Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Tehnologii Informaționale

**RAPORT**

Lucrare de laborator Nr.6

*la Medii Interactive de Dezvoltare a Produselor Soft*

A efectuat: Rosca Andrei

st.gr.TI-144

A verificat: Cojanu Irina

Lector asistent,

Chişinău 2016

**Lucrarea de laborator nr.6**

**Tema:**  Lucru in echipa. Aplicatie complexa

**Scopul lucrării:**

* Crearea unei aplicatii complexe in echipa.
* Divizarea sarcinilor pe membrii echipei

**Formularea condiţiei problemei (sarcina de lucru):**

*Advanced Level* (nota 9 || 10):

* Dezvoltarea unei aplicatii:
  + Web
  + Browser Extension

Incercati sa aplicati cit mai multe din thnologiile noi invatate:

* Integrarea cu baza de date
* Folosirea API
* Cross platform
* User friendly

***Mersul lucrării :***

**Limbajul de programare folosit :** PHP, JavaScript, HTML/CSS

**Framework-uri :** Laravel MVC Framework

**Tehnologii :** AJAX, JSON, Bootstrap, Metro UI

**Aplicație WEB :** Revistă ştiinţifico-practică medicală online a asociației chirurgilor pediatrici din Republica Moldova

**Divizarea task-urilor și lucrul pe GIT:**

**Branch /master/ -** Proiect final

<https://github.com/AndreiRosca/acp>

**Branch /logic-layer/ -** Andrei Roșca

<https://github.com/AndreiRosca/acp/tree/logic-layer>

**Branch /presentation-layer/ -** Pascari Ion

<https://github.com/AndreiRosca/acp/tree/presentation-layer>

**Branch /logic-layer/ -** Andrei Roșca

Responsabilități :

* Crearea model-urilor, si a logicii de interactiune cu baza de date. Crearea structurii bazei de date, implementarea autentificarii in pagina de administrare. Am implementat controllerul ce prelucreaza logica plasarii publicitatii.

Exemplu : Scriptul ce implementeaza logica search-ului in daza de date.

**………………………………………**  
  
<?php

**namespace** App;

**use** Illuminate\Database\Eloquent\Model;

**use** DB;

**class** Search **extends** Model

{

**public** **function** search($keywords, $searchField, $searchMode) {

$query = DB::table('articles')->

select('english\_title', 'romanian\_title',

'authors', 'institution', 'english\_description', 'romanian\_description', 'article\_file\_name',

'article\_file\_mime', 'article\_resume');

$searchTerms = **explode**(" ", $keywords);

**if** ($searchMode == 'and') {

**foreach** ($searchTerms **as** $term)

$query->where($searchField, 'like', '%'.$term.'%');

} **else** {

**foreach** ($searchTerms **as** $term)

$query->orWhere($searchField, 'like', '%'.$term.'%');

}

$articles = $query->get();

$articleList = **array**();

**foreach** ($articles **as** $a) {

$articleList[] = **array**('english\_title' => $a->english\_title,

'romanian\_title' => $a->romanian\_title,

'authors' => $a->authors,

'institution' => $a->institution,

'english\_description' => $a>english\_description,

'romanian\_description' => $a->romanian\_description,

'article\_file\_name' => $a->article\_file\_name,

'article\_resume' => $a->article\_resume);

}

**return** $articleList;

}

}

Exemplu : Scriptul ce implementeaza logica banerilor de publicitate.

**………………………………………**

<?php

**namespace** App\Http\Controllers;

**class** AdController **extends** Controller

{

**public** **function** handleAdInsertion() {

$ad = **new** \App\Ad();

**if**(Input::hasFile('new\_banner\_name')) {

$f = Input::**file**('new\_banner\_name');

$fileName = **date**('Y-m-d\_H-m-s-') . $f->getClientOriginalName();

$ad->insertAd(Input::get('empty\_banner\_id'),

Input::get('link\_of\_banner'), $fileName, $f->getMimeType());

Storage::put('./public/images/' . $fileName,

**file\_get\_contents**($f->getRealPath()));

}

**return** Redirect::away("admin");

}

**public** **function** handleAdDeletion() {

$ad = **new** \App\Ad();

$ad->removeAd(Input::get('existent\_banner\_id'));

**return** Redirect::away("admin");

}

**public** **function** getAdList() {

$ad = **new** \App\Ad();

**return** view('ad\_list', **array**('adList' => $ad->getAdsList()));

}

**public** **function** getAdImage($imageName) {

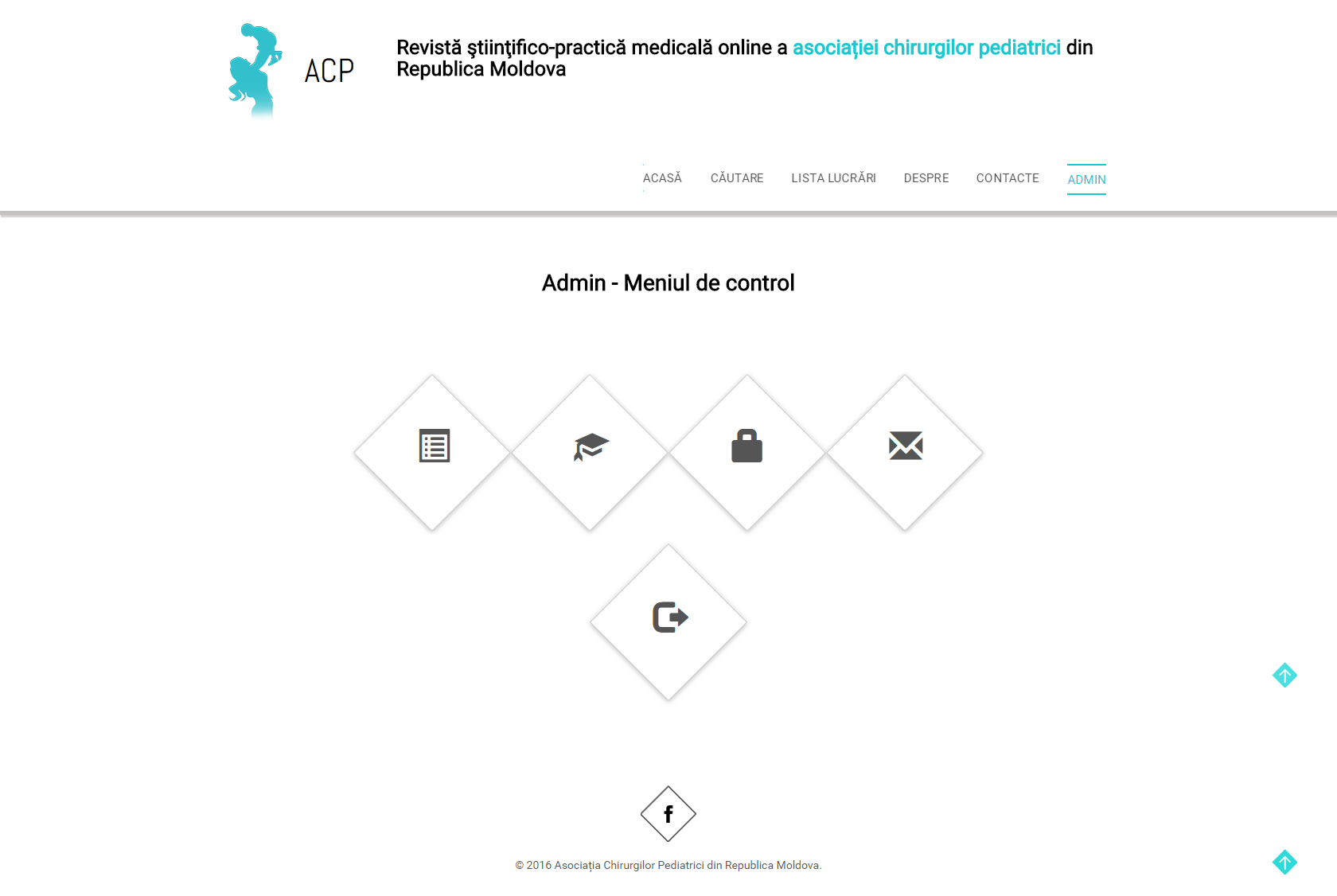
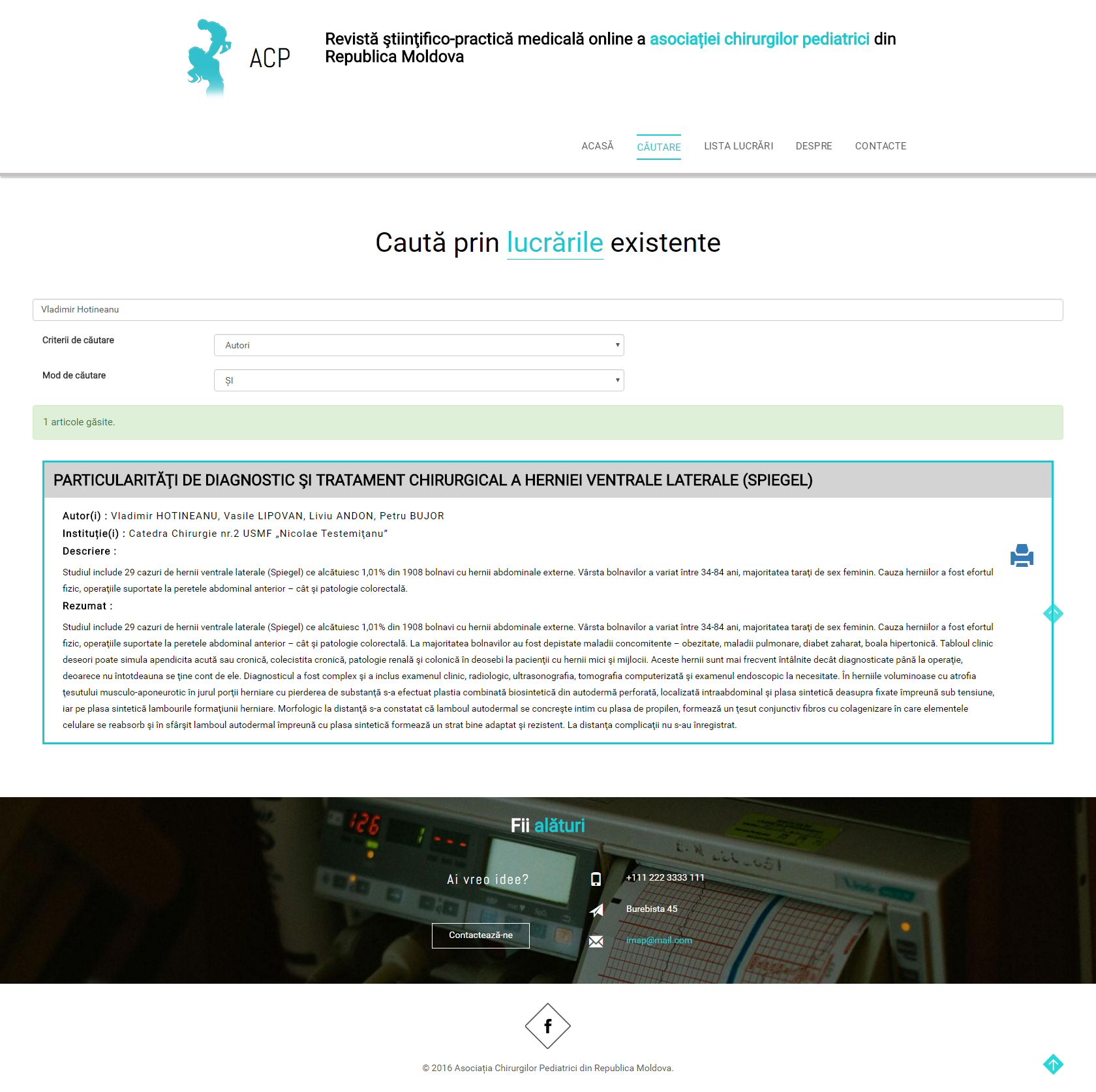
$ad = **new** \App\Ad();

**return** Response::make($ad->getFile($imageName), 200,

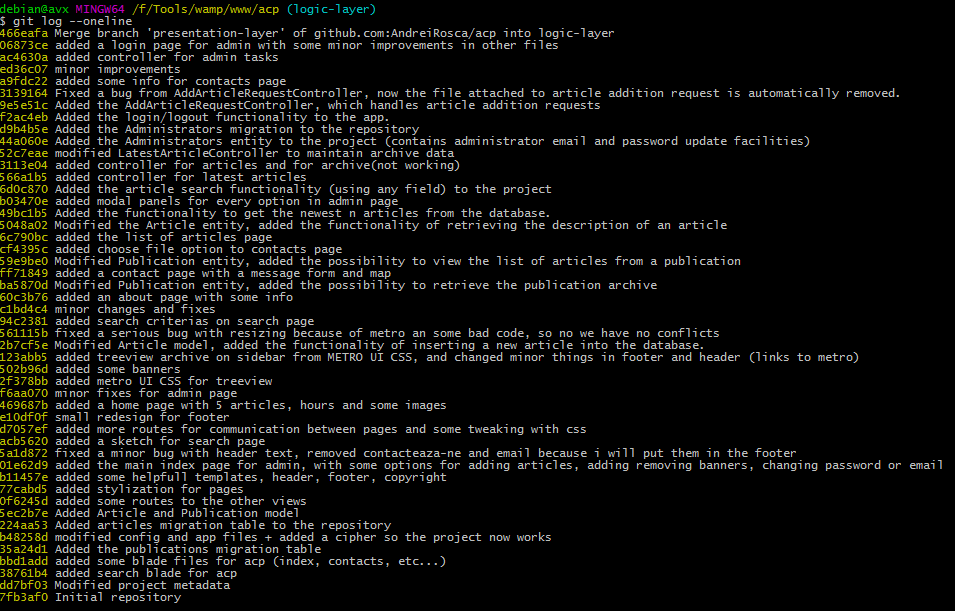
**array**('content-type' => 'image/png'));

}

}



* Lucrul pe un branch cu adunarea a minim 15 commit-uri

****

**Concluzii**

În această lucrare am pus în practică noțiunea de lucru în echipă și anume utilizare GIT-ului pentru a lucra pe branch-uri diferite ca ulterior să facem merge și să avem un proiect cât de cât funcțional. Am obținut deprinderi practice de lucru cu un framework MVC în limbajul de programare PHP. De ce Laravel?

Fiindcă Laravel are numeroase functionalitati care-l transforma intr-un framework extrem de puternic

* Are Facades(clase ale caror metode pot fi apelate ca si metode statice, ex: *Route::get()*, *Input::get()*, *Input::all()* si multe altele).
* Are ORM, deci poti transforma *$articole = DB::table('articole')->get();* in *$articole = Articol::all();*. Un ORM ofera numeroase functionalitati ce-ti confera un control imens asupra aplicatiei.
* Are o comunitate, poti intra pe [http://laravel.io](http://laravel.io/) si sa incepi discuta cu ceilalti programatori care folosesc Laravel daca doressti.
* Design Patterns - Folosind Laravel o sa inveti numeroase design patterns, un subiect frecvent dezbatut printre programatorii php avansati. Dupa cum spun unele voci, dupa ce inveti un framework trebuie sa inveti design patterns, dar de ce sa nu le inveti pe amandoua in acelasi timp?
* Si inca extrem de multe, crede-ma: nu-mi ajunge o zi sa scriu toate functionalitatile!

**Bibliografie**

* <http://php.net/>
* <https://www.mysql.com/>
* Îndrumar metodic pentru lucrările de laborator la MIDPS