

# Proiect AW pentru echivalarea laboratorului

## Tema proiectului

Implementarea unei „agende” sau „director” de contacte – o aplicație web pentru gestionarea contactelor.

## Cerințe

### Generale

Proiectul poate fi scris în orice limbaj de programare, folosind orice infrastructură (framework) disponibilă astăzi, cu condiția ca autorii să înțeleagă principiile de funcționare ale infrastructurii respective – și să implementeze și ceva original.

*Spre exemplu, utilizarea framework-urilor CakePHP, sau Django, pentru python este permisă.*

**Proiectul trebuie să fie munca voastră. În caz contrar, punctajul vostru pentru laborator va fi 0. Tot codul copiat (sau inspirat din) terțe surse trebuie creditat (este suficient link-ul).**

**Tema de proiect, ca și cerințele, este foarte generală – ceea ce înseamnă că nu există niciun motiv plauzibil pentru ca două proiecte să arate asemănător (atât vizual cât și ca și cod).**

Se va încerca oferirea unei experiențe cât mai bune pentru utilizatori – cât mai puține click-uri pentru a ajunge undeva în aplicație, meniuri accesibile. Utilizați AJAX oriunde considerați că ar îmbunătăți experiența utilizatorului.

Toate erorile care apar în funcționarea proiectului trebuie tratate, în așa fel încât utilizatorul să nu primească ecrane albe sau cu erori pe care nu le poate înțelege.

## Organizatorice

Trebuie respectate următoarele deadline-uri intermediare:

- **11.12.2011** – Îmi comunicați la [andrei.avram@gmail.com](mailto:andrei.avram@gmail.com) tehnologia cu care vă doriți să lucrați, pentru a putea face configurările necesare pe server. Motivul pentru care timpul este așa de scurt este că se presupune că faceți proiectul pentru că deja știți să lucrați măcar cu o tehnologie web. Pe 5.12 veți primi un cont propriu, care va avea drepturi de FTP și SSH pe serverul nostru de dezvoltare.
- Serverul rulează Linux (Ubuntu). Este permis să faceți și o aplicație bazată pe tehnologii Microsoft, dar atunci este nevoie să vă asigurați singuri host-are (undeva pe Internet). Dacă sunteți în acest caz *vă rog să mă contactați înainte de 12.12.2011, să căutăm împreună soluții*
- PHP, Perl, Python și eventual Ruby, cu MySQL și Postgresql sunt suportate by default. Django, CakePHP și alte framework-uri, dar și diferite setări rezonabile pentru serverul web sunt
- **23.12.2011** – Partea de creare de utilizatori și autentificare (chiar și către o pagină în care se afișează doar username-ul) este încărcată pe site. Este implementată funcționalitatea CRUD și lista, chiar dacă nu este încă complet funcțională.
- **6.01.2012** – Proiectul final, cu toate funcționalitățile. Se va trimite codul, documentat, și un fișier readme în care se explică structura codului, care au fost provocările întâmpinate și cum s-au depășit.

- **O întâlnire** de evaluare individuală va fi stabilită în ultima săptămână de școală.

## **Punctare**

Punctajul pentru proiect (și, în consecință, pentru laborator), este împărțit astfel:

- 20% respectarea deadline-urilor
- 5% existența fișierului readme
- 50% implementarea tuturor cerințelor
- 25% aprecierea colegilor (vezi mai jos)

## **Note:**

- Pentru contribuții creative și idei deosebite, se poate oferi **bonus**.
- **25% din punctaj** se va da în baza notelor date de colegii voștri, care **vor testa proiectele voastre** în ultima săptămână (între 6 și 13 ianuarie). Pentru asta, proiectul nu trebuie să conțină niciunde numele vostru, și vom anonimiza URL-ul.

## **Structuri de date**

### **Contact**

Pentru fiecare persoană pentru care se vor reține date, se vor reține minim

- Nume și prenume
- Adresa de email
- Numărul de telefon
- Adresa poștală
- O fotografie (obligatoriu ca și câmp, dar opțională la completare)
- Username-ul pentru Facebook
- Username-ul pentru Twitter

### **Utilizator**

Pentru fiecare utilizator al aplicației noastre, vom reține:

- Username
- Parola (într-un format care să nu poată fi folosită de administratorii site-ului)
- Adresa de email
- Data ultimei autentificări

## **Funcționalități**

### **Multi – user**

Aplicația va fi multi-user – fiecare utilizator va avea acces doar la contactele proprii, după ce trece printr-un proces de autentificare. Fiecare utilizator va avea contul lui, în care va avea ca minime acțiuni:

- Dezautentificarea
- Schimbarea parolei (dacă este cazul, vezi nota de mai jos)

Indiferent dacă utilizatorii sunt locali sau din alte surse de utilizatori, se vor păstra înregistrări pentru fiecare autentificare și deautentificare a utilizatorului, împreună cu IP-ul de la care s-a făcut accesare.

## Login

Utilizatorilor neautentificați li se va prezenta un formular de autentificare.

## Creeare utilizator nou

În cazul în care un utilizator vrea să-și creeze un utilizator nou (pentru cazul cu bază de date proprie, nu cu OAuth), utilizatorului îi va fi prezentat un formular de înscriere, care, după ce va fi validat că este corect și complet, va crea utilizatorul și îl va autentifica.

*Notă: Gestiunea utilizatorilor se poate face în orice formă, fie printr-un tabel în baza de date (soluția clasică), fie prin OAuth, Facebook, Google login.*

## Lista contactelor

Va afișa o listă a tuturor contactelor persoanei respective, cu următoarele funcționalități:

- paginare\* - dacă sunt mai mult de X contacte, se vor afișa doar X contacte, și un meniu de navigare între paginile de contacte.
- filtrare / căutare\* - selectarea doar a acelor contacte care corespund criteriilor de căutare – printr-un formular special. Filtrarea se va face cel puțin după nume și prenume (opțional)
- link-uri către pagina de detalii / editare / ștergere pentru fiecare contact, și un link pentru adăugarea de noi contacte

## Pagina de detalii

Va afișa detaliile pentru contactul respectiv, inclusiv fotografia (dacă contactul nu are fotografie, atunci se va afișa o fotografie default). Pagina va avea și următoarele funcționalități:

- Dacă adresa este introdusă, va folosi Google Maps (sau un alt serviciu de mapping, Bing, Yahoo ...) pentru a afișa o hartă interactivă cu un marker pe adresa indicată în câmpul adresă al persoanei
- Dacă contul de Facebook este introdus, va afișa ultimul status de pe Facebook al persoanei (opțional, sunt colegi care nu folosesc Facebook, iar status-urile sunt limitate la persoane care sunt relaționate pe Facebook).
- Dacă contul de Twitter este introdus, va afișa ultimul status de pe Twitter al persoanei (indiferent dacă aveți sau nu aveți Twitter, puteți să folosiți stream-ul public al unei persoane publice – <https://twitter.com/#!/BARACKOBAMA>, <https://twitter.com/#!/StewieGriffin>, sau al meu <https://twitter.com/#!/yetibalauru>)

## CRUD

Adăugare, Modificare și Ștergerea unui contact. Pentru Adăugare și Modificare trebuie implementate următoarele funcționalități:

- validarea informațiilor din formularul de introducere, cu afișarea eventualelor erorilor într-un loc în care rezolvarea lor să fie intuitivă
- upload-ul unei fotografii (dacă o fotografie nouă nu este încărcată se păstrează cea veche). Această funcționalitate poate fi implementată ca un formular separat.

Pentru Ștergere:

- cererea care va face efectiv ștergerea (fie ea și AJAX) se va face prin POST
- va fi cerută o formă de confirmare înainte de a face ștergerea
- în cazul în care contactul are o fotografie, la ștergerea contactului fotografia va fi și ea ștersă de pe disc.