“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	**PES1UG22CS697**
Dhruthan MN	**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.

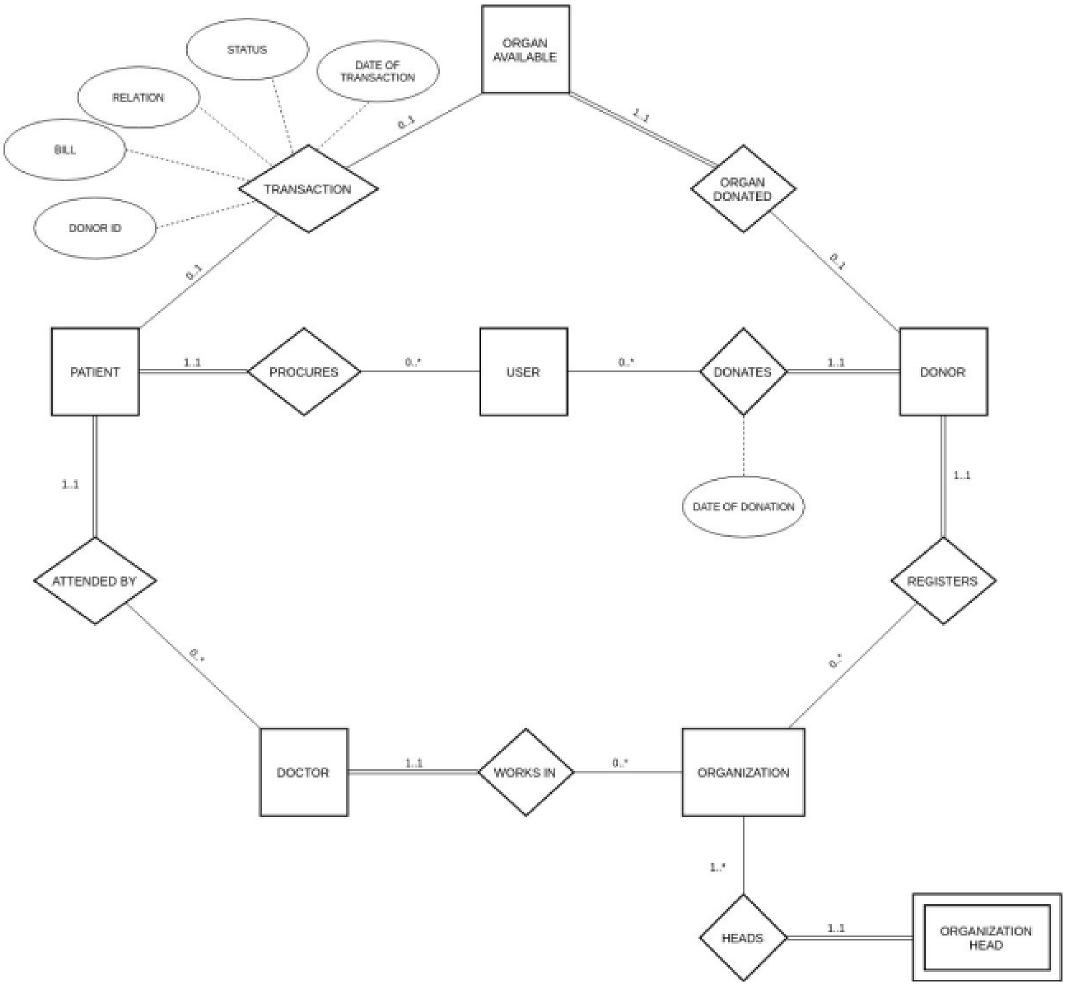
Basic Steps in Implementation :

- 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.

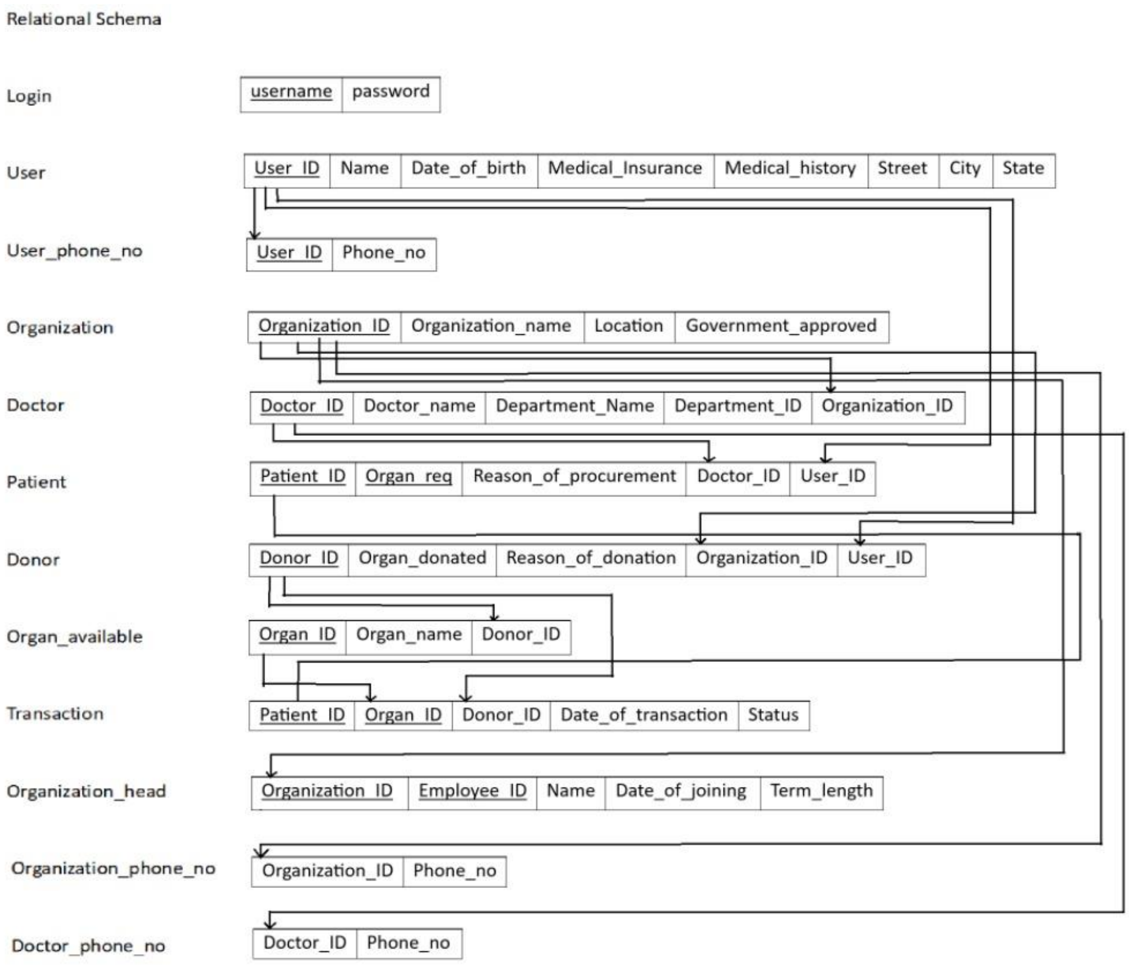
Technologies Used:

- MY
- HTML
- CSS
- Python
- Flask

ER Diagram



Relational Schema



Relational Schema

Login

username	password

User

User_ID	Name	Date_of_birth	Medical_Insurance	Medical_history	Street	City	State

User_phone_no

User_ID	Phone_no

Organization

Organization_ID	Organization_name	Location	Government_approved

Doctor

Doctor_ID	Doctor_name	Department_Name	Department_ID	Organization_ID

Patient

Patient_ID	Organ_req	Reason_of_procurement	Doctor_ID	User_ID

Donor

Donor_ID	Organ_donated	Reason_of_donation	Organization_ID	User_ID

Organ_available

Organ_ID	Organ_name	Donor_ID

Transaction

Patient_ID	Organ_ID	Donor_ID	Date_of_transaction	Status

Organization_head

Organization_ID	Employee_ID	Name	Date_of_joining	Term_length

Organization_phone_no

Organization_ID	Phone_no

Doctor_phone_no

Doctor_ID	Phone_no

Triggers

1) Trigger for adding Donor information to Log table.

```
delimiter //
create trigger ADD_DONOR_LOG
after insert
on Donor
for each row
begin
insert into log values
(now(), concat("Inserted new Donor",
cast(new.Donor_Id as char)));
end //
delimiter ;
```

2) Trigger for adding “Update” action information in Log table.

```
create trigger UPD_DONOR_LOG
after update
on Donor
for each row
begin
insert into log values
(now(), concat("Updated Donor Details",
cast(new.Donor_Id as char)));
end //
delimiter ;
```

3) Trigger for adding “Delete” action information in Log table.

```
create trigger DEL_DONOR_LOG
after delete
on Donor
for each row
begin
insert into log values
(now(), concat("Deleted Donor ",
cast(old.Donor_Id as char)));
end //
delimiter ;
```

4) Trigger for adding “Add patient” action information in Log table

```
create trigger ADD_PATIENT_LOG
after insert
on Patient
for each row
begin
insert into log values
(now(), concat("Inserted new Patient
", cast(new.Patient_Id as char)));
end //
delimiter ;
```

5) Trigger for adding “Update information” action information in Log table

```
create trigger UPD_PATIENT_LOG
after update
on Patient
for each row
begin
insert into log values
(now(), concat("Updated Patient Details
", cast(new.Patient_Id as char)));
end //
delimiter ;
```

6) Trigger for adding “Delete information” action information in Log table

```
create trigger DEL_PATIENT_LOG
after delete
on Donor
for each row
begin
insert into log values
(now(), concat("Deleted Patient ",
cast(old.Donor_Id as char)));
end //
delimiter ;
```

7) Trigger for adding “Add transaction” action information in Log table

```
create trigger ADD_TRASACTION_LOG
after insert
on Transaction
for each row
begin
insert into log values
(now(), concat("Added Transaction ::
Patient ID : ", cast(new.Patient_ID as
char), "; Donor ID :
" ,cast(new.Donor_ID as char)));
end //
delimiter ;
```

Transactions

1) Whenever a donor is added to the Donor Table, a corresponding organ must be added to the Organ_available table. So the two insert commands must be atomic. We have created the following transcation for this purpose

```
-- 1. start a new transaction
START TRANSACTION;

-- 2. insert into Donor table
INSERT INTO Donor values ( _ , _ , _ , _ , _ );

-- 3. insert into Organ_available table
INSERT INTO Organ_available ( _ , _ );

-- 4. commit changes
COMMIT;
```

2) Whenever a transaction takes place, the record corresponding to that Organ_ID must be deleted from Organ_available table. So the insert and delete commands must be atomic. We have created the following transaction for this purpose.

```
-- 1. start a new transaction
START TRANSACTION;

-- 2. insert into Donor table
INSERT INTO Transaction values ( _ , _ , _ , _ ,
_ );

-- 3. delete from Organ_available table
DELETE FROM Organ_available where Organ_ID = _;

-- 4. commit changes
COMMIT;
```

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.

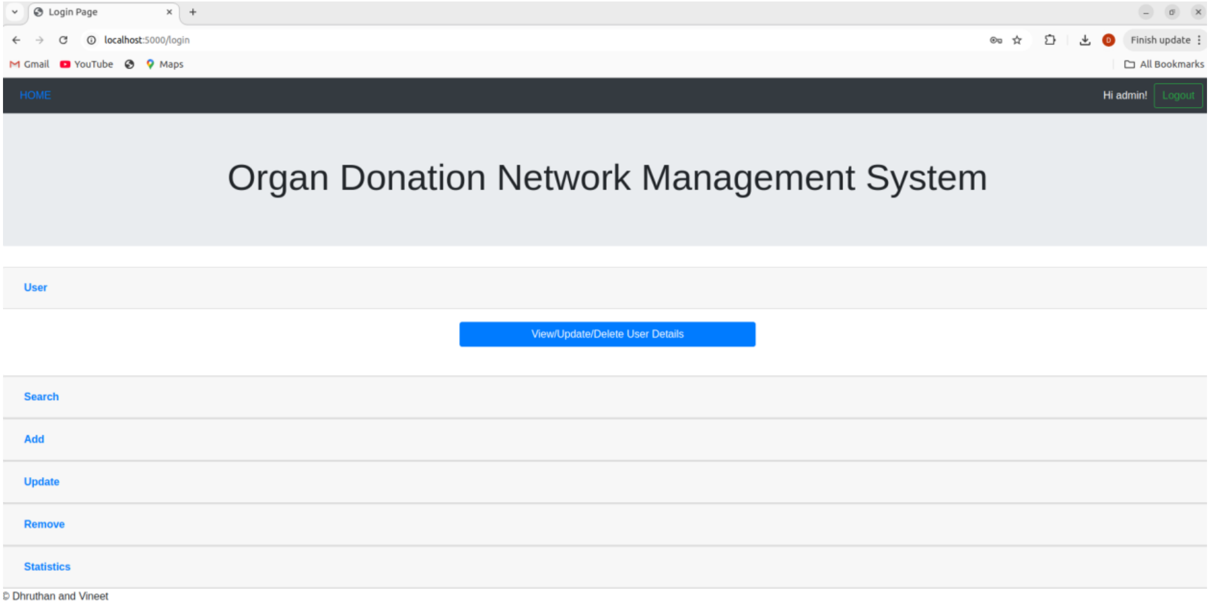
---

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
```

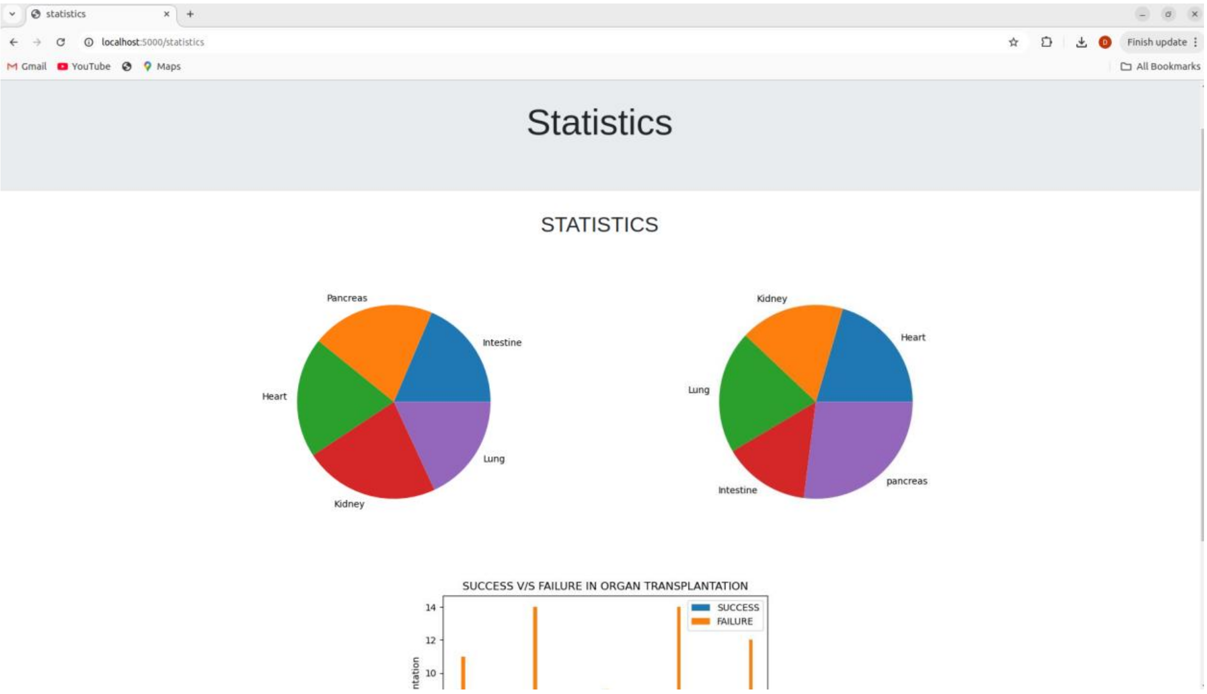
Explanation: This query counts the number of unsuccessful transactions (indicated by Status = 0) for a specific organ by joining the Transaction and Donor tables.

Screenshots



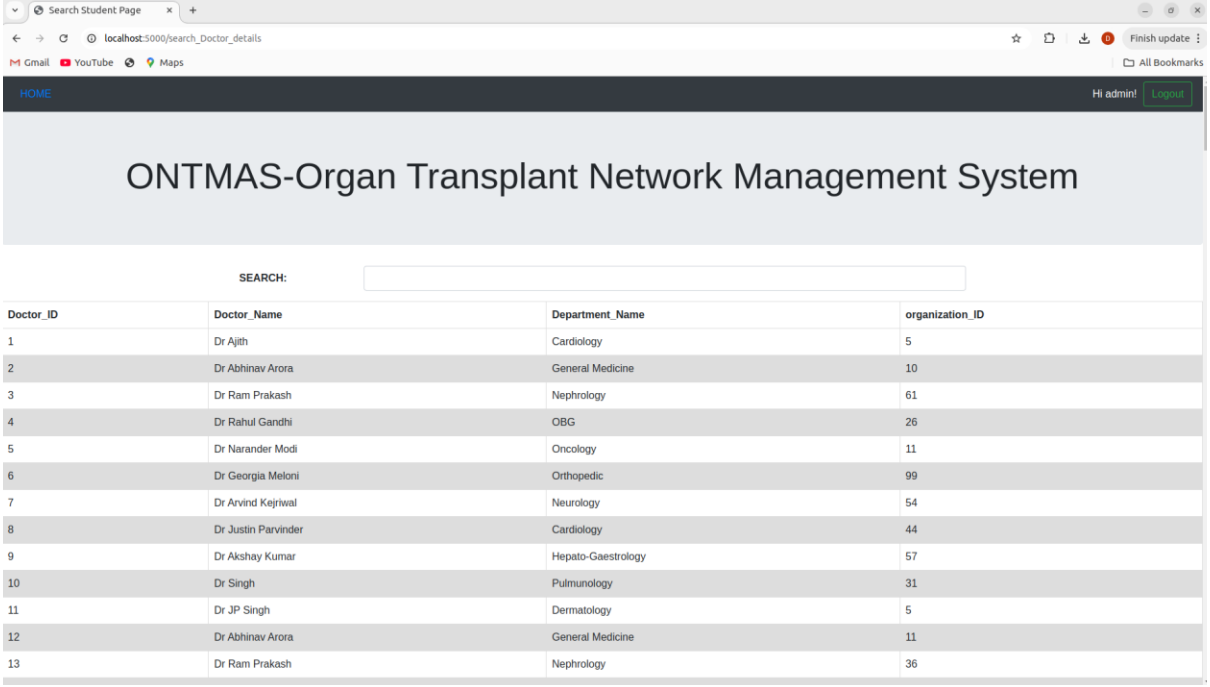
Organ Donation Network Management System

© Dhruthan and Vineet



Statistics

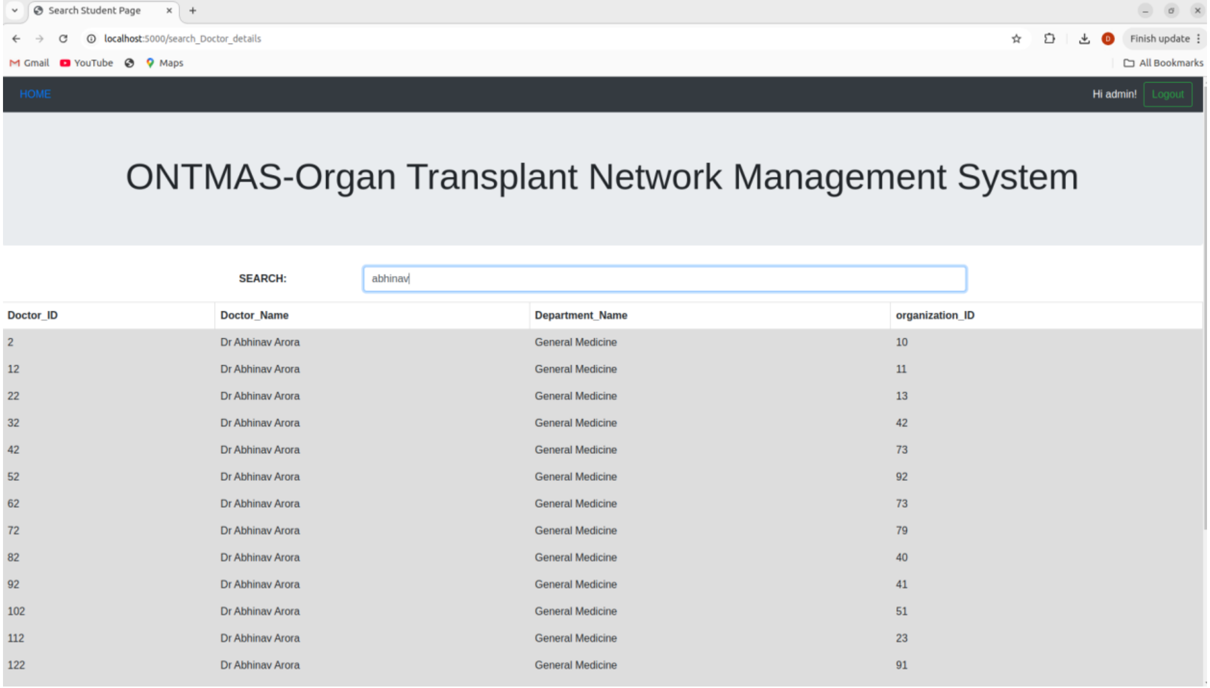
STATISTICS



ONTMAS-Organ Transplant Network Management System

SEARCH:

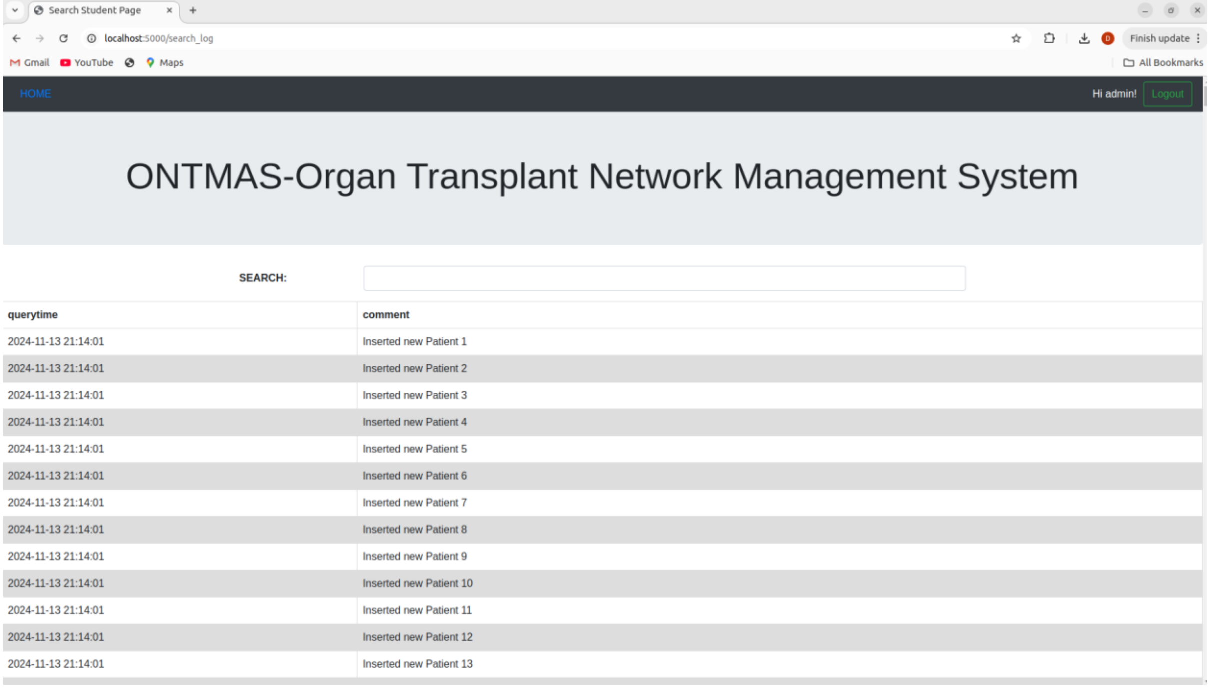
Doctor_ID	Doctor_Name	Department_Name	organization_ID
1	Dr Ajith	Cardiology	5
2	Dr Abhinav Arora	General Medicine	10
3	Dr Ram Prakash	Nephrology	61
4	Dr Rahul Gandhi	OBG	26
5	Dr Narander Modi	Oncology	11
6	Dr Georgia Meloni	Orthopedic	99
7	Dr Arvind Kejriwal	Neurology	54
8	Dr Justin Parvinder	Cardiology	44
9	Dr Akshay Kumar	Hepato-Gaestrology	57
10	Dr Singh	Pulmunology	31
11	Dr JP Singh	Dermatology	5
12	Dr Abhinav Arora	General Medicine	11
13	Dr Ram Prakash	Nephrology	36



ONTMAS-Organ Transplant Network Management System

SEARCH: abhinav

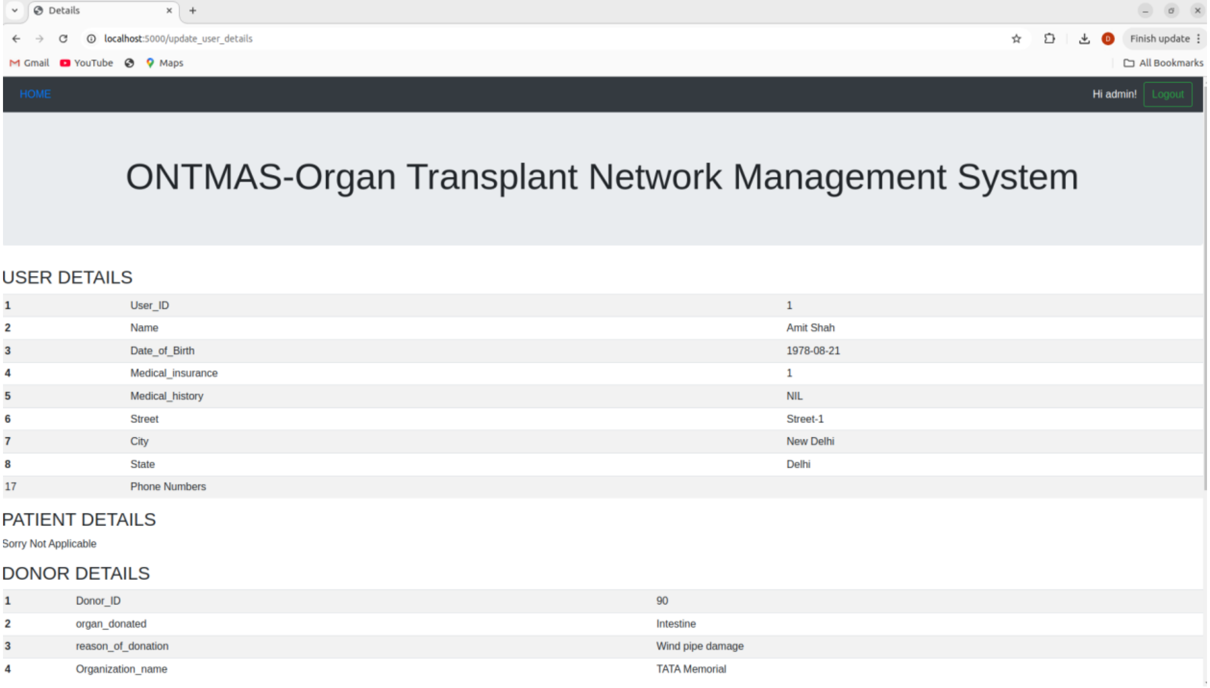
Doctor_ID	Doctor_Name	Department_Name	organization_ID
2	Dr Abhinav Arora	General Medicine	10
12	Dr Abhinav Arora	General Medicine	11
22	Dr Abhinav Arora	General Medicine	13
32	Dr Abhinav Arora	General Medicine	42
42	Dr Abhinav Arora	General Medicine	73
52	Dr Abhinav Arora	General Medicine	92
62	Dr Abhinav Arora	General Medicine	73
72	Dr Abhinav Arora	General Medicine	79
82	Dr Abhinav Arora	General Medicine	40
92	Dr Abhinav Arora	General Medicine	41
102	Dr Abhinav Arora	General Medicine	51
112	Dr Abhinav Arora	General Medicine	23
122	Dr Abhinav Arora	General Medicine	91



ONTMAS-Organ Transplant Network Management System

SEARCH:

querytime	comment
2024-11-13 21:14:01	Inserted new Patient 1
2024-11-13 21:14:01	Inserted new Patient 2
2024-11-13 21:14:01	Inserted new Patient 3
2024-11-13 21:14:01	Inserted new Patient 4
2024-11-13 21:14:01	Inserted new Patient 5
2024-11-13 21:14:01	Inserted new Patient 6
2024-11-13 21:14:01	Inserted new Patient 7
2024-11-13 21:14:01	Inserted new Patient 8
2024-11-13 21:14:01	Inserted new Patient 9
2024-11-13 21:14:01	Inserted new Patient 10
2024-11-13 21:14:01	Inserted new Patient 11
2024-11-13 21:14:01	Inserted new Patient 12
2024-11-13 21:14:01	Inserted new Patient 13



ONTMAS-Organ Transplant Network Management System

USER DETAILS

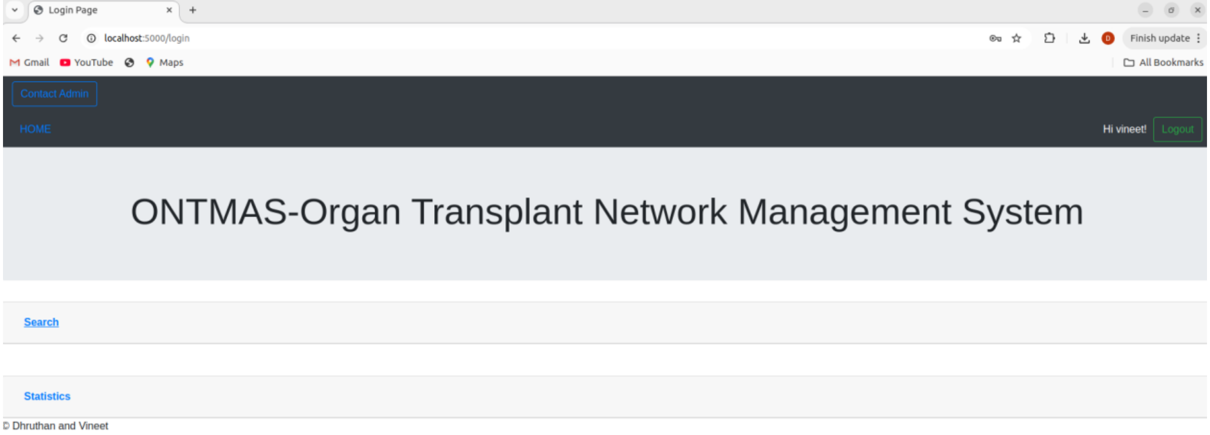
1	User_ID	1
2	Name	Amit Shah
3	Date_of_Birth	1978-08-21
4	Medical_insurance	1
5	Medical_history	NIL
6	Street	Street-1
7	City	New Delhi
8	State	Delhi
17	Phone Numbers	

PATIENT DETAILS

Sorry Not Applicable

DONOR DETAILS

1	Donor_ID	90
2	organ_donated	Intestine
3	reason_of_donation	Wind pipe damage
4	Organization_name	TATA Memorial



ONTMAS-Organ Transplant Network Management System

© Dhruthan and Vineet