

# **Proiect Baze de Date**

**Student: Subtirica Gabriel**

**Grupa: 331AB**

## Descriere cerinta:

Elaborarea unui proiect ce consta in obtinerea unui site/aplicatii impreuna cu o baza de date pentru tinerea evidentei unei companii ce se ocupa cu organizarea de evenimente.

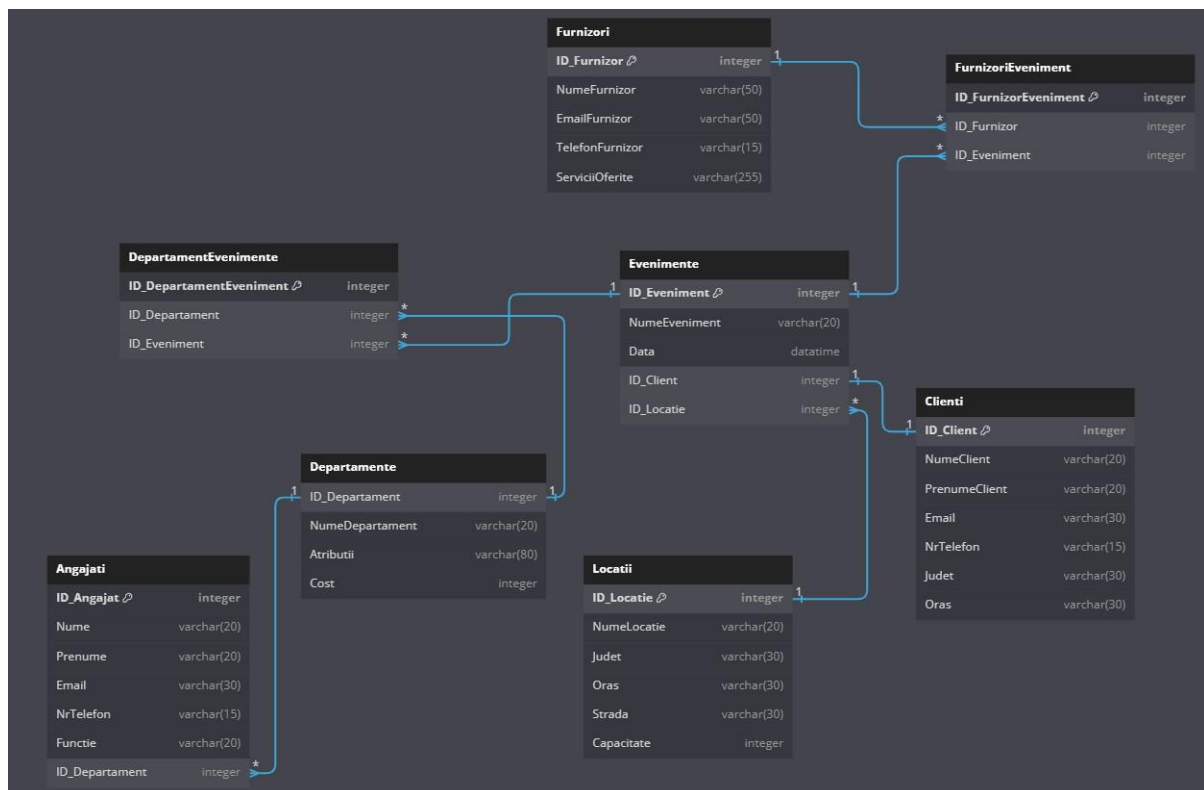
## Etape de proiectare

### *Baza de date*

Pentru elaborarea bazei de date am considerat urmatoarele tabele, alaturi de continutul aferent acestora:

- **Furnizori:** ID\_Furnizor(cheie primara), NumeFurnizor, EmailFurnizor, TelefonFurnizor, ServiciiOferite;
- **Clienti:** ID\_Client(cheie primara), NumeClient, PrenumeClient, Email, NrTelefon, Judet, Oras;
- **Departamente:** ID\_Departament(cheie primara), NumeDepartament, Atributii, Cost;
- **Angajati:** ID\_Angajat(cheie primara), Nume, Prenume, Email, NrTelefon, Functie, ID\_Departament(cheie externa);
- **Locatii:** ID\_Locatie(cheie primara), NumeLocatie, Judet, Oras, Strada, Capacitate;
- **Evenimente:** ID\_Eveniment(cheie primara), NumeEveniment, Data, ID\_Client(cheie externa), ID\_Locatie(cheie externa);
- **DepartamentEvenimente**(tabela de legatura): ID\_DepartamentEveniment(cheie primara), ID\_Departament(cheie externa), ID\_Eveniment(cheie externa);
- **FurnizoriEvenimente**(tabela de legatura): ID\_FurnizorEveniment(cheie primara), ID\_Furnizor(cheie externa), ID\_Evenimente(cheie externa);

## Identificare tabele – Stabilirea campurilor si a tipurilor de date



### Relatii intre tabele:

Clienti – Evenimente: one to one

Locatii – Evenimente: one to many

Departamente – Angajati: one to many

Furnizori + Evenimente: many to many => FurnizoriiEvenimente(tabel de legatura)

- Furnizori – FurnizoriiEveniment: one to many
- Evenimente – FurnizoriiEveniment: one to many

Departamente + Evenimente: many to many => DepartamentEvenimente(tabel de legatura)

- Departamente – DepartamentEvenimente: one to many
- Evenimente – DepartamentEvenimente: one to many

## **Lista de functionalitati:**

Firma noastră este specializată în organizarea evenimentelor private și conferințelor pentru clienții noștri.

Pentru a începe procesul de organizare, fiecare client este nevoit să ofere o serie de informații personale (care sunt înregistrate în „Clineti”), după care are posibilitatea să-și aleagă un singur tip de eveniment pe care dorește să-l organizeze, cum ar fi o petrecere privată sau o conferință corporativă. Odată ce clientul decide asupra tipului de eveniment, este necesar să precizeze detaliile esențiale, cum ar fi numele evenimentului, data la care dorește să aibă loc evenimentul și locația unde va fi organizat.

Echipa noastră se asigură că fiecare eveniment este unic și personalizat pentru a se potrivi cerințelor și așteptărilor clientului. În același timp, aceeași înregistrare în „Evenimente” ar stabili o conexiune în tabela „DepartamentEvenimente”, indicând departamentele și angajații care se vor ocupa de organizarea evenimentului. De asemenea, furnizorii evenimentului ar fi legați de eveniment folosind tabela „FurnizoriEveniment”, unde ID-urile acestora și a evenimentelor ar fi înregistrate.

## **Constrangeri:**

Un client îți poate cumpara serviciile pentru un eveniment, iar fiecare eveniment este asociat cu un singur client;

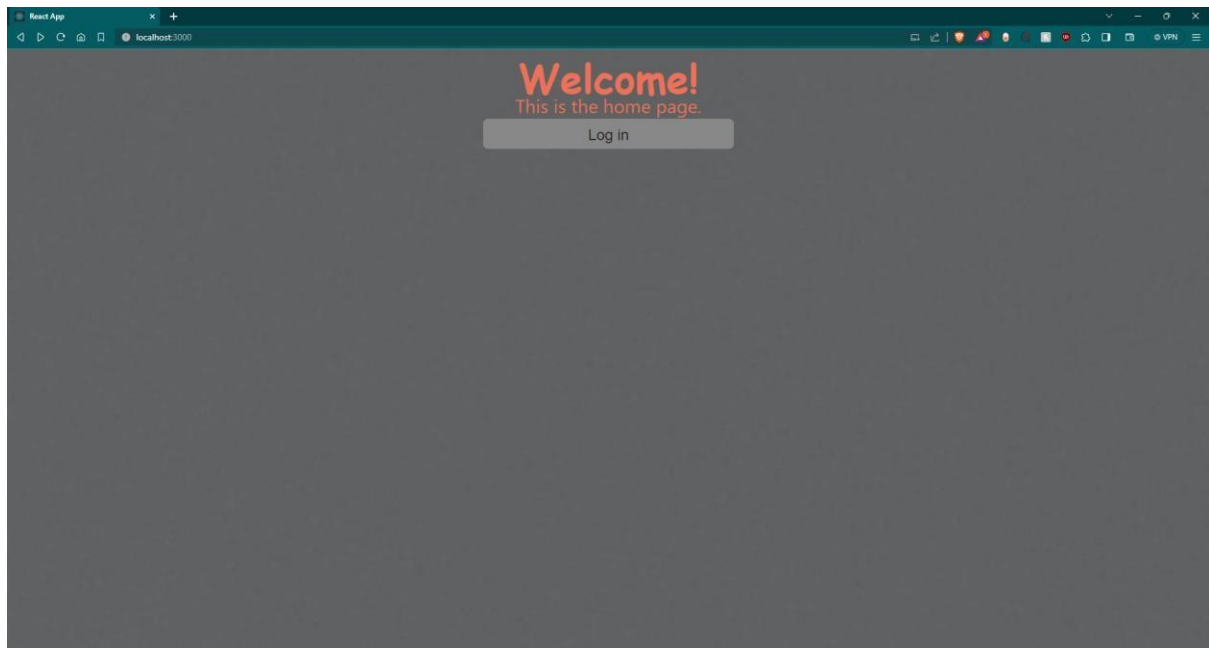
La un eveniment lucrează mai multe departamente, iar un departament poate fi chemat la mai multe evenimente;

Un eveniment poate avea loc într-o singură locație, dar o locație poate găzdui mai multe evenimente;

Într-un departament pot lucra mai mulți angajați, dar un angajat nu poate lucra în mai multe departamente;

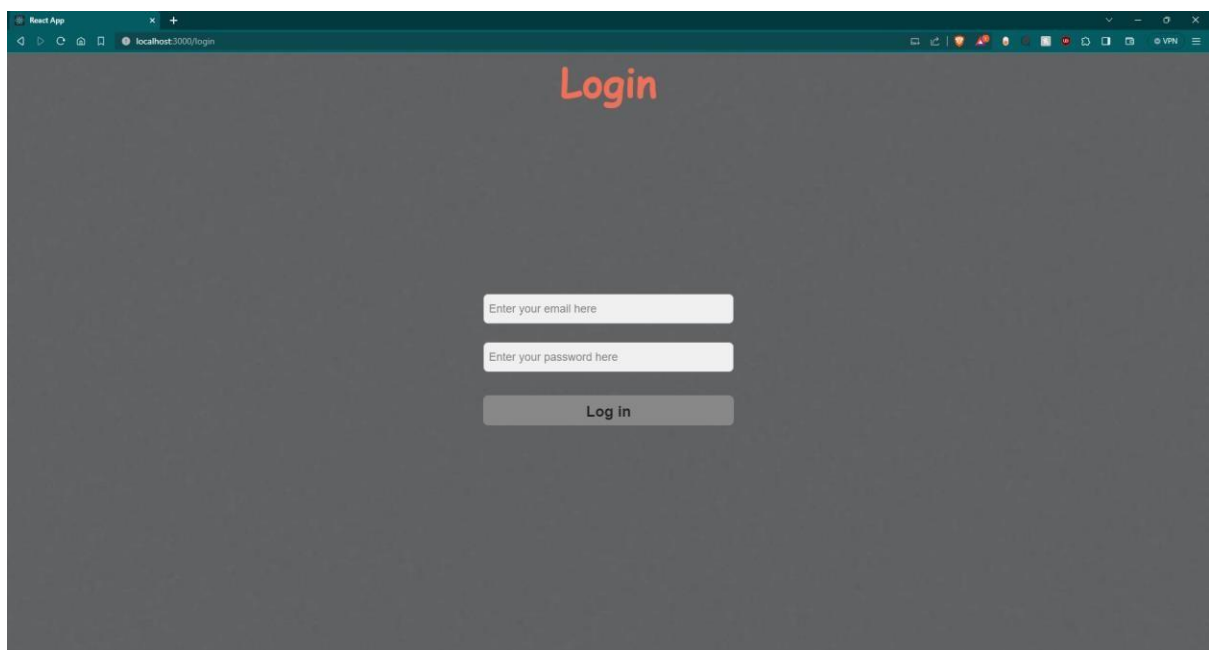
Un furnizor poate oferi servicii mai multor evenimente, iar un eveniment primește serviciile mai multor furnizori;

## Pagina de start a aplicatiei:

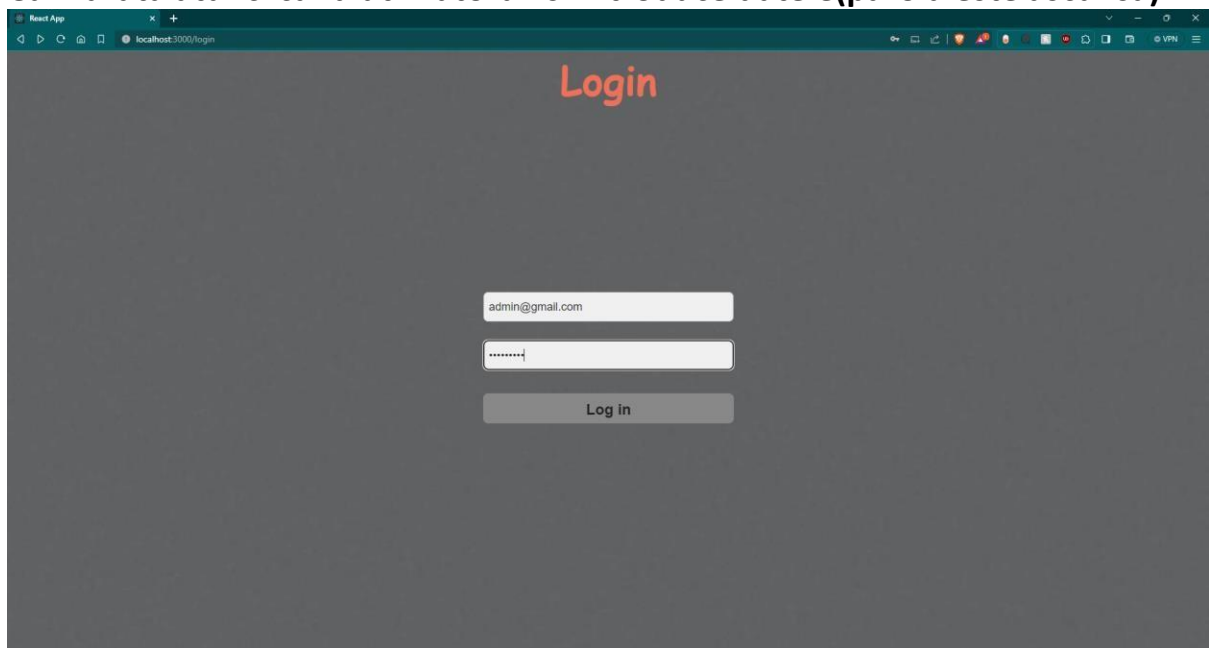


Apas pe butonul „Log in” si il trimite pe utilizator la pagina de login, unde este nevoit sa isi introduca datele(email si parola). Daca are deja cont o sa il trimita mai departe, in home(unde poate observa tabelele sau poate face restul operatiilor). Daca este un utilizator nou, ii sunt salvate datele si la urmatoarea logare cu aceste date o sa aibe acces la restul partii din aplicatie:

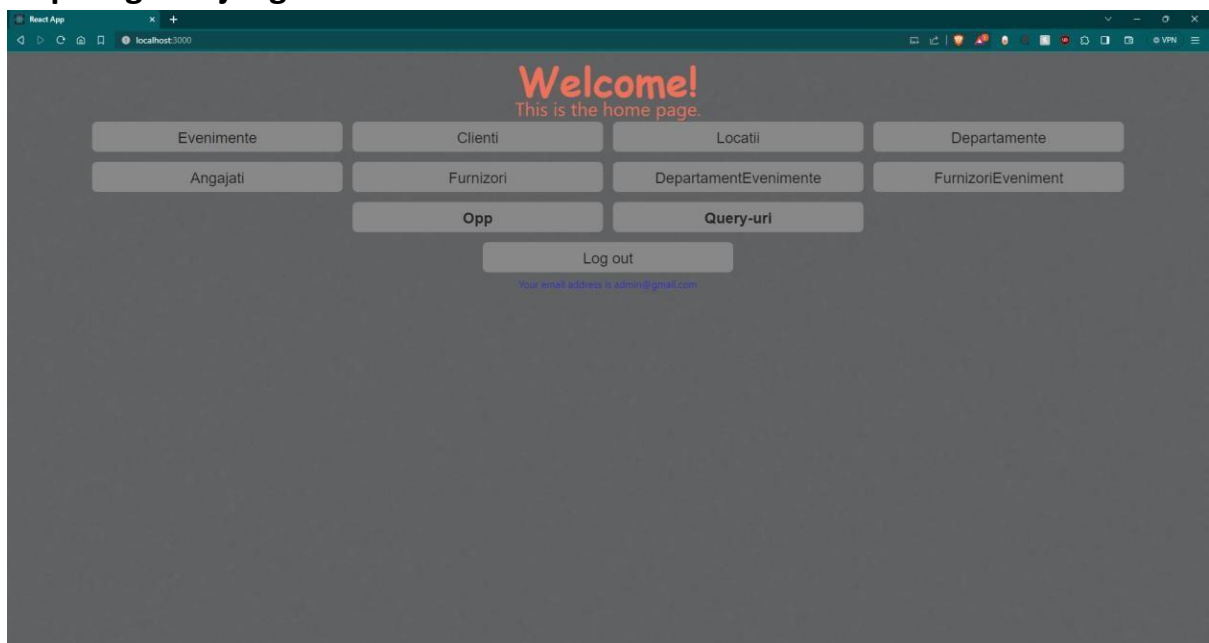
## Login:



Cum arata atunci cand utilizatorul isi introduce datele(parola este ascunsa):



Dupa logare ajungem in fereastra home:



Primele 8 butone de sus reprezinta denumirile tabelelor din baza mea de date, iar la interactiunea cu acestea se afiseaza continutul in timp real:

## Exemplu pe tabela Evenimente:

The screenshot shows a web application interface. At the top, there's a "Welcome!" message and a navigation bar with buttons: "Evenimente", "Clienti", "Locatii", "Departamente", "Angajati", "Furnizori", "DepartamentEvenimente", and "FurnizoriEveniment". Below the navigation bar is a table with 5 columns: "ID\_Eveniment", "NumeEveniment", "Data", "ID\_Client", and "ID\_Locatie". The table contains 9 rows of data. At the bottom, there are buttons for "Opp", "Query-uri", and "Log out", along with a small text link "Your email address is admin@gmail.com".

ID_Eveniment	NumeEveniment	Data	ID_Client	ID_Locatie
1	Nunta	2023-12-01	1	5
2	Conferinta IT	2023-11-15	2	2
3	Petrecere corporate	2023-11-20	3	4
4	Lansare produs	2023-11-25	4	3
5	Aniversare	2023-12-10	5	3
6	Revelion	2023-12-31	6	6
7	Lansare firma	2023-12-15	3	5
8	Petrecere de Craciun	2023-12-22	4	5
9	Nunta lui Marcel	2024-01-09	3	2

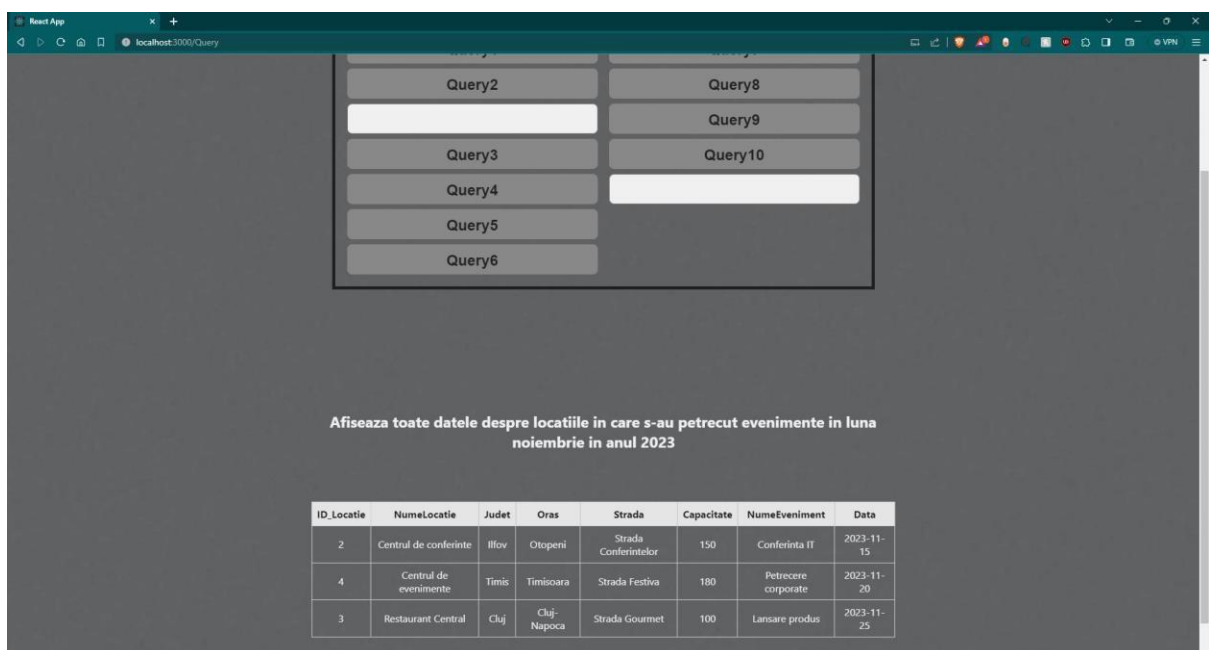
Butonul „Opp” trimite utilizatorul pe o noua fereastră in care poate introduce, edita sau sterge date din tabelele Departamente si Locatii:

The screenshot shows the "Opp" (Options) page. On the left, there are buttons for "Home" and "Query-uri". The main content area is divided into two columns. The left column is for "ID Departament:" and includes input fields for "ID Departament", "Nume Departament:", "Atributii:", and "Cost:", followed by buttons for "Insert Departament", "Update Departament", and "Delete Departament". The right column is for "ID Locatie:" and includes input fields for "ID Locatie", "Nume Locatie:", "Judet:", "Oras:", "Strada:", and "Capacitate:", followed by buttons for "Insert Locatie", "Update Locatie", and "Delete Locatie".

De asemenea, butonul „Query-uri” trimite utilizatorul pe o pagina in care se pot afisa interogările:



La apasare unui buton din cele denumite „Query” se afiseaza tabelul ce respecta interogările respective. Query-ul 2 este reprezentat de interogarea simpla variabila, iar Query-ul 10 este ce complexa variabila





La introducerea valorii „200” in textboxul de sub Query 2 si la apasarea butonului:

The screenshot shows a web application interface with a grid of buttons labeled Query2 through Query10. Query2 and Query10 are highlighted with white borders, indicating they are active or selected. Below the grid, there is a text prompt in Romanian: "Afiseaza toate datele despre Evenimentele ce au avut loc in Locatii cu o capacitate de minim 200 de locuri". Below this prompt is a table with 7 columns: ID\_Eveniment, NumeEveniment, Data, Email Client, ID\_Locatie, NumeLocatie, and Capacitate. The table contains one row of data.

ID_Eveniment	NumeEveniment	Data	Email Client	ID_Locatie	NumeLocatie	Capacitate
6	Revelion	2023-12-31	6	6	Club de noapte Elegant	250

Insert, Update, Delete:

Insert:

```
const handleSubmit = (event) => {
  event.preventDefault();
  // console.log('ceva');
  fetch('http://localhost:3080/departamente', {
    method: 'post',
    headers: { 'Content-Type': 'application/json' },
    // body: JSON.stringify({ x:3 })
    body: JSON.stringify({
      departament: ID_Departament,
      numeDepartament: NumeDepartament,
      atributii: Atributii,
      cost: Cost
    })
  });
};
```

```
app.post('/departamente', (req, res) => {
  const departament = req.body;
  console.log(departament);
  Request.query('INSERT INTO Departamente (ID_Departament, NumeDepartament, Atributii, Cost) VALUES (${departament.departament}, '${departament.numeDepartament}', '${departament.atributii}', ${departament.cost})', function (err, result) {
    if (err) {
      console.log(err);
      res.status(500).send(err);
    } else {
      res.status(200).send('Departament adăugat cu succes');
    }
  });
});
```

## Update:

```
const handleUpdate = (event) => {
  event.preventDefault();

  // Collect the data from input fields
  const departament = {
    ID_Departament,
    NumeDepartament,
    Atributii,
    Cost,
  };

  fetch('http://localhost:3080/departamente', {
    method: 'put',
    headers: { 'Content-Type': 'application/json' },
    // body: JSON.stringify({ x:3 })
    body: JSON.stringify(
      departament
    )
  });
};
```

```
app.put('/departamente', (req, res) => {
  const departament = req.body;
  console.log(departament);
  Request.query('UPDATE Departamente SET NumeDepartament = '${departament.NumeDepartament}', Atributii = '${departament.Atributii}', Cost = '${departament.Cost}' WHERE ID_Departament = '${departament.ID_Departament}', function (err, result) {
    if (err) {
      console.log(err);
      res.status(500).send(err);
    } else {
      res.status(200).send('Departament actualizat');
    }
  });
});
```

## Delete:

```
const handleDelete = (event) => {
  event.preventDefault();

  // Collect the data from input fields
  const departament = {
    ID_Departament,
    NumeDepartament,
  };

  fetch('http://localhost:3080/departamente', {
    method: 'delete',
    headers: { 'Content-Type': 'application/json' },
    // body: JSON.stringify({ x:3 })
    body: JSON.stringify(
      departament
    )
  });
};
```

```
app.delete('/departamente', (req, res) => {
  const departament = req.body;
  console.log(departament);
  Request.query('DELETE FROM Departamente WHERE ID_Departament = '${departament.ID_Departament}' OR NumeDepartament = '${departament.NumeDepartament}', function (err, result) {
    if (err) {
      console.log(err);
      res.status(500).send(err);
    } else {
      res.status(200).send('Departament sters');
    }
  });
});
```

## Interogările simple și complexe:

### Query1:

```
SELECT ID_Eveniment, NumeEveniment, Data, ID_Client, L.Oras
FROM Evenimente INNER JOIN
Locatii L ON Evenimente.ID_Locatie = L.ID_Locatie WHERE L.Oras = 'Brasov'
```

### Query2:

```
SELECT Evenimente.ID_Eveniment, Evenimente.NumeEveniment,
Evenimente.Data, Evenimente.ID_Client, Evenimente.ID_Locatie,
Locatii.NumeLocatie, Locatii.Capacitate
FROM Evenimente
INNER JOIN Locatii ON Evenimente.ID_Locatie = Locatii.ID_Locatie
WHERE Locatii.Capacitate >= 150;
```

Afiseaza toate datele despre Evenimentele ce au avut loc in Locatii cu o capacitate de minim 150 de locuri

### Query3:

```
SELECT D.ID_Departament, D.NumeDepartament, D.Atributii, D.Cost,
COUNT(De.ID_Eveniment) AS NumerEvenimente
FROM DepartamentEvenimente De
INNER JOIN Departamente D ON De.ID_Departament = D.ID_Departament
GROUP BY D.NumeDepartament, D.Atributii, D.Cost, D.ID_Departament
HAVING COUNT(De.ID_Eveniment) > 2;
```

Afiseaza toate datele despre departamentele care au contribuit la minim 2 evenimente

**Query4:**

```
SELECT Locatii.ID_Locatie, Locatii.NumeLocatie, Locatii.Judet, Locatii.Oras,  
Locatii.Strada, Locatii.Capacitate, Evenimente.NumeEveniment,  
Evenimente.Data  
  
FROM Locatii  
  
INNER JOIN Evenimente ON Locatii.ID_Locatie = Evenimente.ID_Locatie  
  
WHERE Evenimente.Data LIKE '2023-11%';
```

Afiseaza toate datele despre locatiile in care s-au petrecut evenimente in luna noiembrie in anul 2023

**Query5:**

```
SELECT Furnizori.ID_Furnizor, Furnizori.NumeFurnizor, Furnizori.EmailFurnizor,  
Furnizori.TelefonFurnizor, Furnizori.ServiciiOferite  
  
FROM Furnizori  
  
INNER JOIN FurnizoriEveniment ON Furnizori.ID_Furnizor =  
FurnizoriEveniment.ID_Furnizor  
  
WHERE FurnizoriEveniment.ID_Eveniment = 6;
```

Afiseaza toate datele despre furnizorii ce au oferit servicii evenimentului 'Revelion'

**Query6:**

```
SELECT Departamente.ID_Departament, Departamente.NumeDepartament,  
Departamente.Atributii, Departamente.Cost  
  
FROM Departamente  
  
INNER JOIN Angajati ON Departamente.ID_Departament =  
Angajati.ID_Departament  
  
WHERE Angajati.Nume LIKE 'Pop%'
```

Afiseaza toate datele despre departamentele cu angajatii al caror nume incepe cu 'Pop'

**Query7:**

```
SELECT Angajati.ID_Angajat, Angajati.Nume, Angajati.Prenume, Angajati.Email,  
Angajati.NrTelefon, Angajati.Functie, Departamente.NumeDepartament FROM  
Angajati INNER JOIN Departamente ON Angajati.ID_Departament =  
Departamente.ID_Departament WHERE Departamente.Cost < (SELECT  
AVG(Departamente.Cost) FROM Departamente)
```

Afiseaza datele despre angajatii al caror departament are costul mai mic decat media costurilor tuturor departamentelor

**Query8:**

```
SELECT Angajati.Nume, Angajati.Prenume, Angajati.Email, Angajati.NrTelefon,  
Angajati.Functie FROM Angajati INNER JOIN Departamente ON  
Angajati.ID_Departament = Departamente.ID_Departament WHERE  
Departamente.Cost = ( SELECT MAX(Cost) FROM Departamente)
```

Afiseaza toate datele despre angajatii ce lucreaza la departamentul cu costul cel mai mare

**Query9:**

```
SELECT Angajati.ID_Angajat, Angajati.Nume, Angajati.Prenume, Angajati.Email,
Angajati.NrTelefon, Angajati.Functie, Departamente.NumeDepartament FROM
Angajati INNER JOIN Departamente ON Angajati.ID_Departament =
Departamente.ID_Departament WHERE Departamente.ID_Departament IN
(SELECT ID_Departament FROM Departamente WHERE Departamente.Cost >
(SELECT AVG(Departamente.Cost) FROM Departamente))
```

Afiseaza datele despre angajatii care lucreaza in departamentele ce au un cost mai mare decat costul mediu pentru un departament

**Query10:**

```
SELECT Clienti.ID_Client, Clienti.NumeClient, Clienti.PrenumeClient,
Clienti.Email, Clienti.NrTelefon, Clienti.Judet, Clienti.Oras FROM Clienti
INNER JOIN ( SELECT Evenimente.ID_Client FROM Evenimente INNER JOIN
Locatii ON Evenimente.ID_Locatie = Locatii.ID_Locatie WHERE
Locatii.Capacitate = ${req.body.capacity1}) AS Subcereri ON
Clienti.ID_Client = Subcereri.ID_Client;
```

Afiseaza datele despre clientii care au platit pentru evenimentele ce au avut loc in locatiile cu capacitatea precizata