

# Tema proiectului: **Gestiunea recoltei unui fermier**

**Nume:** FLOREA ALEXANDRA

**Grupa:** 1307B

**Profesor Coordonator:** Cătălin Mironeanu

## **Scopul aplicației:**

Rolul aplicației este de a ajuta fermierii tineri și cei mai experimentați să își gestioneze mai bine terenurile, culturile și recolta. Aceștia vor putea să introducă în baza de date diverse date, care împreună vor alcătui un sistem de gestiune și de monitorizare a terenurilor lor. În subpunctele ce urmează se va explica baza de date de la baza proiectului, împreună cu rolurile și constrângerile datelor introduse.

## **Descrierea bazei de date:**

Într-o zonă de câmpie există mai mulți fermieri. Fermierii cultivă diferite tipuri de plante pe diferite terenuri. Un fermier poate deține mai multe terenuri. Fiecărui teren îi corespunde o cultură. Mai mulți fermieri pot avea același tip de cultură. Culturile dețin informații despre planta cultivată, tipul culturii și data cultivării. Fiecare plantă are un soi unic. Se înregistrează data la care e cultivată fiecare tranșă de plante. Fiecare arie cultivată beneficiază de un tip de seminte, în funcție de preferința fermierului. Semintele au o calitate, un producător și un preț specific pe kg care se vinde. Soiurile au un nume unic și ajung la maturizare într-un anumit timp. Un fermier poate planta același tip de cultură pe mai multe terenuri cu aceeași suprafață, la aceeași dată.

Nu se pune problema afectării recoltei, scăderea suprafeței cultivate, schimbarea prețului, ambiguității legăturii dintre planta-soi-seminte, necultivării unui teren, stabilirea unui număr de terenuri pe care le poate deține un fermier. Toate terenurile înregistrate vor fi cultivate.

## **Tehnologiile folosite:**

Pentru realizarea proiectului au fost necesare:

Pentru gestionarea bazei de date (Oracle SQL):

- **Oracle SQL Developer Data Modeler**
  - Pentru crearea entităților, tabelor și realizarea de legături între acestea la nivel logic și relațional pentru a ușura vizibilitatea și de a avea o interfață în care se pot modela ușor concepte de baze de date
- **SQL Developer**
  - Pentru rularea scripturilor de creare de tabele, inserare, ștergere și modificare de date, verificarea constrângerilor și pentru a avea o bază de date la care se conectează proiectul.

Pentru partea de Backend (Python):

- **Pycharm cu interpretor Python 3.6**, la care am adăugat modulele:
  - **Flask version 2.0.3**
    - permite dezvoltarea rapidă a aplicațiilor web, este ușor de înțeles și oferă o structură flexibilă)
    - este un framework web pentru Python
  - **Jinja2 version 3.0.3**
    - este un motor de șabloane pentru limbajul de programare Python
    - este adesea folosit în cadrul aplicațiilor web Flask pentru a genera dinamic conținut HTML
  - **cx\_Oracle 8.3.0**
    - este un modul Python pentru accesul la baze de date Oracle
    - permite conectarea la baze de date Oracle și executarea de operațiuni precum interogări SQL

Pentru partea de Frontend (html, css):

- **Visual Studio Code**
  - A avut rol în rularea și testarea de coduri HTML și CSS, fiind o interfață în care se pot gestiona ușor erorile, se pot folosi extensii precum **LIVE SERVER** sau **Prettier – code Formatter** ce ajută la observarea mai ușoară a problemelor de sintaxă și rularea mai rapidă a codului

## Conectarea la baza de date:

În proiect am folosit contul personal pentru a mă conecta la baza de date, motiv pentru care în caz că baza de date nu se poate deschide, trebuie modificate datele utilizatorului. Utilizatorul trebuie să beneficieze de o bază de date Oracle, de la care să dețină hostul, portul, service\_name-ul, user și parola. Primele 3 sunt folosite ca parametrii la funcția `cx_Oracle.makedsn()`, cu rolul de a se conecta la baza de date Oracle. A doua funcție (`cx_Oracle.connect()`) este folosită pentru a conecta utilizatorul la propria baza de date, unde acesta poate manipula tabelele. Secvența de cod în care se realizează conexiunea este următoarea:

```
dsn_tns = cx_Oracle.makedsn('host', 'port', service_name='service_name')
conn = cx_Oracle.connect(user='user', password='password', dsn=dsn_tns)
```

## Structura și inter-relationarea tabelor:

În urma descrierii realizate mai sus, au rezultat diagramele din figura 1. Din text se deduce că între tabela FERMIER și tabela CULTURĂ ar trebui să fie o relație MANY-TO-MANY, însă am decis să normalizez problema creată de această relație încă din modelul LOGIC, prin crearea unei noi tabele – TEREN. Astfel, se crează o relație ONE-TO-MANY între tabela FERMIER și tabela TEREN, deoarece: “Un fermier poate deține mai multe terenuri, însă un teren poate fi deținut de un singur fermier.”, și o

relație ONE-TO-MANY între tabela CULTURĂ și tabela TEREN, deoarece: ”Un teren are o singură cultură și o cultură poate fi pe mai multe terenuri. ”

Între CULTURĂ și PLANTĂ se realizează o relație ONE-TO-ONE deoarece: ”Pe o cultură poate fi plantat doar un tip de planta, iar un anumit tip de planta poate fi cultivat pe un singur tip de cultura”. Legătura dintre PLANTĂ și SOI este o legătură ONE-TO MANY, pentru că o plantă poate avea mai multe soiuri, însă un soi este doar al unei plante.

În final mai există o legătură ONE-TO-MANY între SEMINTE și SOI, pentru că un soi poate beneficia de mai multe tipuri de semințe, însă un anumit tip de semințe aparțin doar unui soi specific.

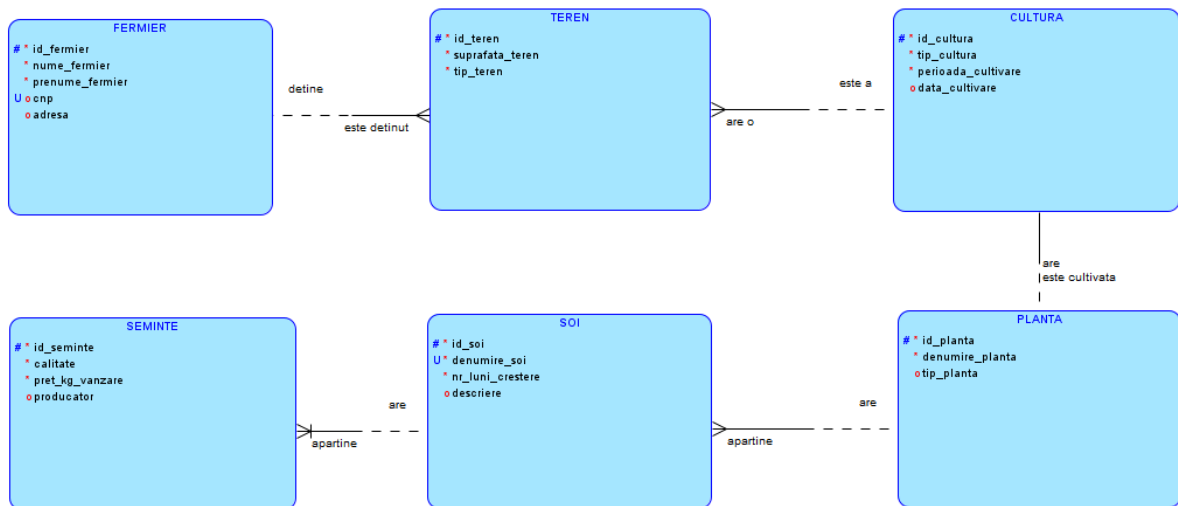


Figura 1. Modelul Logic

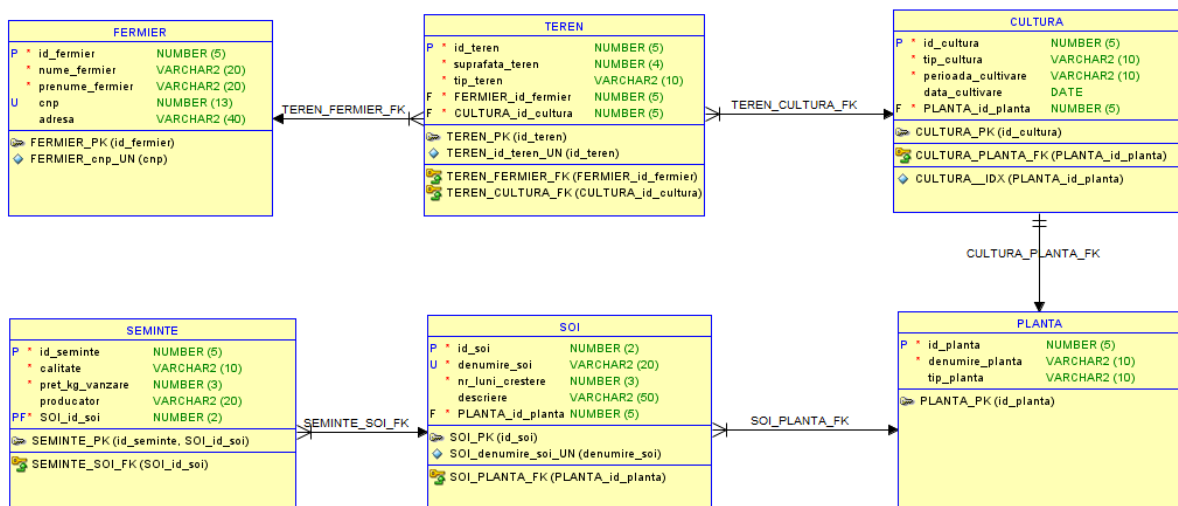


Figura 2. Modelul Relațional

În figura 2 se observă că s-au adăugat și date despre cheile primare și cele străine. Aceste detalii și motivele pentru care s-au ales în acest fel, vor fi atinse în subpunctul următor.

## **Descrierea constrangerilor folosite si de ce au fost acestea necesare:**

### **1. ENTITATEA FERMIER**

- **ID\_fermier**

- este cheie primară (obligatorie, unică)
- este valoare numerică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate ( înafară de cnp, dar acesta este opțional), din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Nume\_fermier**

- valoare șir de caractere, obligatorie
- are o constrângere de tip regex deoarece numele nu ar trebui să conțină cifre sau alte caractere:  
**regexp\_like(nume\_fermier, '[A-Z][a-z]{1,}')**

- **Prenume\_fermier**

- valoare șir de caractere, obligatorie
- are o constrângere de tip regex deoarece prenumele nu ar trebui să conțină cifre sau alte caractere:  
**regexp\_like(nume\_fermier, '[A-Z][a-z]{1,}')**

- **CNP\_fermier**

- valoare numerică, opțională, unic
- unic deoarece acesta este scopul său, de identificator
- acest camp este opțional deoarece se poate ca fermierul să își dorească să păstreze aceste date confidențiale

- **Adresa**

- valoare șir de caractere, opțională
- asemenea câmpului anterior, adresa este opțională din motive de confidențialitate

## 2. ENTITATEA TEREN

- **ID\_teren**

- valoare numerică, obligatorie, unică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Suprafata\_teren (ha)**

- valoare numerică, obligatorie
- este necesar sa se stie suprafata de teren pentru a putea realiza ulterior calcule pe baza ei

- **Tip\_teren**

- valoare șir de caractere, obligatorie
- are doar 4 valori posibile (arabil, sera, pomicol, viticol ) deoarece acestea sunt tipurile principale de terenuri ce pot fi cultivate, valoarea default = *arabil*
- este obligatoriu deoarece aceste detalii vor fi necesare pentru a avea sens inserările din restul tabelelor

- *ID\_fermier (cheie străină către "Fermier")*

*-după cum am discutat și mai sus, cele două tabele au relație ONE-TO-MANY și a fost necesară folosirea unei chei străine*

- *ID\_cultura (cheie străină către "Cultura")*

*-după cum am discutat și mai sus, cele două tabele au relație ONE-TO-MANY și a fost necesară folosirea unei chei străine*

## 3. ENTITATEA CULTURA

- **ID\_cultura**

- valoare numerică, obligatorie, unică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Tip\_cultura**

- valoare șir de caractere, obligatorie
- are doar 3 valori posibile (cereale, legume, fructe) deoarece acestea sunt tipurile principale de culturi ce pot fi cultivate
- este obligatoriu deoarece aceste detalii vor fi necesare pentru a avea sens inserările din restul tabelelor

- **Perioada\_cultivare**

- valoare șir de caractere, obligatorie
- are doar 4 valori posibile (iarna, primăvara, vara, toamna) deoarece acestea sunt tipurile principale de perioade de cultivare
- este obligatoriu deoarece aceste detalii vor fi necesare pentru a avea sens inserările din restul tabelelor

- **Data\_cultivare**

- valoare tip dată, obligatorie
- data la care se cultivă respective cultură
- este necesar să se știe suprafața de teren pentru a putea realiza ulterior calcule pe baza ei

- *ID\_planta (cheie străină către "Planta")*

*-după cum am discutat și mai sus, cele două tabele au relație ONE-TO-ONE și a fost necesară folosirea unei chei străine*

#### **4. ENTITATEA PLANTA**

- **ID\_planta**

- valoare numerică, obligatorie, unică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Denumire\_planta**

- valoare șir de caractere, obligatorie
- are o constrângere de tip regex deoarece denumirea nu ar trebui să conțină cifre sau alte caractere: **regex\_like(denumire\_planta, '[A-Z][a-z\s]')**

- **Tip\_plantă** (anuală, bianuală, perenă etc.)

- valoare șir de caractere, opțională
- este opțional pentru că nu este o dată de care depind neapărat celelalte tabele, deși poate fi utilă(de exemplu în cazul pomilor sau viței de vie care se plantează o singură dată)

## 5. ENTITATEA SOI

- **ID\_soi**

- valoare numerică, obligatorie, unică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Denumire\_soi**

- valoare șir de caractere, obligatorie, unică
- are o constrângere de tip regex deoarece denumirea nu ar trebui să conțină cifre sau alte caractere: **regex\_like(denumire\_soi, '[A-Z][a-z\s]')**
- este unică deoarece nu pot exista mai multe soiuri cu aceeași denumire

- **Descriere**

- valoare șir de caractere, opțională
- stă la latitudinea userului dacă dorește sau nu să insereze informații utile despre respectivul soi

- **Nr\_luni\_crestere**

- valoare numerică, obligatorie
- este necesar sa se stie suprafata de teren pentru a putea realiza ulterior calcule pe baza ei

- *ID\_soi (cheie străină către "Planta")*

*- după cum am discutat și mai sus, cele două tabele au relație ONE-TO-MANY și a fost necesară folosirea unei chei străine*

## 6. ENTITATEA SEMINTE

- **ID\_semințe**

- valoare numerică, obligatorie, unică
- este activat autoincrementul

S-a ales cheie primară deoarece la restul valorilor nu există constrângeri de unicitate din cauza unicității și a ușurinței introducerii datelor am ales să adaug autoincrement.

- **Calitate**

- valoare șir de caractere, obligatorie
- este obligatorie pentru că prețul poate depinde de calitate

- **Producator**

- valoare șir de caractere, opțional
- opțional deoarece de multe ori producătorul poate fi însuși fermierul care plantează

- **Pret\_kg\_vanzare (lei)**

- valoare numerică, obligatorie
- valoare necesară pentru a face calcule ulterioare

- **ID\_soi (cheie străină către "Soi")**

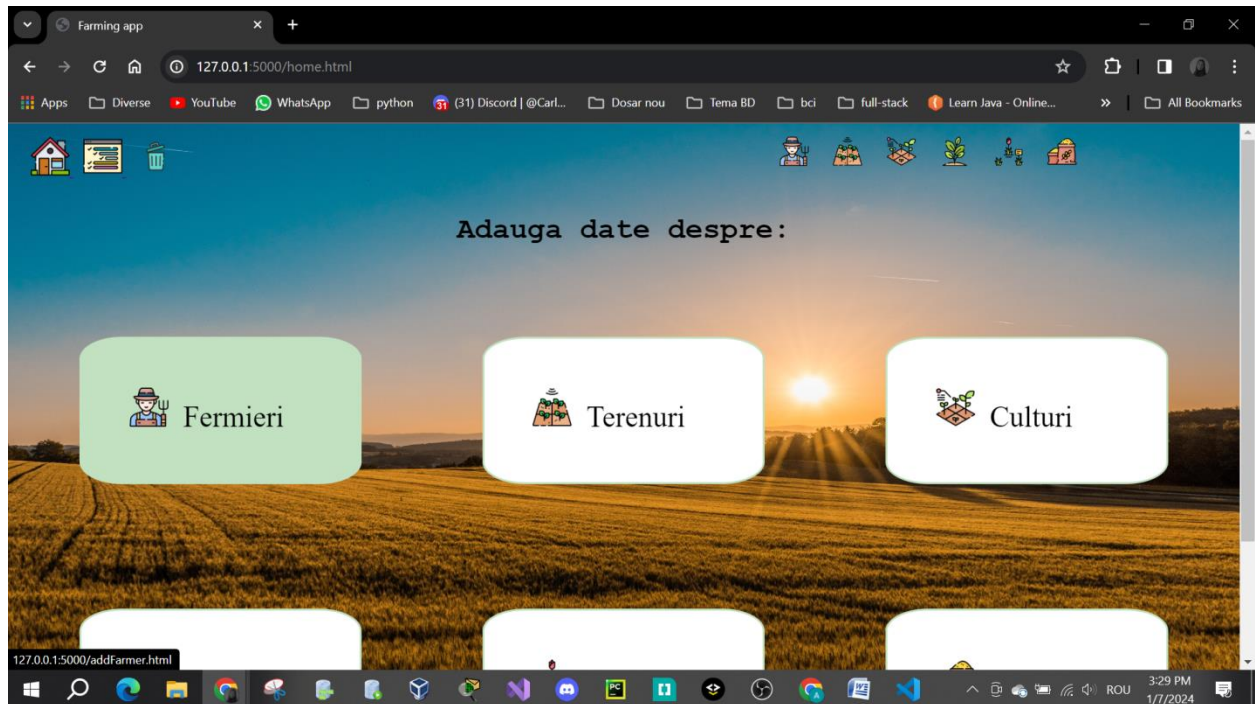
*- după cum am discutat și mai sus, cele două tabele au relație ONE-TO-MANY și a fost necesară folosirea unei chei străine*

**\*\* in fisierul cu scriptul de inserare date de test sunt adaugate comentarii pentru fiecare tip de inserare valida/invalida**

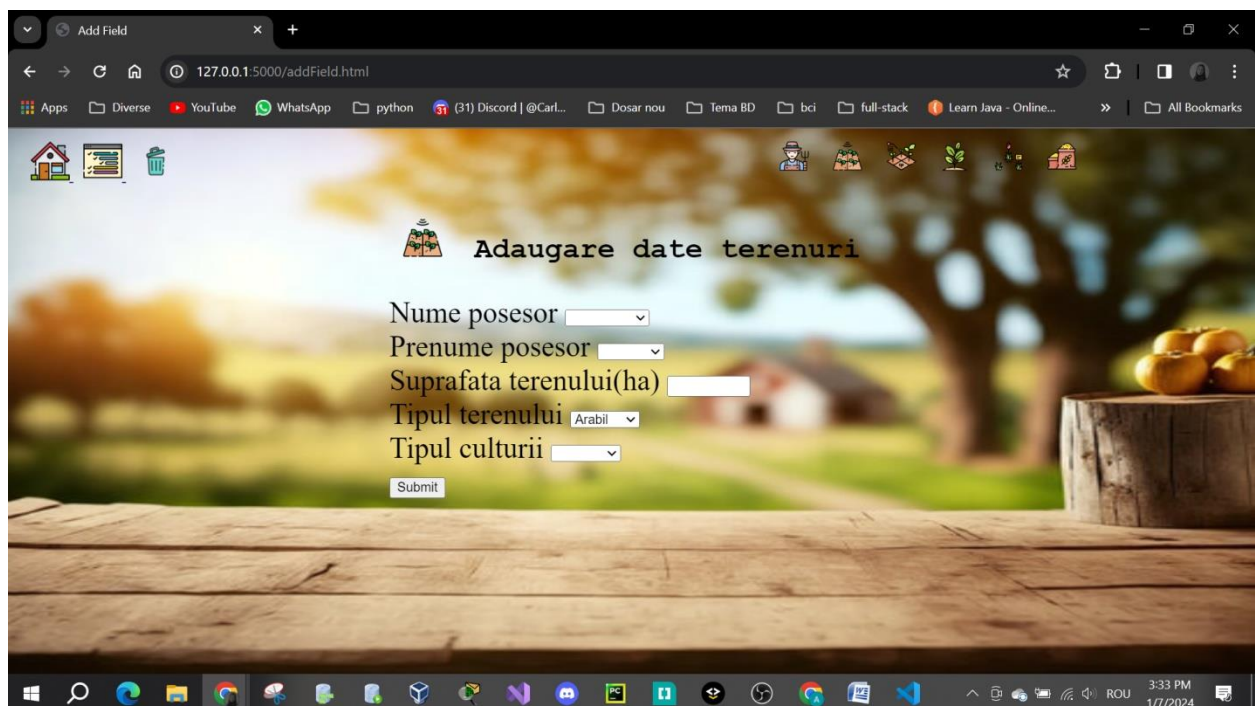


## Capturi de ecran cu interfața aplicației și secvențe de cod:

Homepage ( pagina de selectare a tabelii în care se vor introduce date)



Pagina de adăugare de date pentru terenuri:



Codul în Python pentru pagina anterioară:

```
Project ▾
  ▾ Tema de casa BD C:\Users\...
    ▾ Tema de casa BD
      ▾ Scripturi
      ▾ static
      ▾ templates
      ▾ vc14
      ▾ adrci.exe
      ▾ adrci.sym
      ▾ BASIC_LICENSE
      ▾ BASIC_README
      ▾ genezi.exe
      ▾ genezi.sym
      ▾ main.py
      ▾ oci.dll
      ▾ oci.sym
      ▾ ocjdbc19.dll
      ▾ ocjdbc19.sym
      ▾ ociw32.dll
      ▾ ociw32.sym
      ▾ ojdbc8.jar
      ▾ oramysql19.dll
      ▾ oramysql19.sym
  Run main x

main.py
56 @app.route(rule='/addField.html', methods=['GET', 'POST'])
57 def addField():
58     cursor.execute("SELECT nume_fermier FROM fermier")
59     nume_fermier_options = [row[0] for row in cursor.fetchall()]
60     cursor.execute("SELECT prenume_fermier FROM fermier")
61     prenume_fermier_options = [row[0] for row in cursor.fetchall()]
62     if request.method == 'POST':
63         nume_fermier = request.form['nume_fermier']
64         prenume_fermier = request.form['prenume_fermier']
65         suprafata teren = request.form['suprafata teren']
66         tip_teren = request.form['tip_teren']
67         tip_cultura = request.form['tip_cultura']
68         cursor.execute(
69             "SELECT id_fermier FROM fermier WHERE nume_fermier = :nume_fermier AND prenume_fermier = :prenume_fermier"
70             {"nume_fermier": nume_fermier, "prenume_fermier": prenume_fermier})
71         fermier_id_fermier = cursor.fetchone()[0]
72
73         cursor.execute("SELECT id_cultura FROM cultura WHERE tip_cultura = :tip_cultura",
74             {"tip_cultura": tip_cultura})
75         cultura_id_cultura = cursor.fetchone()[0]
76
77         cursor.execute("SELECT teren_seq.NEXTVAL FROM DUAL")
78         id_teren = cursor.fetchone()[0]
79
80     viewData()
```

Cod în HTML pentru aceeași pagină (formularul):

```
File Edit Selection View Go Run ...
Tema de casa BD
EXPLORER
  ▾ TEMA D...
    ▾ templates
      ▾ addField.html
      ▾ addCrop.html
      ▾ addFarmer.html
      ▾ addField.html
      ▾ addPlant.html
      ▾ addSeed.html
      ▾ addVariety.html
      ▾ deleteTables.html
      ▾ home.html
      ▾ viewData.html
    ▾ vc14
    ▾ adrci.exe
    ▾ adrci.sym
    ▾ BASIC_LICENSE
    ▾ BASIC_README
    ▾ genezi.exe
    ▾ genezi.sym
    ▾ main.py
    ▾ oci.dll
    ▾ OUTLINE
    ▾ TIMELINE
  ▾ deleteTables.html
  ▾ addField.html
  ▾ home.css
  ▾ background2.jpg

63 <p class="title-text">
64     </img>
65     <b>Adaugare date terenului</b>
66 </p>
67
68 <div class="form-style">
69     <form method="POST" action="/addField.html">
70         <label for="nume_fermier">Nume posesor</label>
71         <select id="nume_fermier" name="nume_fermier">
72             <option></option>
73             {% for option in nume_fermier_options %}
74                 <option value="{{ option }}">{{ option }}</option>
75             {% endfor %}
76         </select><br>
77         <label for="prenume_fermier">Prenume posesor</label>
78         <select id="prenume_fermier" name="prenume_fermier">
79             <option></option>
80             {% for option in prenume_fermier_options %}
81                 <option value="{{ option }}">{{ option }}</option>
82             {% endfor %}
83         </select><br>
84         <label for="suprafata teren">Suprafata terenului(ha)</label>
85         <input name="suprafata teren" type="number" class="form-control is-valid" id="suprafata teren" min="0" max="1000000">
86         <label for="tip_teren">Tipul terenului</label>
87         <select id="tip_teren" name="tip_teren" value="arabil" required>
88             <option value="arabil">Arabil</option>
89             <option value="sera">Sera</option>
90             <option value="pomicol">Pomicol</option>
91             <option value="viticol">Viticol</option>
92         </select><br>
93         <label for="tip_cultura">Tipul culturii</label>
```

Pagina de afișare a datelor:

127.0.0.1:5000/viewData.html

Apps Diverse YouTube WhatsApp python (31) Discord | @Carl... Dosar nou Tema BD bci full-stack Learn Java - Online... All Bookmarks

Icons: Home, Mail, Trash, User, Farm, Tractor, Plant, Garden, House, Shop

### Date despre fermieri, terenuri si culturi

Numele fermierului	Prenumele fermierului	Tipul terenului	Suprafața terenului	Tipul Culturii	Denumirea plantei	Perioada cultivării	Data cultivării
Enache	Tudor	arabil	15 ha	legume	Morcov	primavara	2023-03-10 00:00:00
Gavrilescu	Mihai	sera	5 ha	legume	Rosie	primavara	2023-02-28 00:00:00
Ionescu	Andrei	arabil	4 ha	cereale	Grau	toamna	2022-10-10 00:00:00
Ionescu	Andrei	pomicol	8 ha	fructe	Par	toamna	2012-09-17 00:00:00
Ionescu	Andrei	arabil	2 ha	legume	Morcov	primavara	2023-03-10 00:00:00
Ionescu	Andrei	arabil	20 ha	cereale	Grau	toamna	2022-10-10 00:00:00
Nicolescu	Marian	viticol	2 ha	fructe	Strugure	primavara	2017-03-20 00:00:00

### Date despre semințe și soiuri

Taskbar: Windows, Search, Edge, File Explorer, Chrome, VS Code, Docker, Telegram, PC, Teams, OBS, Clock, 3:30 PM 1/7/2024