	1	t2.jpg	1395071315		10
☐		Edit		Copy		Delete	3	1	t2.jpg	1395071337		4
☐		Edit		Copy		Delete	4	2	jy.jpg	1395072642		1
☐		Edit		Copy		Delete	5	1	jy.jpg	1395073515	super	9
☐		Edit		Copy		Delete	6	2	IMG_0242.JPG	1395184338	descr	2
☐		Edit		Copy		Delete	7	2	IMG_1876.JPG	1398591291	catel	0
☐		Edit		Copy		Delete	8	3	IMG_0172.JPG	1398595823		0

Imaginea este vizibila acum in pagina de home .



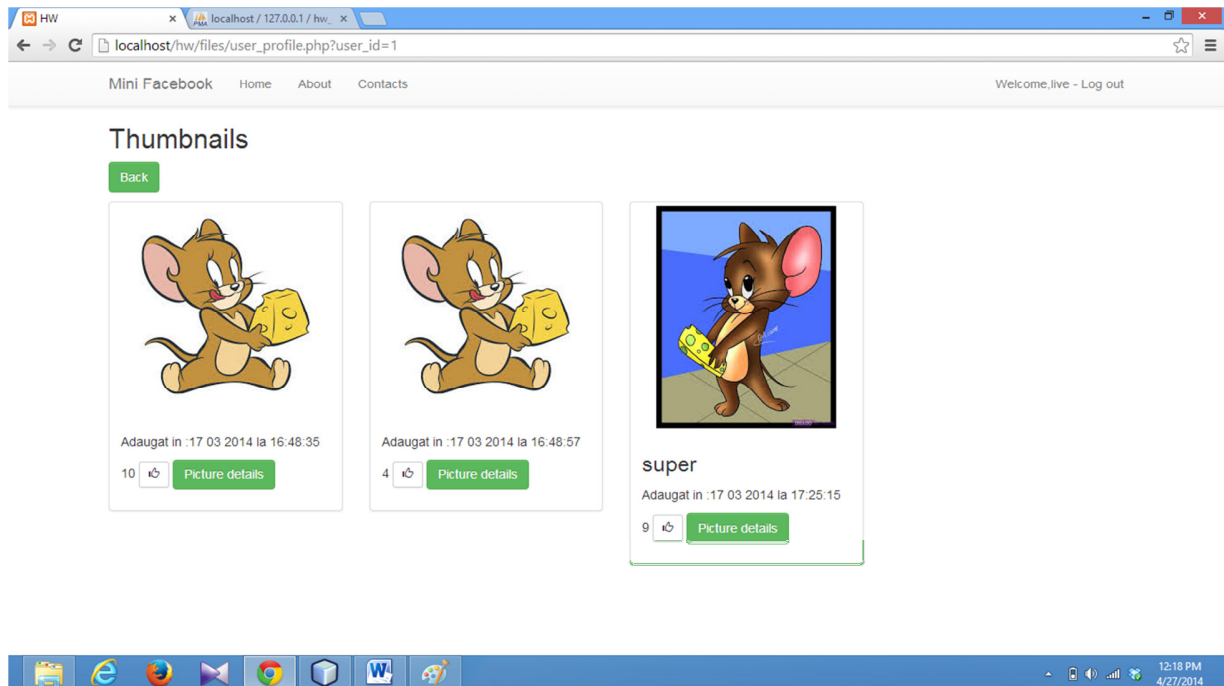
Aveti posibilitatea de a accesa pagina de contacts unde va apare o lista cu profilele userilor .



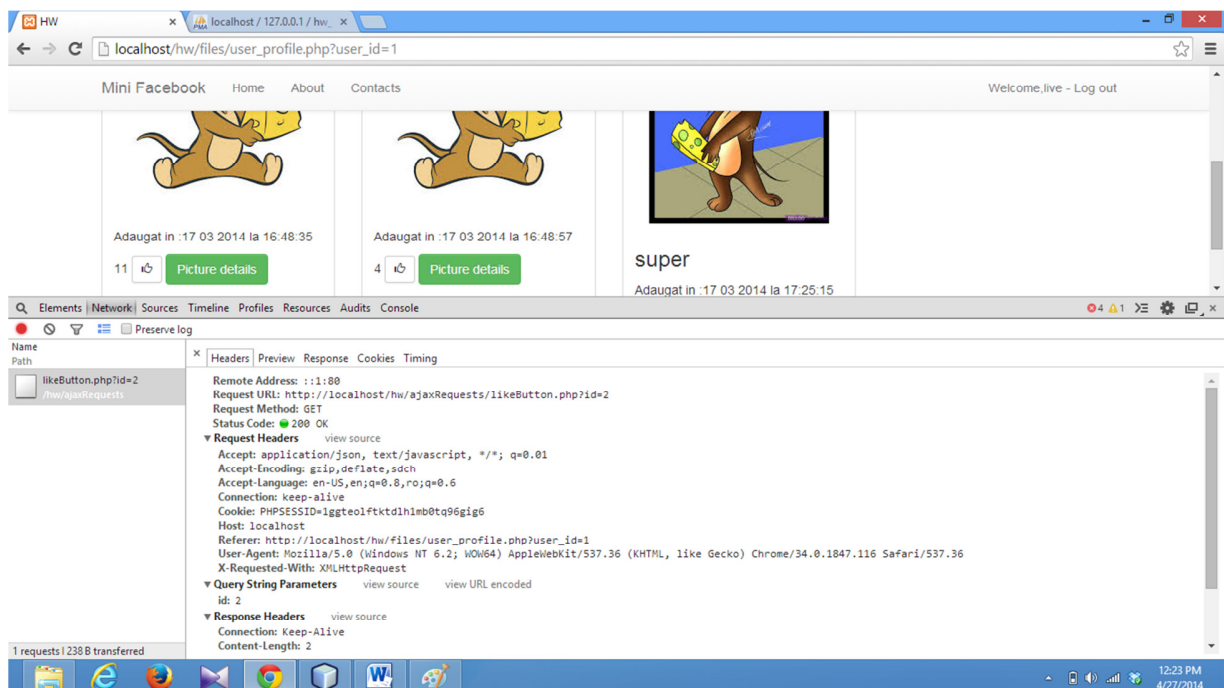
Pentru accesarea profilului unui user se face click pe linkul – vezi profil - . In acest link se trimite si id-ul userului pe care se face click.

Acest link va deschide pagina user_profile in care se face un GET pentru a prinde id-ul userului dorit .

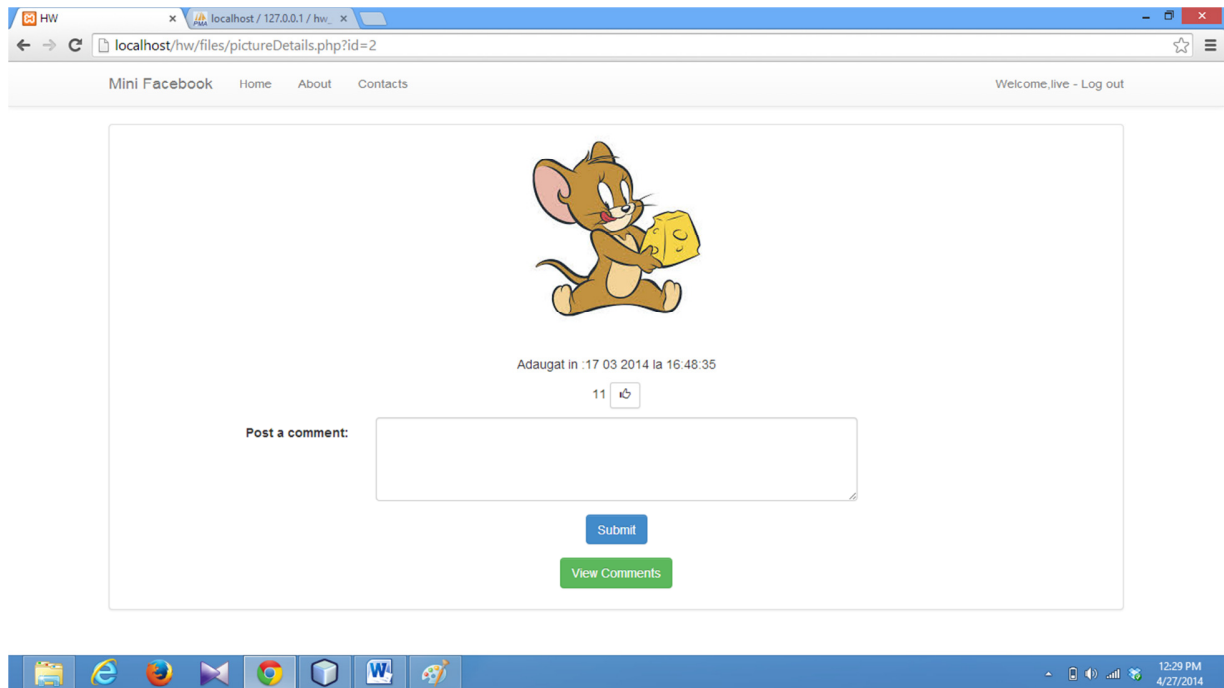
Deschidem profilul userului 'popica'.



Se deschide profilul acestui user si se afiseaza toate pozele acestuia . Butonul de like executa un call AJAX care va updata numarul actual de likeuri la poza respective si va trimite ca raspuns numarul nou de likeuri .



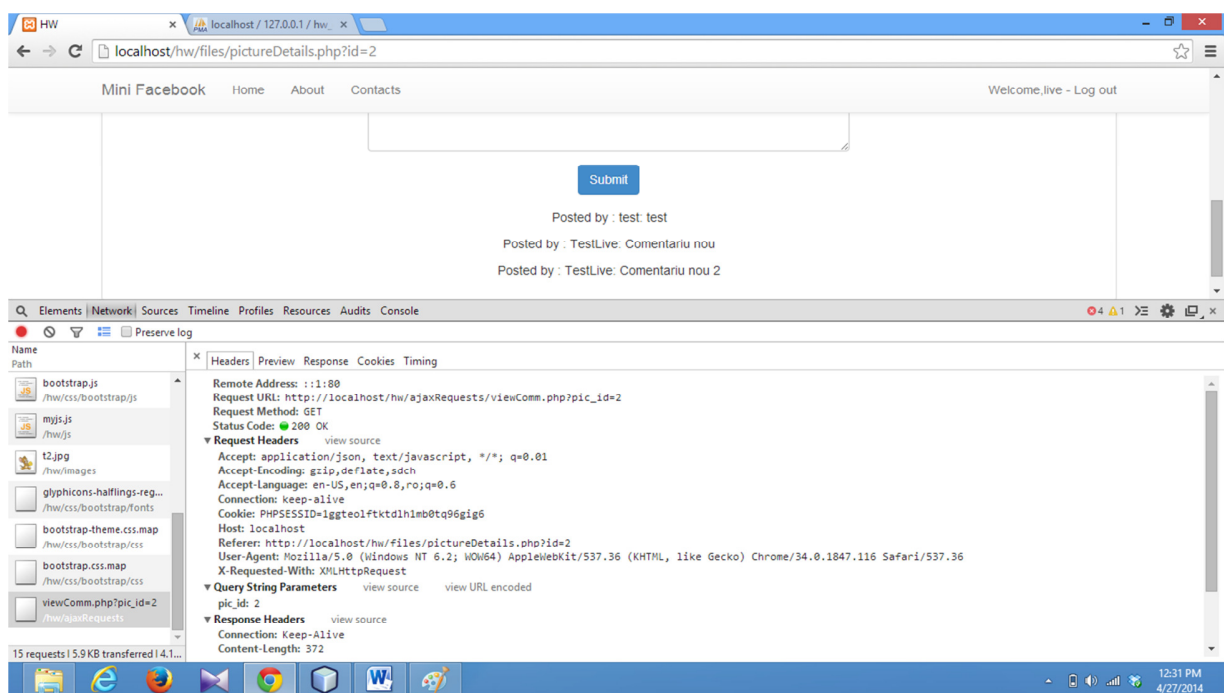
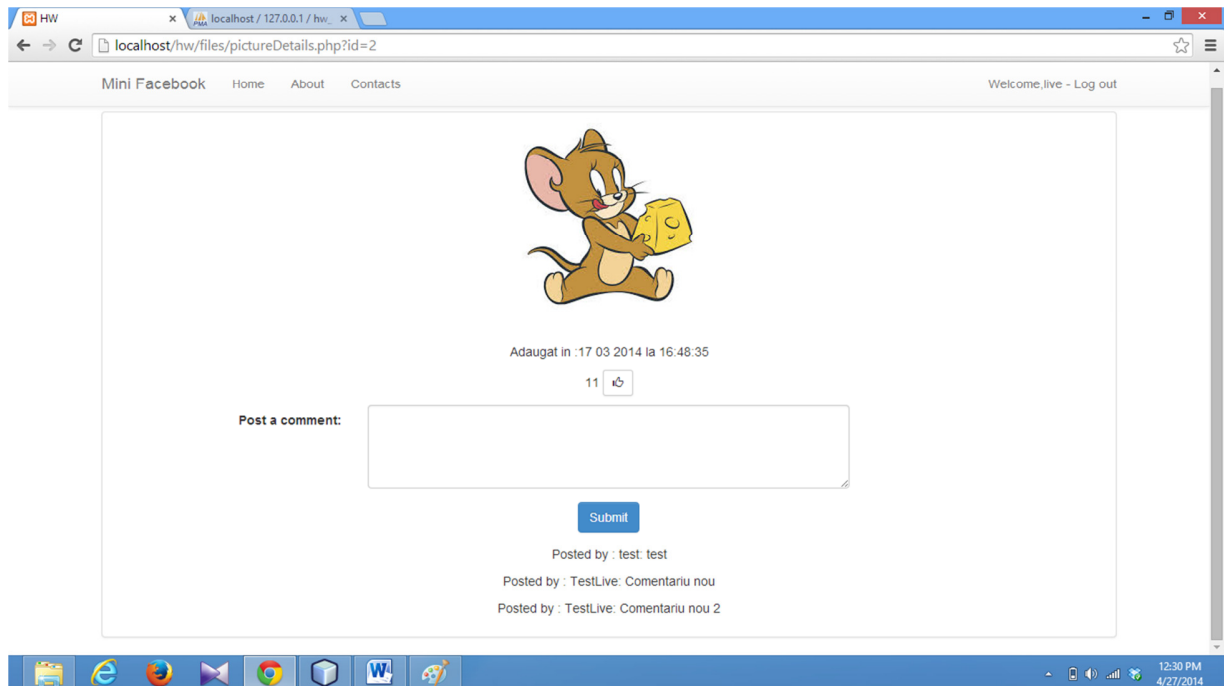
Fiecare poza are optiunea de a o deschide la o dimensiune mai mare si a vizualiza descrierea pozei si comentariile adaugate la poza .



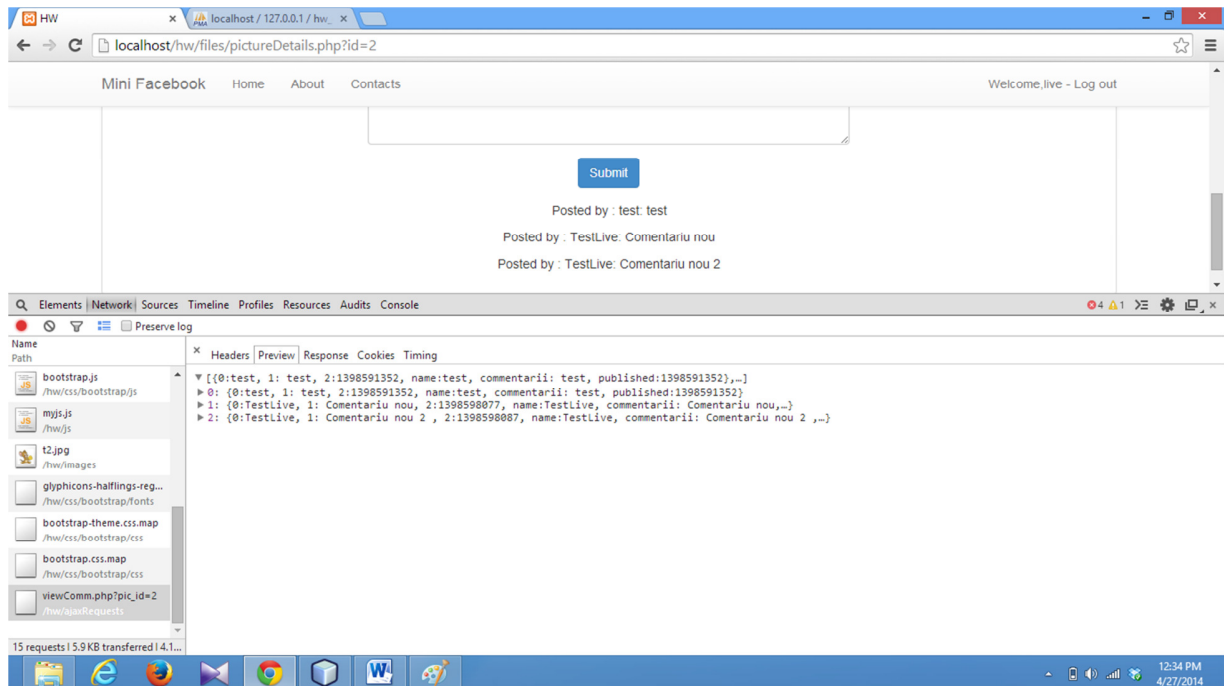
Din nou cele doua optiuni :

- submit
- view comments

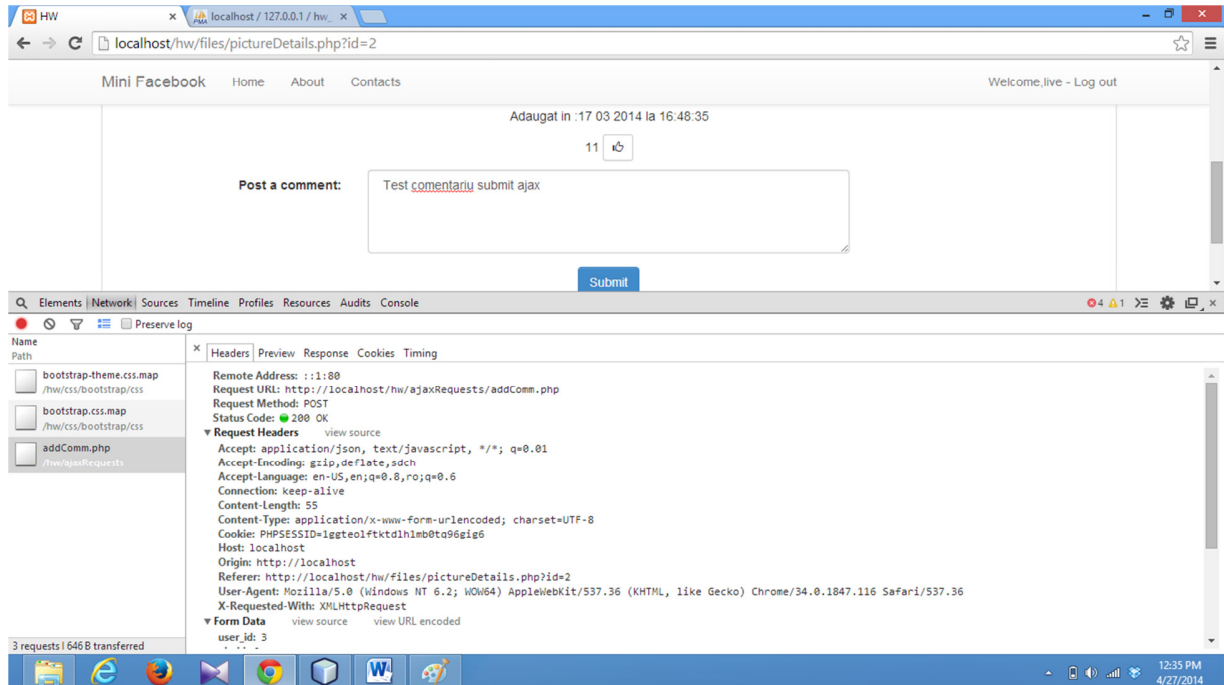
sunt actionate de functii javascript care fac call de AJAX.



De data aceasta raspunsul primit este un json (array de obiecte) ce contine informatii despre comentariu si userul care l-a adaugat.



Deasemenea submitul comentariului este tot prin AJAX :



De data aceasta metoda prin care am trimis datele este POST.

Pentru realizarea acestui proiect m-am folosit atat de tehnologii server-side cat si client-side .

Am optat pentru o organizare a fisierelor cat mai simpla si usor de folosit . :

Root

- ajaxRequests

- components

- css

- files

- images

- js

- cele mai importante fisiere le-am lasat in root