CS204 Week 2 Report

Group 11

NGO Database

Week 2 :

In this week we have completed the NGO side of the website the part where NGO will monitor it's database. Functionalities like the donation data and member data monitoring has been added to the NGO side of the website. Some of the functionality for the User side has been done as well.

NGOs can set up their profile on our website. It can create a new profile giving all the details regarding it's functionality. They can change any data from the profile using the profile page. NGO's can look for new applicants that want to join their endeavour. User's can choose an NGO and apply for that. The respective NGO will receive the details of the same and will have to ability to say yes or no to the particular applicant.

Contact Page has been added for the NGO's so that any user of the website can contact them regarding any queries.

Description of the work in this Week :

Donation facility: This facility enables registered user to donate any amount to any particular NGO to use this functionality user need to login and at his dashboard he will be greeted with feature of donation utilizing this feature user can select any NGo of his choice based upon their works and donate any amount to it.Each donation

activity will be tracked by the website using donor id which is generated once user donated any amount.

Channelizing donation records: Once a user donated to any NGO the website will record his donation history the same will be visible to the user when the user logged in and arrives at the dashboard. At the dashboard he can track on which day and time he has donated to which NGo also how much he donated. Our website also records the donation information for NGO convenience also i.e. once any registered user donated to any ngo then that particular ngo will be having a record of donation i.e at what date and time which user has donated to them.

Integration of contact us page: After the completion of frontend implementation the backend integration of contact us page also completed using contact us page any NGO or user can reach out to the core committee for any type of issue raised.

Reusing Function: Previously due to javascript asynchronous nature we were facing issues with the return statement define in controllers due to which at each new route we need to create new connection and utilize new connection every time but now we had user Promise to resolve asynchronous nature and able to reuse a single connection multiple times to perform queries.

Member Functionality : Now User goes to its dashboard and then goes to the NGO to see a list of all NGO, He/she will see a button Join on the NGO, in case he/she has not joined it till now . As soon as the user clicks on it , He/she gets redirected to a page where he has to enter some details like email and password along with the description of why he wants to join that NGO. Remaining details are

being fetched through user table through use of email and password.After this as soon as the complete details of a user is gathered(Both from user table and that gathered during registration form), We put its complete detail in another table named member

Improved Code Structure a little bit : There is a file made named connection.js in which Several functions like execute, executeAndReturn, These will be used to run query and each one has different significance , If we only want to run query like Insertion etc then we call execute along with query as a parameter. If we also want some data to be returned then we call executeAndReturn along with a query as a parameter. This will return a promise, which could be resolved or rejected wherever it is being used. This improves the structure of code as it is removing redundancy.

Database :

We have implemented the backend part of the ngo signup and login where we have taken ngo-email-id as the primary key and we are using this to access all the table regarding ngo. we have created two table for donor because in case the user delete the account the amount get deleted from the user donor but the ngo needs the amount and if there is a ngo_donor table then the data will be stored there also and ngo can get data from that tables.

Donor Table :

```
create table donor(
donor_id int(125) primary key auto incremented not null,
donor_name varchar(125) not null,
user_email varchar(125)  not null,
Amount int(125) not null,
Ngo_name varchar(125) not null,
RegDate timestamp NULL DEFAULT CURRENT_TIMESTAMP,
constraint foreign Key(user_email) references signup(user_email) on delete cascade on update cascade
)
```

```
create table donor_ngo(
donor_id int(125) primary key not null,
donor_name varchar(125) not null,
ngo_id int(125) not null,
Amount int(125) not null,
Ngo_name varchar(125) not null,
RegDate timestamp NULL DEFAULT CURRENT_TIMESTAMP,
constraint foreign Key(Ngo_id) references ngo(Ngo_id) on delete cascade on update cascade
)
```