# Laborator 7

**IMPORTANT!** În taskurile de mai jos, considerăm numele aplicației, ca fiind "aplicatie" deci căile cererilor vor începe, dupa localhost cu "/aplicatie". Când în exercițiu se vorbește despre o cerere, ne referim la restul linkului, de dupa stringul "/aplicatie".

## TASK 1 (1p) PROIECT!!!!

Utilizatorul va trebui să primească un mail de confirmare în momentul în care se înregistrează pe site. Taskul se va realiza în felul următor:

* Veți modifica modelul utilizator pentru a mai adăuga un câmp numit **cod**. Acesta va fi de tip sir de caractere cu lungime de maxim 100. Veti lăsa acest camp si cu posibilitatea de a fi nul (pentru utilizatorii deja creati). Veți mai crea si un câmp boolean numit **email\_confirmat**, nenul, implicit fals.
* La înregistrarea unui utilizator veți genera un string aleator care va fi setat în câmpul **cod** al utilizatorului
* Veți crea un template pentru e-mailul de confirmare în care veți ura "bun venit" utilizatorului pe site. În template se va specifica numele, prenumele și username-ul acestuia, cât și linkul de confirmare al e-mailului. Linkul de confirmare va avea formatul: /confirma\_mail/cod/ unde cuvântul cod va fi înlocuit cu codul generat, de exemplu:

/confirma\_mail/mdj0iSSA1957a1pQ

* **(Later edit) Template-ul e-mailului va contine si o imagine cu logo-ul site-ului (sau ce altceva vreti voi)**
* La click pe linkul respectiv se cauta in tabelul de utilizatori codul, si pentru utilizatorul respectiv se confirma e-mailul (se setează câmpul email\_confirmat pe True). Iar în pagina afișată se va afișa un mesaj de confirmare.
* Dacă un utilizator încearcă să se logheze înainte de a confirmat e-mailul, i se va afișa un mesaj de eroare.
* Utilizatorii se vor putea loga după ce au e-mailul confirmat.

## TASK 2 (1p) PROIECT!!!!

Veți implementa un sistem de promoții (realizat prin trimitere de mailuri), în următorul mod:

* Veți crea un tabel numit vizualizări unde veți salva ultimele N produse vizualizate de către utilizatori. De exemplu, N poate fi egal cu 5. Tabelul va conține id-ul utilizatorului, id-ul produsului și data vizualizării.
* Când se introduce în tabel o înregistrare cu o vizualizare nouă, se verifică dacă pentru utilizator nu se depășesc deja N vizualizări. Dacă se depășesc, se sțerge cea mai veche vizualizare.
* Veți crea un tabel numit promoții care va avea câmpurile nume, data\_creare, data\_expirare și minim înca 2 câmpuri alese de voi.
* Veți alege o caracteristică a produselor pentru care doriți să faceți promoții, de exemplu, categoria (dar poate fi absolut orice coloană din tabelul de produse). Se consideră că o promoție e valabilă pentru mai multe categorii în același timp. Veți face minim 2 template-uri de e-mail care vor conține **mesaje personalizate pentru categoriile pentru care se fac promoțiile**. Aceste template-uri vor trebui să aibă în conținut următoarele variabile:
  + subiectul mesajului
  + data expirării promotiei
  + eventuale alte date relevante
* Va exista o pagină, numită /promotii care va avea un formular cu câmpurile: subiect și mesaj, nume promotie, timp promotie, un choice field multiplu cu categoriile pentru care dorim să facem promotiile (veti pune doar categoriile pentru care aveti template-uri). Implicit sunt toate selectate, și câmpuri corespunzătoare coloanelor alese de voi
* La submit, se vor introduce datele unei noi promotii în tabel. În același timp se vor selecta pentru categoriile pentru care ați ales să faceți promoții utilizatorii care au avut minim K vizualizări cu produse din acea categorie, unde K e ales de voi (K < N)
* Folosind functia send\_mass\_mail() veți grupa userii pentru fiecare categorie în parte și veți trimite printr-o singură comandă mailurile cu template-urile pentru toate promoțiile.

## TASK 3 (1p) PROIECT!!!!

Pentru cerintele urmatoare se va folosi functia mail\_admins(). Veti defini minim 2 administratori de site in settings.py. Mailurile se vor trimite și in format text si html. In formatul html, subiectul mesajului va fi scris si in textul mesajului cu un tag h1 de culoare rosie(deci stilizat). Se vor implementa următoarele măsuri de securitate pentru site:

* dacă un user încearcă să se logheze de 3 ori în mai puțin de 2 minute pe site și eșuează de fiecare dată, se va trimite un mail către administratori cu subiectul "Logari suspecte", username-ul cu care s-a incercat logarea și ip-ul de pe care s-a incercat logarea.
* dacă un user încearcă să se înregistreze cu username-ul admin, se va trimite un mesaj către administratorii site-ului cu subiectul "cineva incearca sa ne preia site-ul" si adresa de e-mail de pe care s-a incercat inregistrarea. Inregistrarea va fi, evident, refuzată
* veti alege voi o zonă de cod pe care o veti pune in try... except si dacă aceasta dă eroare se va trimite un e-mail către administratori cu eroarea. În mesajul html, eroarea va avea background rosu.

## TASK 4 (1p) PROIECT!!!!

Veți implementa un sistem de Logging pentru proiectul vostru. Sistemul de logging trebuie să aibă formatare atât pentru modul verbose cât și simple.

Sistemul de logging va implementa toate cele 5 nivele: DEBUG, INFO, WARNING, ERROR și CRITICAL.

Veți afișa mesajele de tip WARNING, ERROR și CRITICAL în consolă, in formatul simple. În rest pentru fiecare tip de mesaje veti avea alt fișier de log: debug.log, info.log, warning.log, error.log, critical.log. În fișiere veți afișa mesajele în formatul verbose.

Folosind:

logger = logging.getLogger('django')

veți alege în codul vostru câte 2 locuri pentru fiecare dintre 5 nivele unde să transmiteți mesaje (adică două locuri în care trimiteți mesaje critice, două locuri cu eroare etc.)