# Laborator 2

În exercițiile 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", cum e evidentiat aici: http://localhost:8000/aplicatie/cerere

1) Creati o functie în views.py numita aduna\_numere. Aceasta va trebui asociată unor cereri care incep cu /pag și se termină cu cifre (de exemplu: /pag/123, /pag/abc1234). Vom considera cererile drept continand doar litere si cifre. Raspunsul va fi un text de forma: "numar cereri: [nr] suma numere: [suma]" unde se înlocuiesc zonele de text scrise cu paranteze drepte. Numarul de cereri se referă practic la cate cereri s-au potrivit cu formatul. "Suma numere" va contine suma intregilor maximali de la finalul cererilor. De exemplu, pentru cererile consecutive de mai jos, aratam ce se afiseaza, (sau nu scriem nimic daca cererea nu se potriveste si nu afiseaza).

/pag/123

numar cereri:1 suma numere: 123

/pag/abcd

-eroare

/pag/12abcd

-eroare

/pag/abc1234

numar cereri:2 suma numere: 1357

/pag/12ab23

numar cereri:3 suma numere: 1380

/pag/0

numar cereri:4 suma numere: 1380

2) Creati o functie în views.py numita afiseaza\_liste. Aceasta va trebui asociată unor cereri care incep cu /liste si sunt urmate de un querystring cu parametri. Pentru fiecare parametru diferit se va genera o alta lista neordonata prin tagul <ul> (in raspunsul HTML) cu valorile asociate. De exemplu /liste?a=10&b=20&a=15&b=11&a=8&b=123 va genera doua liste (una pentru parametrul a si alta pentru parametrul b) fiecare avand ca itemuri in lista valorile parametrilor ca mai jos:

**a :**

* 10
* 15
* 8

**b:**

* 20
* 11
* 123

3) Creati o functie în views.py numita numara\_nume. Aceasta va trebui asociată unor cereri care incep cu /nume\_corect urmat de un "/" urmat de un string care va avea urmatorul format:

- Primul va fi prenumele care va incepe cu litera mare, va fi urmat de oricate litere mici, va avea sau nu o linioara, va avea sau nu o litera mare urmata de litere mici

- dupa prenume vom avea un spatiu simulat prin caracterul "+" (vedeti ca se transforma automat)

- numele format din litera mare urmata de zero sau mai multe litere mici.

Vor fi acceptate doar cererile care sunt nume corecte. Pagina va afisa numarul de nume primite.

4) Creati o functie în views.py numita cauta\_subsir. Aceasta va trebui asociată unor cereri care incep cu /subsir urmat de un "/" urmat de oricate cifre, urmat de un subsir format doar din literele a si b mici, urmat iar din cifre. Se va afisa in pagina lungimea subsirului.

5) Se da dictionarul:

d={

"lista":[

{

"a":10,

"b":20,

"operatie":"suma"

},

{

"a":40,

"b":20,

"operatie":"diferenta"

},

{

"a":25,

"b":30,

"operatie":"suma"

},

{

"a":40,

"b":30,

"operatie":"diferenta"

},

{

"a":100,

"b":50,

"operatie":"diferenta"

}

]

}

Creati un template care afiseaza o lista neordonata cu operatiile din dictionar. Functia render va transmite care pagina chiar acest dictionar. Template-ul va afisa o lista neordonata ca mai jos:

* 10+20=30
* 40-20=20
* 25+30=55
* 40-30=10
* 100-50=50