Project Proposal

CS 203

DATABASE SYSTEMS

**NAIKI**

Members

Ovaiz Ali 18K-0137

Zaeem Ahmed 18K-0166

Muhammad Hassan 18K-

Teacher: Dr. Zulfiqar

20thDecember 2020

# 1) DESIGN PHASE:

## a) Proposal:

**i) Introduction and brief description.**

As per the latest poverty estimates, 24% of Pakistan's population lives below the national poverty line; which includes 31% in rural areas and 13% in urban areas. Hence, there is a desperate need for an automated efficient system for Pakistan which connects privileged with the underprivileged people of Pakistan.

This project will provide a platform for people to donate whatever they are fortunate with (blood, money, clothes, time, knowledge), for the betterment of the lower and underprivileged class people who are living hand to mouth conditions or have lost their home due to natural disasters or any reason and for the betterment of the environment of the country. Moreover, it also encourages people to donate more and more as the user will also receive points after each donation depending upon the amount and type of donation made which can later be redeemed for free or discounted items at the selected store (or these points can be donated again in the form of cash).

Therefore, NAIKI is a system which takes Donors and the Help seekers on board in order to support those in need.

**ii) What functions should the system perform? That is, inventory control, billing,**

**ordering, etc.**

A needy person will call on the NGO's customer care or visit them in person with their request of the need. The staff at NGO will lodge their needs into the system and then search it in the donation database whether any of the donors is donating what is required by the needy. This will lead to 2 outcomes; (1) The request is matched with the donation and the donor is sent a message that their donation has been accepted, (2) The request is not matched as none of the donors is donating what is needed. In this case the request will be uploaded in the database and when any match is available, the needy person will be contacted by the NGO staff.

A donor will login into their account and then publish the ad/status of their donation. After it’s published, the database will automatically search for matching needs as per the donation. (1) If it is matched the donor will be prompted and asked if he wants to donate to the person prompted. If he chooses yes, the NGO staff will be notified that the request for the need has been accepted by the donor. If he chooses no, then the ad/status of donation will be published. (2) If it is not matched then the ad/status will automatically be published and whenever the matching requests will be made, the donor will be notified if he wants to donate to this person or not.

(A donor will login and search for the available requests on the database and then he can choose to donate to any kind of request that is being posted.

**iii) Front-End and Back-End Technologies**

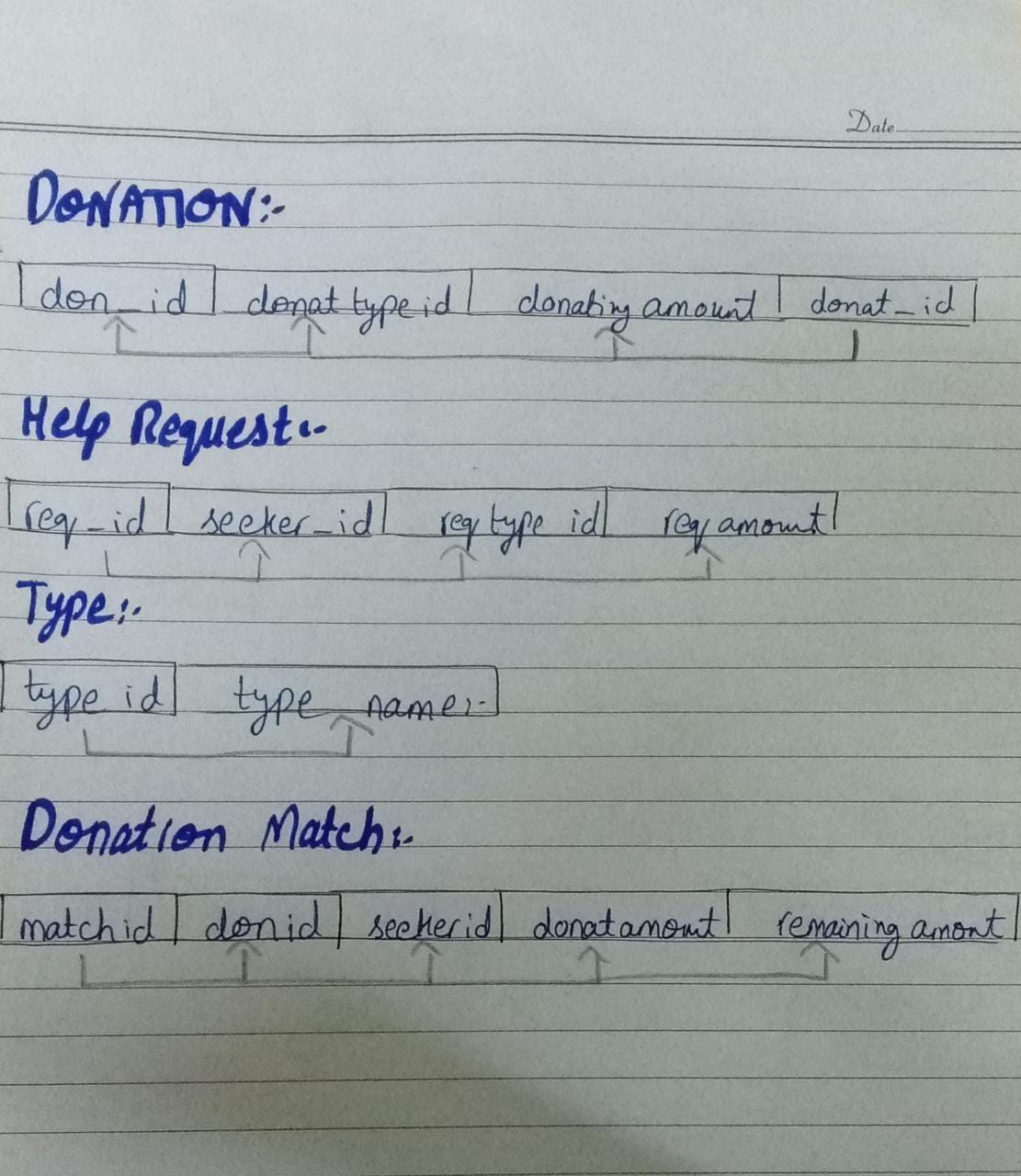
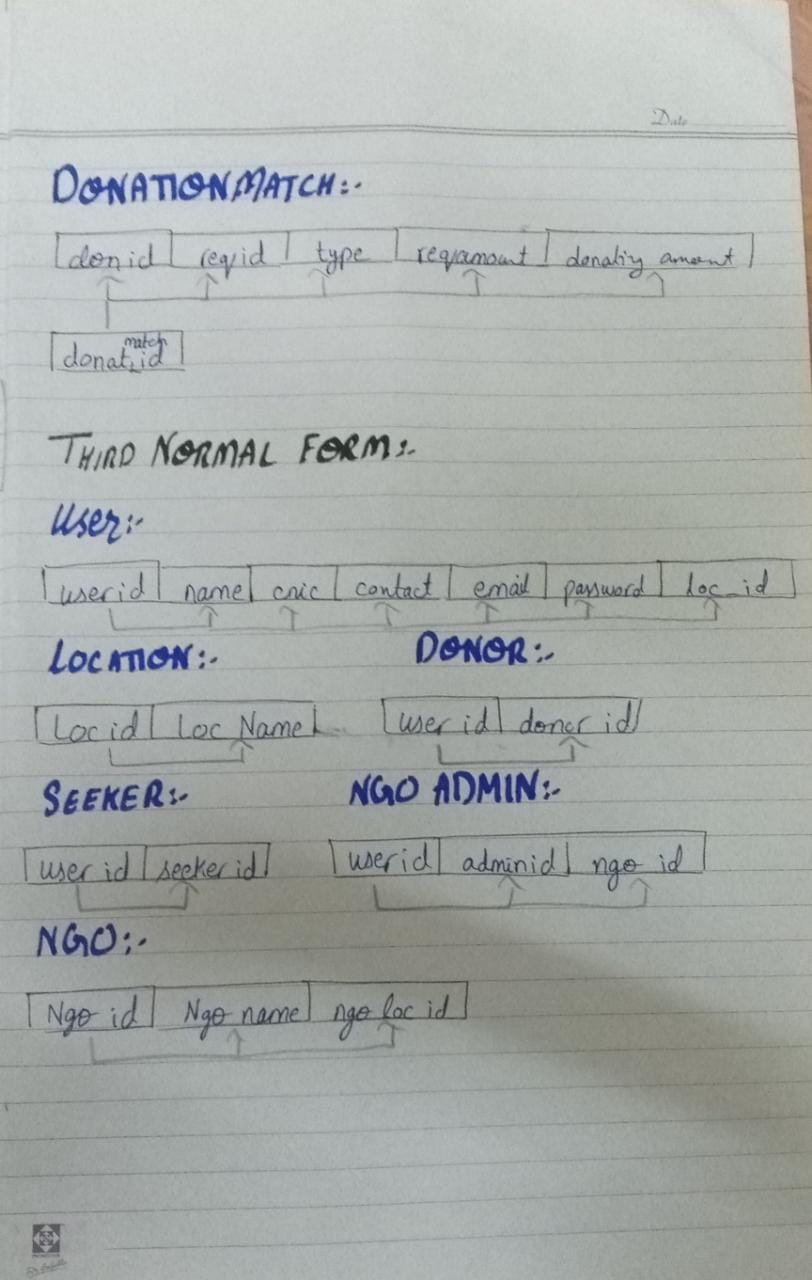
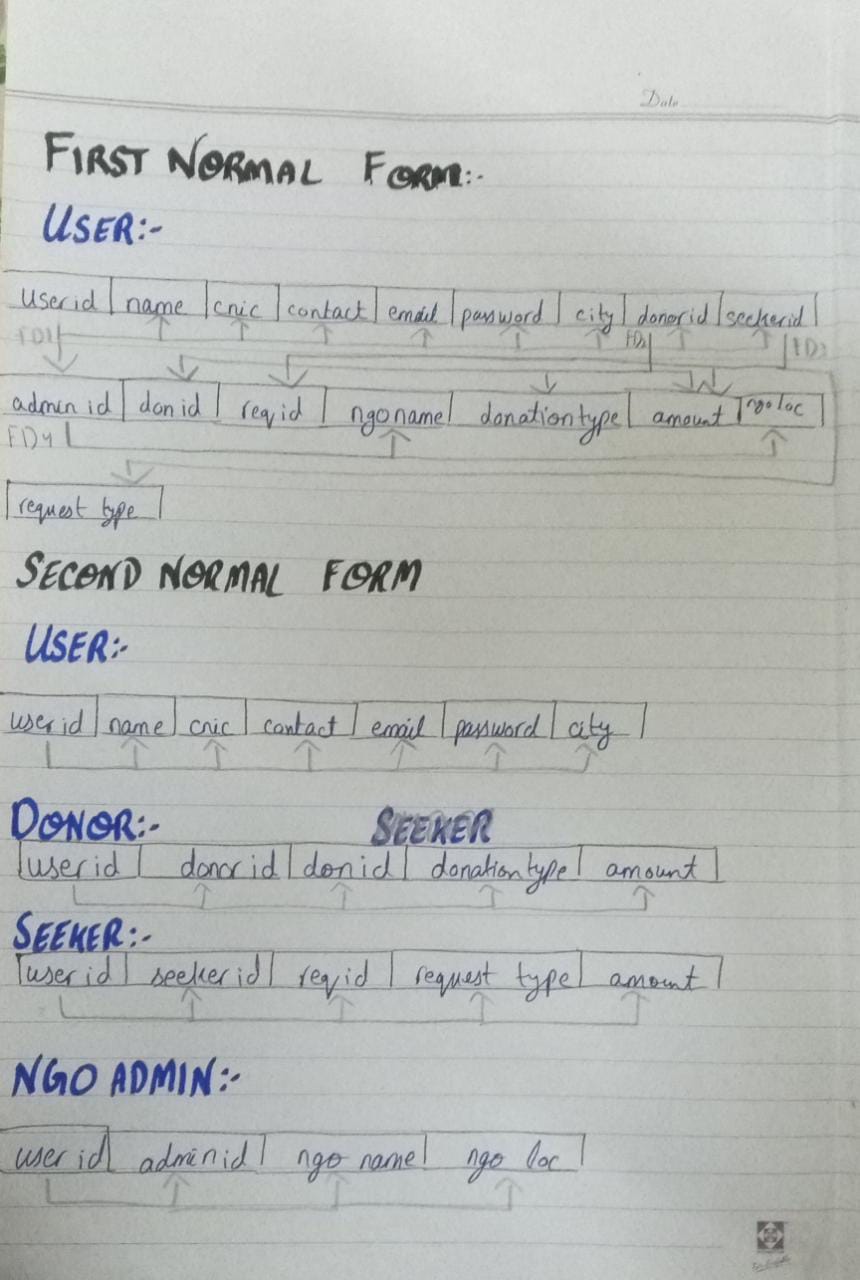
Front-End Technologies

* HTML
* CSS
* JavaScript

Back-End Technologies

* Node Js
* Express Js
* MySQL

## b) Normalization:

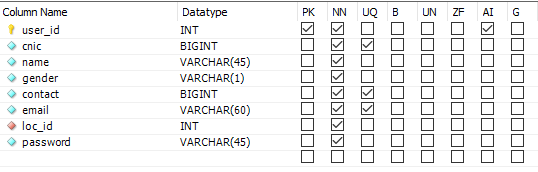


## c) Entity Relationship Design:

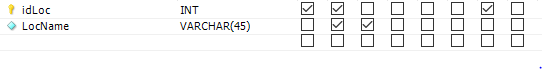
**i) Describe your entities. You must have enough entities to insure your project is not a "toy" system. GENERAL RULE: you should have about 5 entities and 7 or 8**

**tables.**

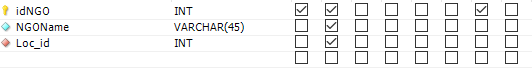
* **Entities**
  + Donor (Person who donates on our portal)
  + Help Seeker (Person who receives donation from our portal)
  + NGO (Registered NGO’s on our portal, in order to place donation request or help out us if we are short of resources)
  + Sys\_User (General users of the portal, it covers all of them donors, help seekers and NGO admins)
  + Location (To keep track of the location of stakeholders, in order to carry out further proceedings)
  + NGO Employee/Admin (Representative from registered NGO’s on our portal which manages all the donation requests and donations accumulated)
  + Donation Request (Donation request being placed are tracked to fulfill our help seekers necessity)
  + Donation Type (Specified types of donations which our portal deals in)
* **Tables**
* Sys\_user (all details of user)



* Location (data of cities)



* Ngo (data of NGOs in the system)



* Ngo\_emp (data of employees working in ngos)



* Donor (data of all the donors)



* Seeker (data of all the help seekers)



* Donat\_type (data of all the things donation can be made)



* Donation (data of donations made by donor)



* Donation\_req (data of all help requests made by help seeker)



* Don\_details (data of all donations allocated to the help seeker)

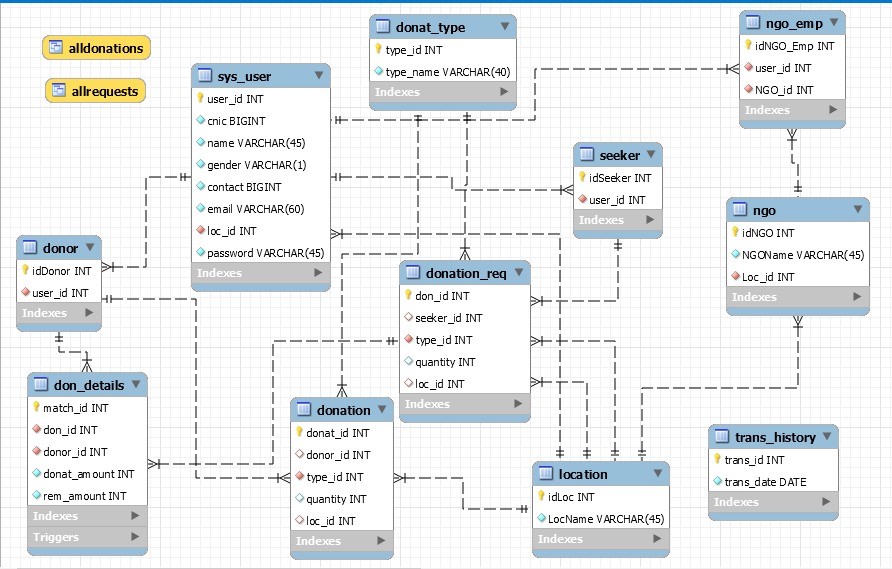


* Trans\_history (data of the time when a donation is allocated)



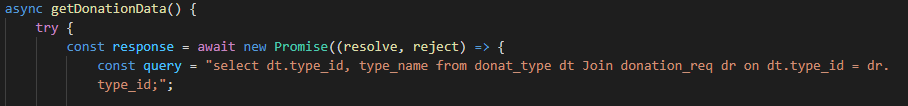
**ii) Draw the E--R Diagram for your database.**

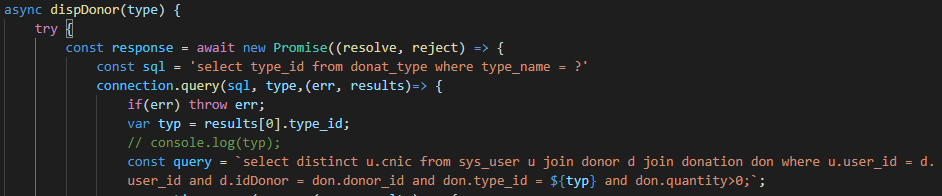
* **Extended Entity Relationship Diagram:**

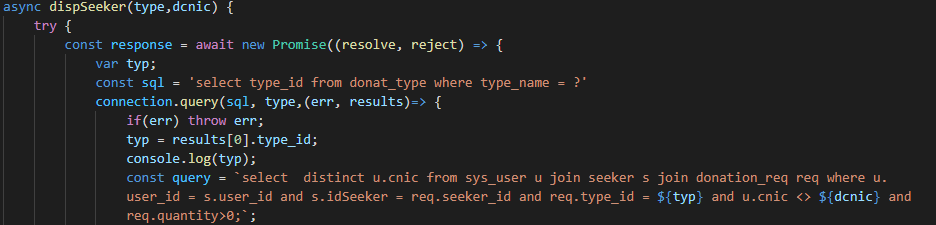


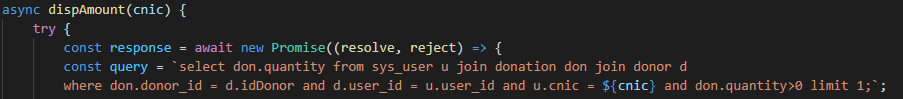
# 2) IMPLEMENTATION PHASE:

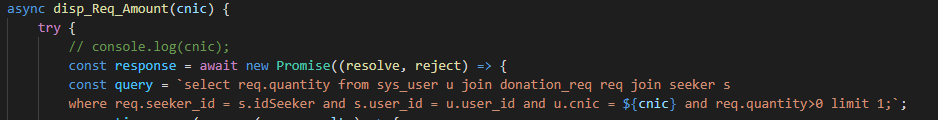
**a) Joins:**

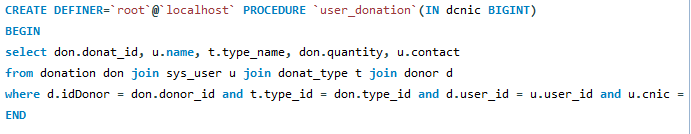


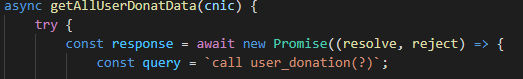


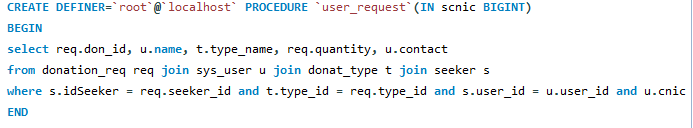


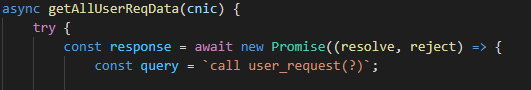






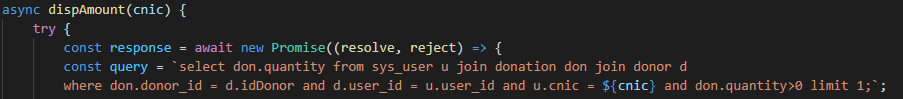


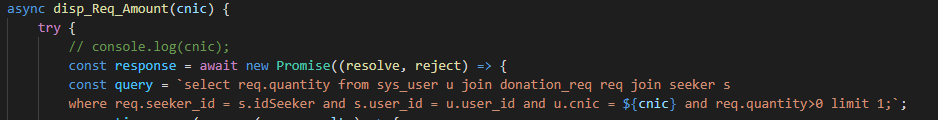




**b) Advanced SQL Queries:**

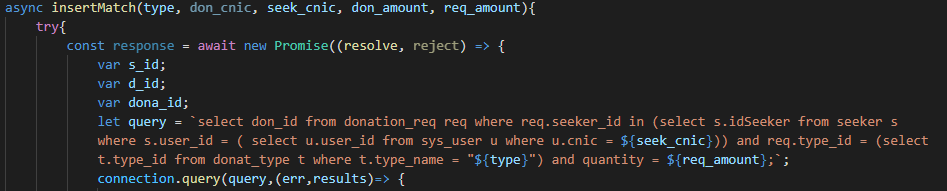
**LIMIT**



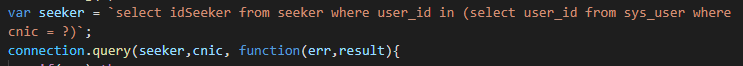


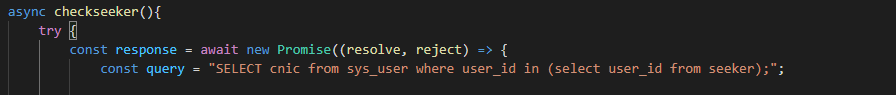


**SUB QUERIES**

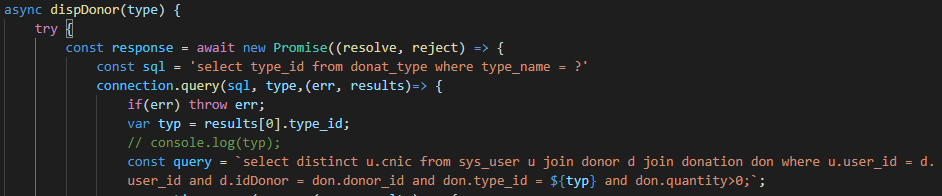


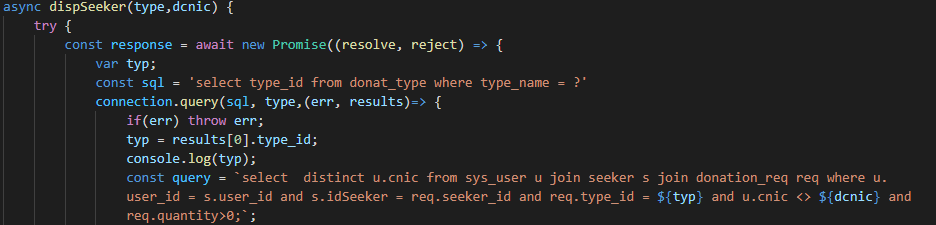






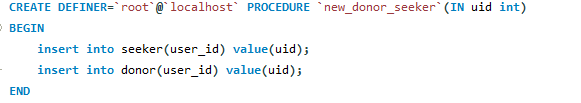
**DISTINCT**

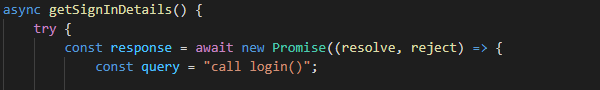


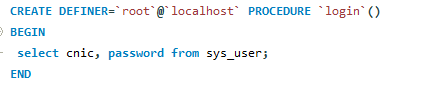


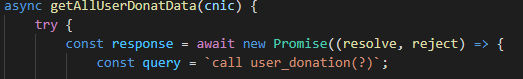
**PROCEDURES**

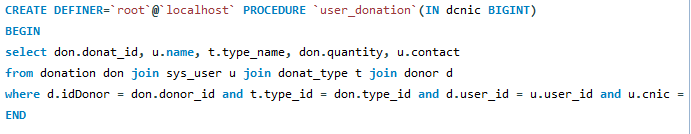


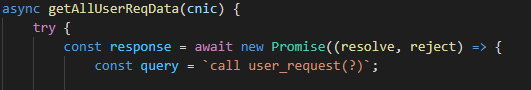


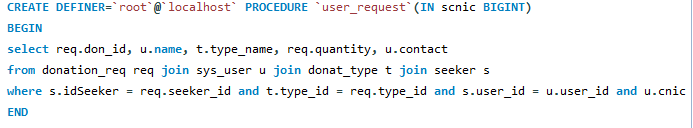


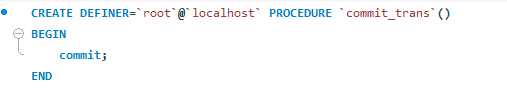




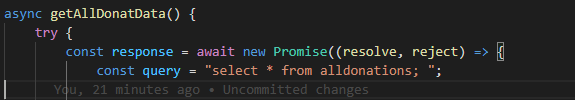




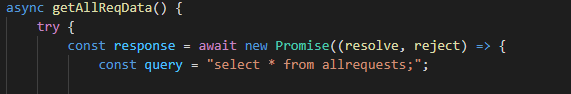




**VIEWS**

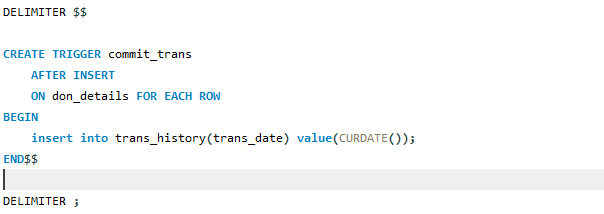




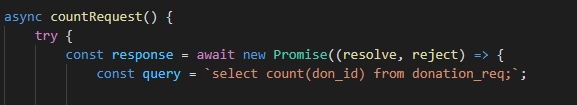


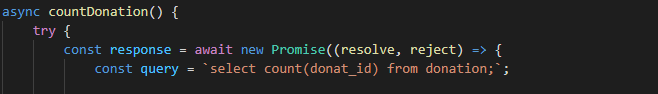


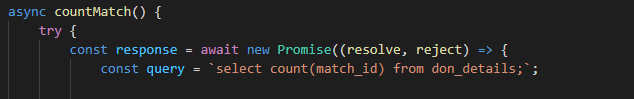
**TRIGGER**



**c) Built-in Functions:**

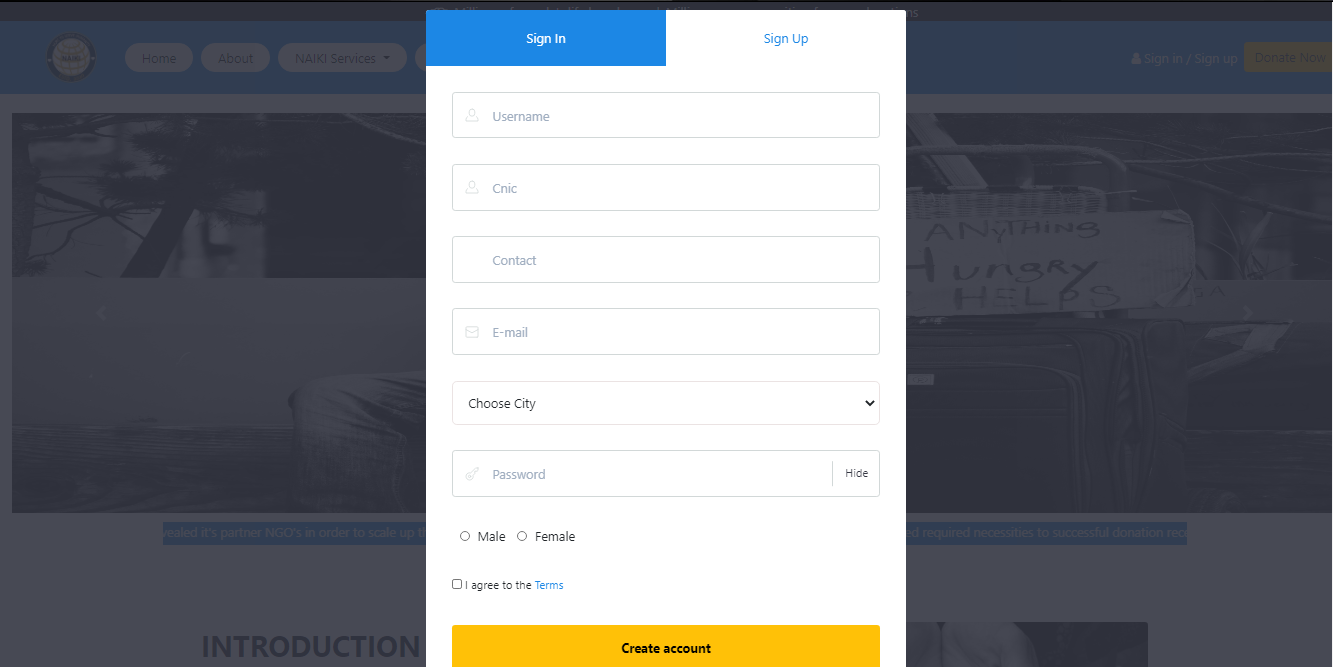


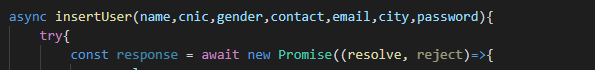




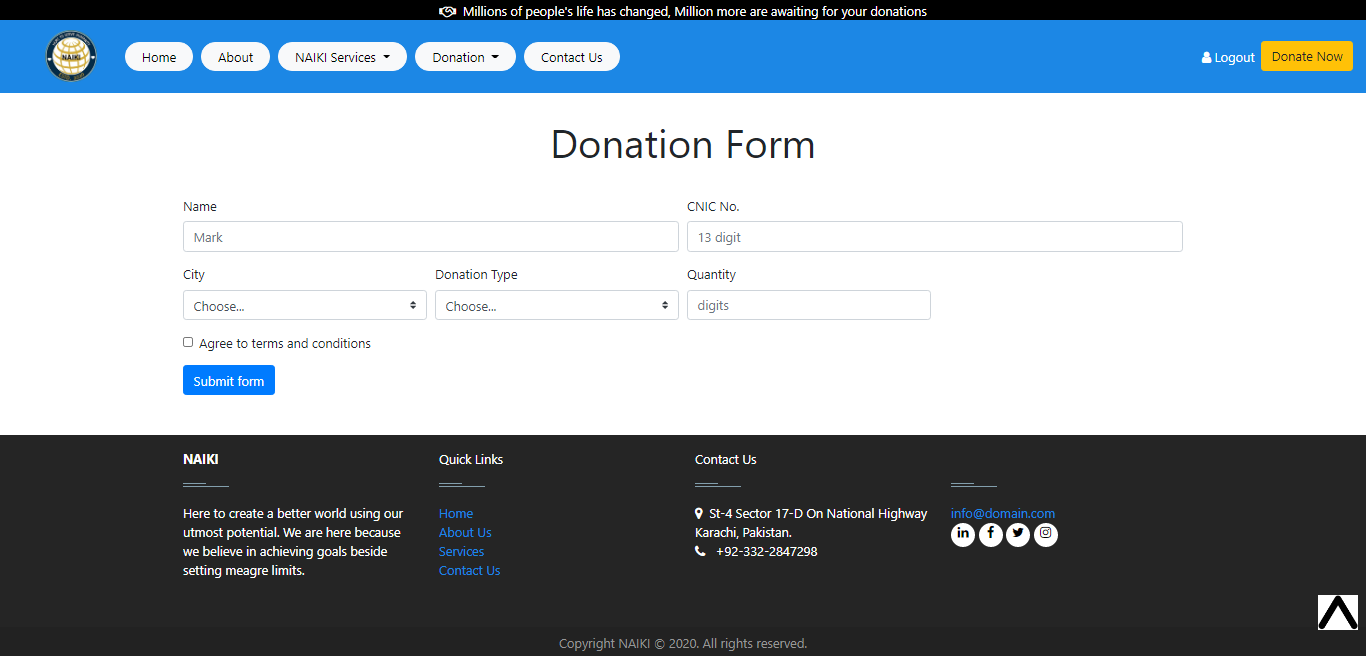
**d) User Input:**

**SIGNUP**



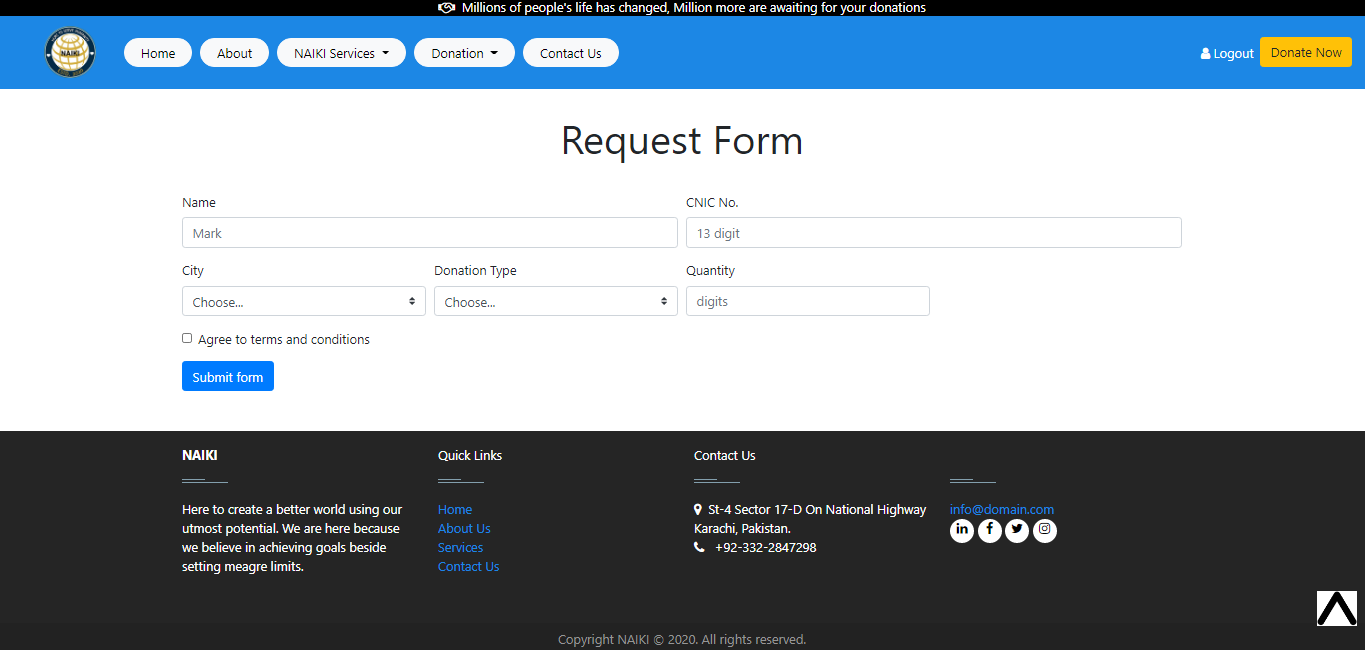


**DONATION FORM FOR DONOR:**



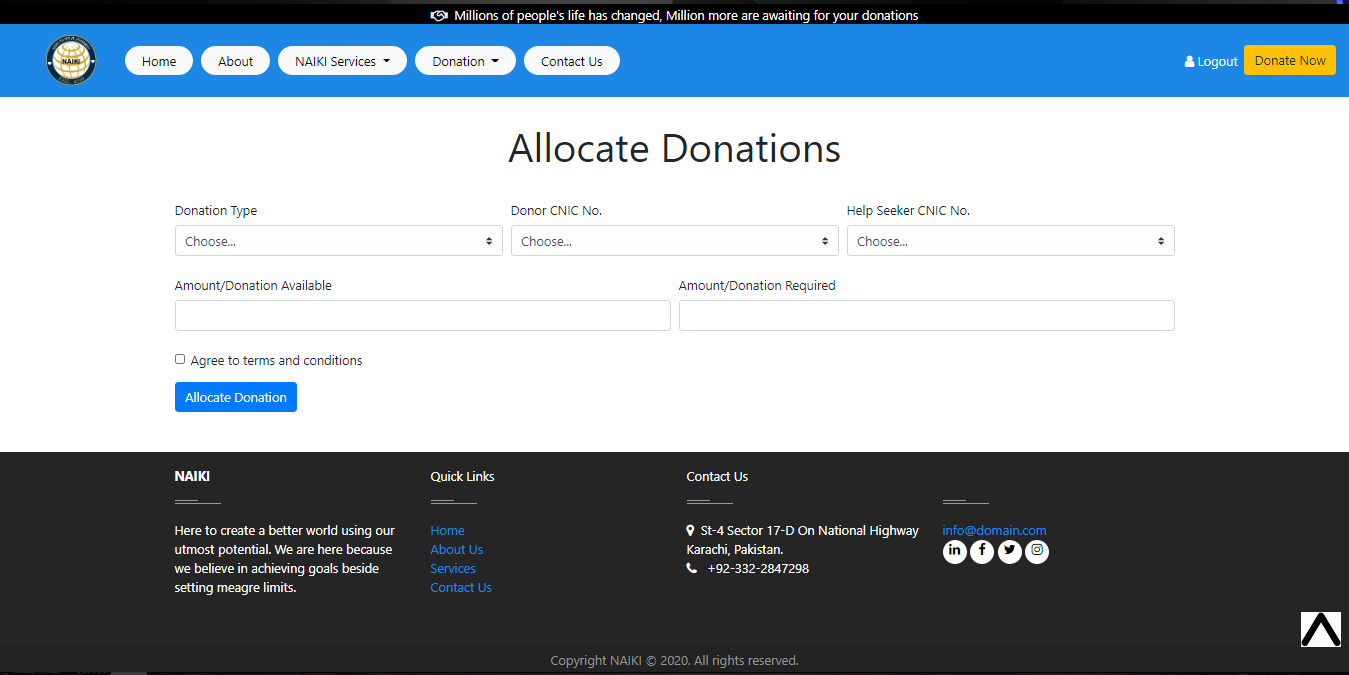


**REQUEST FORM FOR HELP SEEKER**





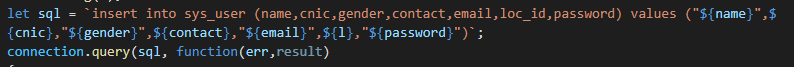
**DONATION ALLOCATION FORM FOR NGO EMPLOYEE**



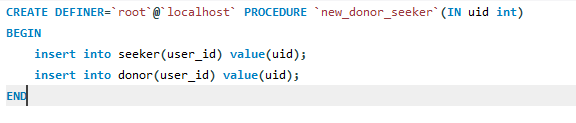


**e) Insert:**

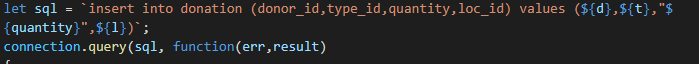
**NEW USER**



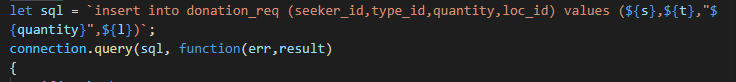
**NEW DONOR AND SEEKER**



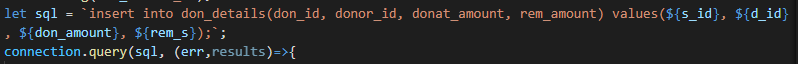
**NEW DONATION**



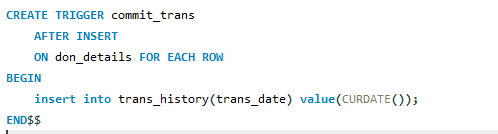
**NEW HELP REQUEST**



**NEW DONATION ALLOCATION**



**ALLOCATION TIME**



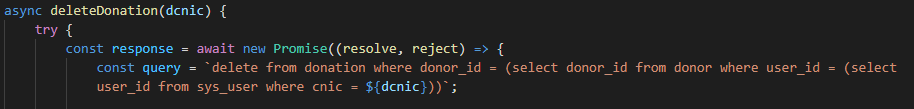
**f) Update:**

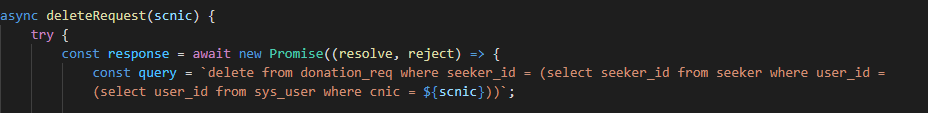
**in function** insertMatch



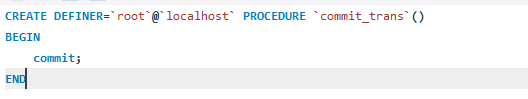


**g) Delete**



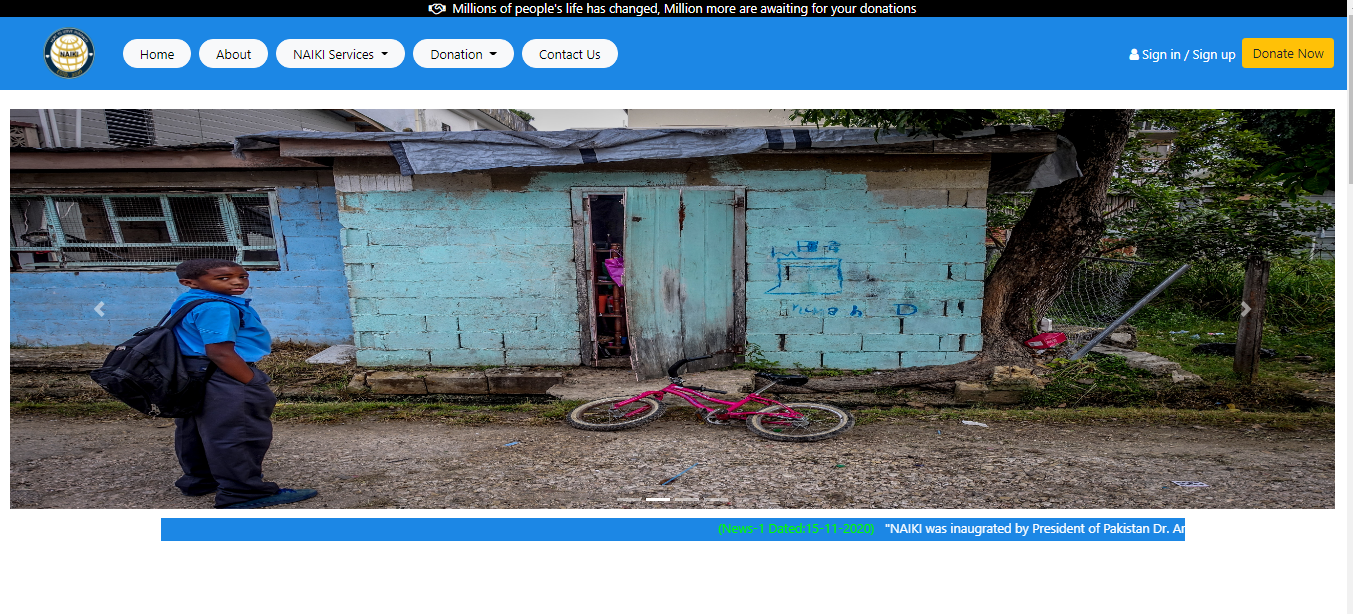


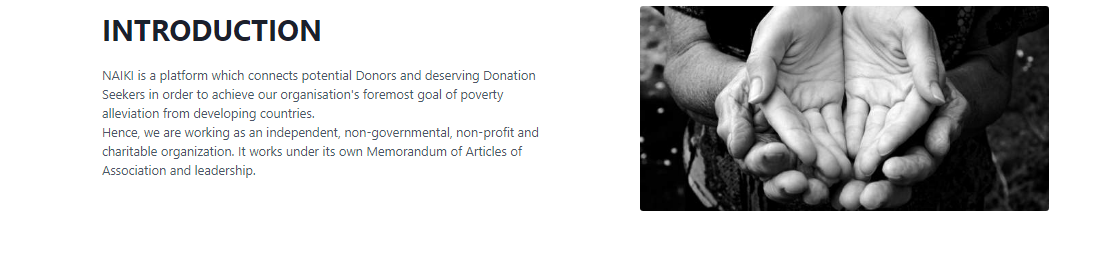
**h) Transaction:**

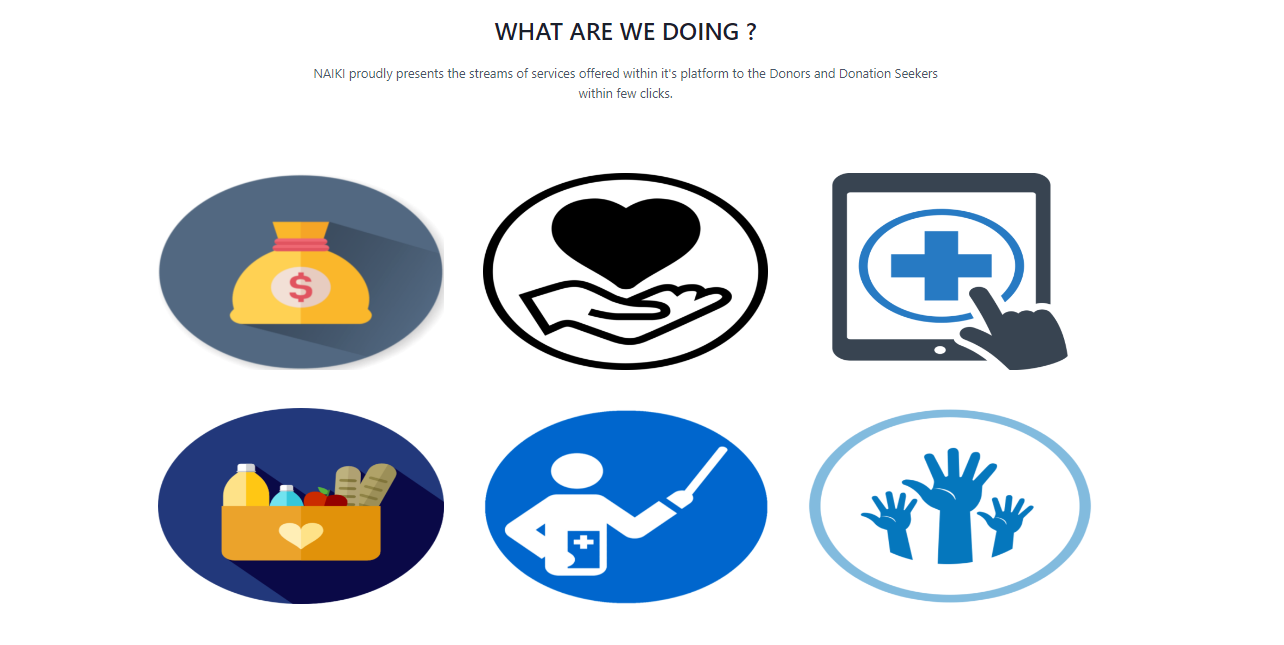


# SCREENSHOTS OF PROJECT

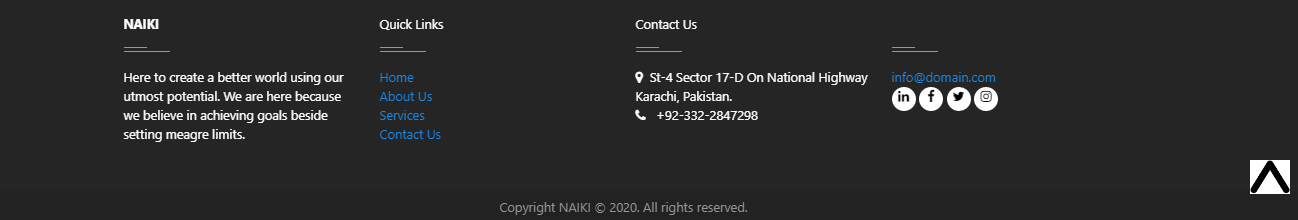
## HOME PAGE



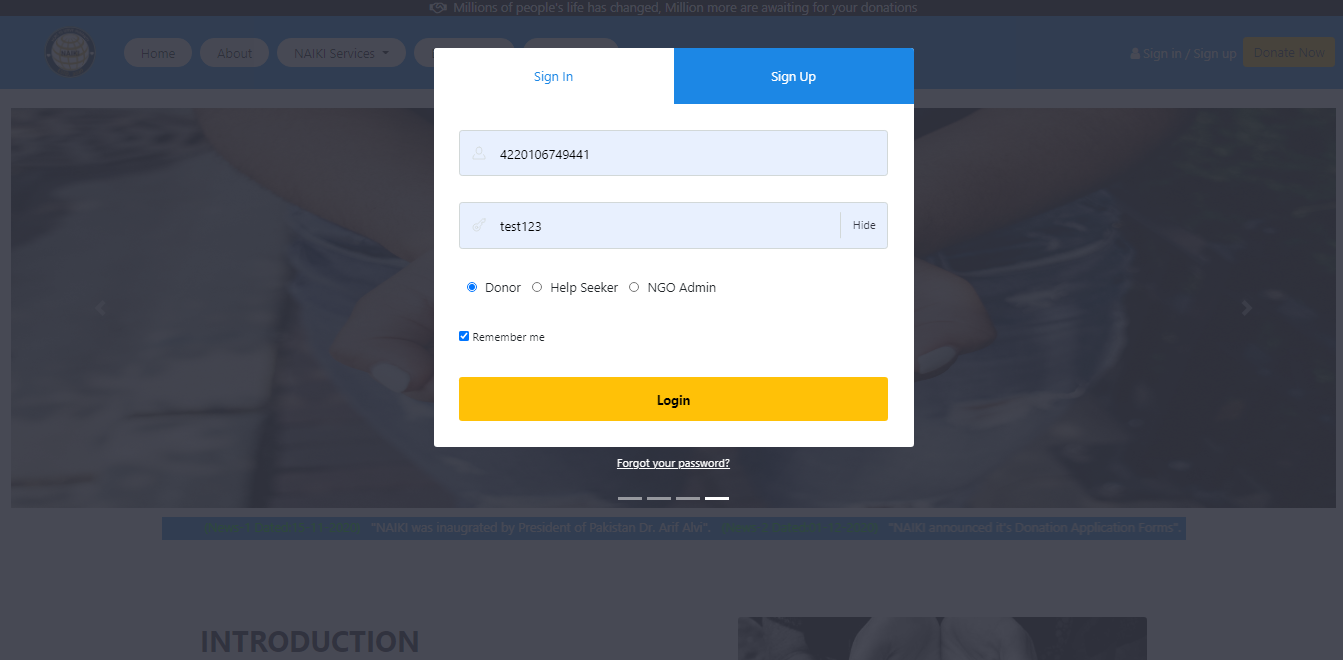








## LOGIN



## SIGN UP

## 

## DONOR

## 

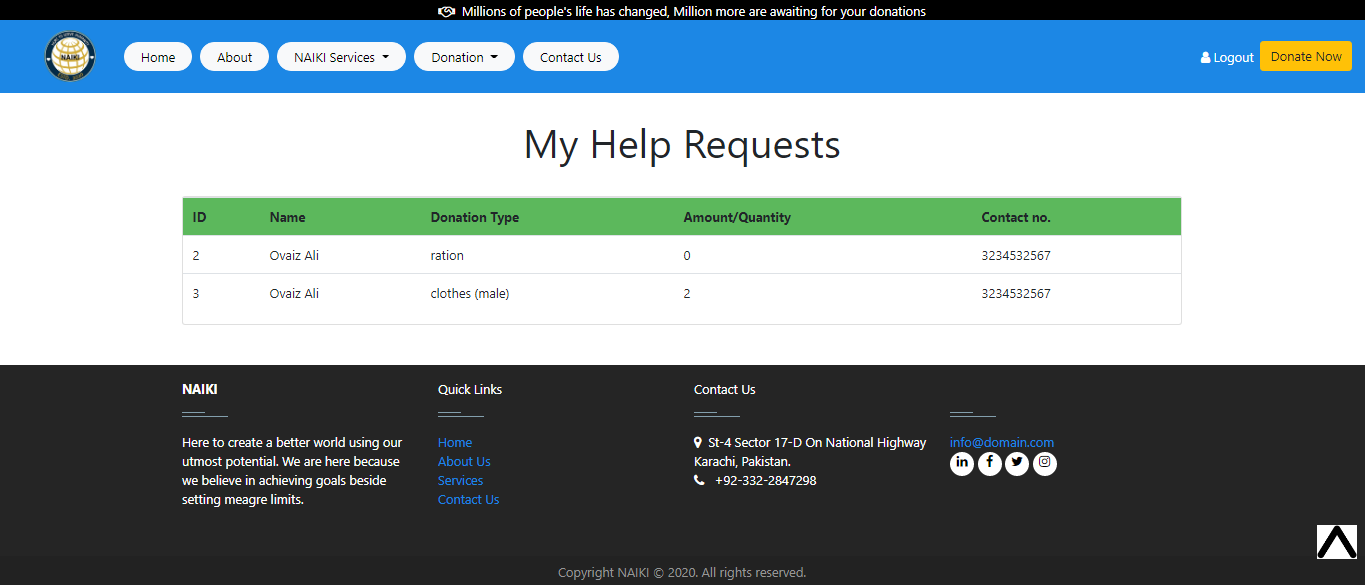
## 

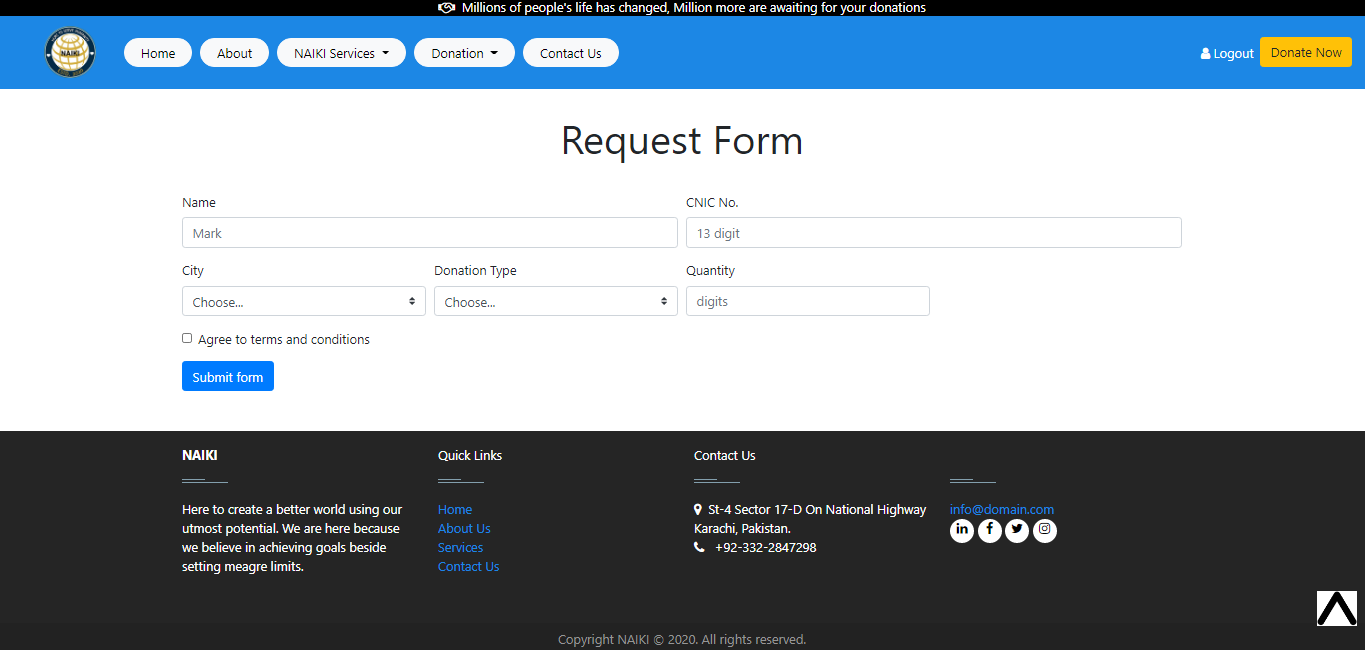
## 

## 

## SEEKER

## 





## NGO ADMIN

