

## Proiect ~Mini Facebook~

Proiectul mini facebook este un proiect ce simuleaza :

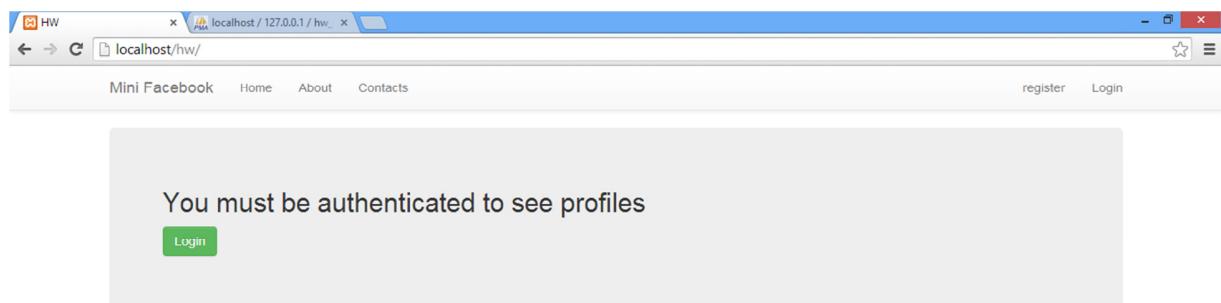
- Inregistrarea unui user
- Creare profil
- Upload de poze profil
- Vizualizare profilului altor useri
- Adaugare like , comentarii pozelor altor useri

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 sunteți încă un utilizator aveți posibilitatea de a va înregistra accesând opțiunea de register :

Copyright @Alex

Daca aveți deja un cont alegeți opțiunea login :

Copyright @Alex

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

Aşa arată tabelul userilor :

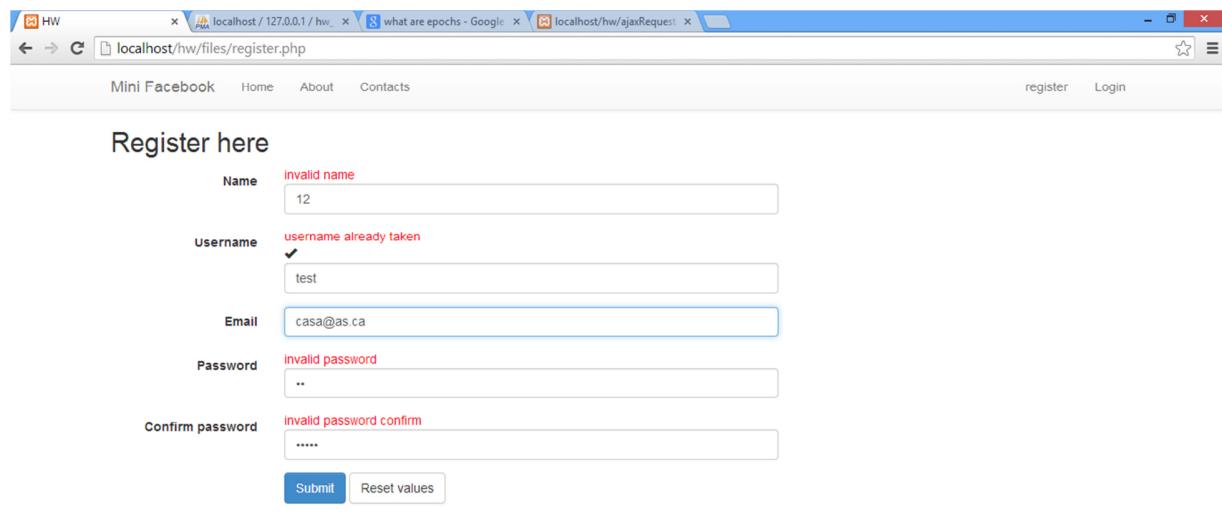
			<b>id</b>	<b>name</b>	<b>username</b>	<b>email</b>	<b>password</b>	<b>created_at</b>
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	popica	alexandru	email@trest	81dc9bdb52d04dc20036dbd8313ed055
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	test	test	test@test2.com	81dc9bdb52d04dc20036dbd8313ed055

Vom crea un nou user cu datele :

- name – TestLive
- username – live
- email – [testlive@minif.co](mailto:testlive@minif.co)
- password – 123456
- created reprezinta epoch timestamp generat în momentul realizării contului

Fiecare dintre campurile de la înregistrare dispune de o validare realizată cu ajutorul Javascriptului și jquery .

Exemplu date scrise gresit :



The screenshot shows a registration form titled "Register here" on a website called "Mini Facebook". The form fields and their validation messages are:

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

At the bottom of the form are two buttons: "Submit" and "Reset values". Below the form, the copyright notice "Copyright @Alex" is visible.

Pentru validarea user-namului am folosit un call de AJAX care verifica baza de date dacă user-namul există și returnează 0 în cazul în care acesta este prezent în baza de date și 1 dacă nu există.

The screenshot shows a web browser window with multiple tabs. The active tab is 'localhost/files/register.php'. The page content is a registration form with fields for Name and Username. The 'Username' field contains 'test' and has a red error message: 'username already taken'. Below the form is a 'register' button and a 'Login' link.

The developer tools Network tab is open, showing a request to 'localhost/hw/ajaxRequests/validations.php?username=test'. The Headers section shows the following details:

- Remote Address: ::1:80
- Request URL: http://localhost/hw/ajaxRequests/validations.php?username=test
- Request Method: GET
- Status Code: 200 OK
- Accept: text/plain, \*/\*; q=0.01
- Accept-Encoding: gzip,deflate,sdch
- Accept-Language: en-US,en;q=0.8,ro;q=0.6
- Connection: keep-alive
- Cookie: PHPSESSID=lggteolftktdlh1mb0tq96gig6
- Host: localhost
- Referer: http://localhost/hw/files/register.php
- User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.116 Safari/537.36
- X-Requested-With: XMLHttpRequest

The Response Headers section includes:

- Content-Type: application/json; charset=UTF-8
- Content-Length: 1
- Date: Mon, 04 Aug 2014 11:21:45 GMT
- Server: Apache/2.4.7 (Ubuntu)

The response body is empty, indicated by the '0' in the preview pane.

This screenshot shows the developer tools Response tab for the same request. The Headers section is identical to the previous screenshot. The Response tab shows the following JSON data:

```
{
  "status": "error",
  "message": "username already taken"
}
```

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

	<input type="button" value="←"/>	<input type="button" value="→"/>		<input type="button" value="▼"/>	<b>id</b>	<b>name</b>	<b>username</b>	<b>email</b>	<b>password</b>	<b>created_at</b>
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		1	popica	alexandru	email@trest	81dc9bdb52d04dc20036dbd8313ed055	1395059582
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		2	test	test	test@test2.com	81dc9bdb52d04dc20036dbd8313ed055	1395072019
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>		3	TestLive	live	testlive@minif.co	e10adc3949ba59abbe56e057f20f883e	1398594452

Un user beneficiaza de toate serviciile site-ului pe cand un simplu vizitator este restrictionat 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:

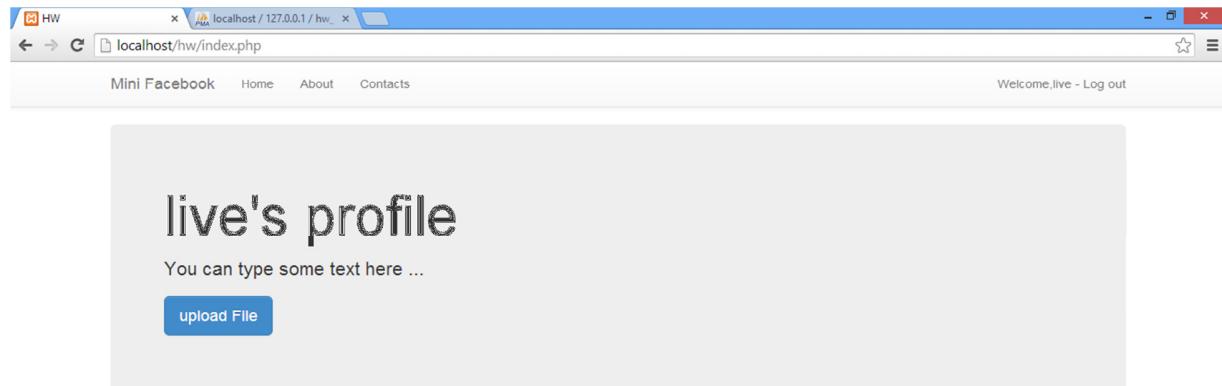
```
$loginClass->setState('user_id', $result[0]['id']);  
$loginClass->setState('username', $result[0]['username']);  
$loginClass->setState('Created', time());  
$loginClass->setState('expire', time() + 1800);
```

Unde setState este metoda clasei "LoginClass"

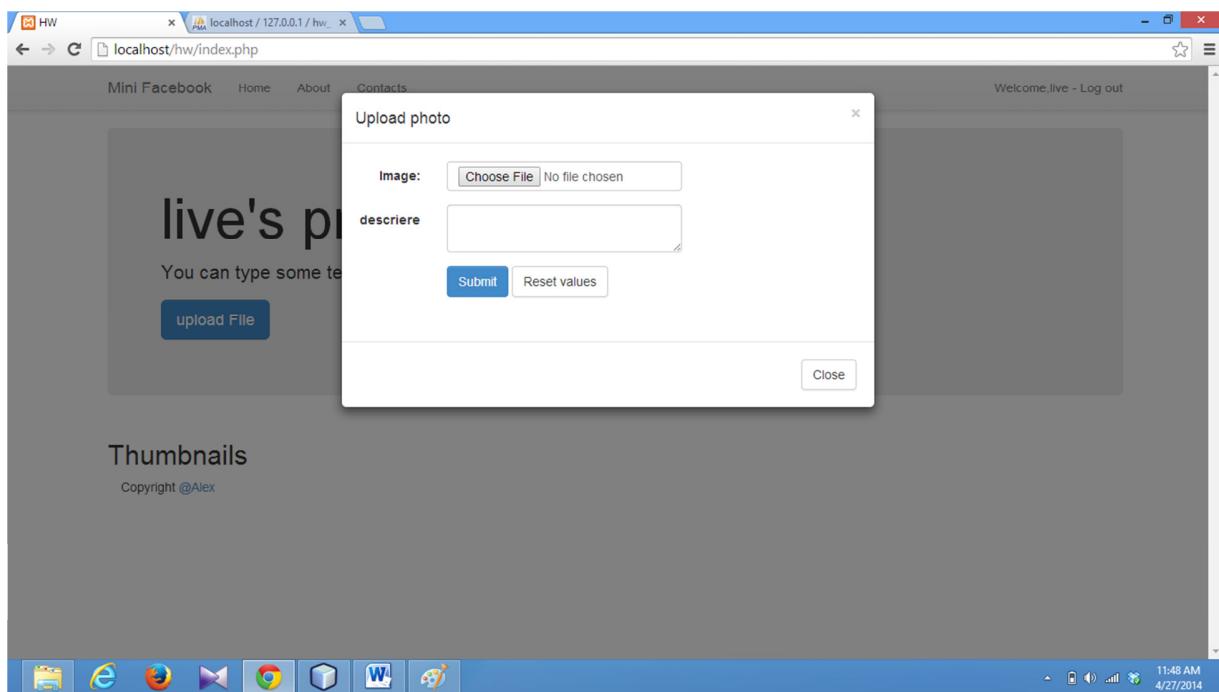
```
public function setState($name,$value)  
{  
    $_SESSION[$name]=$value;  
}
```

O sesiune este valabila pentru 30 de minute.

Folosind userul proaspăt creat , după logare pagina de home se modifica :



Acum se poate folosi butonul de upload pentru adăugarea 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 aceestei 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;
}

```

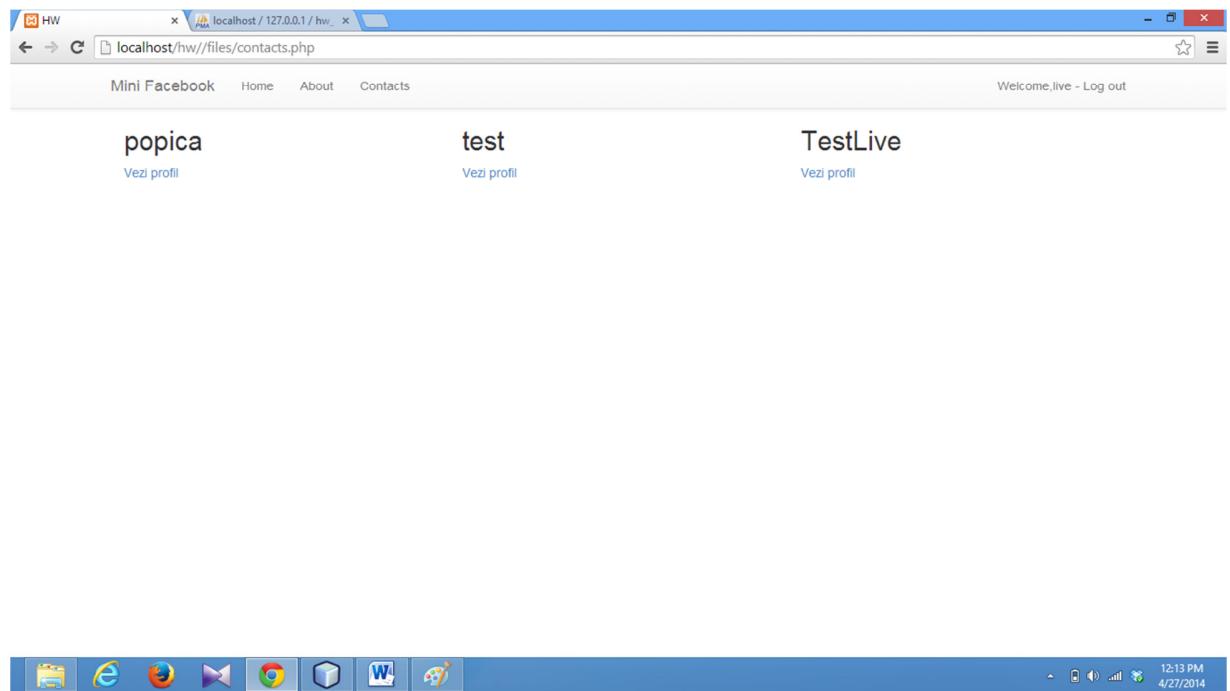
Salvarea presupune stocarea numelui fisierului in baza de date in tabelul "user\_profile" si stocarea imaginii in folderul images din proiect.

	<input type="button" value="←"/>	<input type="button" value="→"/>		<input type="button" value="▼"/>	<b>id</b>	<b>user_id</b>	<b>picture_name</b>	<b>added</b>	<b>description</b>	<b>likes</b>
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	1	1	t2.jpg	1395071315		10
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	1	1	t2.jpg	1395071337		4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	2	2	jy.jpg	1395072642		1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	1	1	jy.jpg	1395073515	super	9
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	2	2	IMG_0242.JPG	1395184338	descr	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	2	2	IMG_1876.JPG	1398591291	catel	0
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	3	3	IMG_0172.JPG	1398595823		0

Imaginea este vizibila acum in pagina de home .

The screenshot shows a web application window titled "HW" running on "localhost / 127.0.0.1 / hw\_". The URL in the address bar is "localhost/hw/index.php". The page header includes "Mini Facebook" links and "Welcome,live - Log out". The main content area displays "live's profile" with a "upload File" button. Below it is a thumbnail image of a black and tan dog, with the caption "Adaugat in :27 04 2014 la 12:50:23" and a "Picture detail" button. At the bottom of the page, there is a footer with "Copyright @Alex" and various system icons.

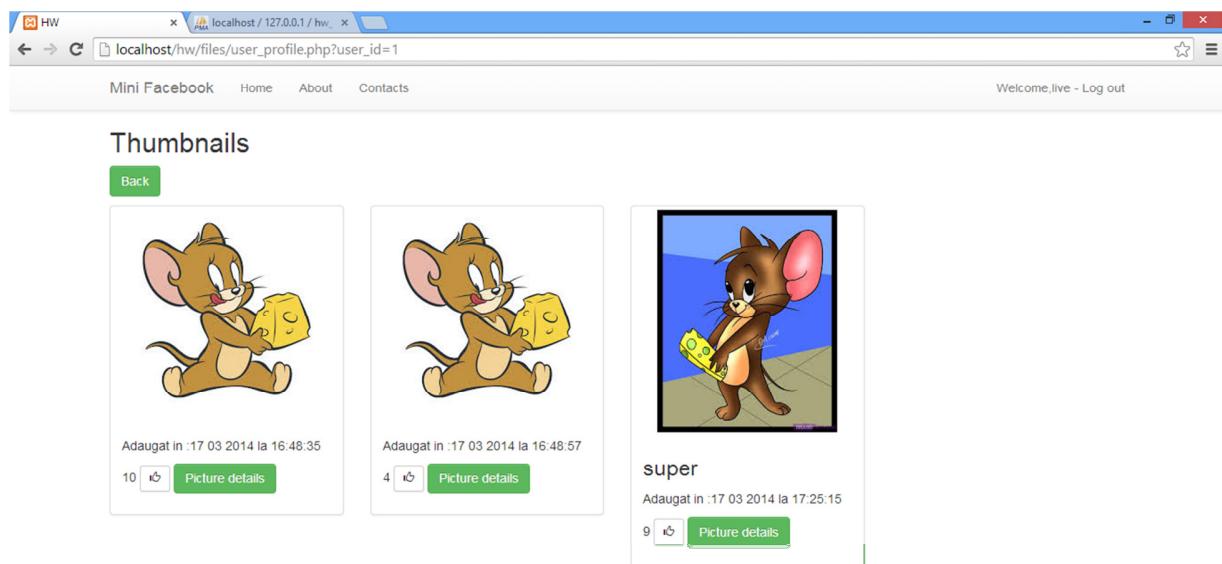
Aveti posibilitatea de a accesa pagina de contacts unde va aparea o lista cu profilele userilor .



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 update numarul actual de likeuri la poza respective si va trimite ca raspuns numarul nou de likeuri .

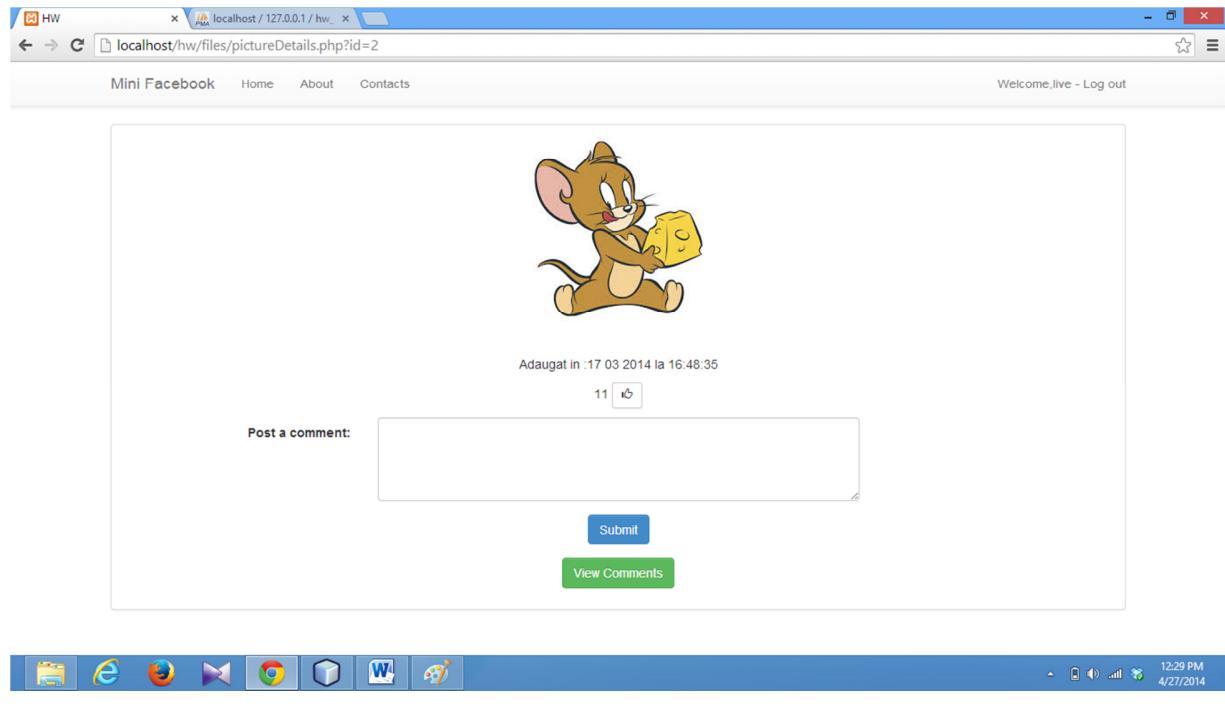
Screenshot of a web browser showing a user profile page. The page displays three images of a cartoon mouse. The first image has a like count of 11, the second of 4, and the third of 9. Each image has a 'Picture details' button below it. The browser's developer tools Network tab is open, showing a request to 'likeButton.php?id=2'. The request details show the following headers:

```

    Remote Address: ::1:80
    Request URL: http://localhost/hw/ajaxRequests/likeButton.php?id=2
    Request Method: GET
    Status Code: 200 OK
    Request Headers:
        Accept: application/json, text/javascript, */*; q=0.01
        Accept-Encoding: gzip,deflate,sdch
        Accept-Language: en-US,en;q=0.8,ro;q=0.6
        Connection: keep-alive
        Cookie: PHPSESSID=lggeteoiftktdlh1mb0tq96gig6
        Host: localhost
        Referer: http://localhost/hw/files/user_profile.php?user_id=1
        User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/34.0.1847.116 Safari/537.36
        X-Requested-With: XMLHttpRequest
    Query String Parameters:
        id: 2
    Response Headers:
        Connection: Keep-Alive
        Content-Length: 2
  
```

The browser's taskbar at the bottom shows the date and time as 12:23 PM on 4/27/2014.

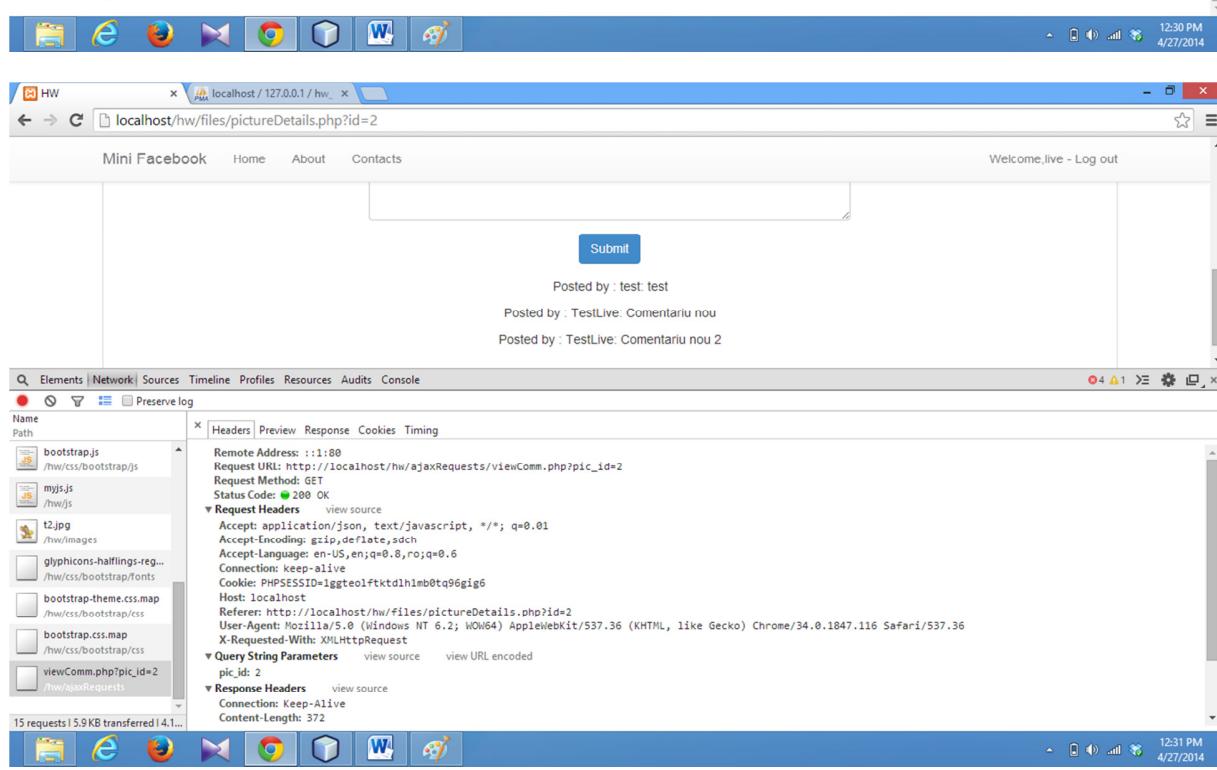
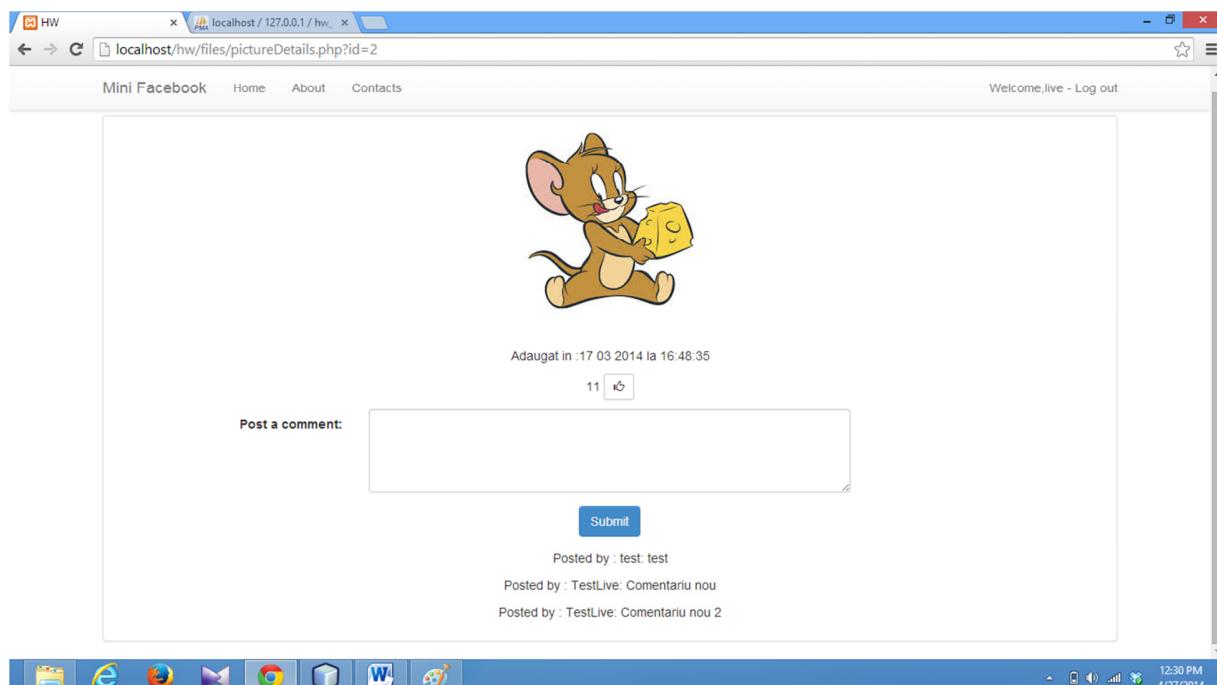
Fiecare poza are optiunea de a o deschide la o dimensiune mai mare si a vizualiza descrierea pozei si comentariile adaugate la poza .



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.

The screenshot shows a browser window with the URL `localhost/hw/files/pictureDetails.php?id=2`. The page content includes a navigation bar with links to Home and About, and a footer with 'Welcome, live - Log out'. The main area displays a post by 'test' with three comments from 'TestLive'. The developer tools Network tab is open, showing requests for Bootstrap files like `bootstrap.js`, `bootstrap-theme.css.map`, and `bootstrap.css.map`, along with a comment submission request to `viewComm.php?pic_id=2`. The status bar at the bottom right shows the date as 4/27/2014.

Deasemenea submitul comentariului este tot prin AJAX :

The screenshot shows a browser window with the same URL as the previous one. The page content is identical, but the developer tools Network tab shows a successful POST request to `addComm.php` with a status code of 200 OK. The request headers and form data are visible, indicating the use of JSON for the comment submission. The status bar at the bottom right shows the date as 4/27/2014.

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