horizontal line

**School of Engineering and Applied Science(SEAS), Ahmedabad University**

**B.Tech Semester IV: Database Management Systems**

**(CSE250)**

**Topic: Organ Bank Database Management System**

**Faculty: Shefali Naik**

**Section-1, Group: 57**

**Students' details**

|  |  |
| --- | --- |
| **Enrollment No.** | **Name of the Student** |
| AU1940124 | Sanskruti Shingala |
| AU1940173 | Priya Bhoraniya |
| AU1940195 | Krinal Boghani |

**1.1 Introduction:**

The Organ Bank Database Management System (DBMS) refers to the technology for creating and managing data of donors, organs, coordinating institutes and recipients for an organ bank. The main aim of this project is to computerize the manual process of managing data of Organ Banks. The process starts when Organ Bank arranges donation camps. In those donation camps, people donate the organs and the record of that is then handed over to DBMS. Organ banks have many coordinating institutes through which it provides service to the needy people. Therefore, Organ Bank matches the data of available organs with the data of recipients. However, organs can be only available after brain death testing of the donor(not necessary in the case of kidney and liver) and therefore a waiting list has been provided. Further, when any organ reaches the organ bank, a matching process has been done and transplanted to the one highest on the waiting list. Hence this whole data can be easily managed by the DBMS. On the basis of this process we have designed the database management system.

Our system can be used by two users i.e. Organ Bank and coordinating hospitals. Both users have certain privileges to watch and manipulate data.

The services of a organ bank can include:

* Handling donor, organ, coordinator, recipient details
* Preparing the waiting list for recipients
* Matching process of organs
* Handling camps

The services of coordinating Hospitals can include:

* Add and manage data of recipient of their institute
* Check Organ Availability
* View information of Camps
* View waiting List

**Used Languages and Softwares**

Front-end ⇒ HTML,CSS,Bootstrap

Back-end ⇒ MySQL

Front-end Database connectivity ⇒ PHP, javascript

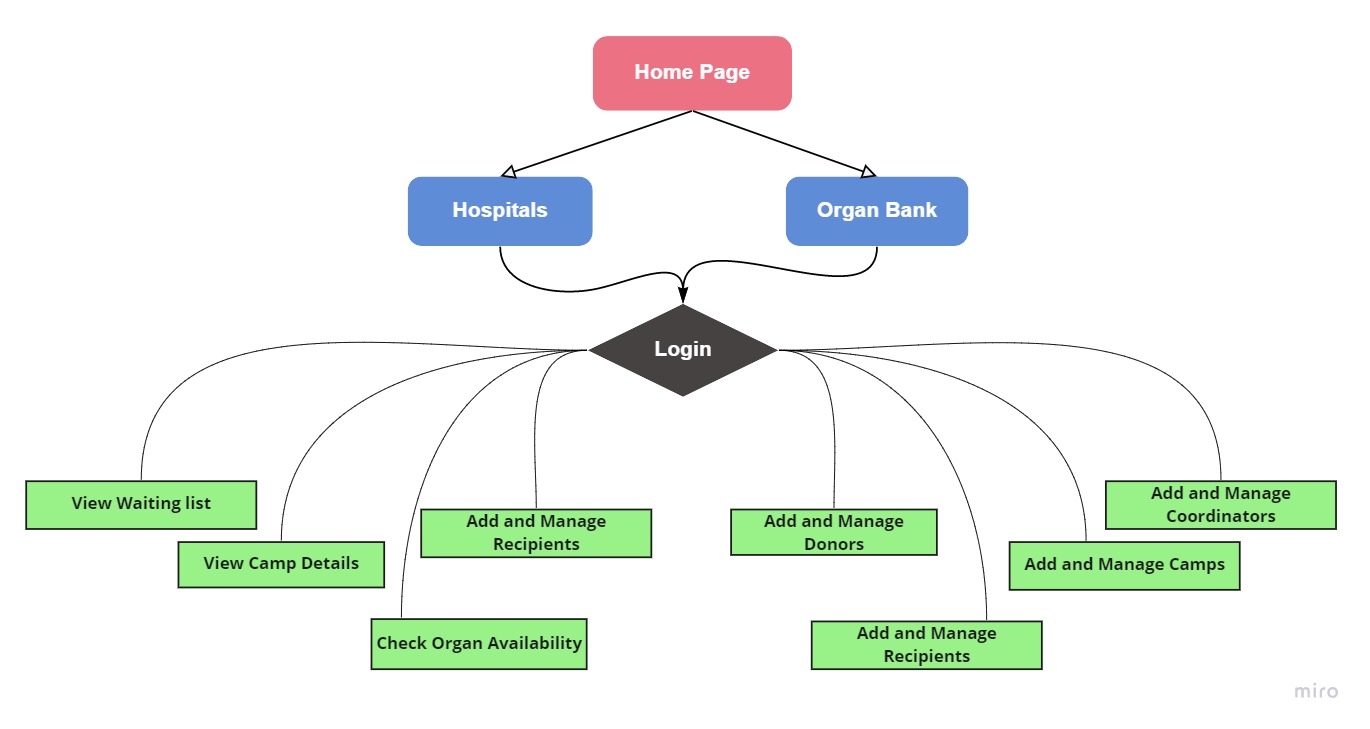
FlowChart and ER Diagram ⇒ MIRO

Data Dictionary Tables ⇒ MySQL Workbench

**1.2 Design of Project**

**1.2.1 Flow of Project**

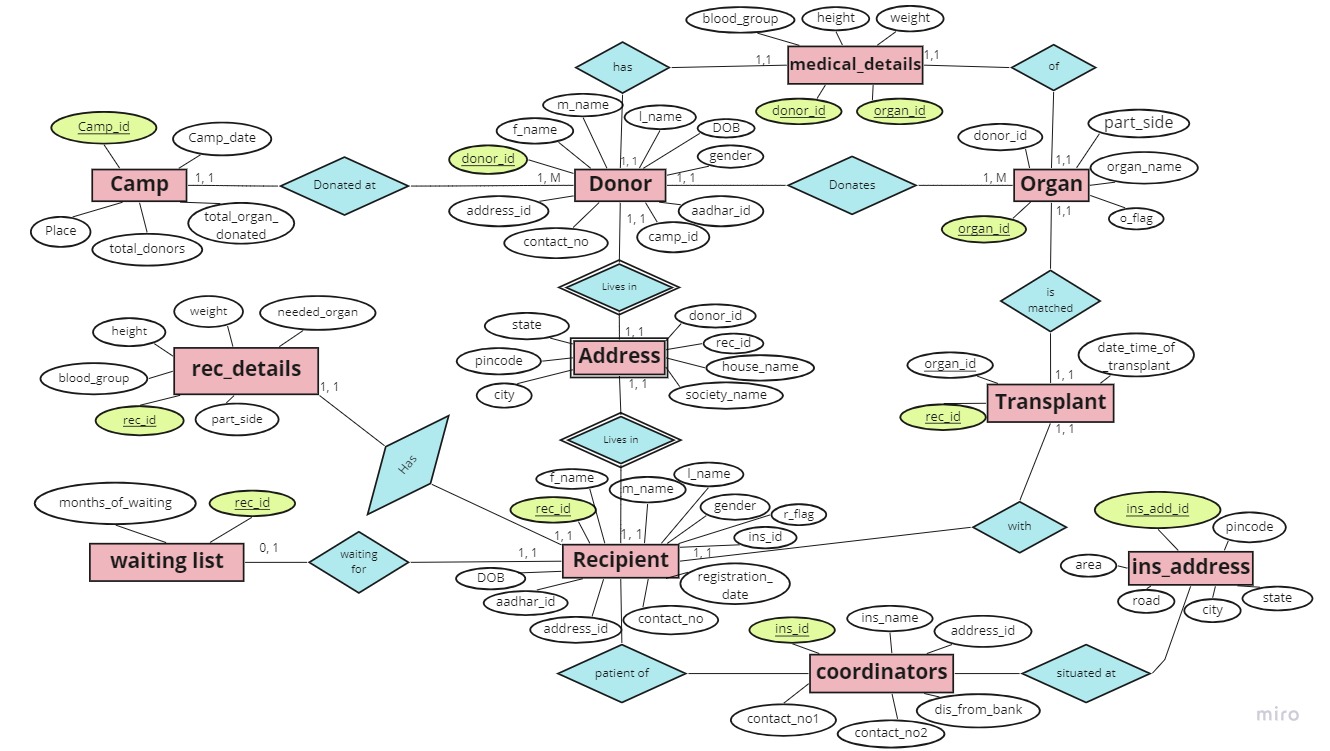
The flow starts from the home page where introductory details of the organ donation bank have been displayed. However, the admin of the bank and hospitals have to log in for managing the data. After successful login, services of hospitals and the bank can be accessible. Hospitals can view the waiting list, data of available organs, camps and can add recipients. However, the admin of the bank has privileges to add, update and delete the data of donors, recipients, organs, and camps. The admin can also do matches of organs for recipients on the waiting list. Further, it also keeps the data of transplants.

****

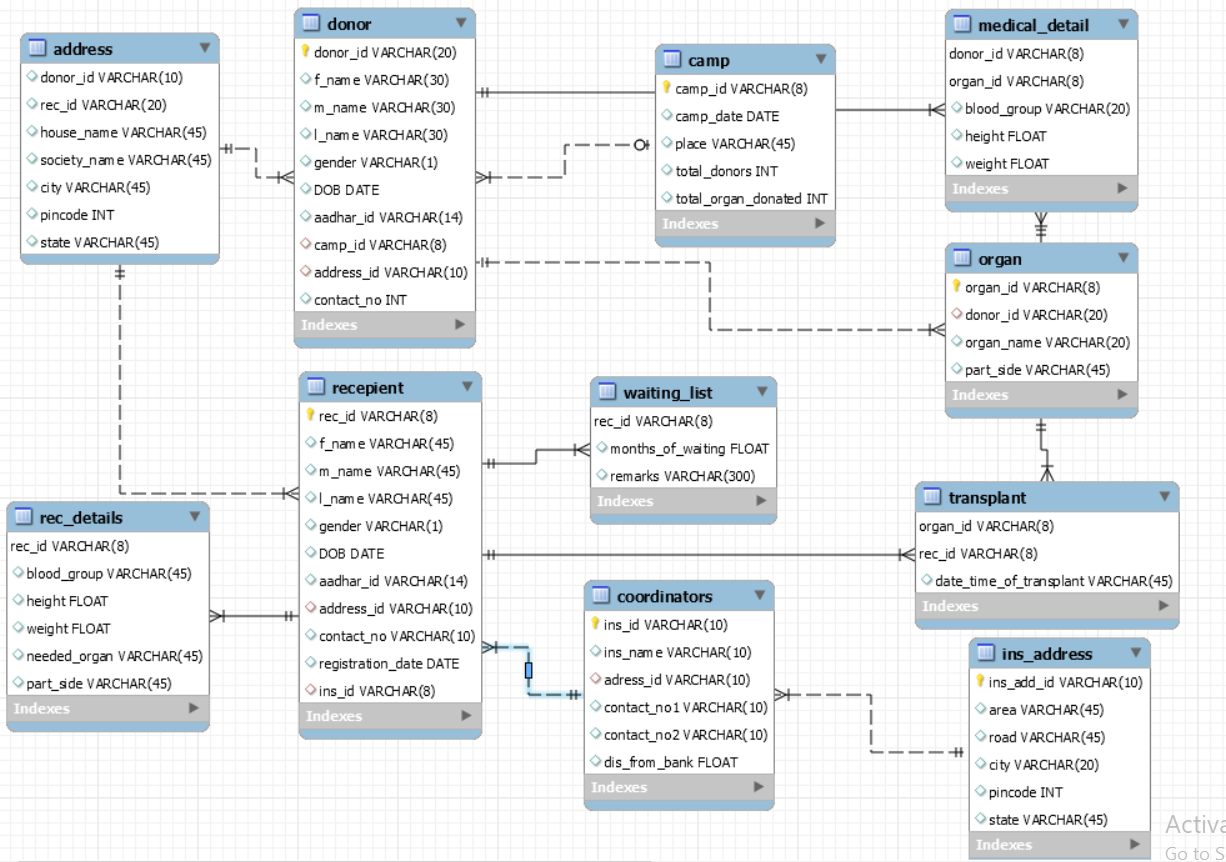
**1.2.2 ER Diagram**

**Relationships**

1. camp→donor (1-M)
2. donor→organ(1-M)
3. Donor→medical\_details (1-1)
4. organ→medical\_details(1-1)
5. coordinators→recipient(1-M)
6. recipient→recipient\_details(1-1)
7. donor→address(1-1)
8. recipient→address(1-1)
9. coordinators→ins\_address(1-1)
10. recipient→transplant(1-0)
11. recipient→waiting\_list(1-0)

****

**1.2.3 Data Dictionary**



**1.2.4 Indexes:**

**camp:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | camp\_id | 5 | A | No |

**coordinators:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | ins\_id | 6 | A | No |

**donor:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | donor\_id | 17 | A | No |

**ins\_address:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | ins\_add\_id | 6 | A | No |

**medical\_detail:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| organ\_id | BTREE | No | No | organ\_id | 18 | A | No |
| PRIMARY | BTREE | Yes | No | donor\_id | 18 | A | No |
| organ\_id | 18 | A | No |

**organ:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | organ\_id | 20 | A | No |

**recipient:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | rec\_id | 12 | A | No |

**recepient\_details**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| **PRIMARY** | **BTREE** | **Yes** | **No** | **rec\_id** | **13** | **A** | **No** |

**transplant**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| rec\_id | BTREE | No | No | rec\_id | 7 | A | No |
| PRIMARY | BTREE | Yes | No | organ\_id | 7 | A | No |
| rec\_id | 7 | A | No |

**waiting\_list**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keyname** | **Type** | **Unique** | **Packed** | **Column** | **Cardinality** | **Collation** | **Null** |
| PRIMARY | BTREE | Yes | No | rec\_id | 7 | A | No |

**1.3 Functionalities used in Backend:**

**1.3.1 Stored Procedures**

1. **To display details of the camps in the hospital page**

DELIMITER $$

CREATE PROCEDURE `camp\_det`()

begin

select camp\_id, place, camp\_date, total\_donors, total\_organ\_donated from camp ;

end$$

DELIMITER ;

$query = "call camp\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)) {

$place = $row['place'];

$camp\_date = $row['camp\_date'];

1. **Display details of the respective organ**

DELIMITER $$

CREATE PROCEDURE `Organ\_data`(IN `organ` VARCHAR(20), IN `part` VARCHAR(30))

BEGIN

select medical\_detail.blood\_group, medical\_detail.height, medical\_detail.weight from medical\_detail left join organ on

organ.organ\_id=medical\_detail. organ\_id where organ.organ\_name=organ and organ.part\_side=part;

END$$

DELIMITER ;

$query = "call Organ\_data('$\_POST[o\_name]','$\_POST[part]')";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)) {

$blood\_group = $row['blood\_group'];

$height = $row['height'];

$weight = $row['weight'];

1. **To display the recipient's details of the respective hospital**

DELIMITER $$

CREATE PROCEDURE `rec\_detailOf\_Hospital`(IN `ins\_id1` VARCHAR(8))

BEGIN

SELECT

recepient.rec\_id, recepient.f\_name, recepient.m\_name, recepient.l\_name, recepient.gender, recepient.DOB, recepient.aadhar\_id, recepient.contact\_no,

recepient.registration\_date, rec\_details.blood\_group, rec\_details.height, rec\_details.weight, rec\_details.needed\_organ, rec\_details.part\_side, address.house\_name, address.society\_name, address.city, address.pincode, address.state

from recepient, rec\_details,address

where trim(recepient.ins\_id)=ins\_id1 and trim(recepient.rec\_id)=trim(rec\_details.rec\_id )and trim(address.rec\_id)=trim(recepient.rec\_id);

END$$

DELIMITER ;

$query = "call rec\_detailOf\_Hospital('$\_SESSION[user]')";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$f\_name = $row['f\_name']; $m\_name = $row['f\_name'];

$l\_name = $row['l\_name']; $gender = $row['gender'];

$DOB = $row['DOB']; $aadhar\_id = $row['aadhar\_id'];

$contact\_no = $row['contact\_no'];

$registration\_date = $row['registration\_date'];

$blood\_group=$row['blood\_group'];$height=$row['height'];

$weight=$row['weight'];$needed\_organ=$row['needed\_organ'];

$part=$row['part\_side'];$house\_name = $row['house\_name'];

$society\_name= $row['society\_name'];$city = $row['city'];

$pincode = $row['pincode'];$state = $row['state'];

1. **To display Camp details in the admin page**

DELIMITER $$

CREATE PROCEDURE `camp\_det`()

begin

select camp\_id, place, camp\_date, total\_donors, total\_organ\_donated from camp ;

end$$

DELIMITER ;

$query = "call camp\_det()";

$query\_run = mysqli\_query($connection, $query

while ($row = mysqli\_fetch\_assoc($query\_run)){

$camp\_id = $row["camp\_id"];

$place = $row['place'];

$camp\_date = $row['camp\_date'];

$total\_donors=$row['total\_donors'];

$total\_organs=$row['total\_organ\_donated']

1. **To display coordinators details in the admin page**

DELIMITER $$

CREATE PROCEDURE `coordinators\_det`()

BEGIN

select coordinators.ins\_name,coordinators.contact\_no1, coordinators.contact\_no2,

ins\_address.area, ins\_address.road, ins\_address.city, ins\_address.pincode,

ins\_address.state, coordinators.dis\_from\_bank

from coordinators, ins\_address

where coordinators.adress\_id = ins\_address.ins\_add\_id;

END$$

DELIMITER ;

$query = "call coordinators\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$ins\_name = $row['ins\_name'];

$contact\_no1 = $row['contact\_no1'];

$contact\_no2= $row['contact\_no1'];

$area = $row['area'];

$road = $row['road'];

$city = $row['city'];

$pincode = $row['pincode'];

$state = $row['state'];

$dis\_from\_bank = $row['dis\_from\_bank'];

1. **To display donor details in the admin page**

DELIMITER $$

CREATE PROCEDURE `donor\_det`()

BEGIN

select donor.f\_name, donor.m\_name, donor.l\_name, donor.gender, donor.DOB, donor.aadhar\_id, donor.contact\_no, address.house\_name, address.society\_name, address.city, address.pincode, address.state

from donor, address

where donor.donor\_id = address.donor\_id;

END$$

DELIMITER ;

$query = "call donor\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$f\_name = $row['f\_name'];

$m\_name = $row['f\_name'];

$l\_name = $row['l\_name'];

$gender = $row['gender'];

$DOB = $row['DOB'];

$aadhar\_id = $row['aadhar\_id'];

$contact\_no = $row['contact\_no'];

$house\_name = $row['house\_name'];

$society\_name= $row['society\_name'];

$city = $row['city'];

$pincode = $row['pincode'];

$state = $row['state'];

1. **To display donor’s medical details in the admin page**

DELIMITER $$

CREATE PROCEDURE `donor\_med\_det`()

BEGIN

select donor.f\_name, donor.m\_name, donor.l\_name, donor.gender, organ.organ\_name, organ.part\_side, medical\_detail.blood\_group, medical\_detail.height, medical\_detail.weight

from donor, organ, medical\_detail

where donor.donor\_id = organ.donor\_id and organ.organ\_id=medical\_detail.organ\_id;

END$$

DELIMITER ;

$query = "call donor\_med\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$f\_name = $row['f\_name'];

$m\_name = $row['m\_name'];

$l\_name = $row['l\_name'];

$gender = $row['gender'];

$organ\_name = $row['organ\_name'];

$part\_side = $row['part\_side'];

$blood\_group= $row['blood\_group'];

$height = $row['height'];

$weight= $row['weight'];

1. **To display recipient details in the admin page**

DELIMITER $$

CREATE PROCEDURE `admin\_rec\_det`()

BEGIN

select recepient.f\_name, recepient.m\_name, recepient.l\_name, recepient.gender, recepient.DOB, recepient.aadhar\_id, recepient.contact\_no, coordinators.ins\_name, recepient.registration\_date, address.house\_name, address.society\_name, address.city, address.pincode, address.state

from recepient, address, coordinators

where recepient.rec\_id = address.rec\_id and recepient.ins\_id =coordinators.ins\_id;

END$$

DELIMITER ;

$query = "call admin\_rec\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$f\_name = $row['f\_name'];

$m\_name = $row['m\_name'];

$l\_name = $row['l\_name'];

$gender = $row['gender'];

$DOB = $row['DOB'];

$aadhar\_id = $row['aadhar\_id'];

$contact\_no = $row['contact\_no'];

$ins\_name = $row['ins\_name'];

$registration\_date = $row['registration\_date'];

$house\_name = $row['house\_name'];

$society\_name= $row['society\_name'];

$city = $row['city'];

$pincode = $row['pincode'];

$state = $row['state'];

1. **To display recipient’s medical details in the admin page**

DELIMITER $$

CREATE PROCEDURE `rec\_med\_det`()

BEGIN

select recepient.f\_name, recepient.m\_name, recepient.l\_name, recepient.gender, rec\_details.needed\_organ, rec\_details.part\_side, rec\_details.blood\_group, rec\_details.height, rec\_details.weight

from recepient, rec\_details

where recepient.rec\_id = rec\_details.rec\_id;

END$$

DELIMITER ;

$query = "call rec\_med\_det()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$f\_name = $row['f\_name'];$m\_name = $row['m\_name'];

$l\_name = $row['l\_name'];$gender = $row['gender'];

$needed\_organ = $row['needed\_organ'];$part\_side = $row['part\_side'];

$blood\_group= $row['blood\_group'];$height = $row['height'];

$weight= $row['weight'];

1. **To display organ and its quantity**

DELIMITER $$

CREATE PROCEDURE `total\_organ`()

BEGIN

select organ\_name,count(organ\_name) from organ group by organ\_name;

END$$

DELIMITER ;

$query = "CALL total\_organ()";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)){

$organ\_name = $row["organ\_name"];

$total = $row['count(organ\_name)'];

1. **The Matching process of the organs of recipient and donor**

DELIMITER $$

CREATE PROCEDURE `matching`(IN `organName` VARCHAR(20), IN `bloodgroup` VARCHAR(10), IN `part` VARCHAR(20))

BEGIN

select distinct organ.organ\_id,medical\_detail.height,medical\_detail.weight from organ,medical\_detail where organ.organ\_name=organName and organ.part\_side=part

and medical\_detail.blood\_group=bloodgroup and organ.organ\_id=medical\_detail.organ\_id and o\_flag=0;

END$$

DELIMITER

$query = "call matching ('$\_POST[o\_name]','$\_POST[blood\_group]','$\_POST[part]')";

$query\_run = mysqli\_query($connection, $query);

while ($row = mysqli\_fetch\_assoc($query\_run)) {

$o\_id = $row['organ\_id'];

$height = $row['height'];

$weight = $row['weight'];

**1.3.2 Executing function**

1. **Number of Organs in bank**

DELIMITER $$

CREATE FUNCTION `count\_organ`( ) RETURNS int(11)

BEGIN

DECLARE r\_organ int;

DEClARE c\_organ

CURSOR FOR

select count(organ\_id) as count from organ;

open c\_organ;

FETCH c\_organ INTO r\_organ;

Return r\_organ;

END$$

DELIMITER ;

$query6="call count\_organ()";

$result6= mysqli\_query($connection,$query6) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result6)){

$r\_organ=$row["r\_organ"];}

1. **Total number of the recipient in the institute**

DELIMITER $$

CREATE FUNCTION `count\_recepiant`(institution\_id varchar(8) ) RETURNS int(11)

DETERMINISTIC

BEGIN

DECLARE r\_count int;

DEClARE c\_count

CURSOR FOR

select count(rec\_id) as count from recepient where ins\_id= institution\_id;

open c\_count;

FETCH c\_count INTO r\_count;

RETURN r\_count;

END$$

DELIMITER ;

$query="call count\_recepiant\_hos('$\_SESSION[user]')";

$result = mysqli\_query($connection, $query) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result)){

$r\_count=$row["r\_count"];}

1. **Total number of the donors in bank**

DELIMITER $$

CREATE FUNCTION count\_donor()

RETURNS INTEGER

DETERMINISTIC

BEGIN

DECLARE r\_donor int;

DEClARE c\_donor

CURSOR FOR

select count(donor\_id) as count from donor;

open c\_donor;

FETCH c\_donor INTO r\_donor;

RETURN r\_donor;

END$$

DELIMITER ;

$query="call count\_donor()";

$result = mysqli\_query($connection, $query) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result)){

$r\_count=$row["r\_donor"];}

1. **Total number of the recipients in bank**

DELIMITER $$

CREATE FUNCTION count\_rec()

RETURNS INTEGER

DETERMINISTIC

BEGIN

DECLARE r\_rec int;

DEClARE c\_rec

CURSOR FOR

select count(rec\_id) as count from recepient;

open c\_rec;

FETCH c\_rec INTO r\_rec;

RETURN r\_rec;

END$$

DELIMITER ;

$query1="call count\_rec()";

$result2 = mysqli\_query($connection, $query1) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result2)){

$r\_rec=$row["r\_rec"];}

1. **Total number of the coordinators of bank**

DELIMITER $$

CREATE FUNCTION count\_coor()

RETURNS INTEGER

DETERMINISTIC

BEGIN

DECLARE r\_coor int;

DEClARE c\_coor

CURSOR FOR

select count(ins\_id) as count from coordinators;

open c\_coor;

FETCH c\_coor INTO r\_coor;

RETURN r\_coor;

END$$

DELIMITER ;

select count\_coor();

$query3="call count\_coor()";

$result3= mysqli\_query($connection, $query3) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result3)){

$r\_coor=$row["r\_coor"];}

1. **Total number of the camps handled by bank**

DELIMITER $$

CREATE FUNCTION count\_camp()

RETURNS INTEGER

DETERMINISTIC

BEGIN

DECLARE r\_camp int;

DEClARE c\_camp

CURSOR FOR

select count(camp\_id) as count from camp;

open c\_camp;

FETCH c\_camp INTO r\_camp;

RETURN r\_camp;

END$$

DELIMITER ;

$query4="call count\_camp()";

$result4= mysqli\_query($connection, $query4) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result4)){

$r\_camp=$row["r\_camp"];}

1. **Total number of transplants of organs by bank**

DELIMITER $$

CREATE FUNCTION count\_transplant()

RETURNS INTEGER

DETERMINISTIC

BEGIN

DECLARE r\_transplant int;

DEClARE c\_transplant

CURSOR FOR

select count(organ\_id) as count from transplant;

open c\_transplant;

FETCH c\_transplant INTO r\_transplant;

RETURN r\_transplant;

END$$

DELIMITER ;

$query5="call count\_transplant()";

$result5= mysqli\_query($connection, $query5) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result5)){

$r\_transplant=$row["r\_transplant"];}

1. **Total number of recipient in waiting list**

DELIMITER $$

CREATE FUNCTION `count\_wait`()

BEGIN

DECLARE r\_wait int;

DEClARE c\_wait

CURSOR FOR

select count(rec\_id) as count from waiting\_list;

open c\_wait;

FETCH c\_wait INTO r\_wait;

Return r\_wait;

end$$

DELIMITER ;

$query7="call count\_wait()";

$result7= mysqli\_query($connection, $query7) or die(mysqli\_error($link));

while ($row = mysqli\_fetch\_assoc($result7)){

$r\_wait=$row["r\_wait"]9;}

**1.3.3 Triggers:**

1. **Donor having age less than 18 is not allowed to donate**

DELIMITER //

create trigger age before insert on donor

for each row

begin

declare msg varchar(200);

declare d1 date;

declare d2 date;

set d1:=new.DOB ;

set d2:= curdate();

if ((timestampdiff(month, d1 , d2)) < 216) then

SET msg = 'Person having age less than 18 can not donate.....';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

END IF;

END//

1. **Update the total number of donors in the `camp` table if a donor is added or deleted**

DELIMITER $$

create trigger inc\_donors\_camp after insert on donor

for each row

begin

UPDATE camp SET total\_donors = total\_donors + 1 where camp.camp\_id = NEW.camp\_id;

end//

------------------------------------------------------------------------------------------------------------

DELIMITER $$

create trigger dec\_donors\_camp before delete on donor

for each row

begin

UPDATE camp SET total\_donors = total\_donors -1 where camp.camp\_id = Old.camp\_id;

end//

1. **Error occurs when the donor’s first name, middle name, last name, or birthdate is not inserted by the a user**

DELIMITER $$

create trigger don\_det before insert on donor

for each row

begin

declare msg varchar(100) ;

if (new.f\_name is null) then

set msg = 'First name of the person can''t be null...';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

elseif(new.m\_name is null) then

SET msg = 'Middle name of the person can''t be null...';

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = msg;

elseif (new.l\_name is null) then

SET msg ='Last name of the person can''t be null...';

SIGNAL SQLSTATE '45002' SET MESSAGE\_TEXT=msg;

elseif (new.DOB is null) then

SET msg = 'Birthdate of the person can''t be null..';

SIGNAL SQLSTATE '45003' SET MESSAGE\_TEXT=msg;

end if;

end//

1. **Error occurs when organ name is not inserted by user**

DELIMITER $$

CREATE TRIGGER `d\_organ` BEFORE INSERT ON `organ`

FOR EACH ROW begin

declare msg varchar(200);

if (new.organ\_name is null) then

SET msg = 'Please add a organ name of the donor...';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

end if;

End

1. **Error occurs when blood group is not inserted by user**

DELIMITER //

create trigger d\_bloodgroup before insert on medical\_detail

for each row

begin

declare msg varchar(200);

if (new.blood\_group is null) then

SET msg = 'Please add a blood group of the donor...';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

end if;

end//

1. **Error occurs when recipient's first name, middle name, last name or birthdate is not inserted by user**

DELIMITER //

create trigger rec\_det before insert on recepient

for each row

begin

declare msg varchar(100) ;

if (new.f\_name is null) then

set msg = 'First name of the person can''t be null...';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

elseif(new.m\_name is null) then

SET msg = 'Middle name of the person can''t be null...';

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = msg;

elseif (new.l\_name is null) then

SET msg ='Last name of the person can''t be null...';

SIGNAL SQLSTATE '45002' SET MESSAGE\_TEXT=msg;

elseif (new.DOB is null) then

SET msg = 'Birthdate of the person can''t be null..';

SIGNAL SQLSTATE '45003' SET MESSAGE\_TEXT=msg;

end if;

end//

1. **Error occurs when blood group or needed organ is not inserted by user**

DELIMITER $$

create trigger rec\_med before insert on rec\_details

for each row

begin

declare msg varchar(200);

if (new.blood\_group is null) then

SET msg = 'Please add a blood group of the recepient...';

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = msg;

elseif (new.needed\_organ is null) then

SET msg = 'Please add a needed organ of recepient...';

SIGNAL SQLSTATE '45001' SET MESSAGE\_TEXT = msg;

end if;

end//

1. **Inserting recipient detail in waiting list when it is inserted into `recipient` table**

DELIMITER $$

CREATE TRIGGER `waiting\_list` AFTER INSERT ON `recepient`

FOR EACH ROW BEGIN

insert into waiting\_list values (recepient.rec\_id,TIMESTAMPDIFF(month,recepient.registration\_date,CURDATE()));

END//

1. **Deleting recipient detail in waiting list when it is inserted into `transplant` table**

DELIMITER $$

CREATE TRIGGER `waiting` AFTER INSERT ON `transplant`

FOR EACH ROW begin

delete from waiting\_list where rec\_id=new.rec\_id;

END//

1. **Deleting recipient detail in a waiting list when it is deleted from `recepient` table**

DELIMITER $$

CREATE TRIGGER `remove\_waiting` AFTER DELETE ON `recepient`

FOR EACH ROW begin

delete from waiting\_list where rec\_id=old.rec\_id;

END//

1. **Value of flag will become 1 in `recepient` and `organ` table after transplantation**

DELIMITER $$

CREATE TRIGGER `transplantation` AFTER INSERT ON `transplant`

FOR EACH ROW begin

update recepient set r\_flag=1 where rec\_id= NEW.rec\_id;

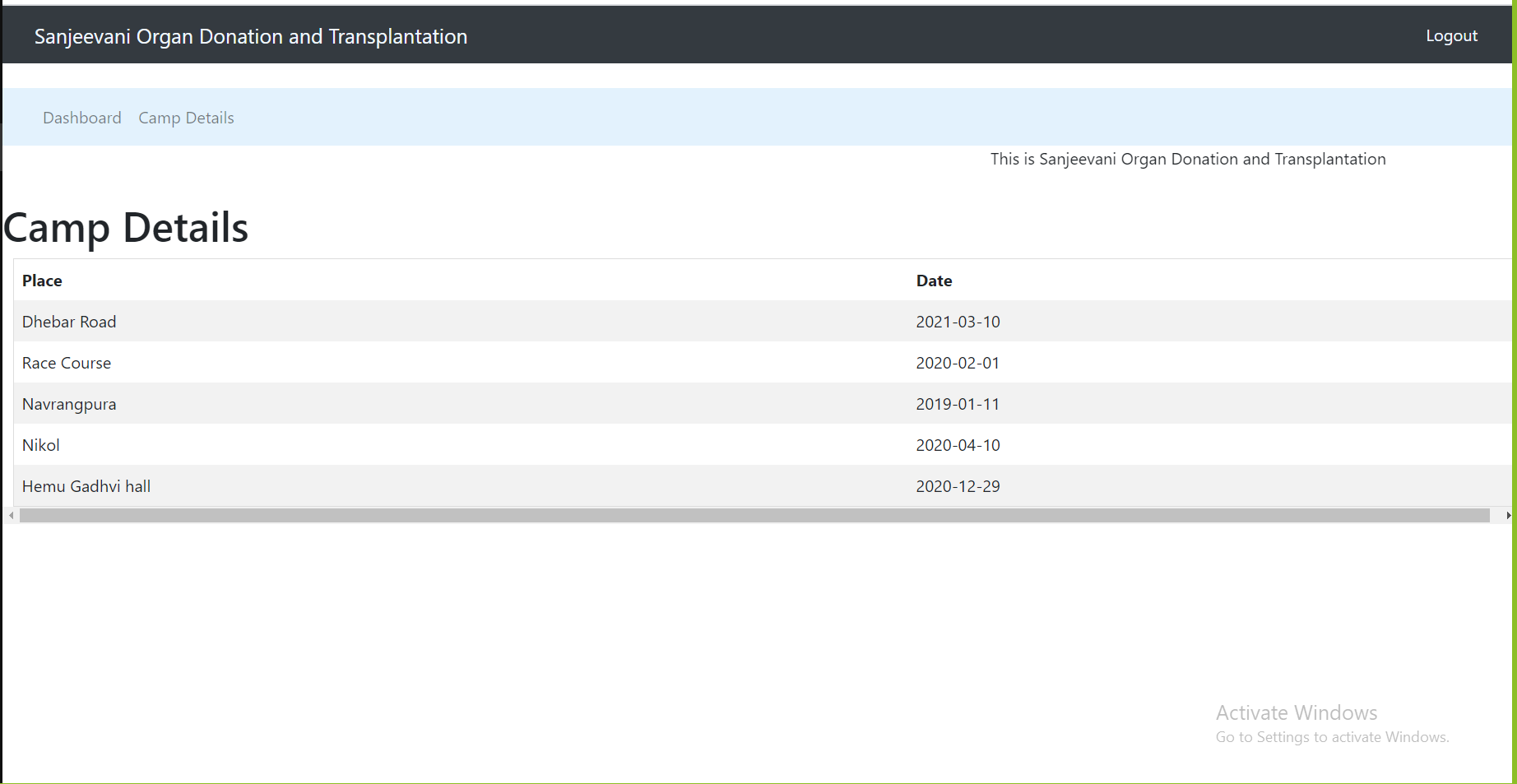
update organ set o\_flag=1 where organ\_id= NEW.organ\_id;

END//

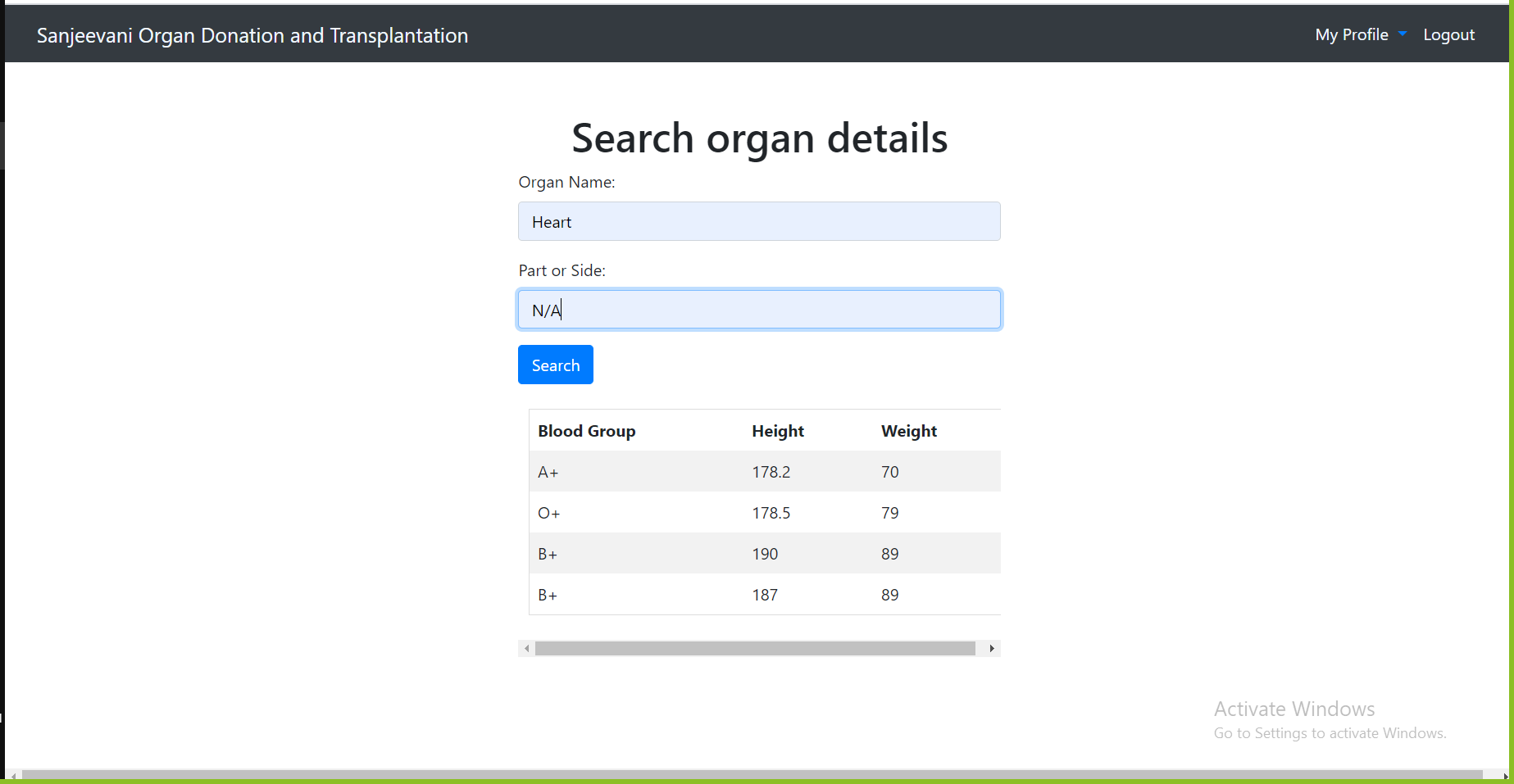
**1.4 Screenshots**

**Procedure :**

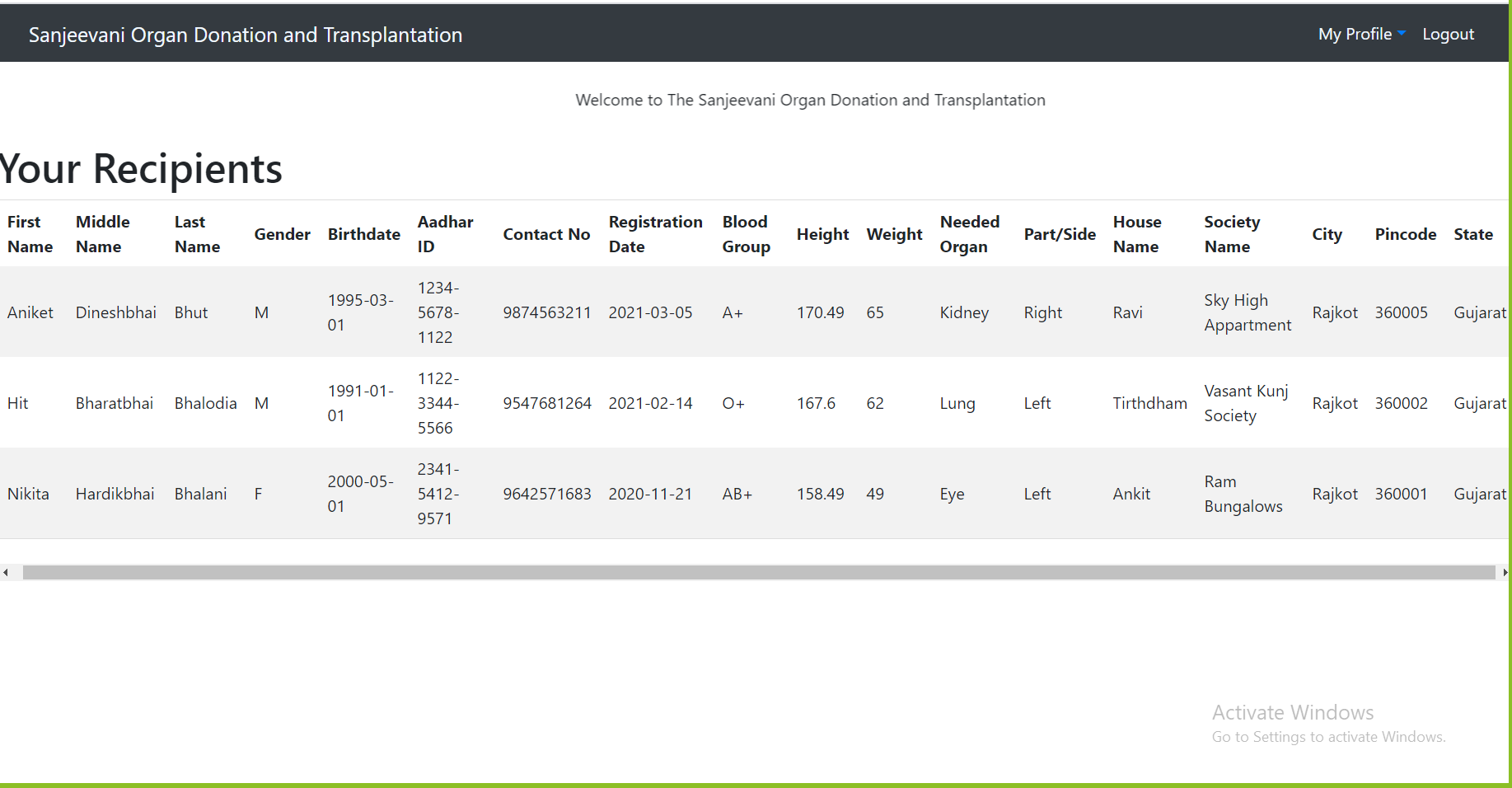
1. **Camp Details for Hospital-**`camp\_det`()

****

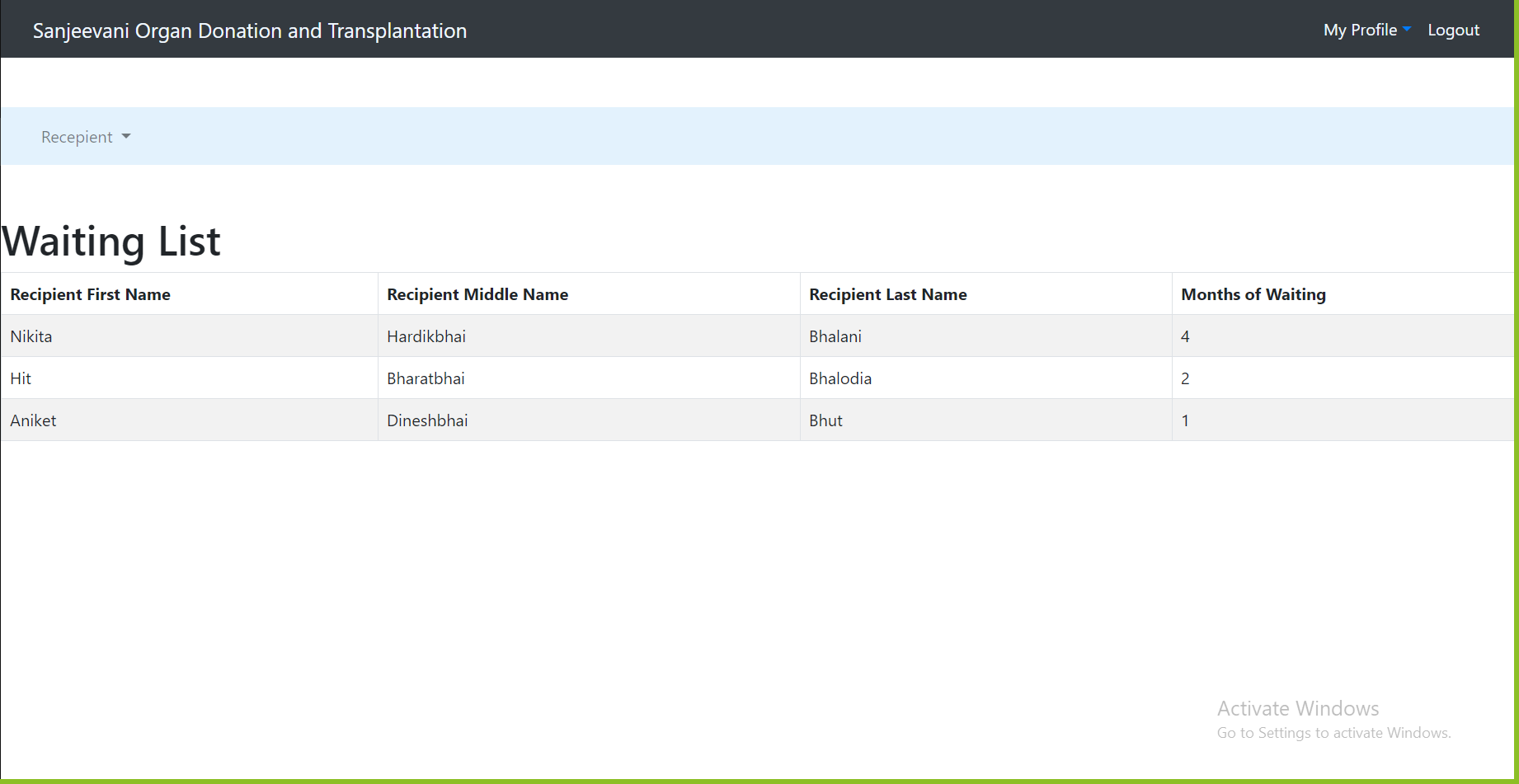
1. **Available organ details for Hospital** -`Organ\_data`(IN `organ` VARCHAR(20), IN `part` VARCHAR(30))

****

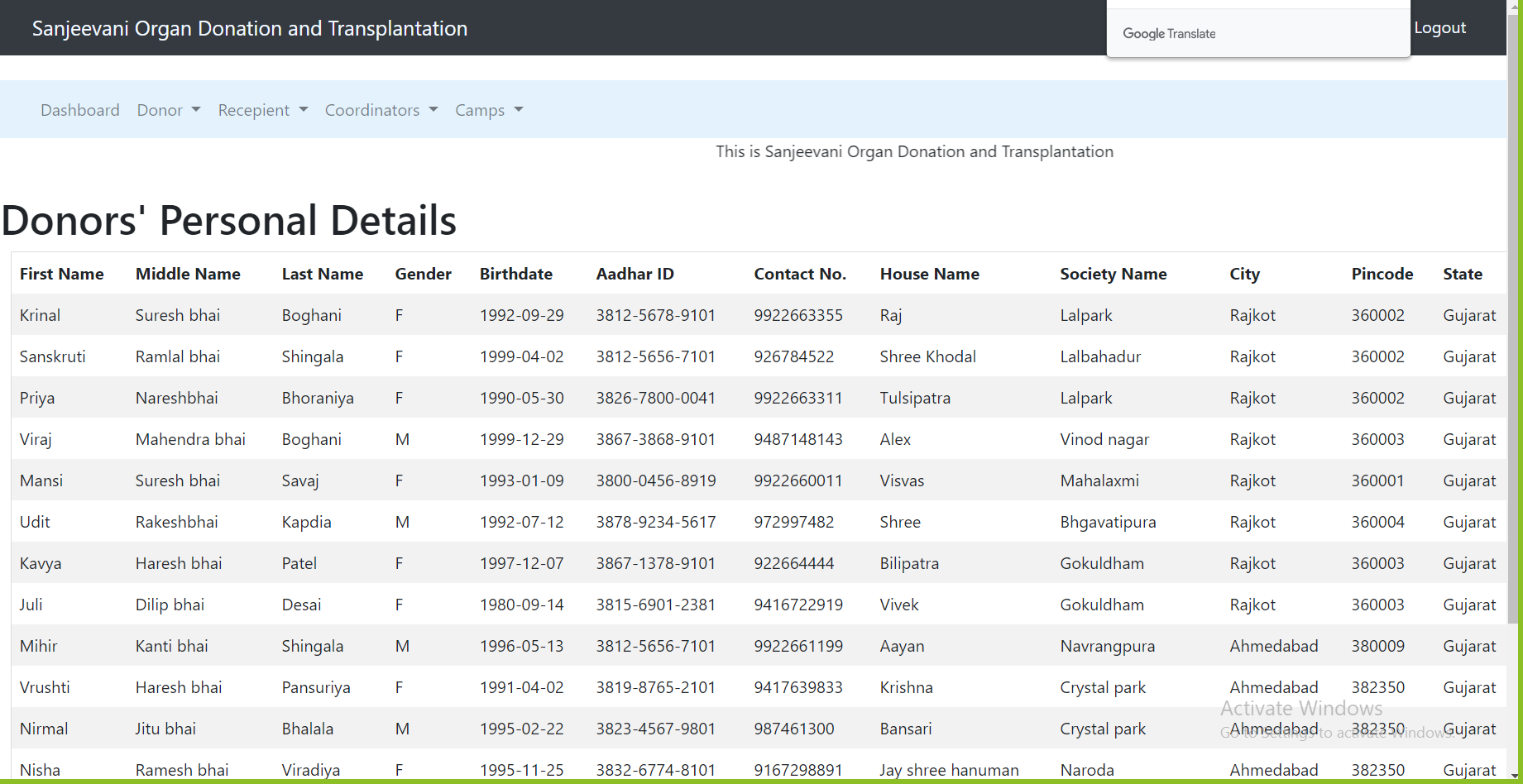
1. **Recipient Details of respective Hospital -**`rec\_detailOf\_Hospital`(IN `ins\_id1` VARCHAR(8))

****

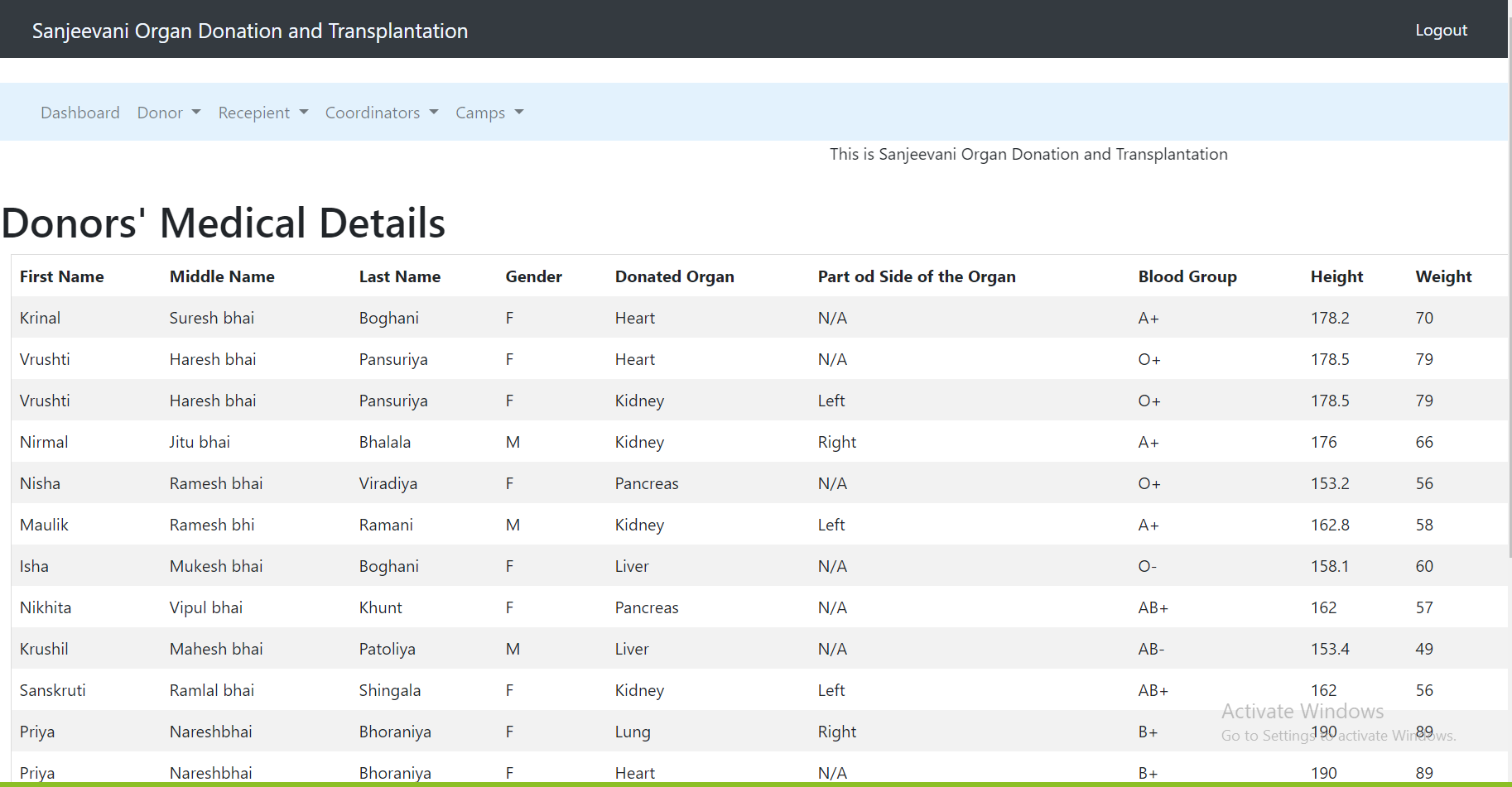
1. **Waiting List for Respective Hospital**

****

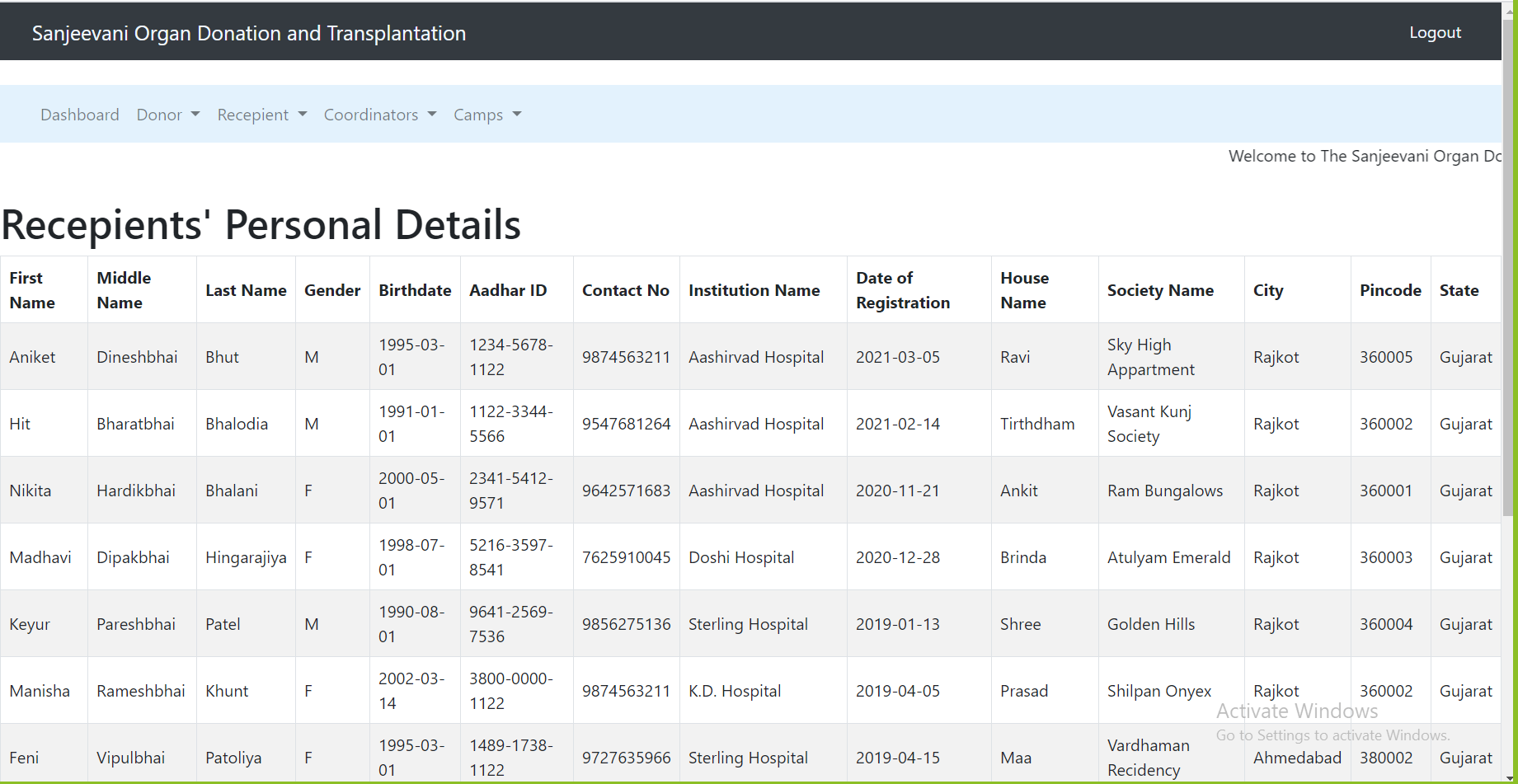
1. **View of Donor’s personal details for Admin -**`donor\_det`()

****

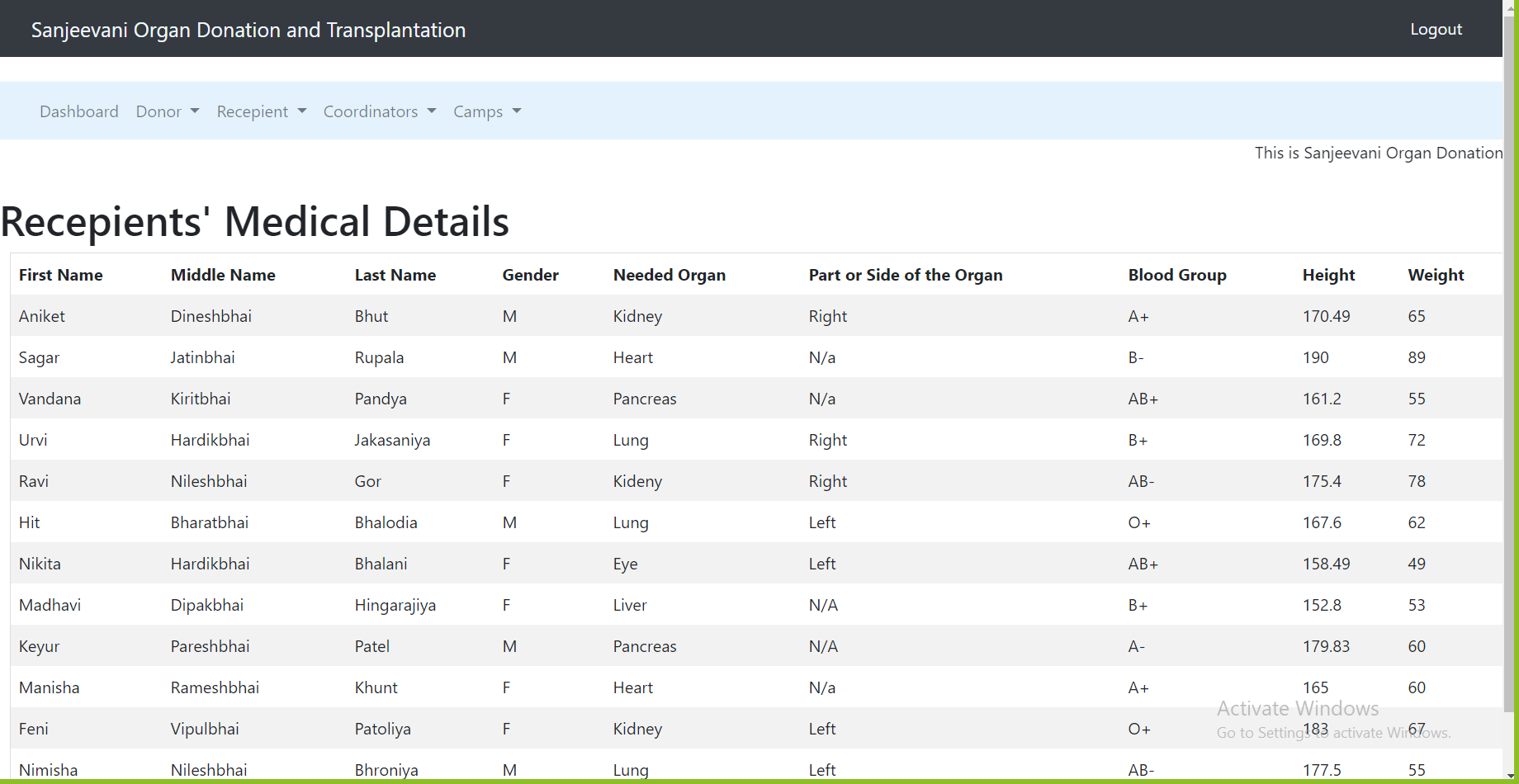
1. **View of Donor’s medical details for Admin -**`donor\_med\_det`()

****

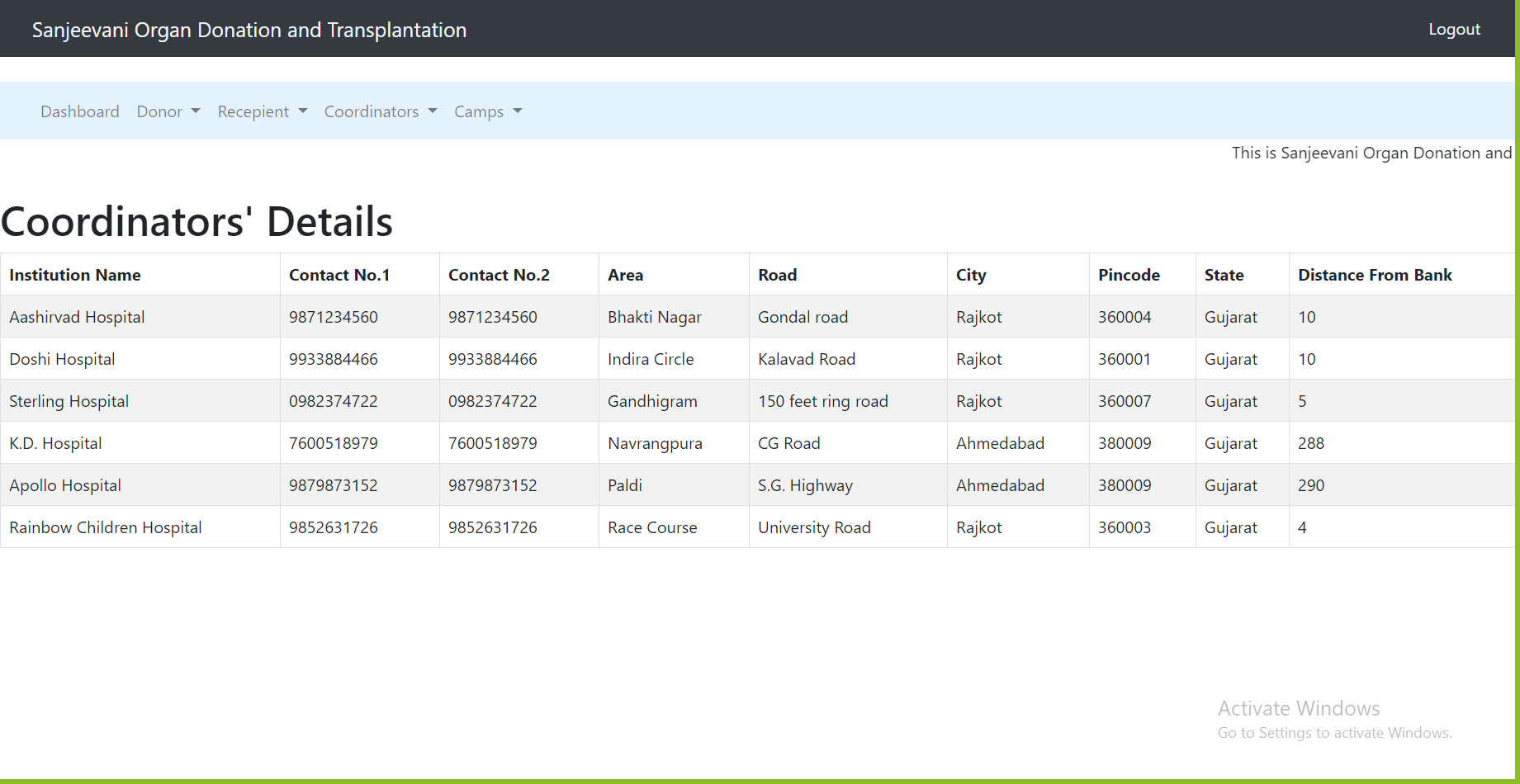
1. **View of Recipients' personal details for Admin -**`admin\_rec\_det`()

****

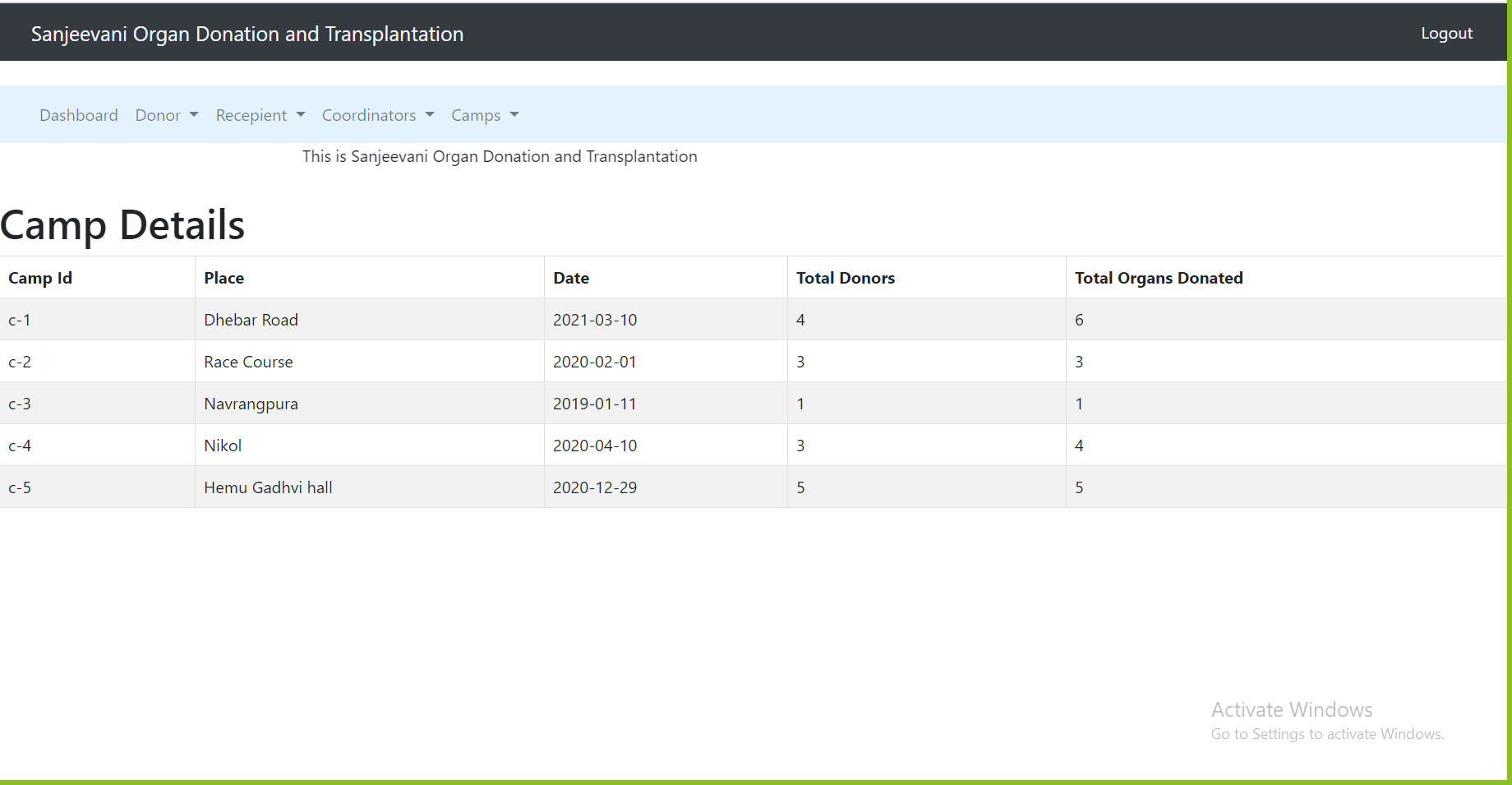
1. **View of Recipients' medical details for Admin -**`rec\_med\_det`()

****

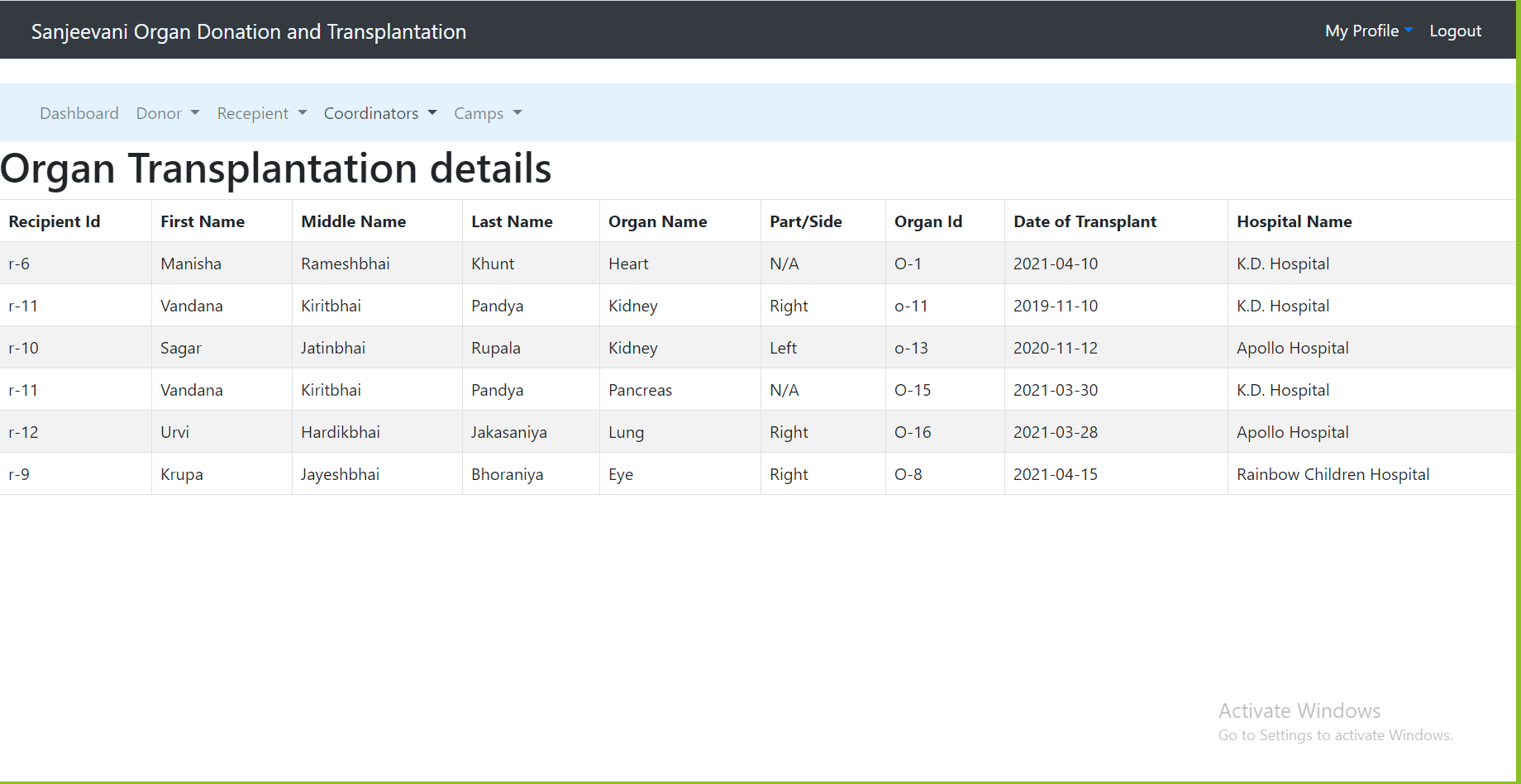
1. **View of coordinators’ details for Admin -**`coordinators\_det`()

****

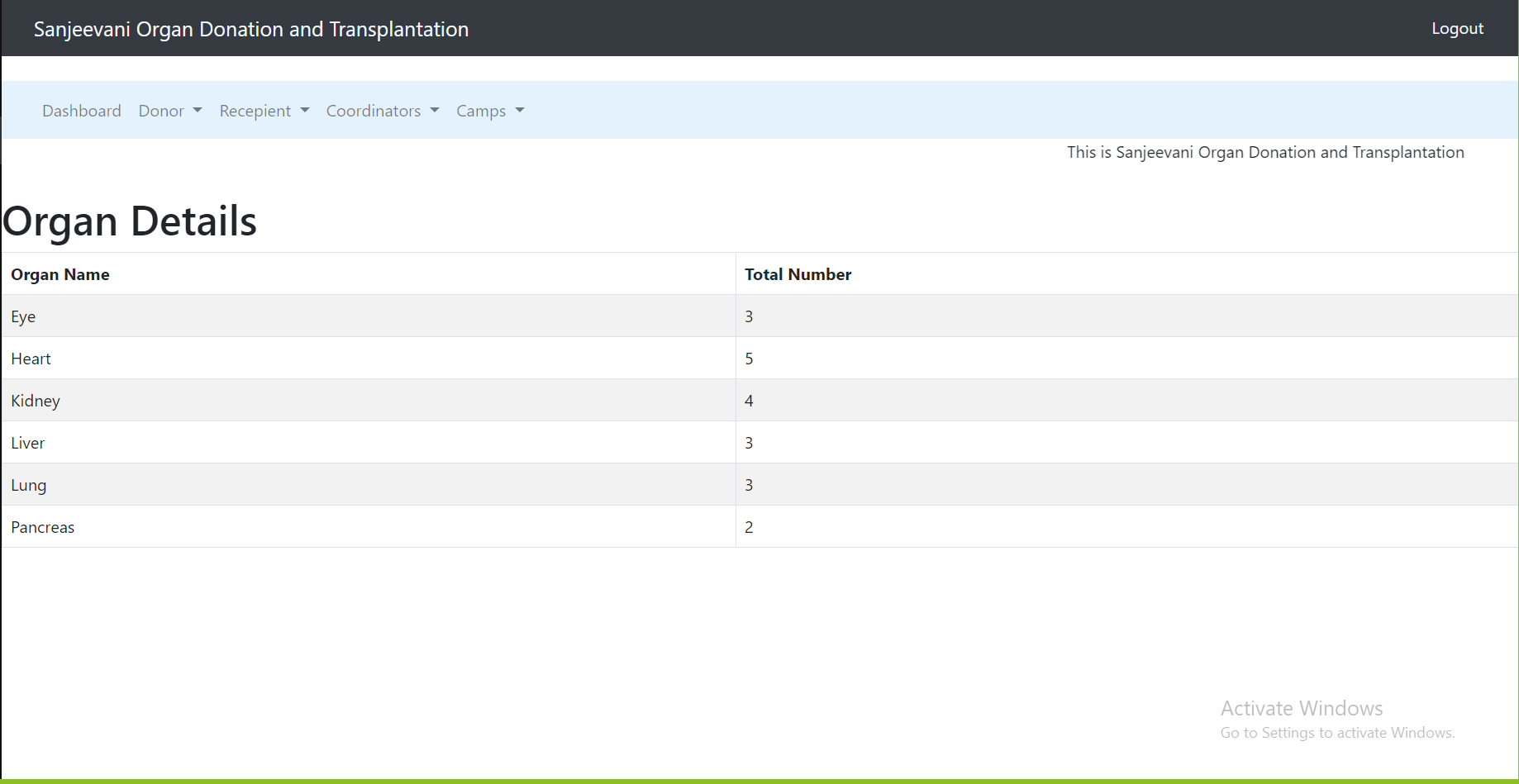
1. **View of Camps’ details for admin -**`camp\_det`()

****

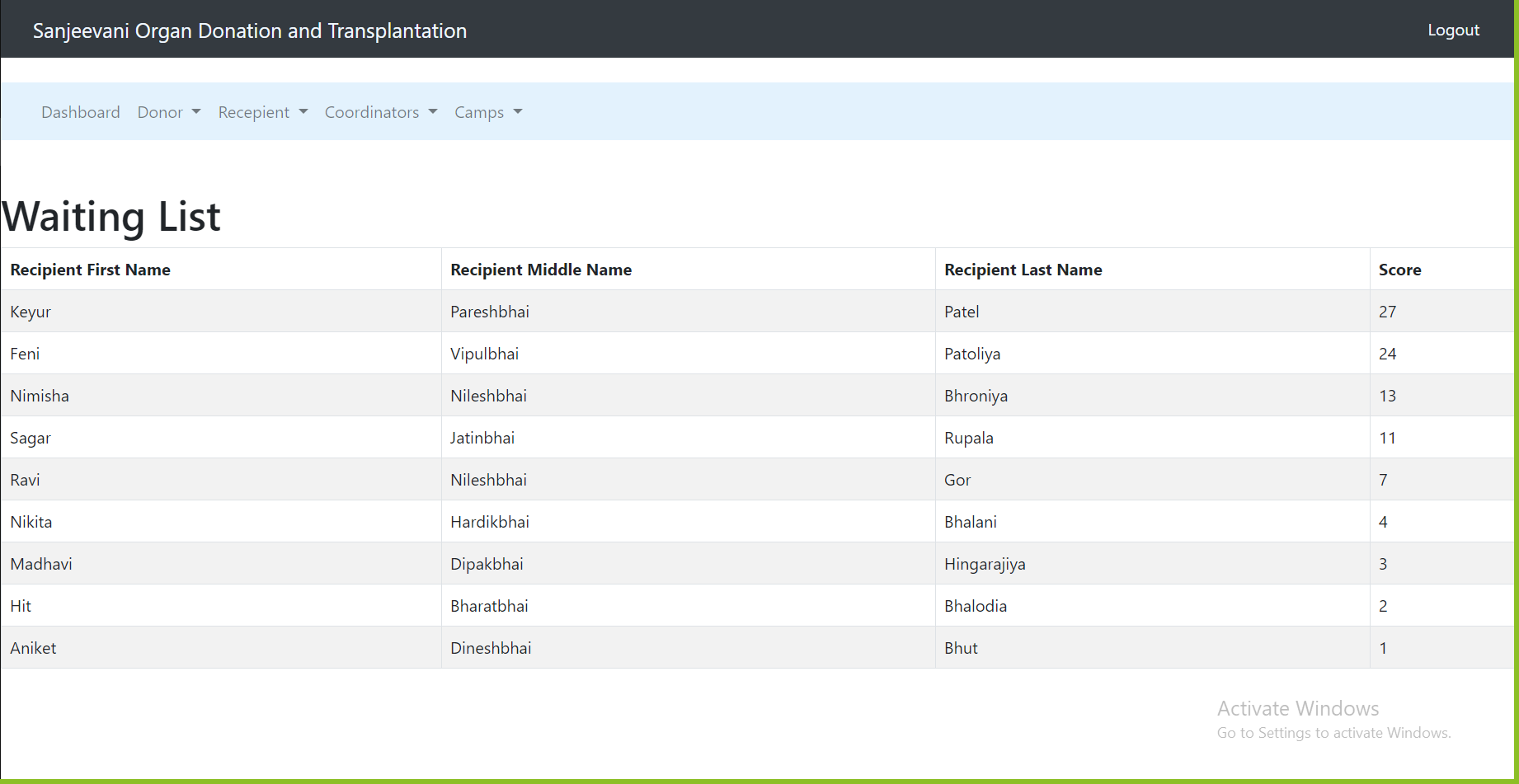
1. **View of total transplantation by the bank for admin**

****

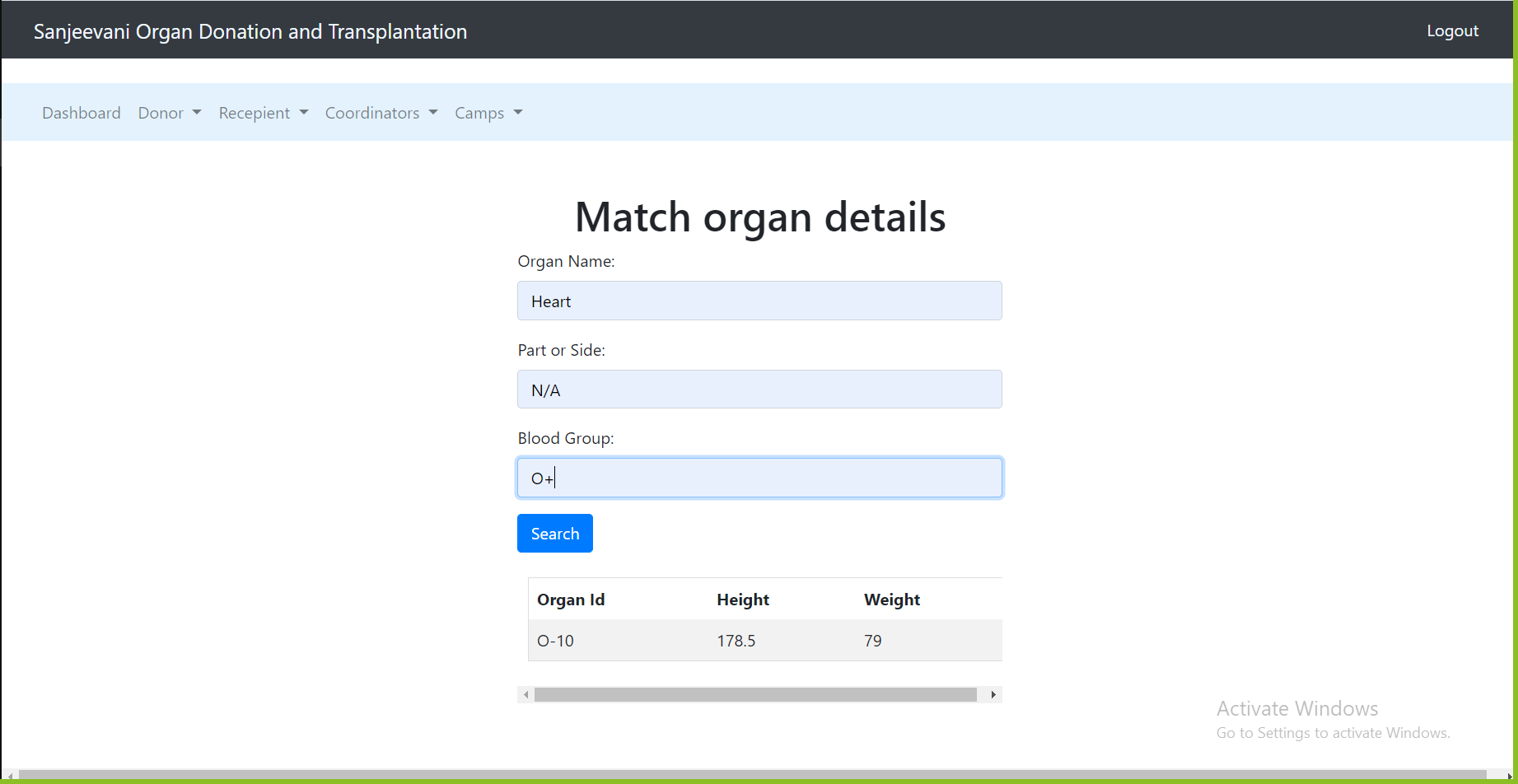
1. **View of the total quantity of respective organ -**`total\_organ`()

****

1. **View of the waiting list of the recipient for admin**

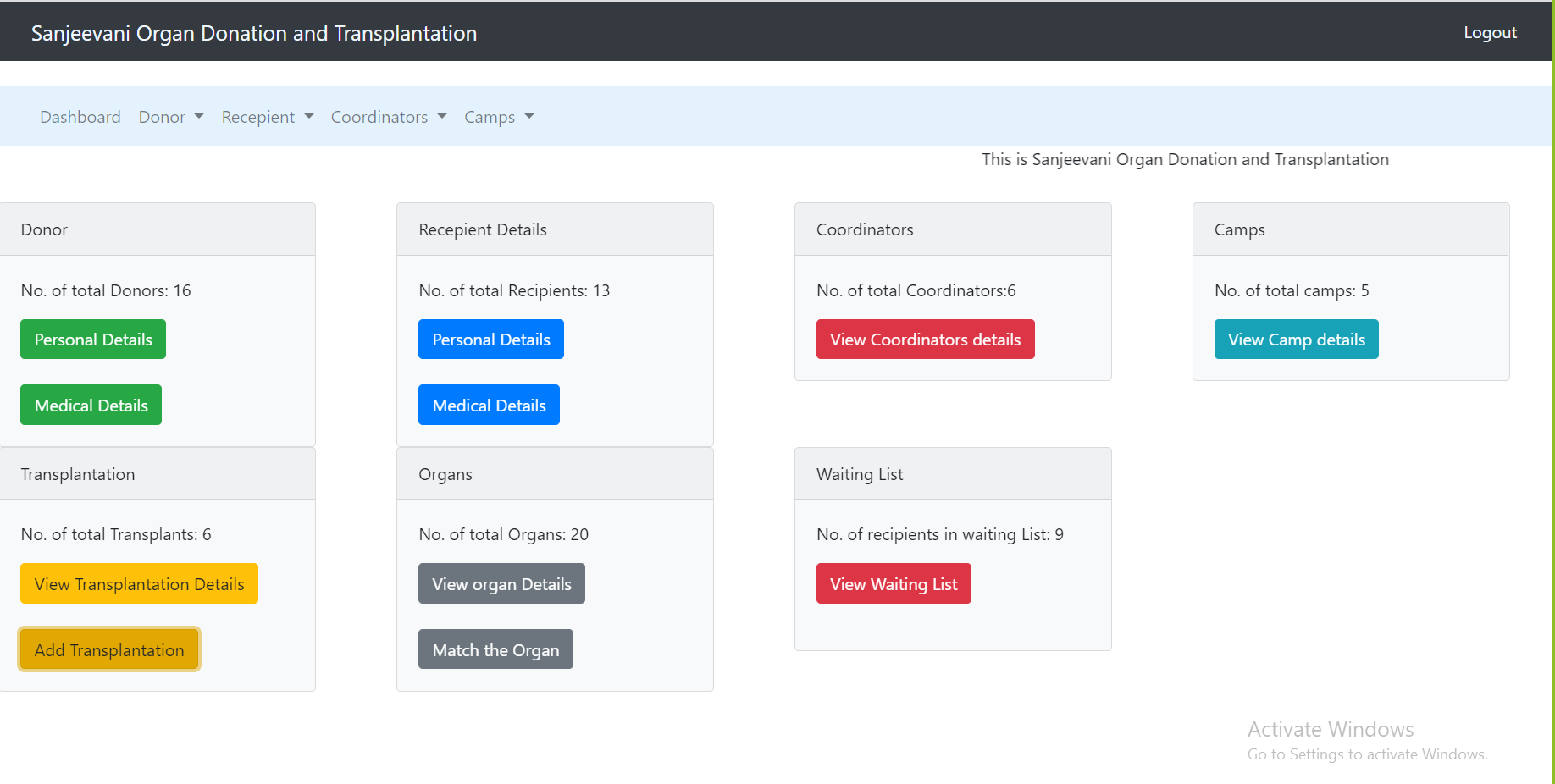
****

1. **Matching Process of Organ -**`matching`(IN `organName` VARCHAR(20), IN `bloodgroup` VARCHAR(10), IN `part` VARCHAR(20))

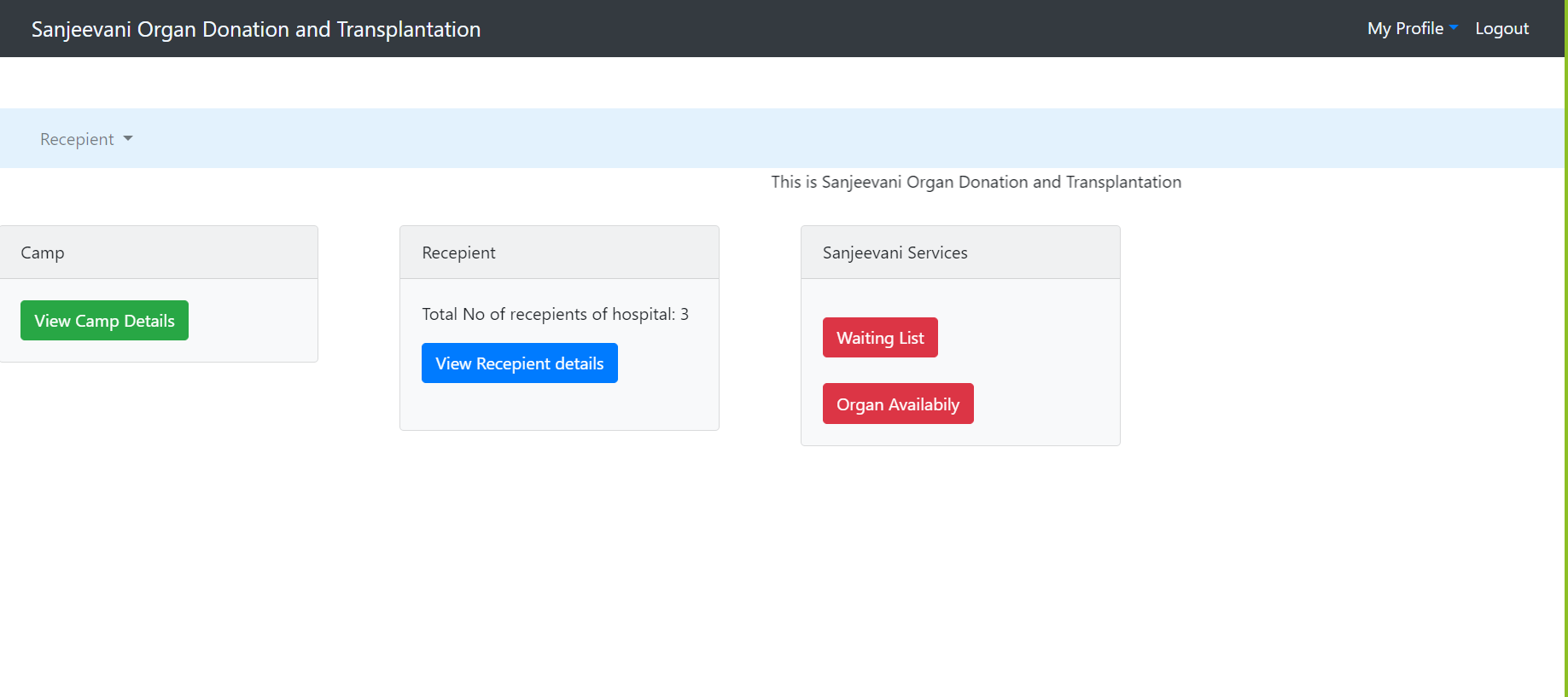
****

**Functions:**

1. **count\_donor(), count\_rec(), count\_coor(), count\_camp(), count\_transplant(), count\_organ()**

****

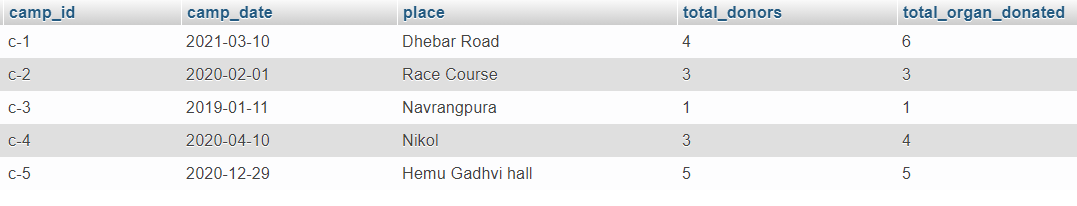
1. **`count\_recepiant`(institution\_id varchar(8) )**

****

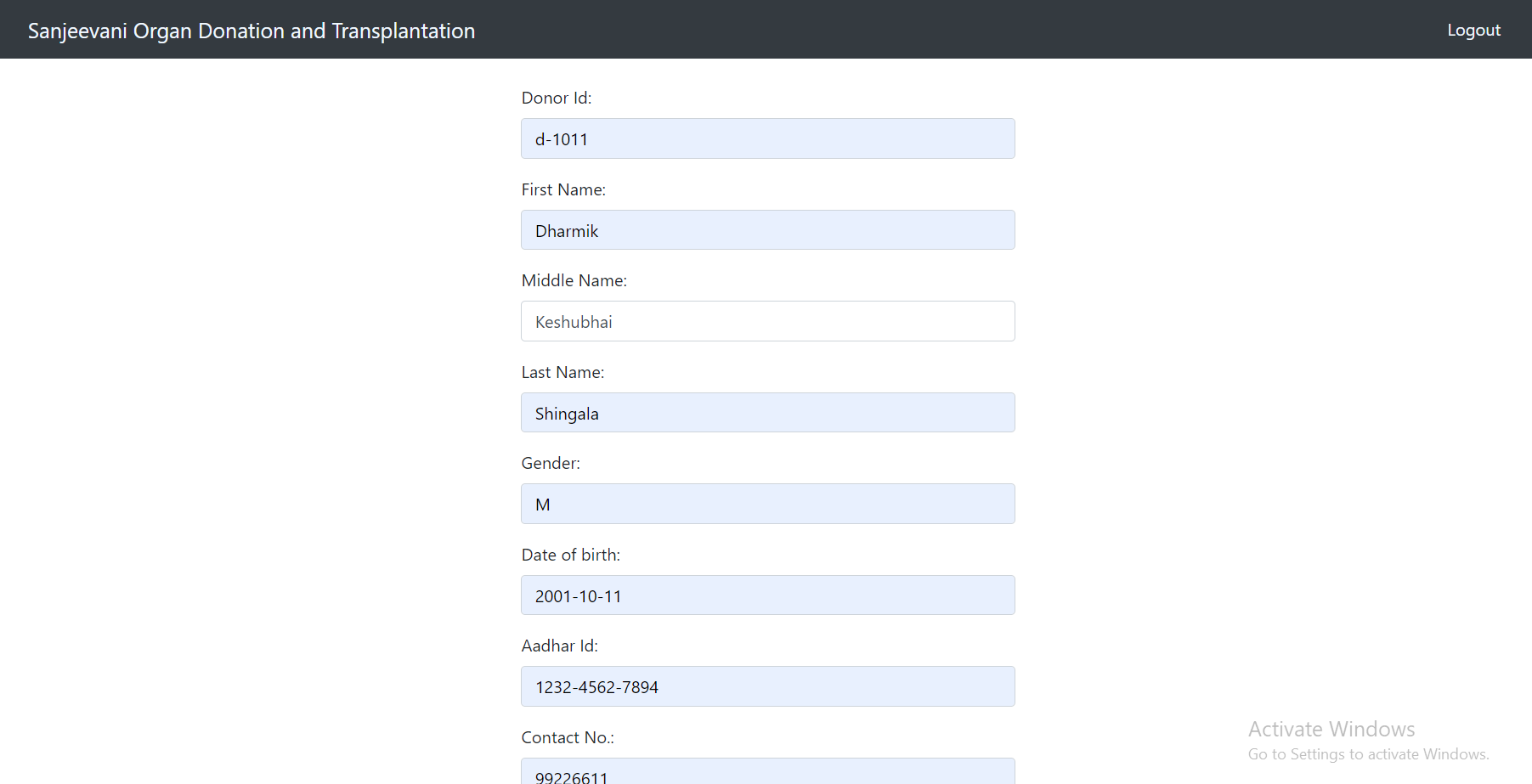
**Triggers:**

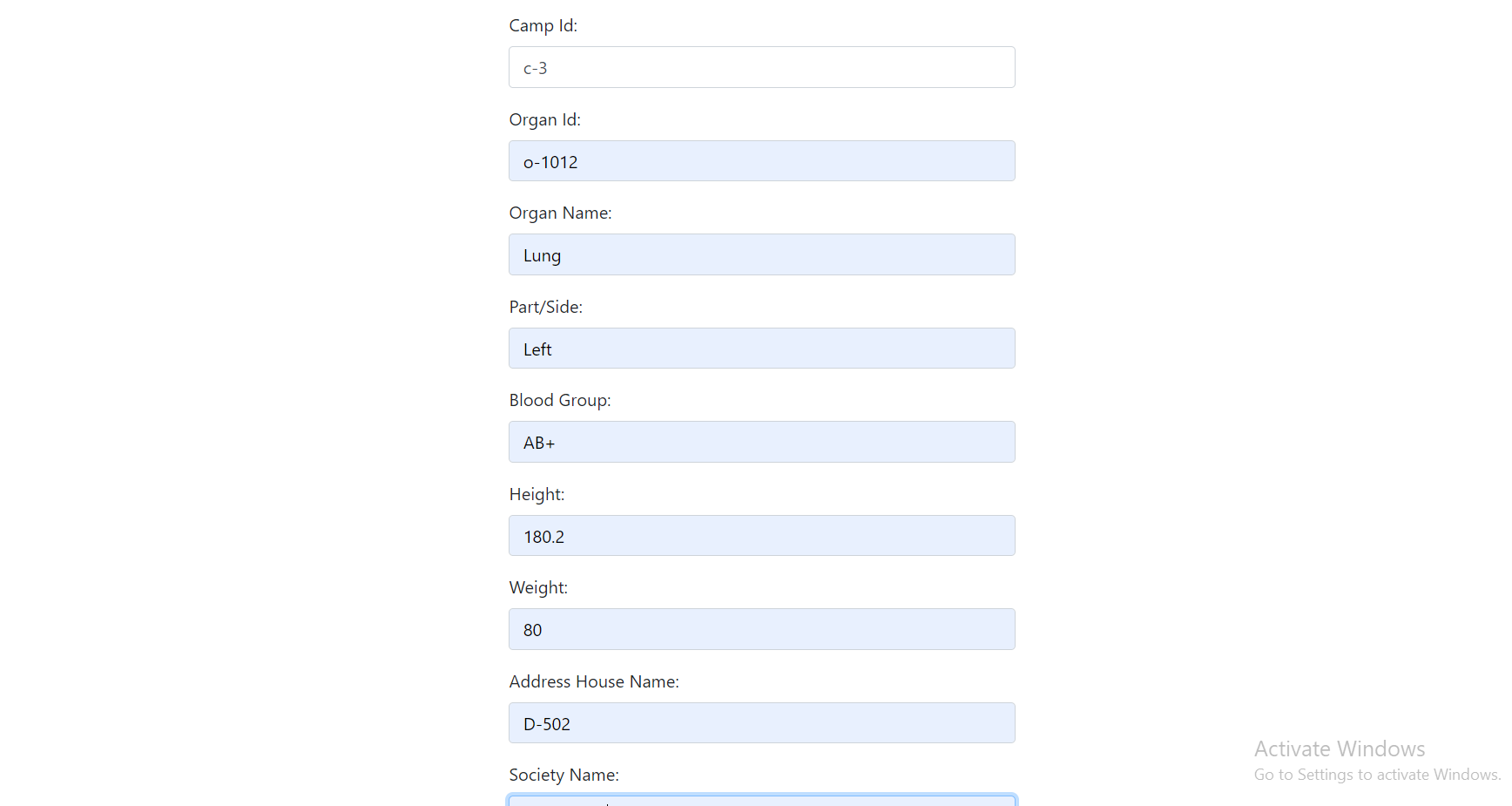
1. **Update the total number of donors in the `camp` table if a donor is added or deleted**

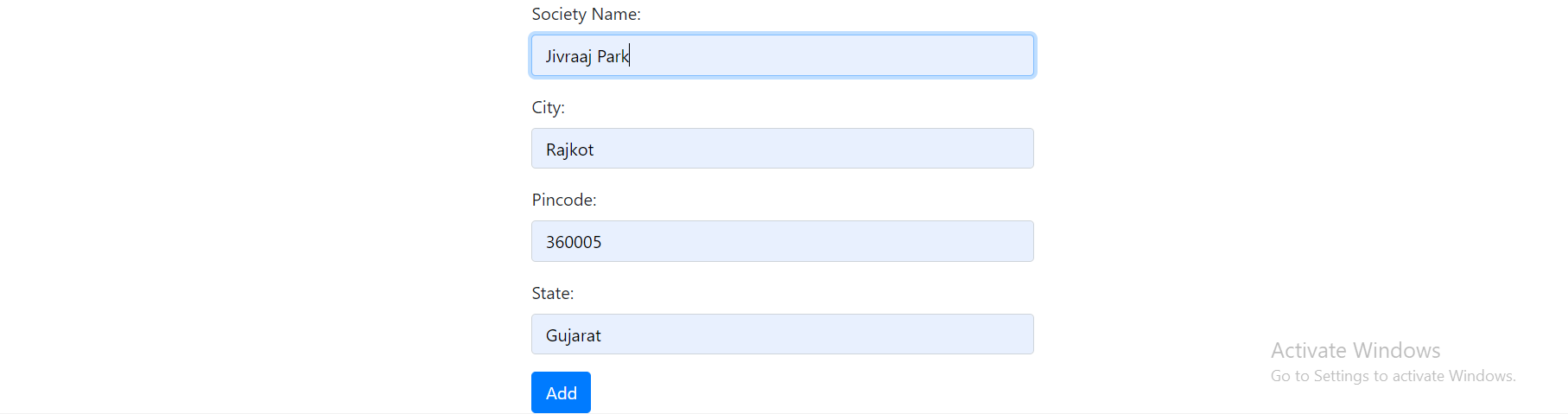
Initial Readings

****

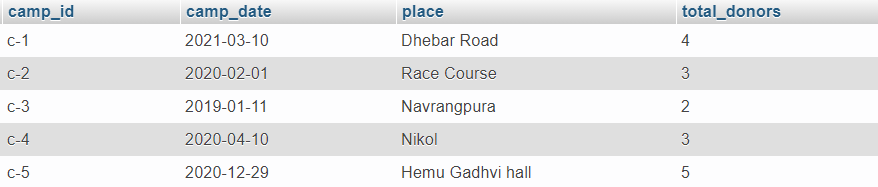
Adding a donor in camp c-3





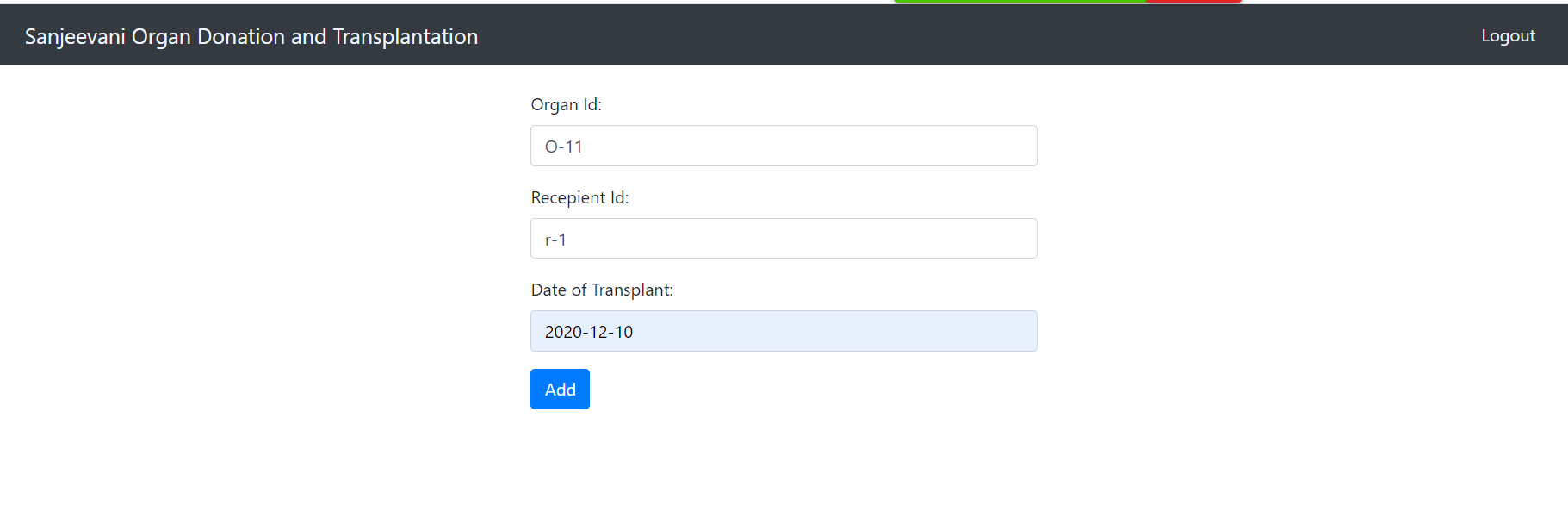


After Trigger Fired:

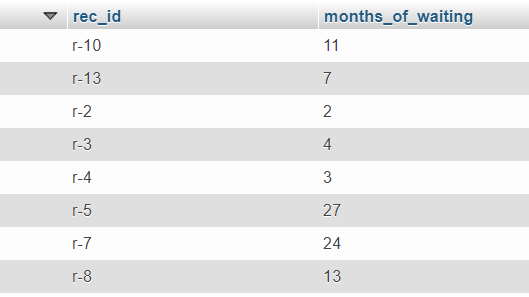


1. **Deleting recipient detail in a waiting list when it is inserted into `transplant` table**

Adding data of transplant

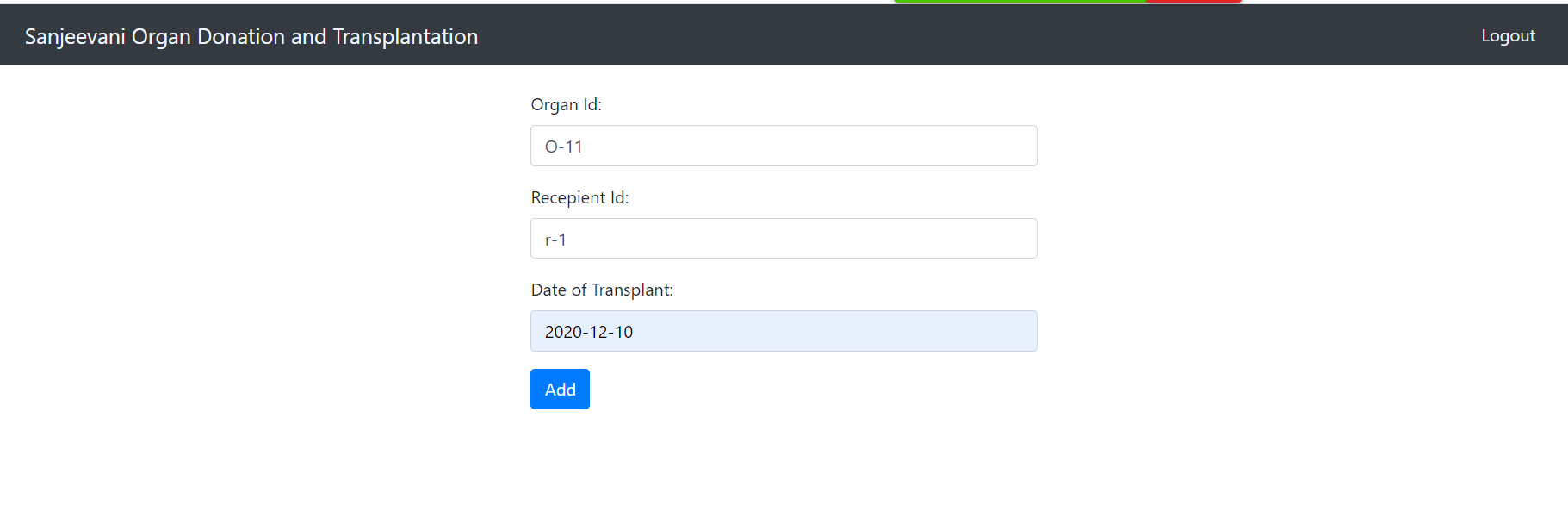
****

Respective Data is deleted on the waiting list table

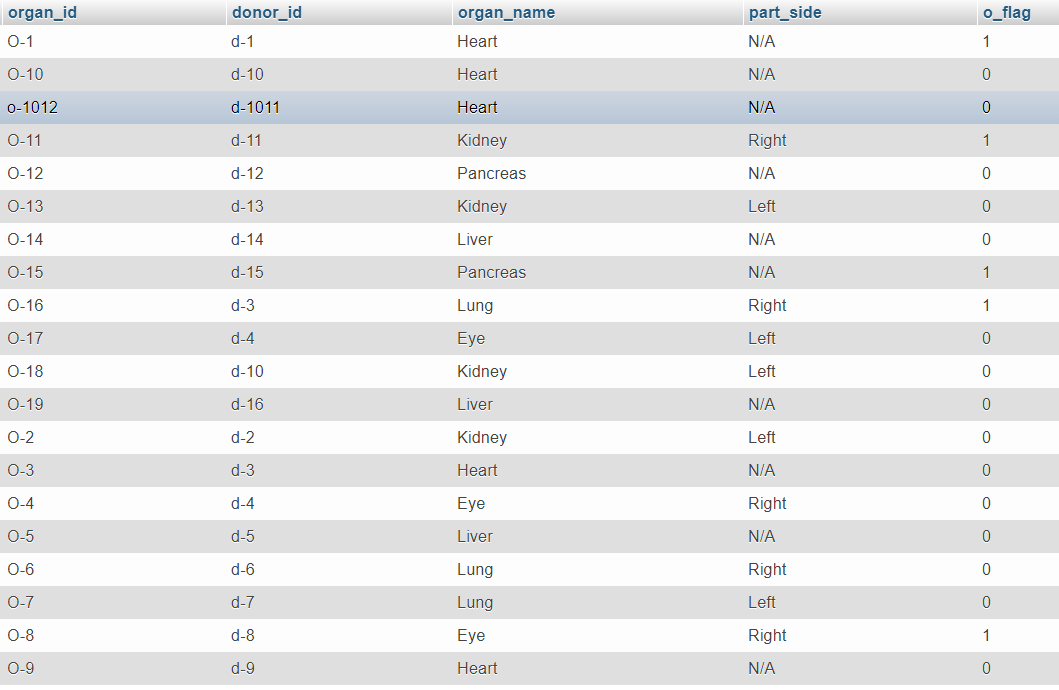
****

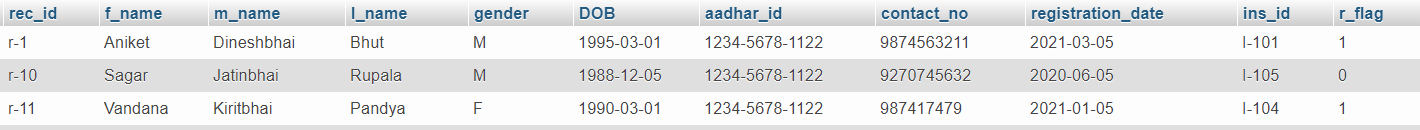
1. **Value of flag will become 1 in `recepient` and `organ` table after transplantation**

After adding data on transplant

****

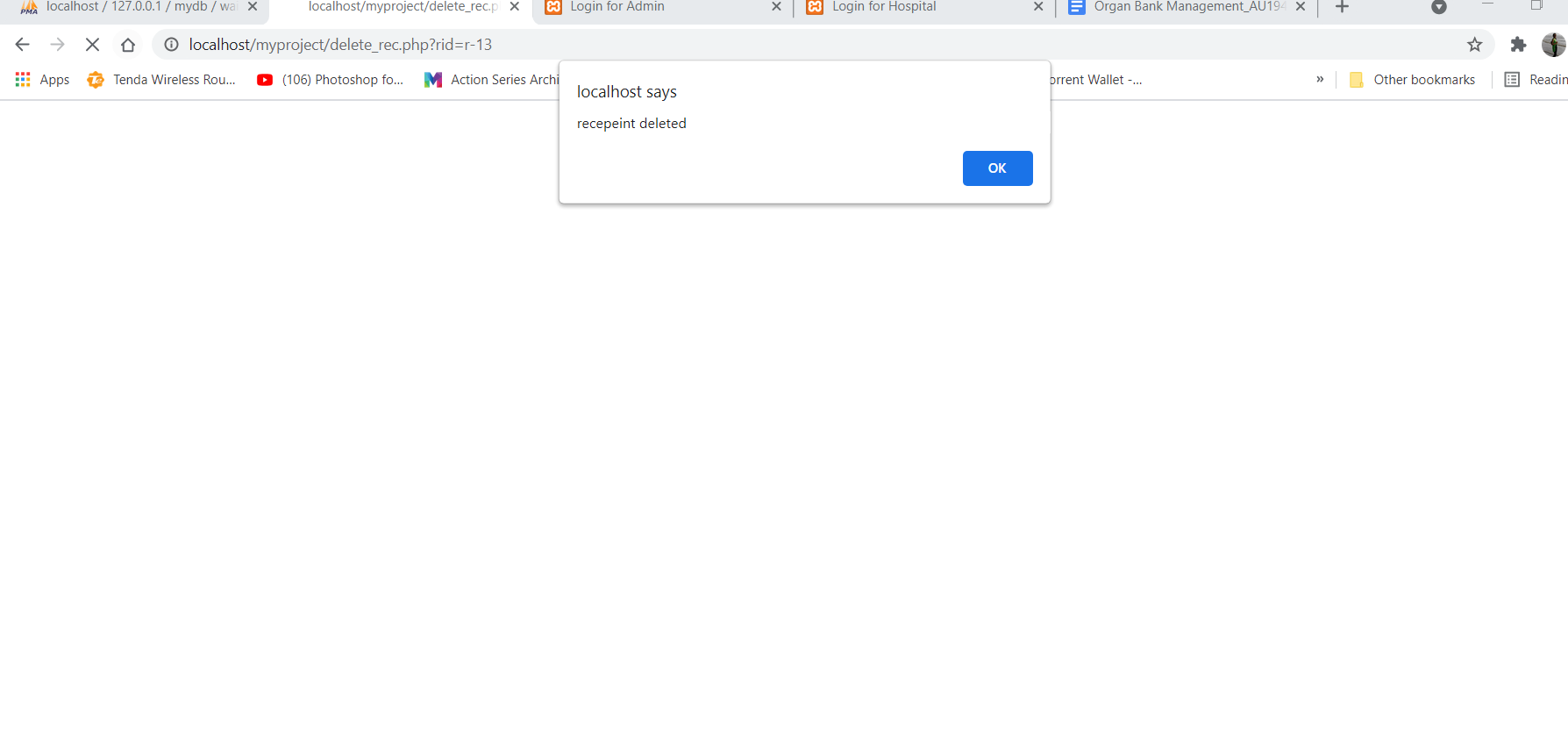
Flag value of respective organ and recipient on their table becomes 1

****

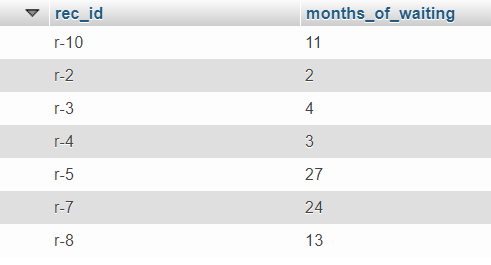
****

1. **Deleting recipient detail in a waiting list when it is deleted from `recepient` table**

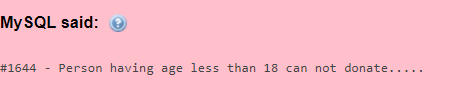
Recipient r-13 Deleted from `recepient`table

****

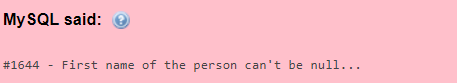
Simultaneously it deleted from the waiting list

****

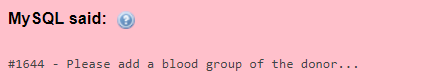
1. **The person below 18 can’t donate the organ**

****

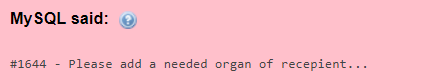
1. **First name of the donor can’t be null**

****

1. **Blood group of the person can’t be null**

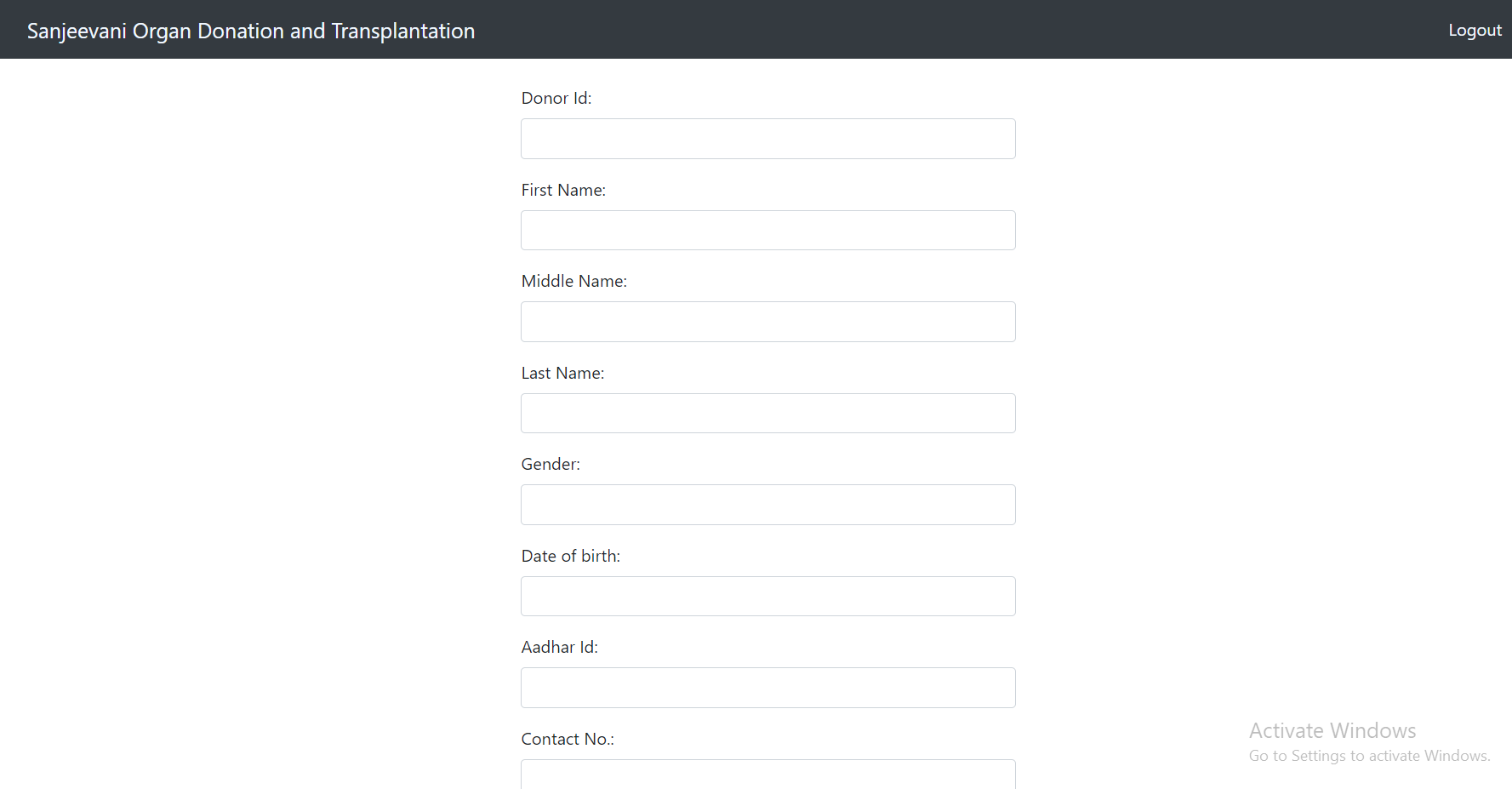
****

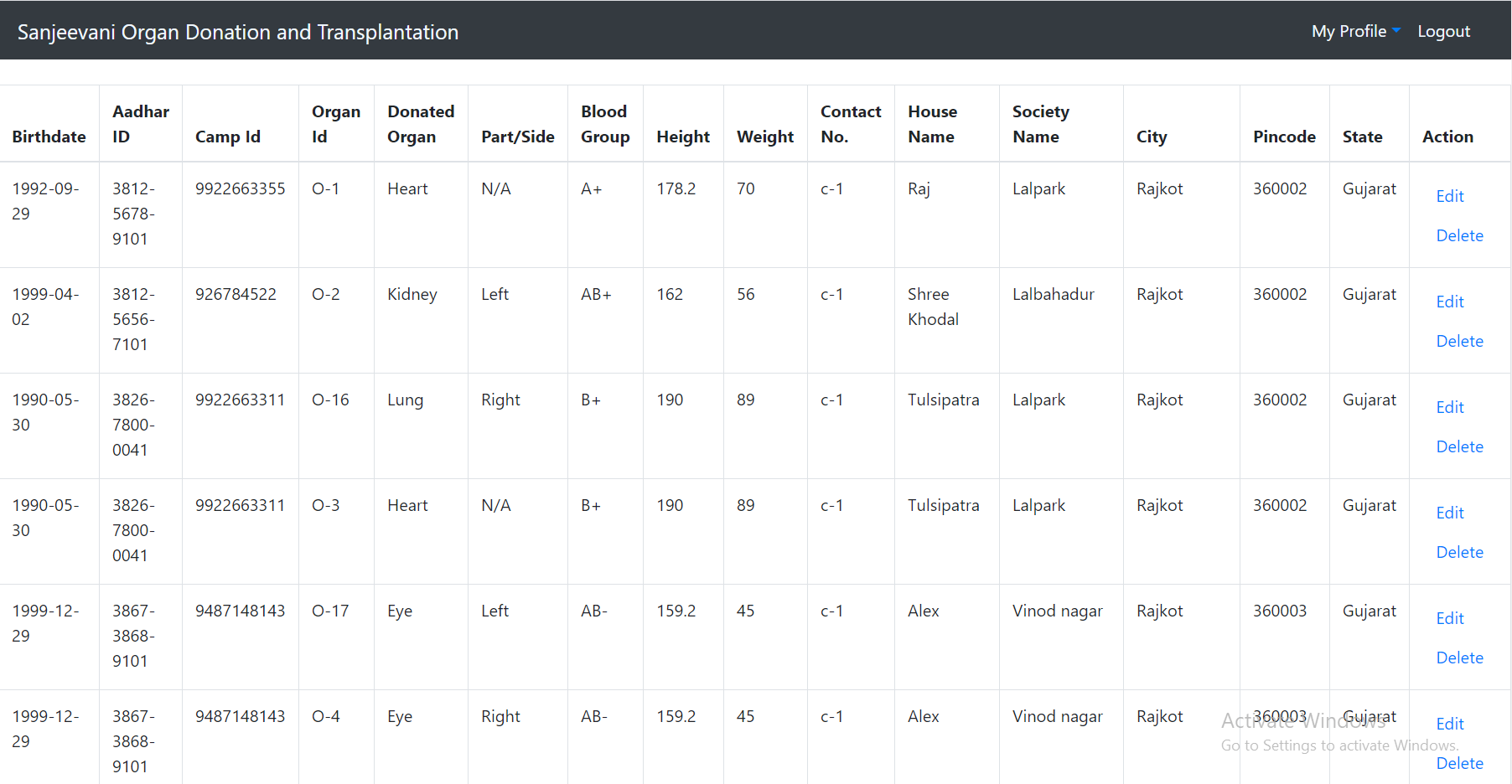
1. **Needed organ of recipient can’t be null**

****

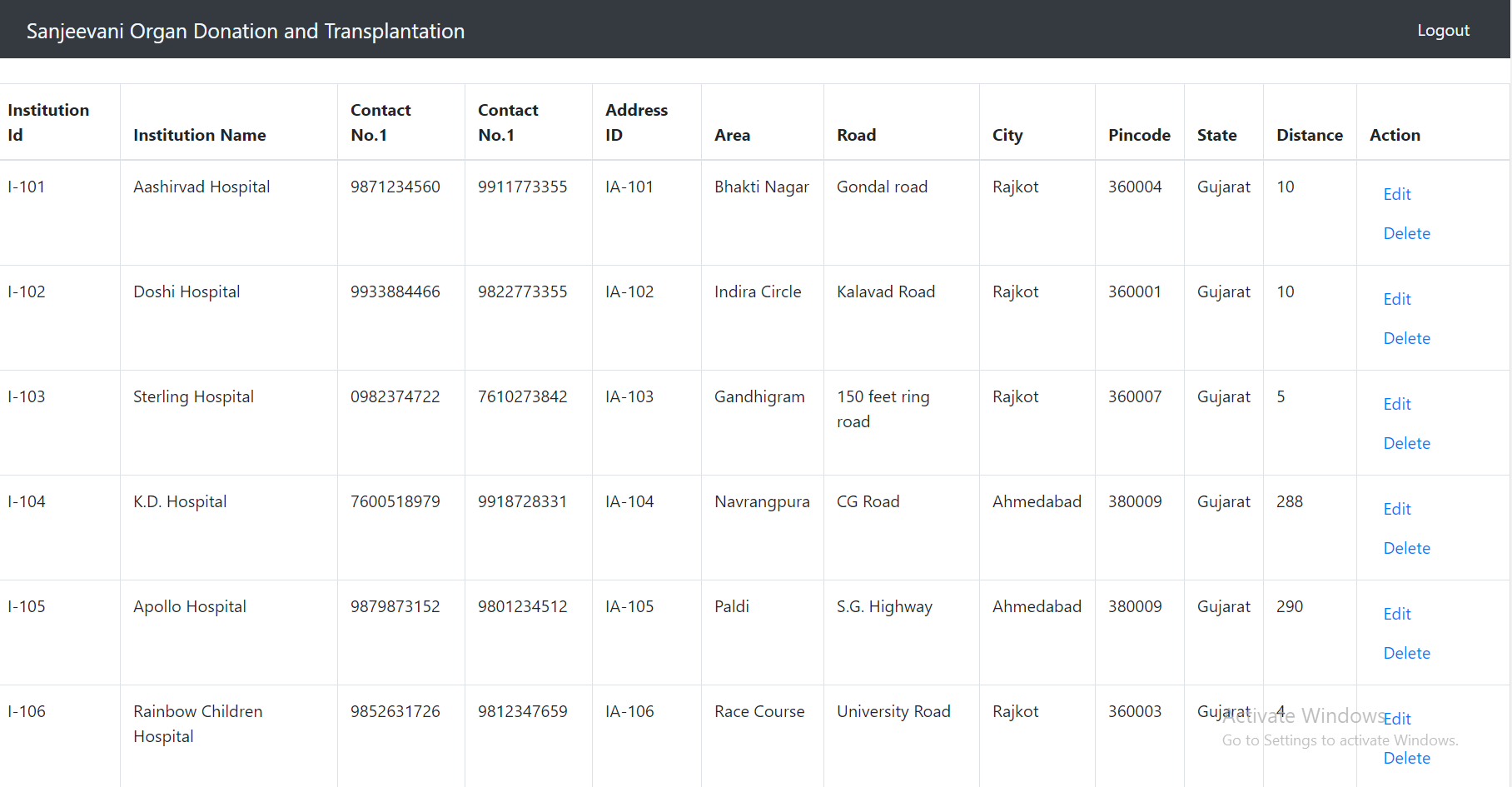
**Other Features:**

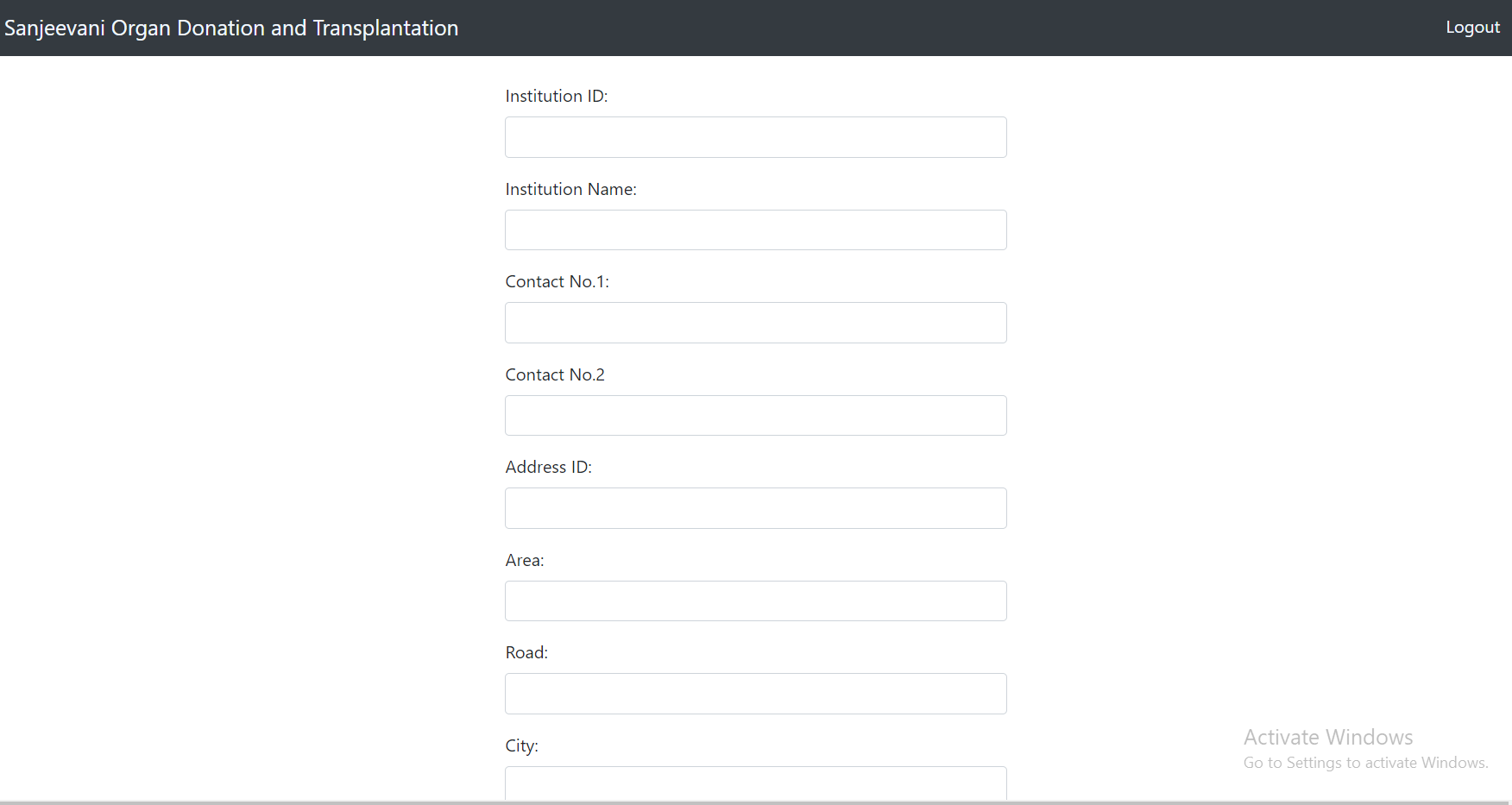
**Admin can add and manage data of Donors:**

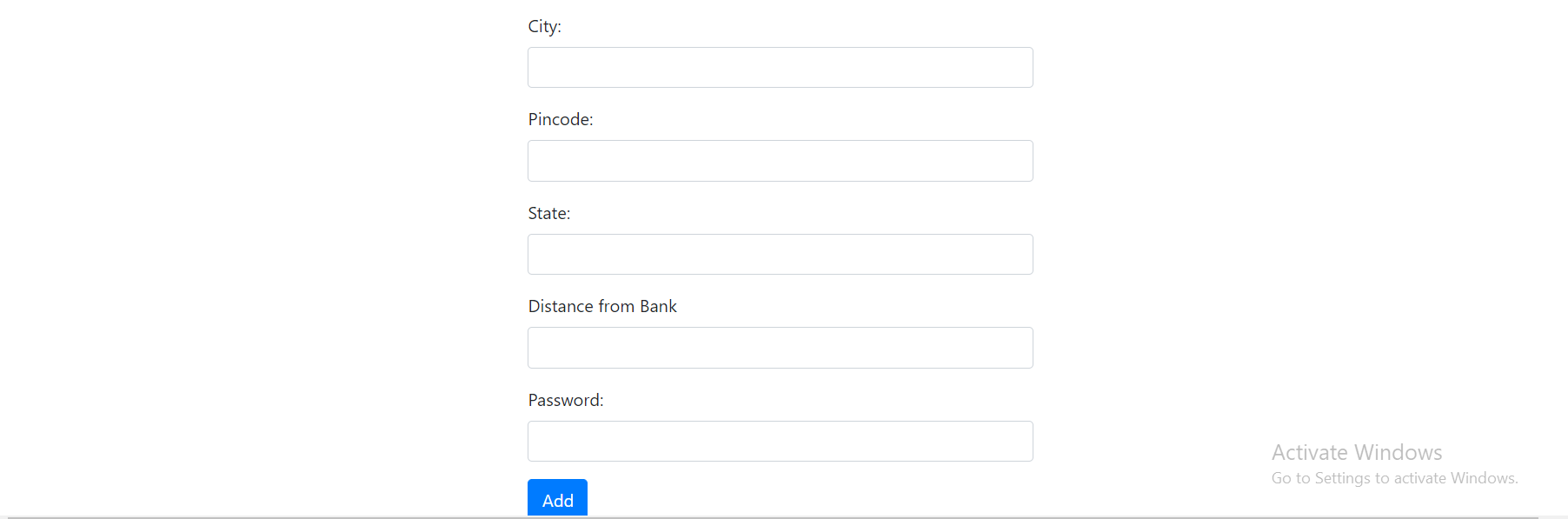
****

****

**Admin can Add and manage Coodinators’ data:**

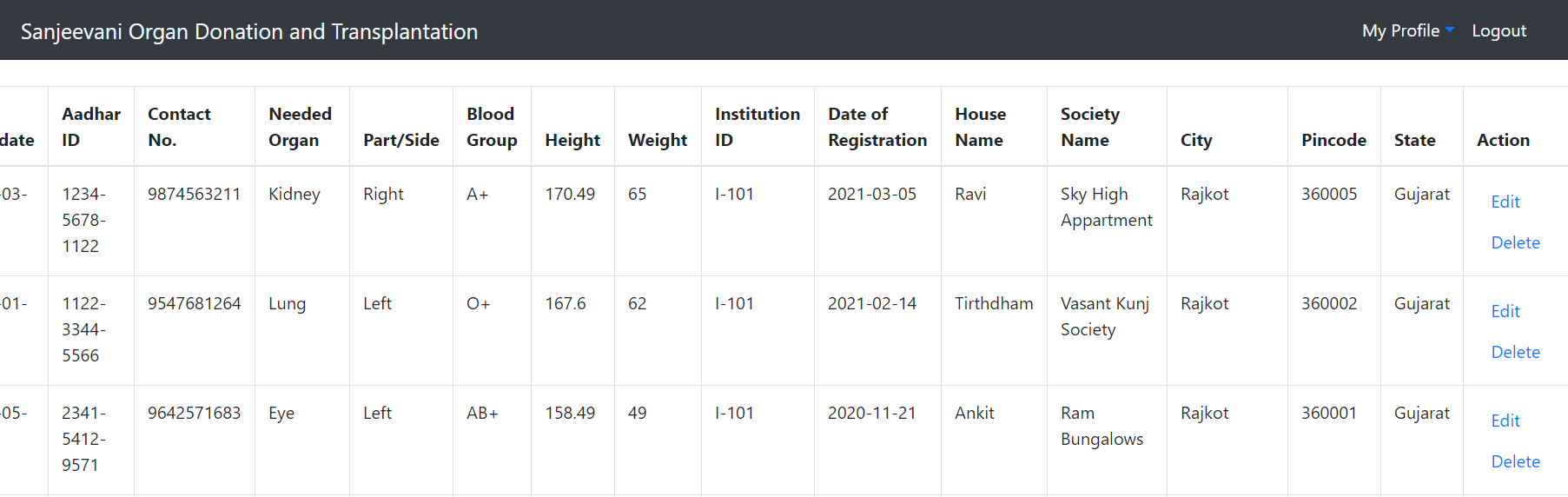
****

****

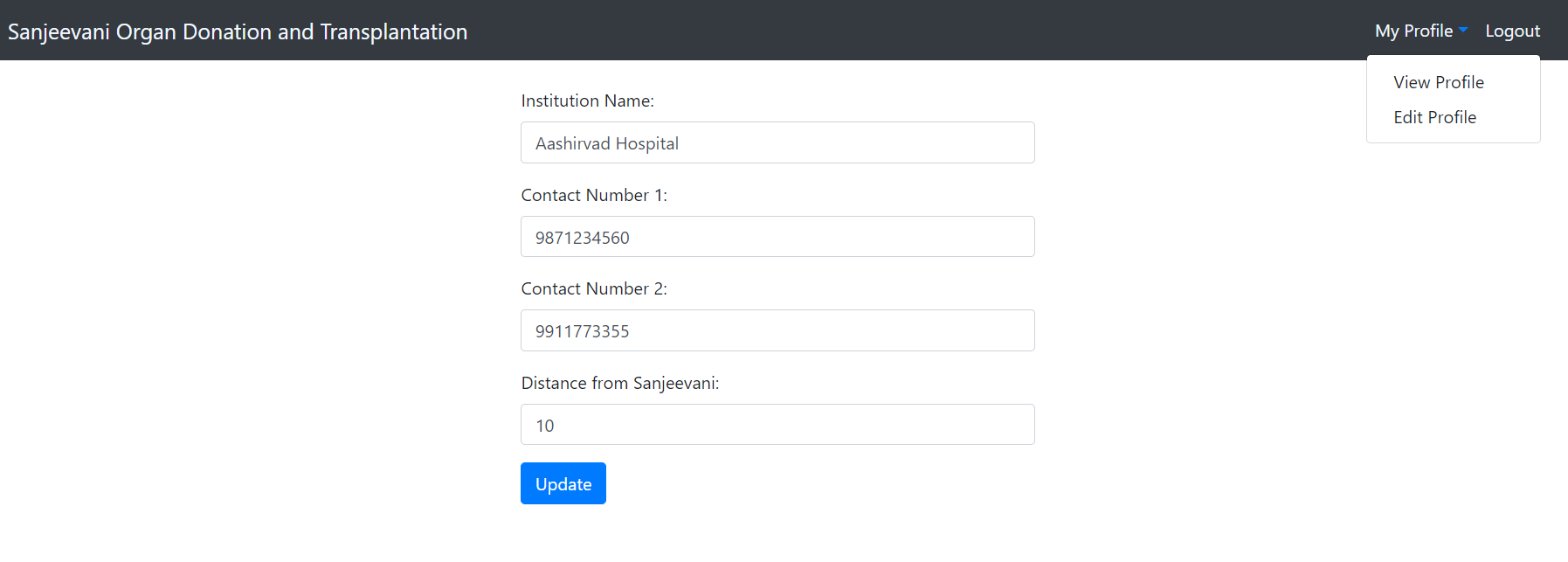
****

Similarly, the admin can add and manage data of camps and recipients …

**Hospital can add and manage data of their recipients**

****

**Hospital can view and edit their profile**

****