**Aplicatia pentru generarea task urile**

browserul: <http://127.0.0.1:5000/>

http://localhost:5173

Comenzi backend:

venv\Scripts\activate

python app.py

comenzi frontend:

npm run dev

autentificare:email = test@test.com, parolă = 1234

backend: http://127.0.0.1:5000

Cheia GPT 4.0:

sk-proj-IK-\_U8AOiNI6SfB69g-u5FaadS0oVg3VcH8XGBLUsBnZHdhyeADGkAmg4hjH83P8EiVg-9qMQgT3BlbkFJspRWunv\_t7d5kFTbCdGfIpj8wIngiUGSlotRoaG5IZ7-qgkAuEiNzATxsPNhPeUU2B3T92Ca0A

Criterii:

Disponibilitate sali (grupa pe categorii: curs, seminar si laborator. Aici poti sa te inspiri de pe orar ugal)

Criterii de generare orar PDF/Excel:

- programul zilnic pentru studenții de la licență va fi în intervalul 8.00-20.00, iar pentru cei de la master de la 16.00 la 20.00; acesta va cuprinde minim 4 ore/zi și maxim 8 ore/zi;

- nu se acceptă mai mult de o pauză (2 ore)/zi. Se recomandă ca studenții să aibă un program fără ferestre. Dacă totuși, din motive obiective (sala nu este disponibilă sau un cadru didactic susține mai multe discipline la același program de studiu etc.) nu se poate evita pauza, aceasta va fi inclusă în cele maxim 8 ore/zi;

- Cursurile sunt susținute cu anul, seminarele și proiectele cu grupa, iar laboratoarele cu subgrupa.

coordonatorul pe domeniu va ține o evidență a sălilor pentru activitățile practice.

se va respecta restricția legată de intervalul orar în care sunt fixate ședințele Consiliului Facultății și anume ziua de miercuri, modulul 4 (ora 14.00). Acest modul va fi liber pentru toate programele de studiu pentru a putea fi folosit pentru diverse activități dedicate studenților (prezentări de firme, workshop-uri etc.);

- în cuprinsul orarului, la curs se vor specifica: denumirea completă, precum și prescurtarea disciplinei, numele și prenumele cadrului didactic și sala; la celelalte activități – acronimul disciplinei, numele cadrului didactic și sala.

git push --set-upstream origin Liliana

Cum mi am facut eu branch si am urcat codul

$ git clone git@github.com:Liliana24znf/Calories\_Counter.git

$ cd Calories\_Counter/

$ git checkout -b Liliana

$ git branch

$ git status

$ git add .

$ git status

$ git commit -m "Prima incercare branch Liliana"

git push --set-upstream origin Liliana

git fetch -- all

git merge origin (branch din care vrem sa luam )

1. **Step reguli**
2. **cum fac cu tabelul?**
3. **Introduc eu denumirea cursurilor, a profesorilor, grupele, ani, divizarea grupelor.**
4. **Baza de date**

**O singura cladire, cursurile pprioritate fata de aplicatiile practice(activitati practice)  
baza de date**

**Engleza!!!!!!!!!**

**Sali, disponibilitate**

**un formal cu Sali si disponibilitate care sa se duca in baza de date   
  
formular: numele profesorului, disciplina1,2,3…, tipul activitate**

**formular: disponibilitatea salilor -toate sa fie in G, sa aiba G in fata, tipul Sali, sala de laborator si sala de curs   
A+numar sala de activitati A01….**

**C- numar sala de curs 6- C1….  
  
  
  
informatiii   
  
!!!!! Reguli documentatie +formular**

**!!!!! Writeless AI -Testare   
!!! model free generative AI- testare**

**!!!!! sa conectez userul cu baza de date, sa ii apara lui doar profesorii adaugati de el**

**!!!! introduc si anul la profesori unde preda**

**sau**

**las fara an, ca sa fie o diferenta de exemplu intre promotii  
  
  
!!! grupe si semigrupe  
  
  
!!! sa imi genereze si pe bucatele- dar problema ar fi la disponibilitate-conditii  
  
!! sa imi afiseze si ceva daca numai sunt profesori sau Sali disponibile?**

**Model free**

**Sali:**

**adauga, stergere, redenumeste**

**Profesor: adauga, sterge, asigneaza curs nou la profesor**

**Genereator:  
conditie pe genereare daca am informatii in baza de date, afisare warning**

**Creeare mode**

**Functia deploy  
sa descarc modeulul generat si dupa sa il pot prelua si antrena**

**Model.save.h5**

**Natural prec. lang. -Keras/ Pytorch https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article**