**“ONTMAS-Organ Transplant Network Management System”**

*Submitted in partial fulfillment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

**in**

**Computer Science & Engineering**

**UE22CS351A – DBMS**

***Submitted by:***

| **Vineet Goel**  **Dhruthan MN** | **PES1UG22CS697**  **PES2UG22CS181** |
| --- | --- |

**August - December 2024**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

FACULTY OF ENGINEERING

**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)

100 feet Ring road, BSK 3rd stage, Hosakerehalli, Bengaluru – 560085

**ONTMAS-Organ Transplant Network Management**

**Problem Statement :** The Organ Transplantation Network Management System is a specialized database management system designed to support and streamline the process of organ transplantation. Organ transplantation involves removing an organ from one individual (the donor) and placing it into another (the recipient) to replace a damaged or missing organ. The donor and recipient may be located at the same site, or organs may be transported from a donor site to another location.

**Basic Steps in Implementation :**

* Each user has an account that can only be created through a government-certified hospital, which will store all the required information as outlined in the Problem Statement.
* Only hospitals are authorized to initiate requests for donations or procurement transactions.
* Government organizations will oversee the matching process between donors and patients, granting approval for transplantation procedures if all regulations are met.
* Statistical data will be collected and analyzed based on the history of transplantation transactions.

**Technologies Used:**

• MY

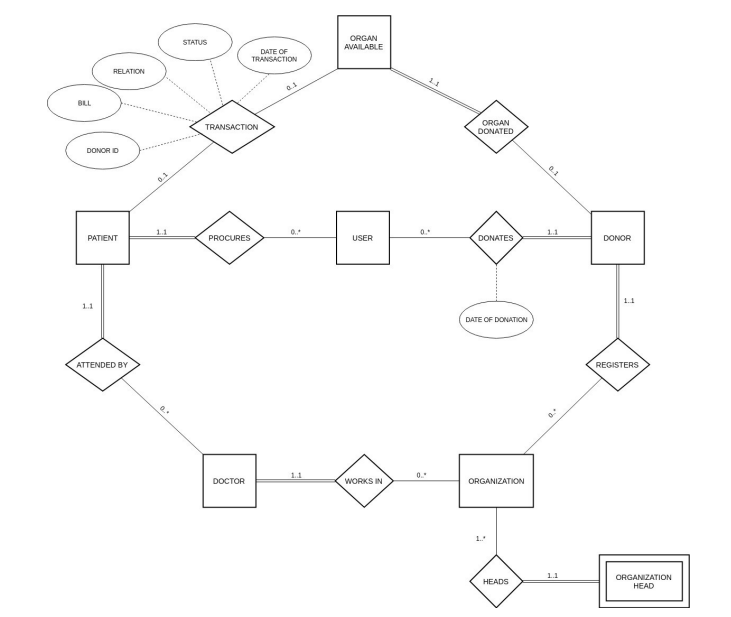
• HTML

• CSS

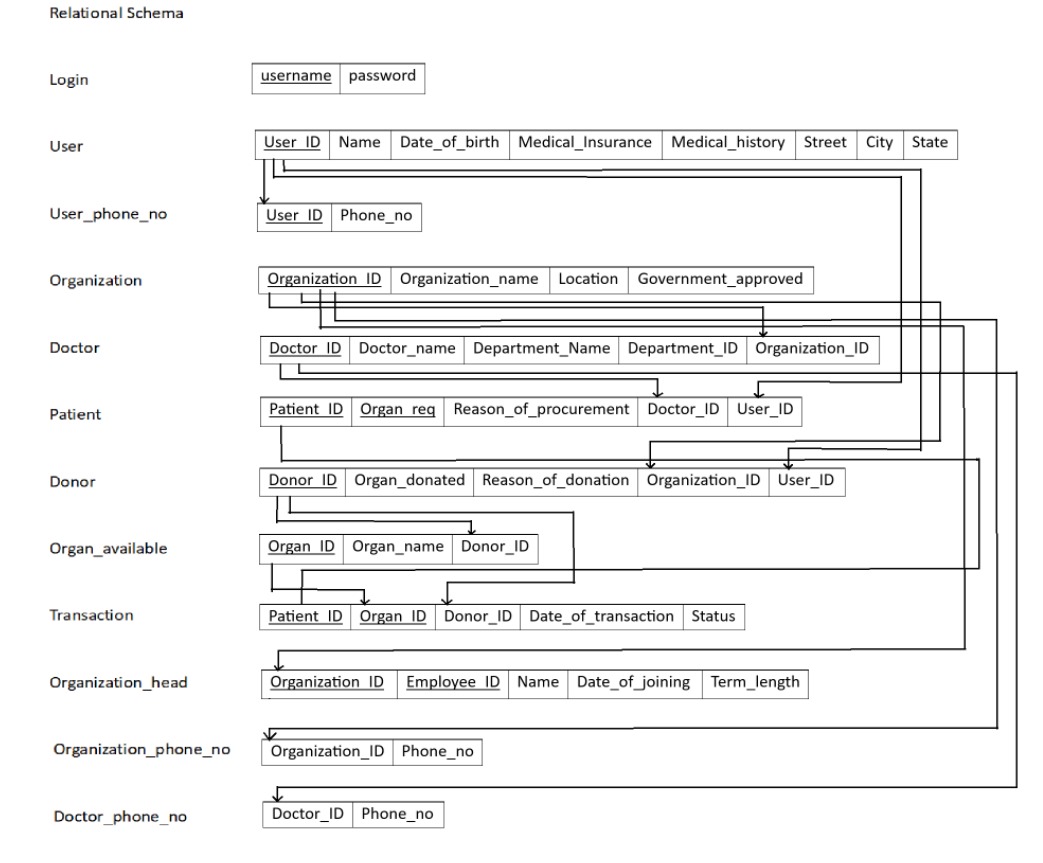
• Python

• Flask

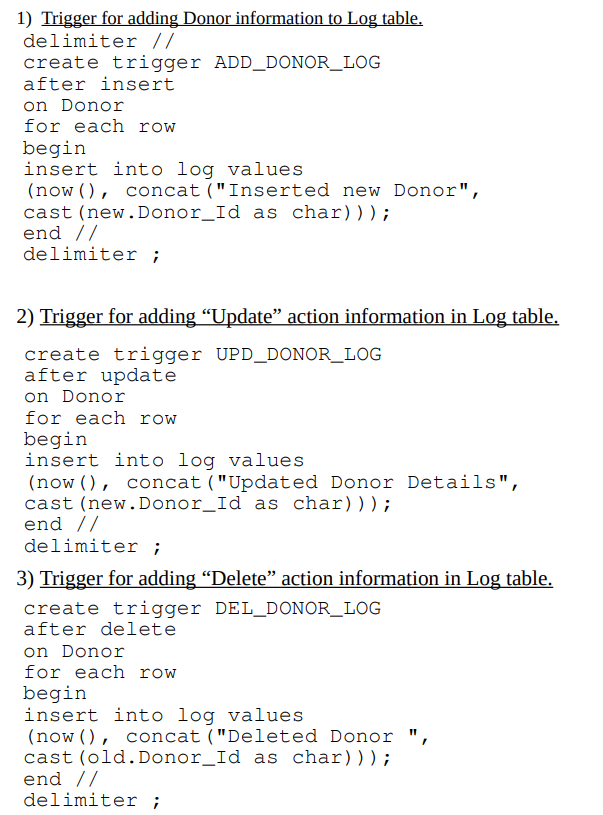
**ER Diagram**

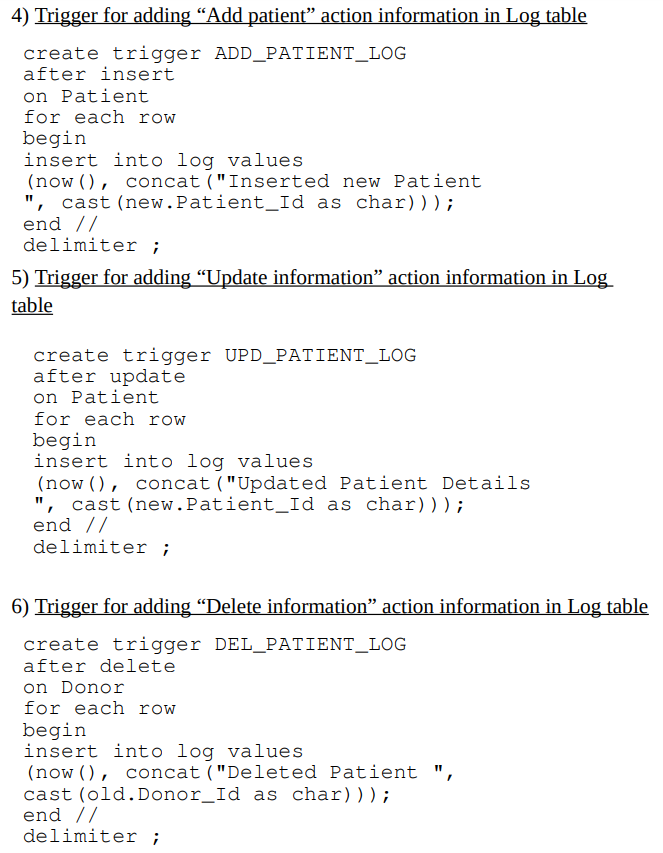
****

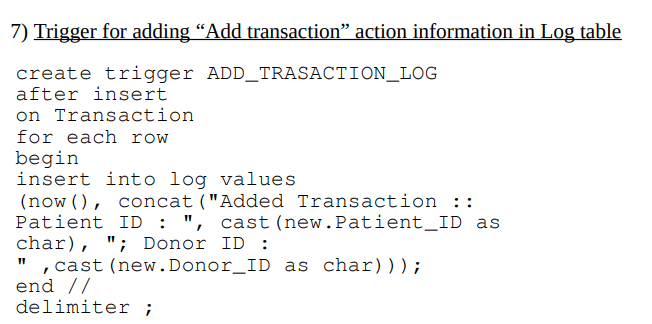
**Relational Schema**

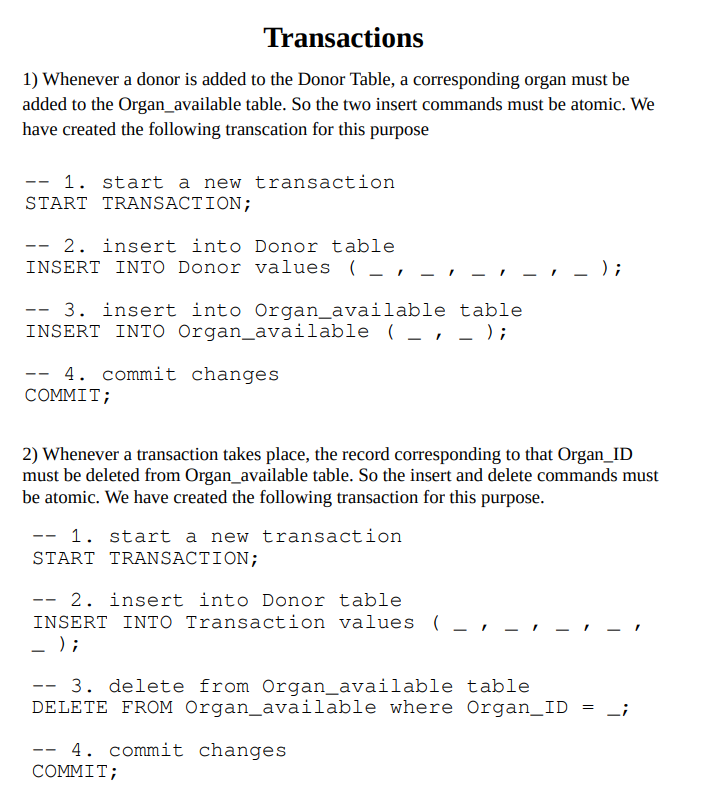
****

**Triggers**









**Queries**

### **1. Login Verification Query**

SELECT \* FROM login WHERE username = '%s'

**Explanation: This query retrieves all details of a user from the login table where the username matches the input provided during login. It is used to verify the user's existence and credentials.**

### **2. Retrieve User Details**

Select \* from User where User.User\_ID = %s

**Explanation: This query fetches all information about a user from the User table based on the provided User\_ID. It is used to display user details.**

### **3. Retrieve User Phone Numbers**

Select \* from User\_phone\_no where User\_ID = %s

**Explanation: This query retrieves all phone numbers associated with a user from the User\_phone\_no table using User\_ID.**

### **4. Retrieve Patient Details**

select Patient\_ID, organ\_req, reason\_of\_procurement, Doctor\_name

from Patient

inner join Doctor on Doctor.Doctor\_ID = Patient.Doctor\_ID and User\_ID = %s

**Explanation: This query retrieves patient details, including their ID, requested organ, reason for procurement, and the associated doctor's name, by joining the Patient and Doctor tables.**

### **5. Retrieve Donor Details**

select Donor\_ID, organ\_donated, reason\_of\_donation, Organization\_name

from Donor

inner join Organization on Organization.Organization\_ID = Donor.Organization\_ID and User\_ID = %s

**Explanation: This query fetches donor details, including ID, donated organ, reason for donation, and associated organization, by joining the Donor and Organization tables.**

### **6. Retrieve Transaction History**

select distinct Transaction.Patient\_ID, Transaction.Donor\_ID, Organ\_ID, Date\_of\_transaction, Status

from Transaction, Patient, Donor

where (Patient.User\_ID = %s and Patient.Patient\_ID = Transaction.Patient\_ID)

or (Donor.User\_Id = %s and Donor.Donor\_ID = Transaction.Donor\_ID)

**Explanation: This query retrieves the transaction history involving patients and donors for the given User\_ID, including transaction details such as organ ID, date, and status.**

### **7. Delete User**

DELETE FROM User where User\_ID = %s

**Explanation: This query deletes a user from the User table based on the provided User\_ID.**

### **8. Retrieve All Data for a Table**

SELECT \* from {id.capitalize()}

**Explanation: This dynamically constructed query retrieves all records from a table specified by the id parameter, which is capitalized for proper formatting.**

### **9. Retrieve Patient Information**

select Patient\_ID, organ\_req, reason\_of\_procurement, Doctor\_name

from Patient

inner join Doctor

on Doctor.Doctor\_ID = Patient.Doctor\_ID and User\_ID = %s

**Explanation: This query retrieves information about patients associated with a specific user, including the requested organ, reason for procurement, and the doctor's name, by joining the Patient and Doctor tables.**

### **10. Retrieve Donor Information**

select Donor\_ID, organ\_donated, reason\_of\_donation, Organization\_name

from Donor

inner join Organization

on Organization.Organization\_ID = Donor.Organization\_ID and User\_ID = %s

**Explanation: This query fetches details of donors related to a specific user, including the donated organ, reason for donation, and the organization they are associated with, by joining the Donor and Organization tables.**

### **11. Retrieve Transaction Information**

select distinct Transaction.Patient\_ID, Transaction.Donor\_ID, Organ\_ID, Date\_of\_transaction, Status

from Transaction, Patient, Donor

where (Patient.User\_ID = %s and Patient.Patient\_ID = Transaction.Patient\_ID)

or (Donor.User\_Id = %s and Donor.Donor\_ID = Transaction.Donor\_ID)

**Explanation: This query retrieves unique transaction records involving both patients and donors associated with a user, including details such as organ ID, transaction date, and status. It cross-references data from the Transaction, Patient, and Donor tables.**

### **12. Retrieve List of Distinct Donated Organs**

select distinct Organ\_donated

from Transaction

inner join Donor on Transaction.Donor\_ID = Donor.Donor\_ID

**Explanation: This query retrieves a list of unique organs donated, by joining the Transaction and Donor tables, to identify all distinct types of organs involved in transactions.**

### **13. Count Successful Transactions for an Organ**

select count(\*)

from Transaction

inner join Donor on Donor.Donor\_ID = Transaction.Donor\_ID

where Organ\_donated = '%s' and Status = 1

**Explanation: This query counts the number of successful transactions (indicated by Status = 1) for a specific organ by joining the Transaction and Donor tables.**

### **14. Count Unsuccessful Transactions for an Organ**

select count(\*)

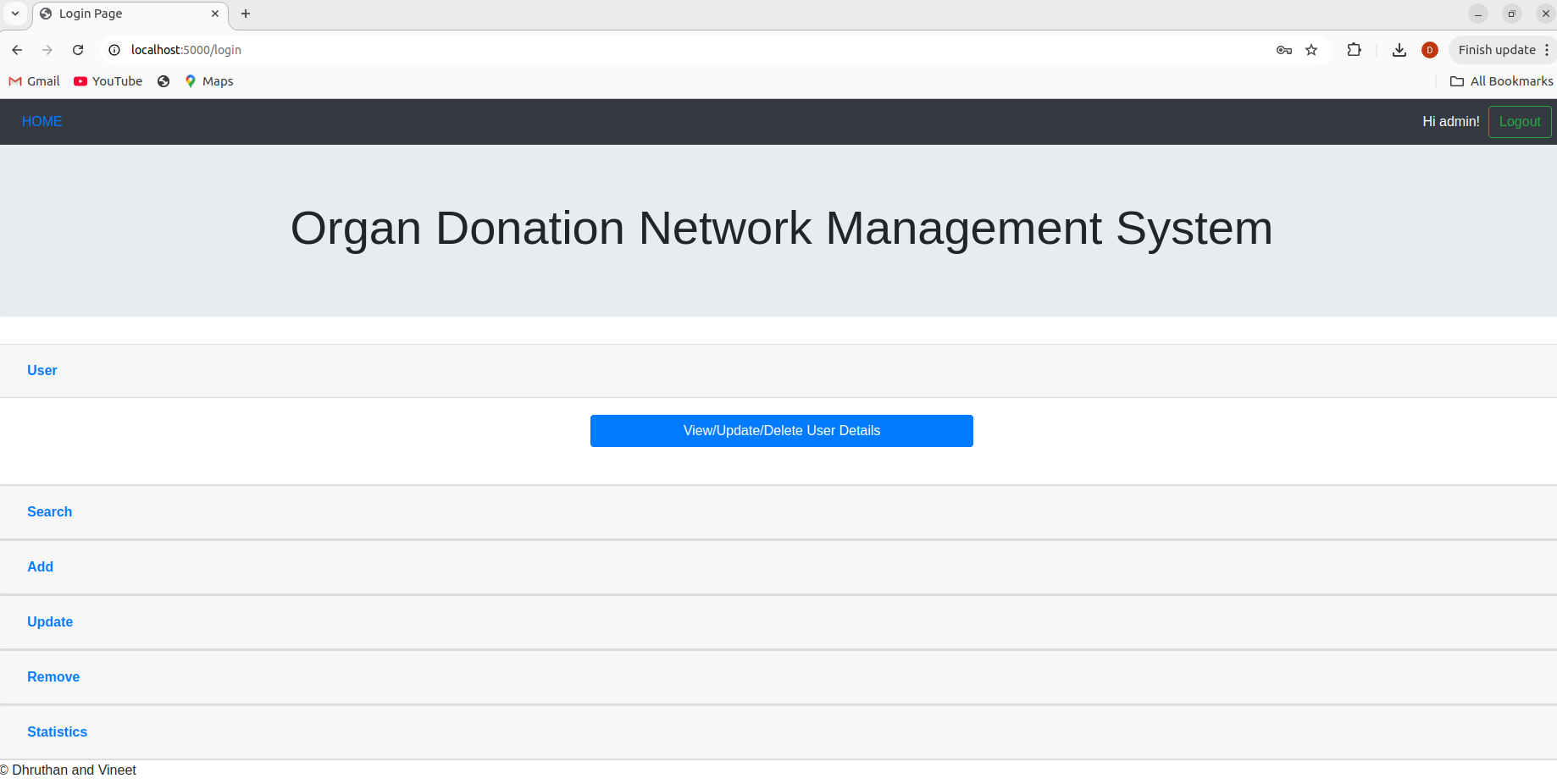
from Transaction

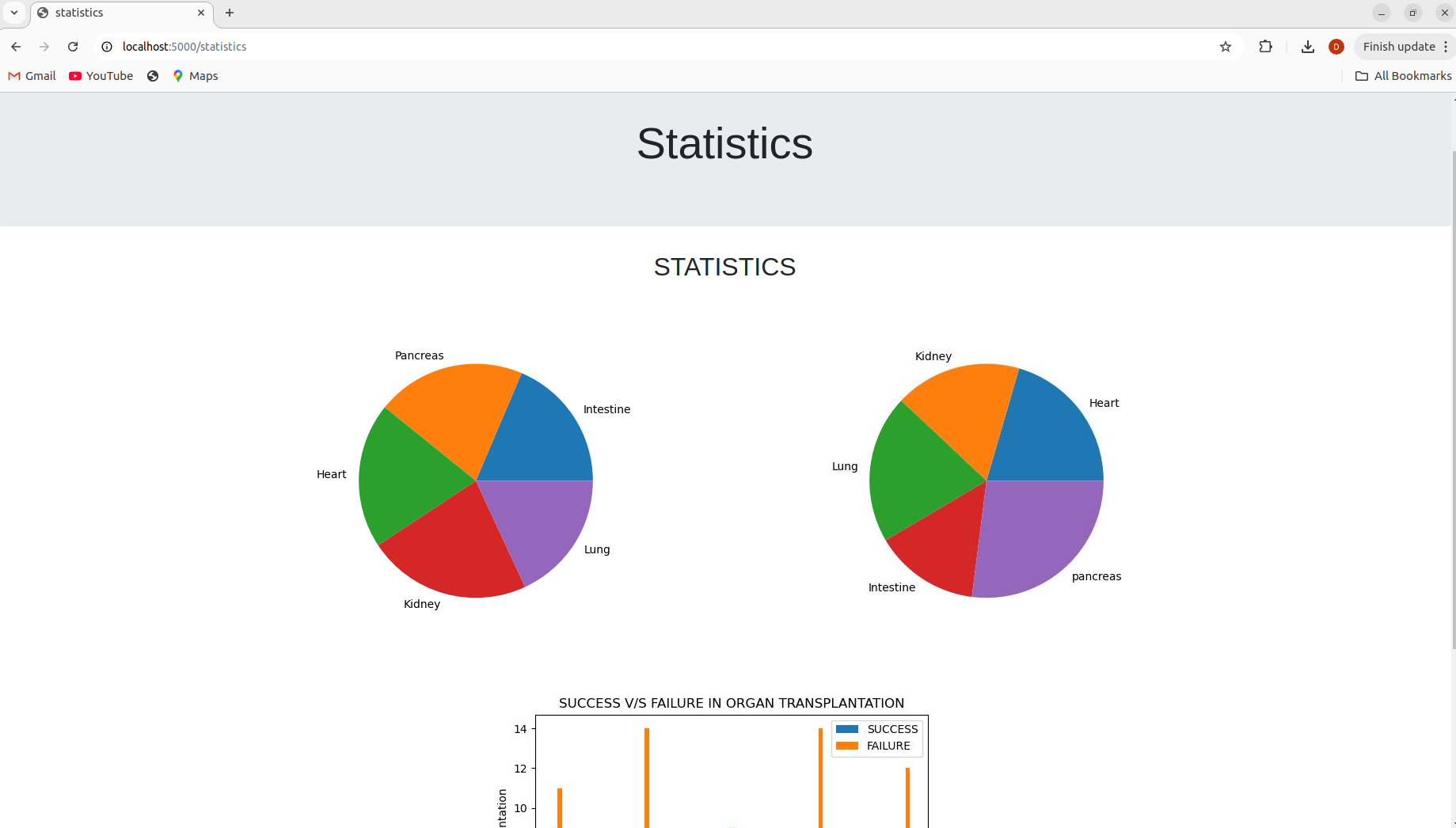
inner join Donor on Donor.Donor\_ID = Transaction.Donor\_ID

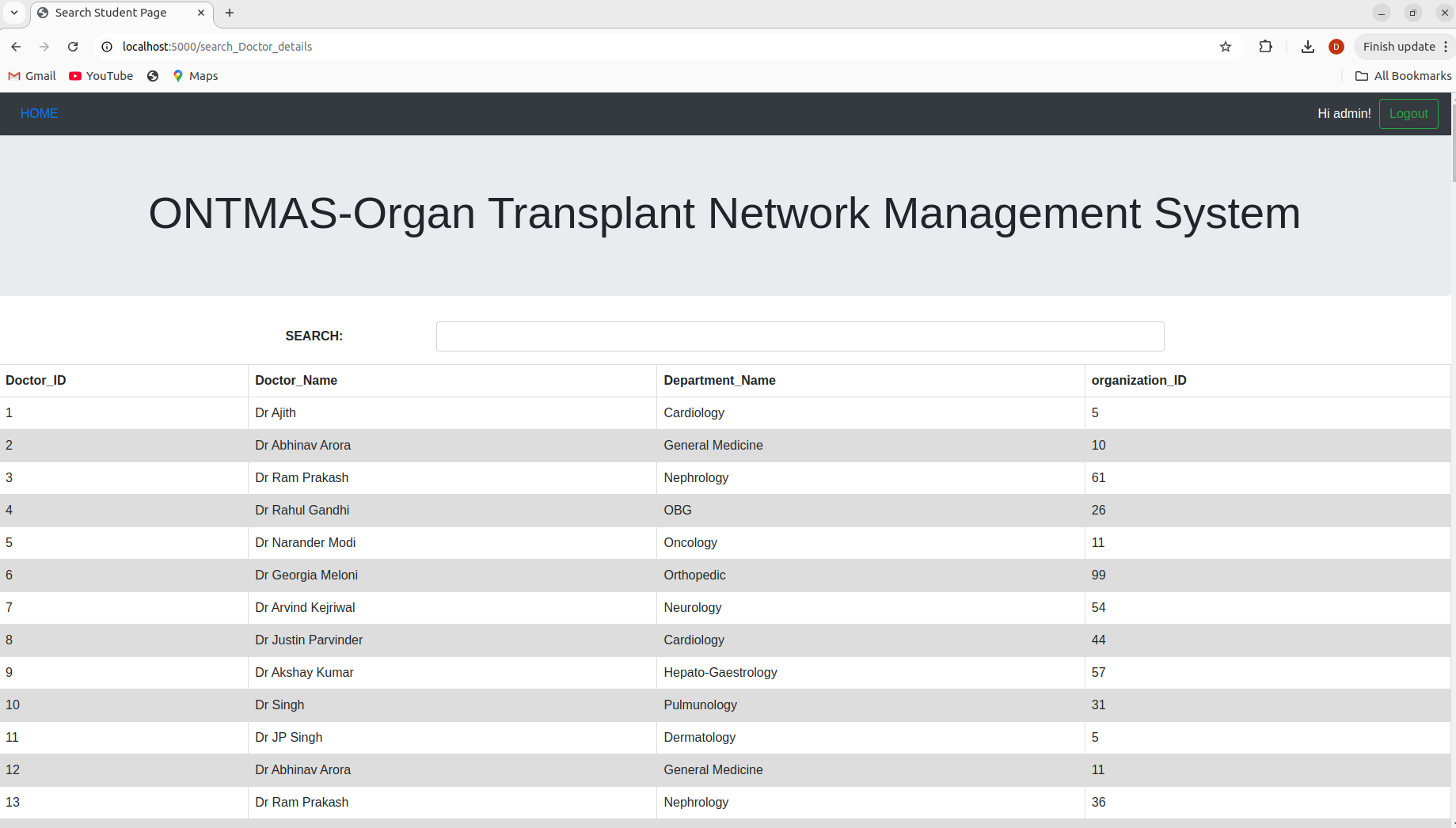
where Organ\_donated = '%s' and Status = 0

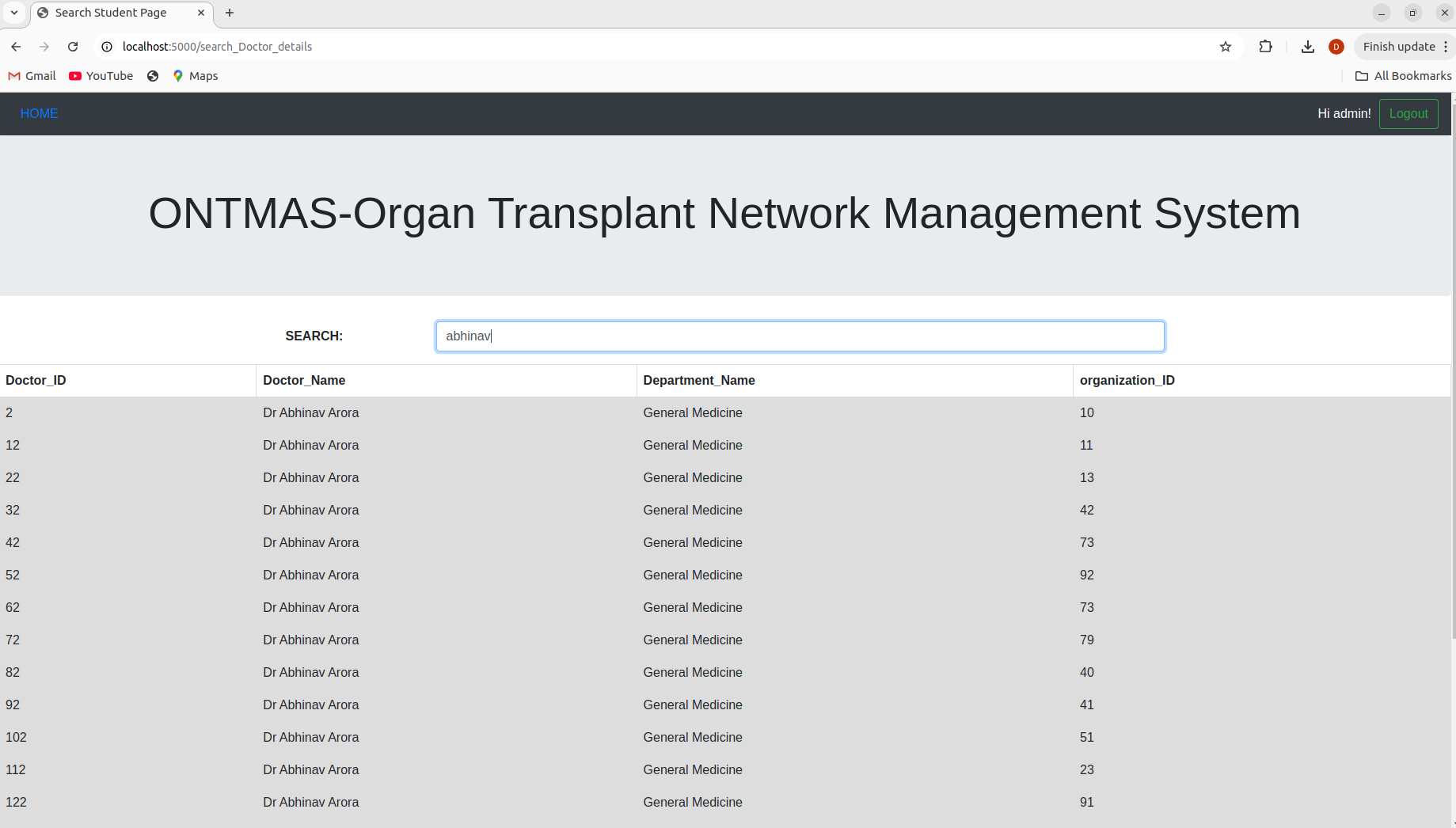
**Explanation: This query counts the number of unsuccessful transactions (indicated by Status = 0) for a specific organ by joining the Transaction and Donor tables.**

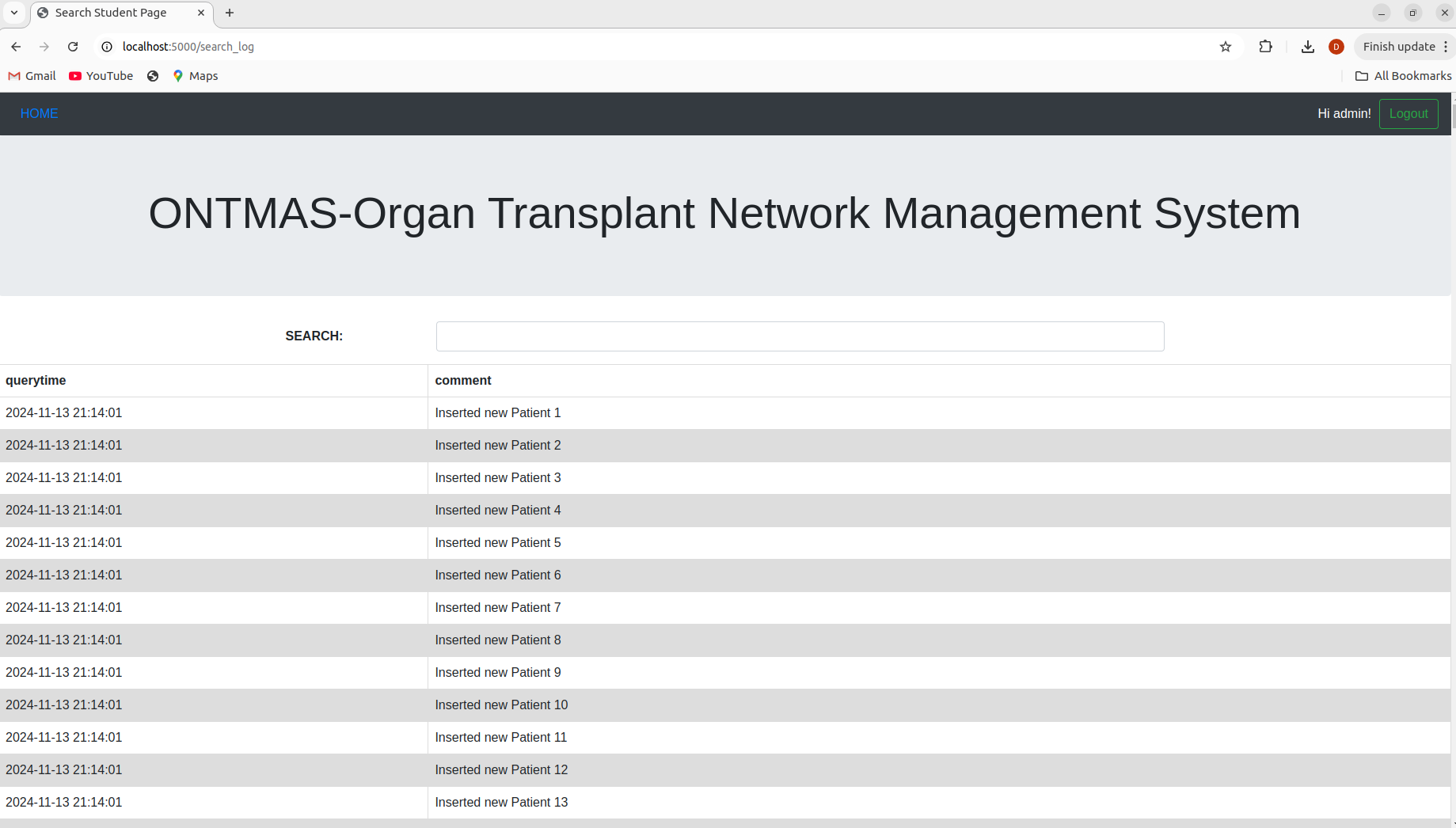
**Screenshots**

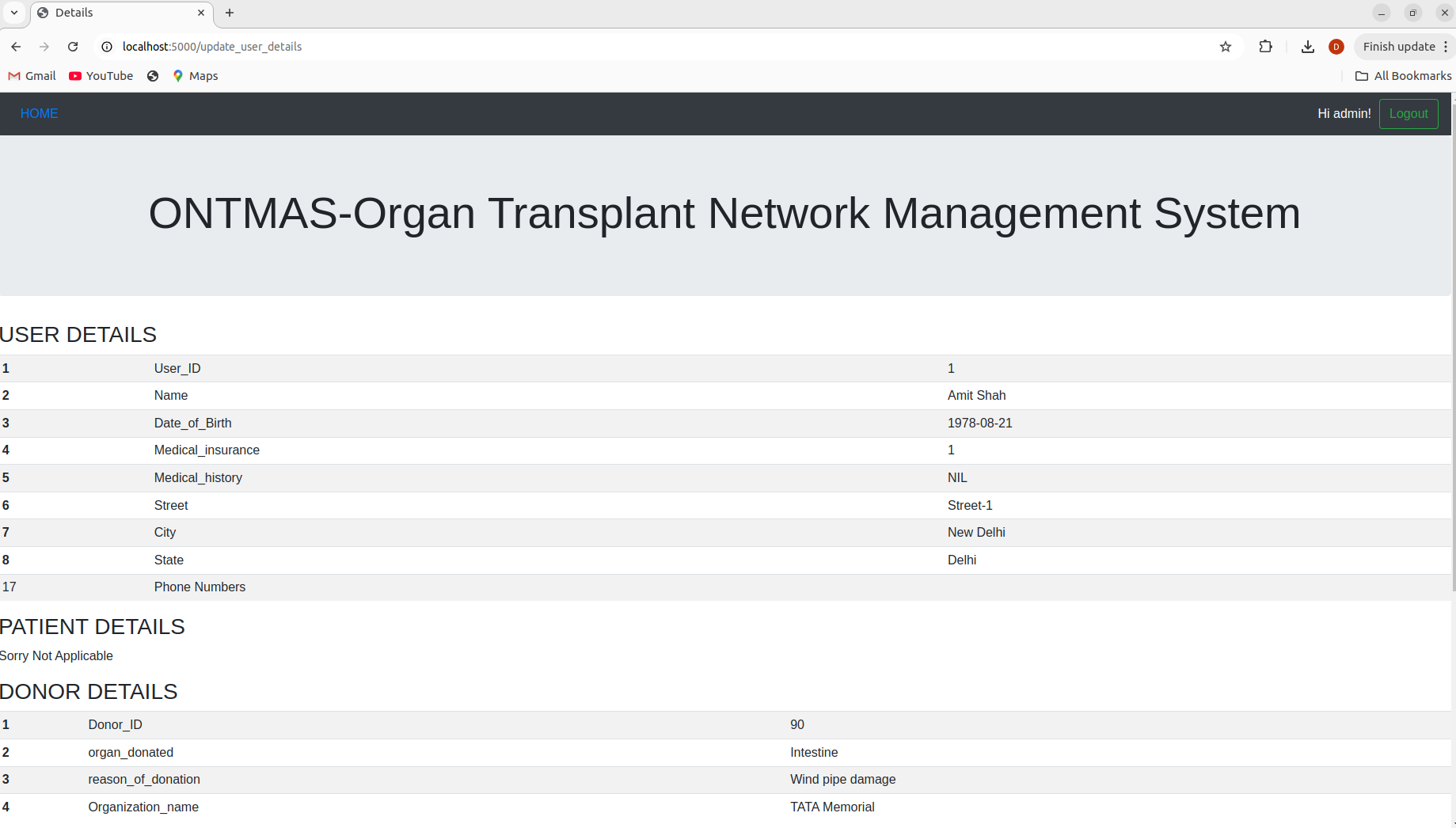
****

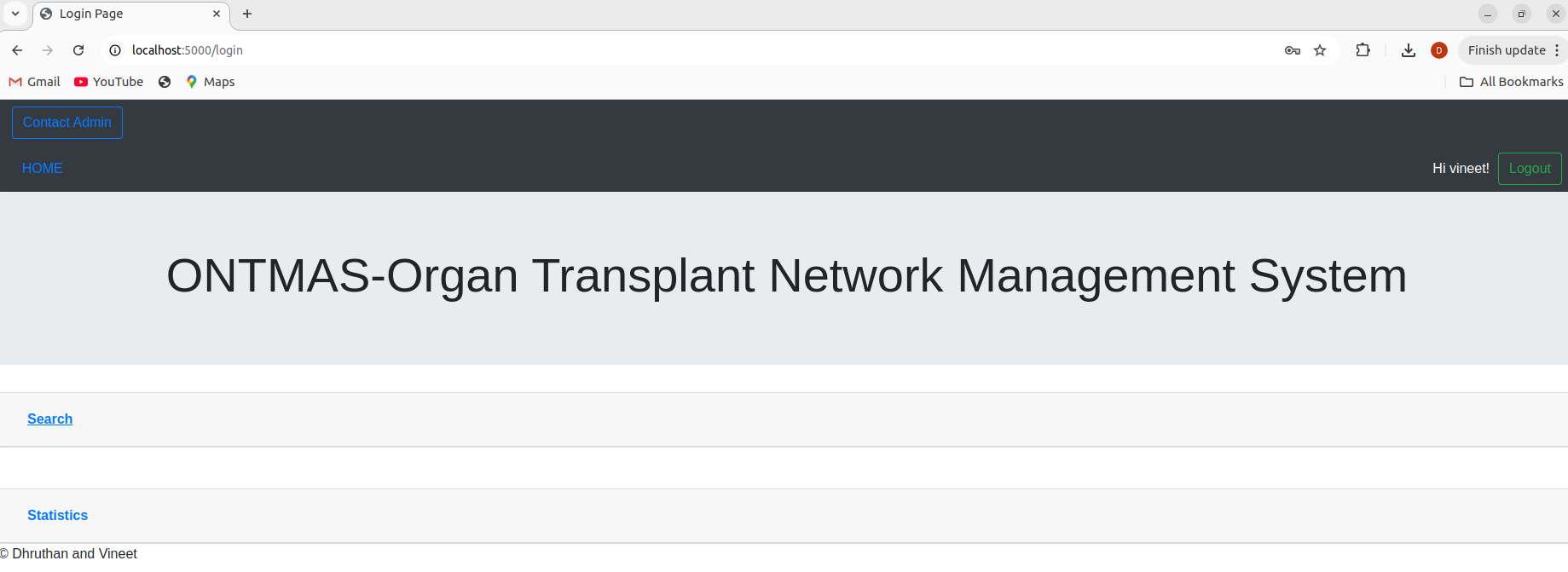
****

****

****

****

****

****