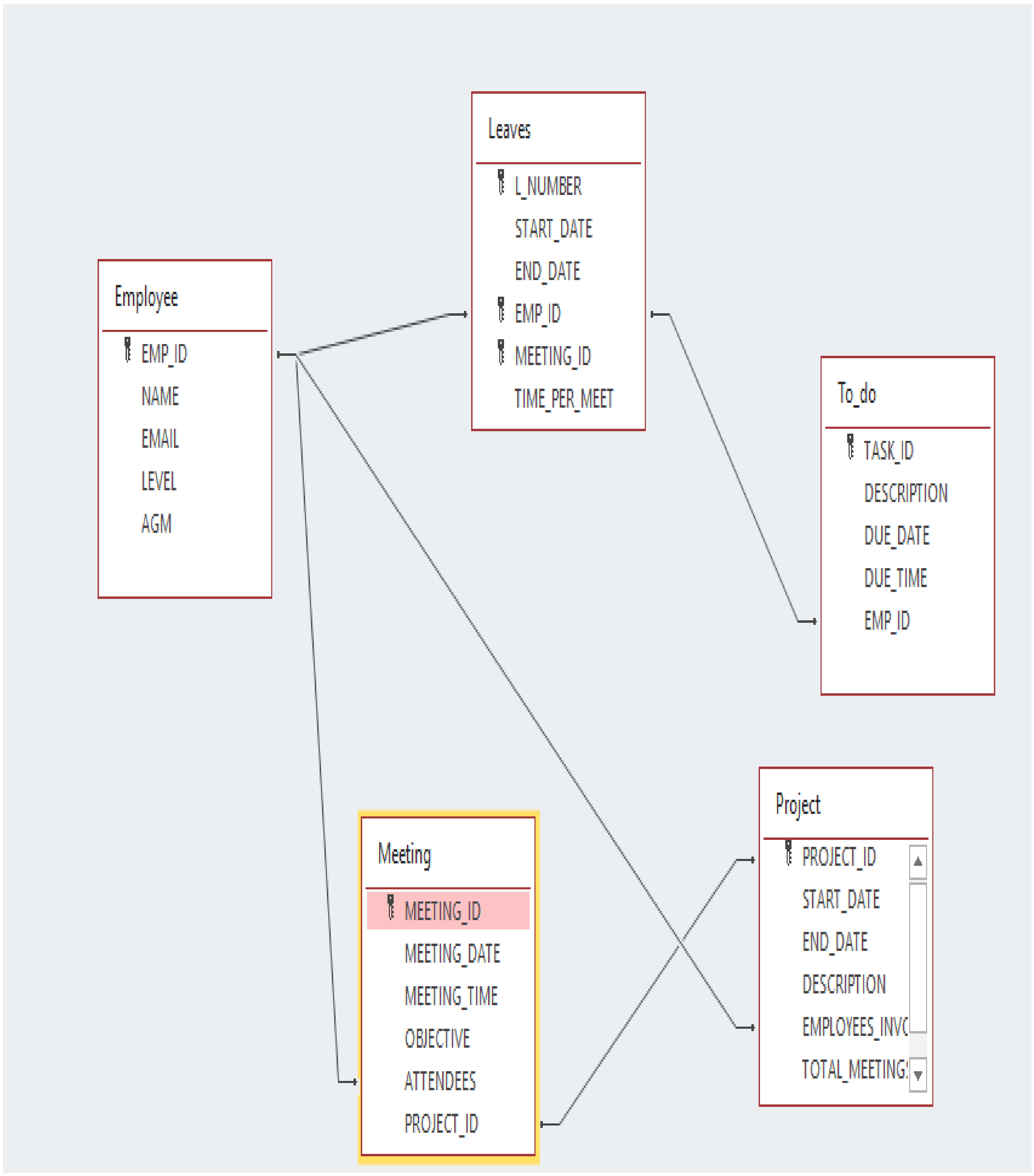
TIME MANAGEMENT SYSTEM

BY:

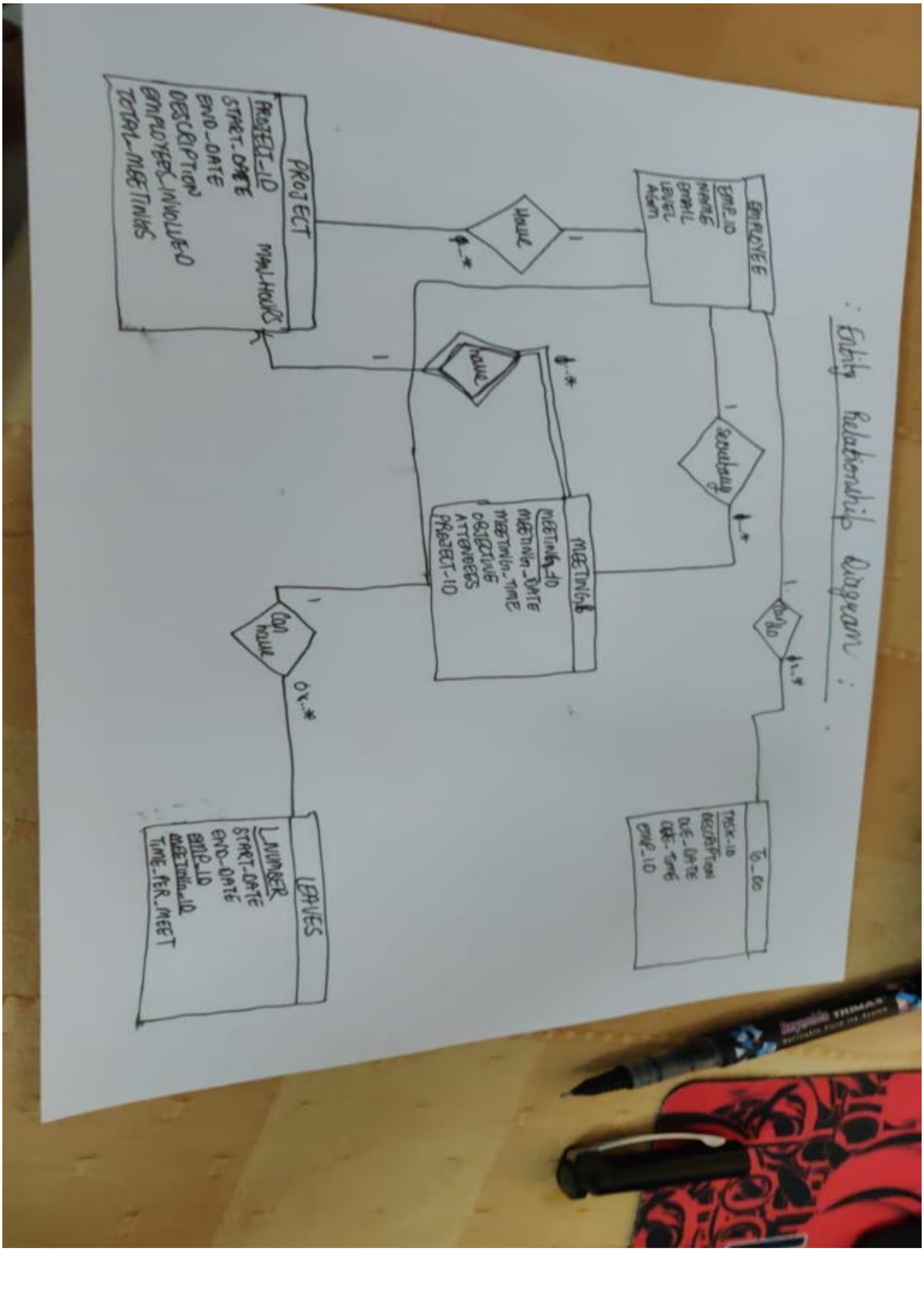
NEIL BHUTADA (181627135)

SHEERSHAK KUMAR (181627123)

SCHEMA DIAGRAM



Entity Relationship Diagram



MySQL Queries for the creation and population of tables.

- create table EMPLOYEE(EMP_ID int NOT NULL , name varchar(20) , email varchar(30) , level varchar(10), AGM numeric(4,2) , primary key(EMP_ID));
- CREATE TABLE PROJECT(PROJECT_ID INT NOT NULL, START_DATE DATE, END_DATE DATE, DESCRIPTION VARCHAR(1000), EMPLOYEES_INVOLVED VARCHAR(100), TOTAL_MEETINGS INT, MAN_HOURS INT, PRIMARY KEY(PROJECT_ID));
- CREATE TABLE MEETING(MEETING_ID INT NOT NULL, MEETING_DATE DATE, MEETING_TIME TIME, OBJECTIVE VARCHAR(1000), ATTENDEES VARCHAR(100), PROJECT_ID INT, PRIMARY KEY(MEETING_ID), FOREIGN KEY(PROJECT_ID) REFERENCES PROJECT(PROJECT_ID));
- CREATE TABLE STATS(EMP_ID INT NOT NULL , MEETING_ID INT NOT NULL, TIME_PER_MEET NUMERIC(4,2), PRIMARY KEY(EMP_ID , MEETING_ID));
- CREATE TABLE TODO(TASK_ID INT NOT NULL, DESCRIPTION VARCHAR(1000) , DUE_DATE DATE , DUE_TIME TIME, EMP_ID INT, PRIMARY KEY(TASK_ID) , FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID));

- CREATE TABLE LEAVES(L_NUMBER INT NOT NULL, START_DATE DATE, END_DATE DATE, EMP_ID INT, PRIMARY KEY(L_NUMBER), FOREIGN KEY(EMP_ID) REFERENCES EMPLOYEE(EMP_ID));

A few insertions that have to made in Employees table because the user can not add or delete the records in Employee table.

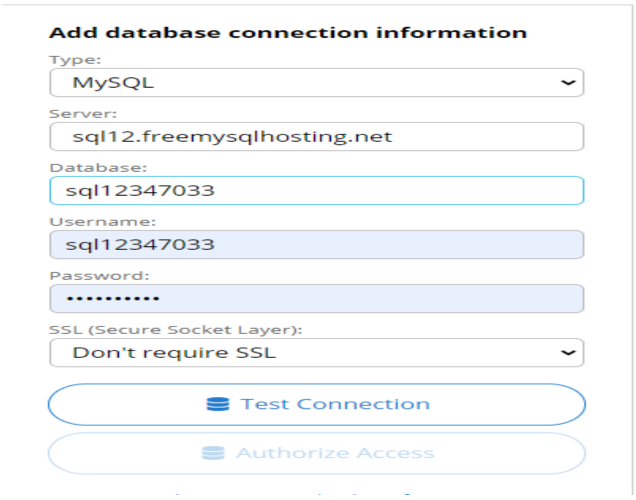
- Insert into Employee values (101, ‘Neil Bhutada’ , ‘neilbhutada@gmail.com’ , ‘Executive’ , 0);
- Insert into Employee values (102, ‘Sheershak Kumar’ , ‘sheershakkumar@gmail.com’ , ‘Executive’ , 0);
- Insert into Employee values (201, ‘Anil Kumble’ , ‘anilkumble@gmail.com’ , ‘Secretary’ , 0);
- Insert into Employee values (202, ‘Rashid Khan’ , ‘rashidkhan@gmail.com’ , ‘Secretary’ , 0);

FRONTEND AND BACKEND CONNECTION

To make our Time Management System we have used AppSheet (owned by Google). AppSheet follows a 'data driven' protocol to make an application. Based on the way the developer structures his/her data, AppSheet uses Natural Language Processing algorithms to create the UI/Front End. Hence immense importance is given to the way the developer structures and relates his data in a database.

In this project we connected our app made in AppSheet to a MySQL cloud database server called 'freemysqlhosting.net' and created our SQL database in "My pHp Admin" using the MySQL services it offers.

In AppSheet to connect our database to our App we have to enter credentials such as Server Type, Server name, Username and Password. It also ask whether our cloud database server uses SSL (Secured Socket Layer) which uses cryptographic messaging for security and validation. But for our case it was not required.



Add database connection information

Type: MySQL

Server: sql12.freemysqlhosting.net

Database: sql12347033

Username: sql12347033

Password: ••••••••••

SSL (Secure Socket Layer): Don't require SSL

The way AppSheet sends data to the cloud sever is via JSON (i.e Java Script Object Notation) , with the help of Web Programming tools :

- AJAX : Whose full form is Asynchronous Java Script And XML. AJAX allows us to perform asynchronous requests and procedures as in this case retrieving and sending data to our My SQL Cloud Sever.
- jQuery : Allows us to write queries to external sources for receiving and sending data.

Source code for establishing a request between cloud server and app

For your reference, I have inserted a hyperlink above which has the source code which establishes a relationship between the app and cloud server.

Out of the numerous lines of code the important functions / parts with examples or chunks are:

`Creation of jQuery Object :` `b=a.jQuery||a.Zepto;`

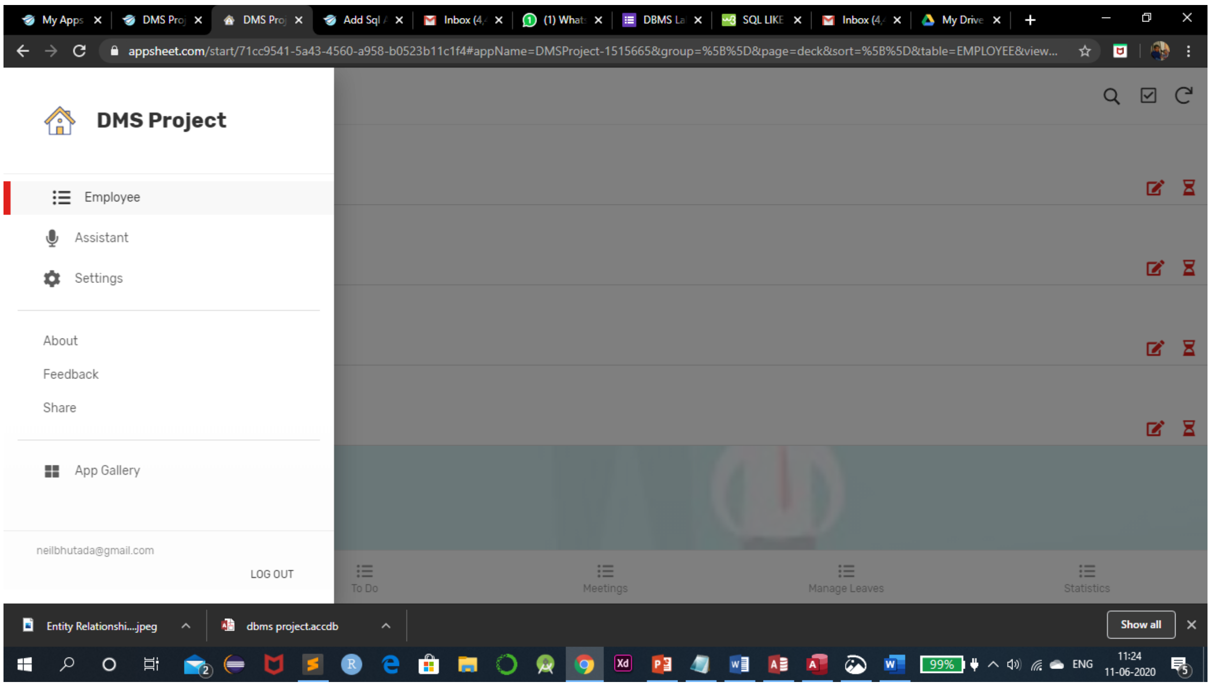
`Receiving data using getResult() :` `b.ua=v.getResult()`

Using AJAX and "POST" Method for sending data:

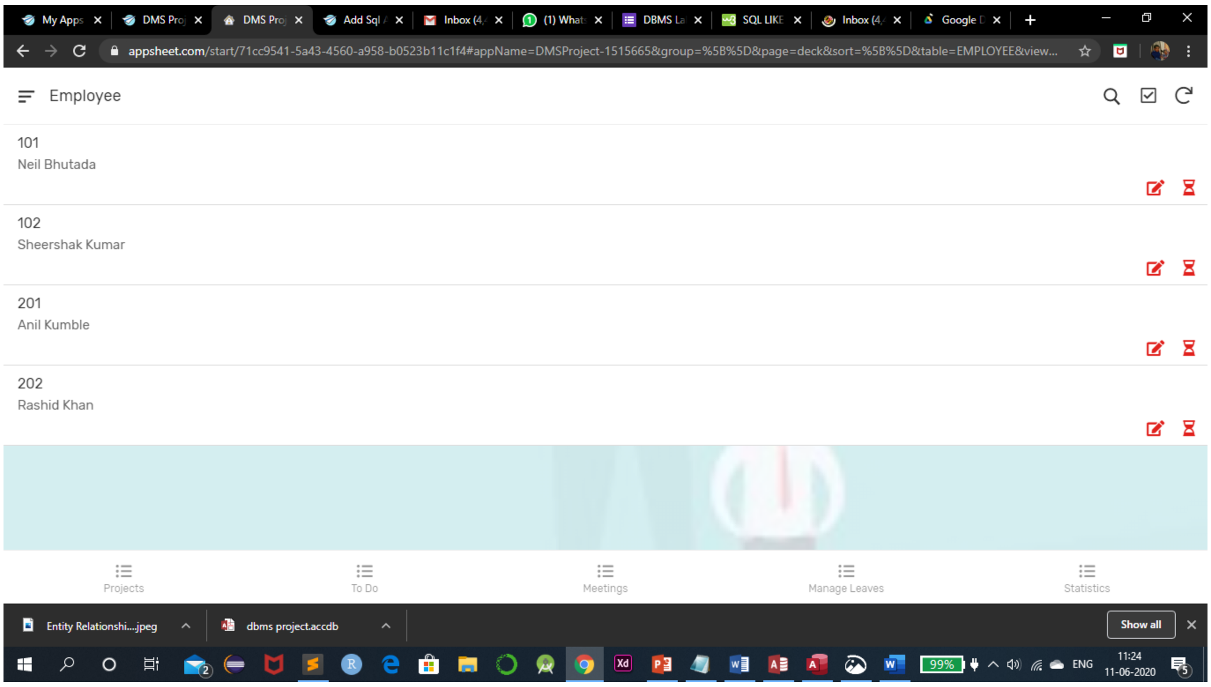
```
ajax({type:"POST",url:"/api/notif/settoken/",data:{token:e,whitelabelGuid:n
,isWhitelabel:t},success:function(){i.default.log("uploaded token for
notification")}
```

FULL IMPLEMENTATION FRONT AND BACKEND

Our App can successfully run on IOS and Android devices and on various web browsers. Screen shots of our web browser version will be included.

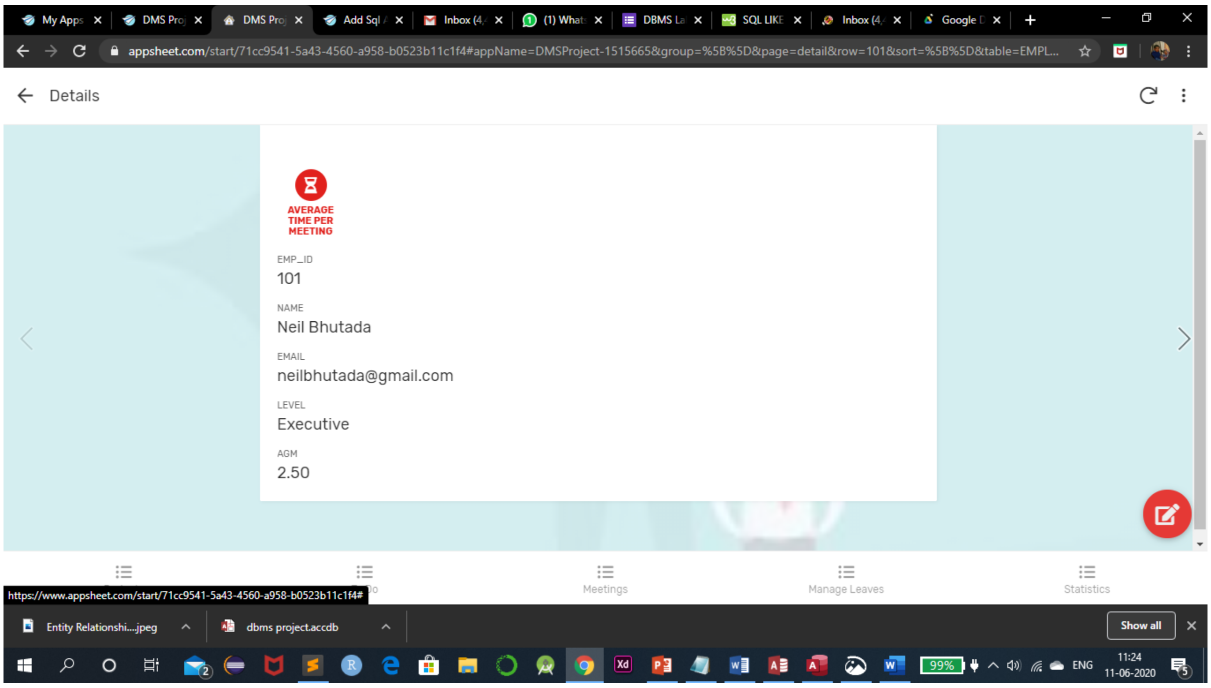


The user can go to the menu bar and would select Employee , the information of all Employees is given. Please note that the Employees in the Employee table cannot be deleted or added by the user. Only updates to the information can be made.



Employee

101	Neil Bhutada
102	Sheershak Kumar
201	Anil Kumble
202	Rashid Khan



Details

AVERAGE TIME PER MEETING

EMP_ID
101

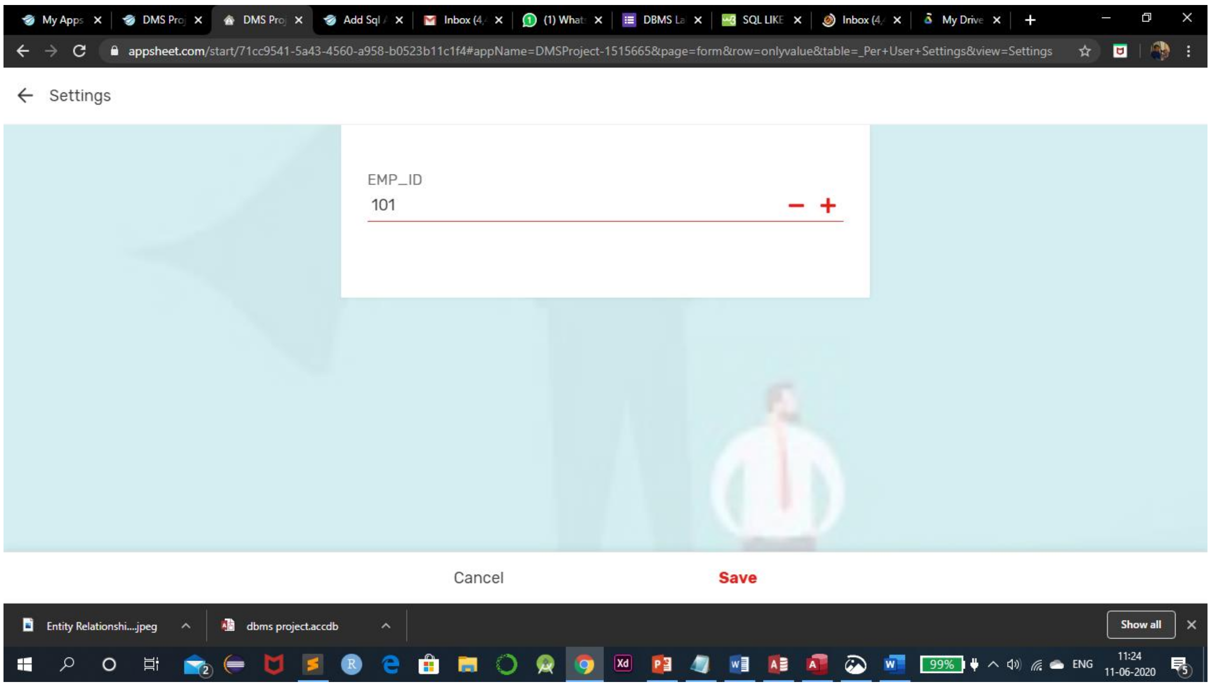
NAME
Neil Bhutada

EMAIL
neilbhutada@gmail.com

LEVEL
Executive

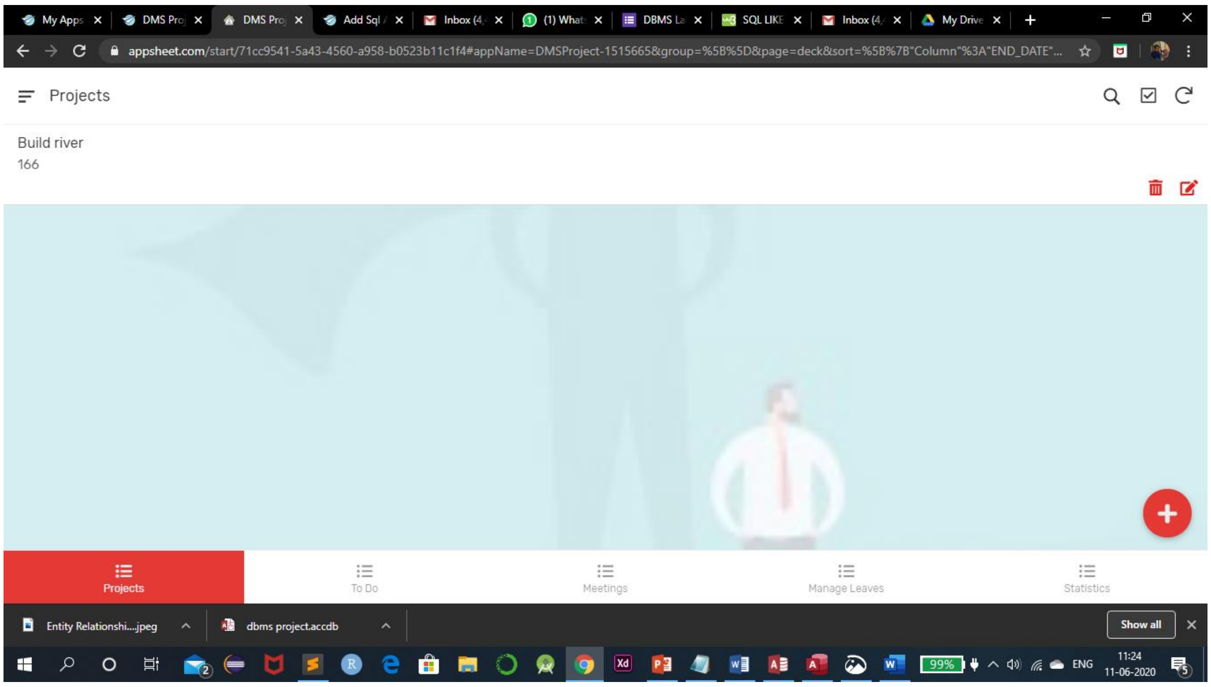
AGM
2.50

The user can re-calculate the Average Time spent per meeting by clicking that half-filed glass icon.



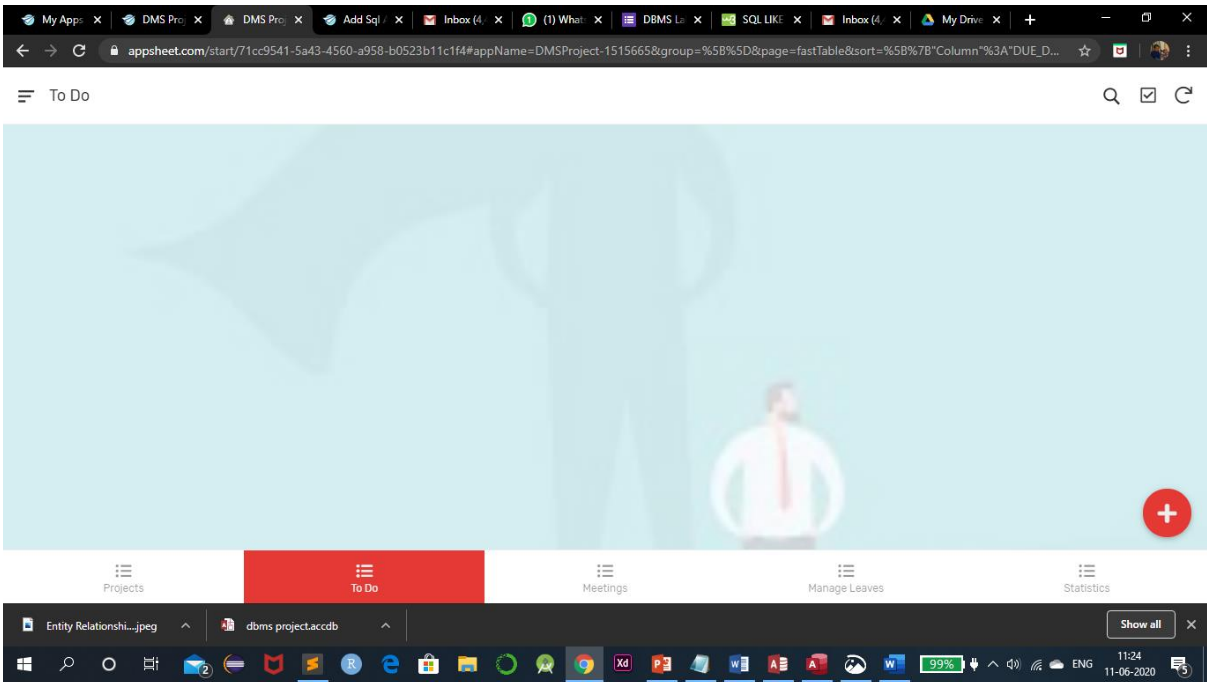
EMP_ID
101

The user can slice values in the app by entering their respective EMP_ID by going to settings in the menu – bar.

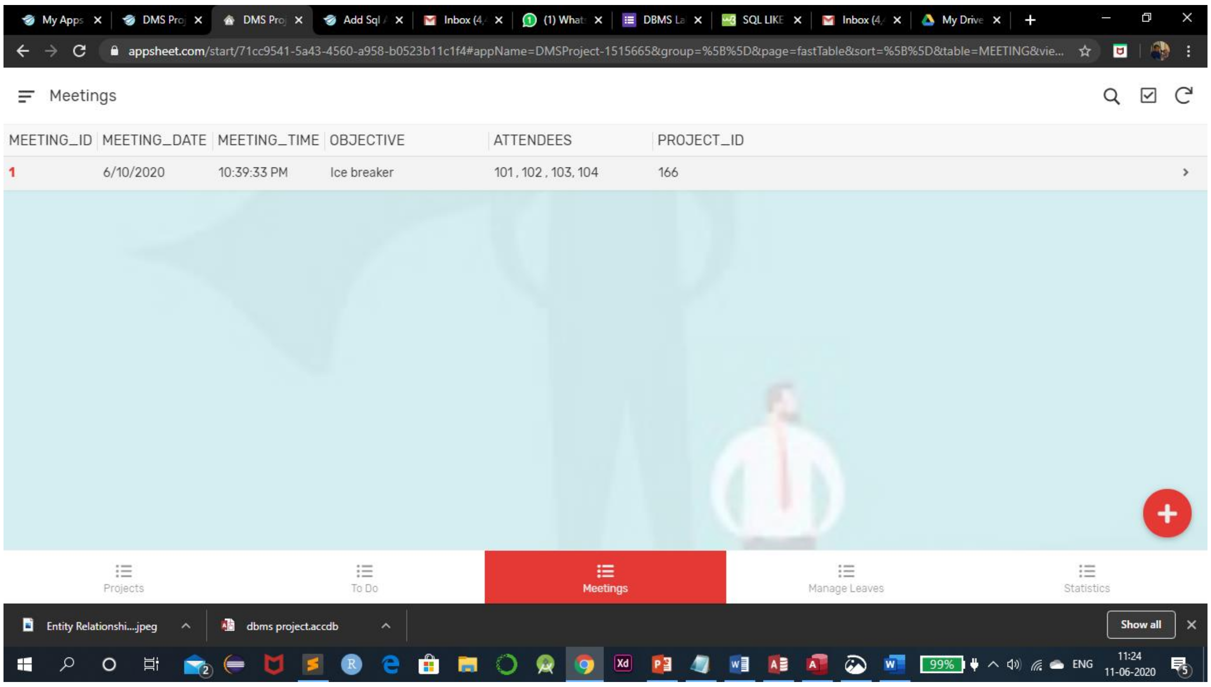


Build river
166

In this tab the user can add/delete/update projects.

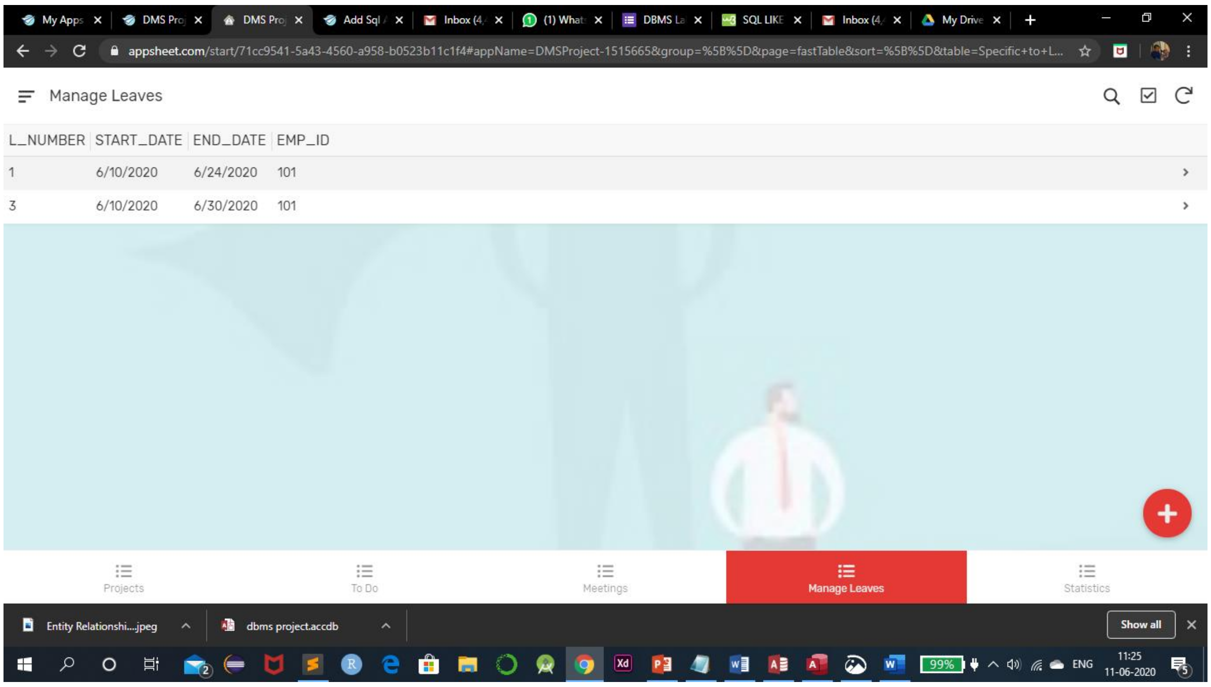


In this tab users can enter Tasks to do and they will receive emails when time approaches.



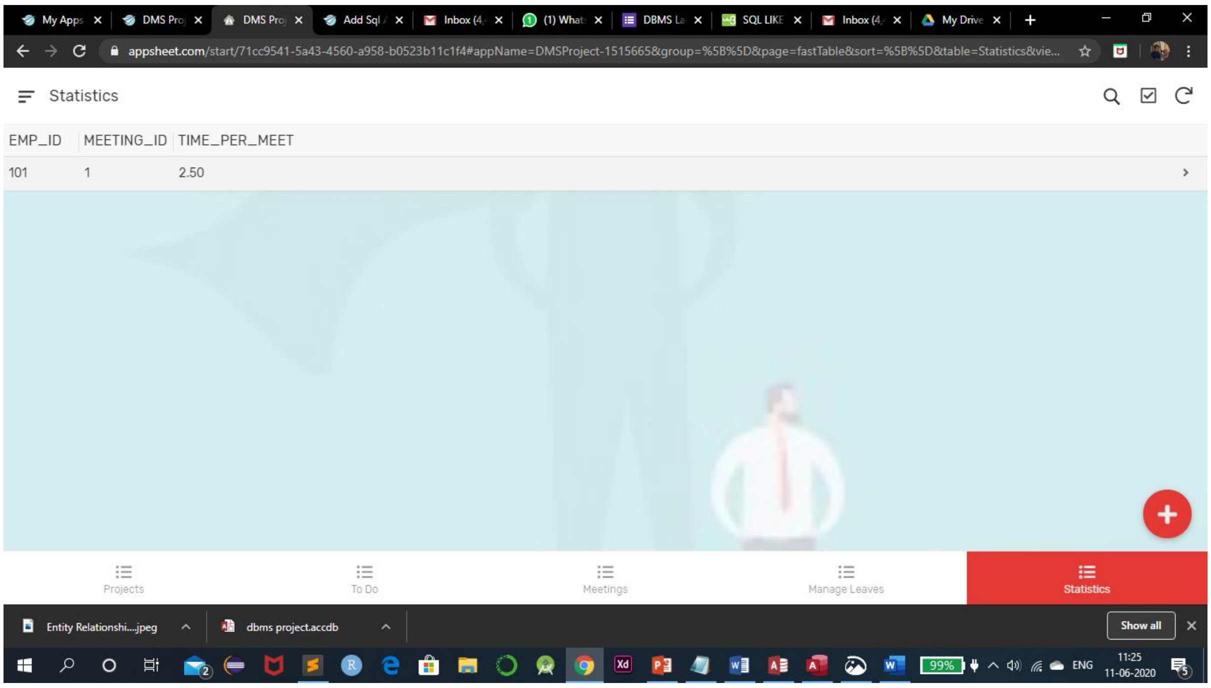
MEETING_ID	MEETING_DATE	MEETING_TIME	OBJECTIVE	ATTENDEES	PROJECT_ID
1	6/10/2020	10:39:33 PM	Ice breaker	101 , 102 , 103, 104	166

Similarly this is the meetings tab.



L_NUMBER	START_DATE	END_DATE	EMP_ID
1	6/10/2020	6/24/2020	101
3	6/10/2020	6/30/2020	101

Here the user can manage their leaves.



EMP_ID	MEETING_ID	TIME_PER_MEET
101	1	2.50

And finally the user can various statistics about the meetings conducted such as time spent per meeting.

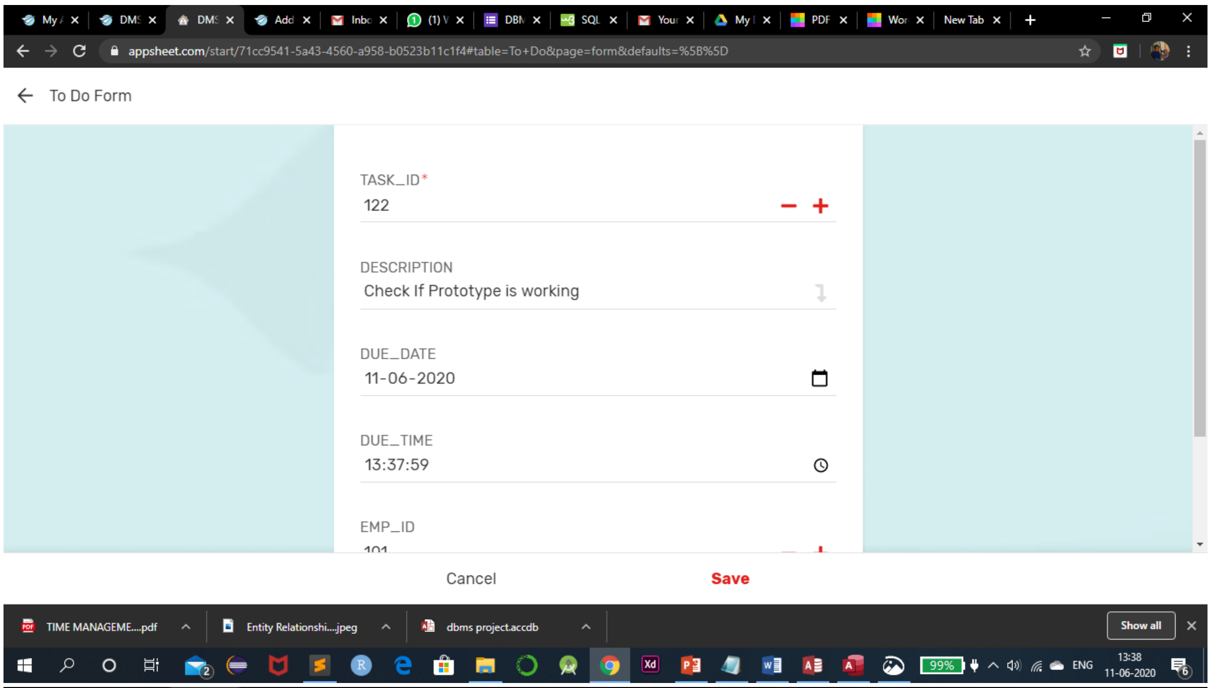
While creating JSON objects for sending and receiving data for particular rows and column in the database. The software refers to a particular column in the following manner [_THISROW][Column Name] where [_THISROW] refers to the row selected in the UI.

AppSheet allows us to use formulas to give functionality to slices and perform actions such as sending emails.

Source code of UI in HTML and Java Script – Click for your reference.

APP WORKING IN REAL TIME

For this purpose, I will insert an item in the To Do section and it be reflected in the TODO table in the MySQL Cloud Database.



To Do Form

TASK_ID*
122

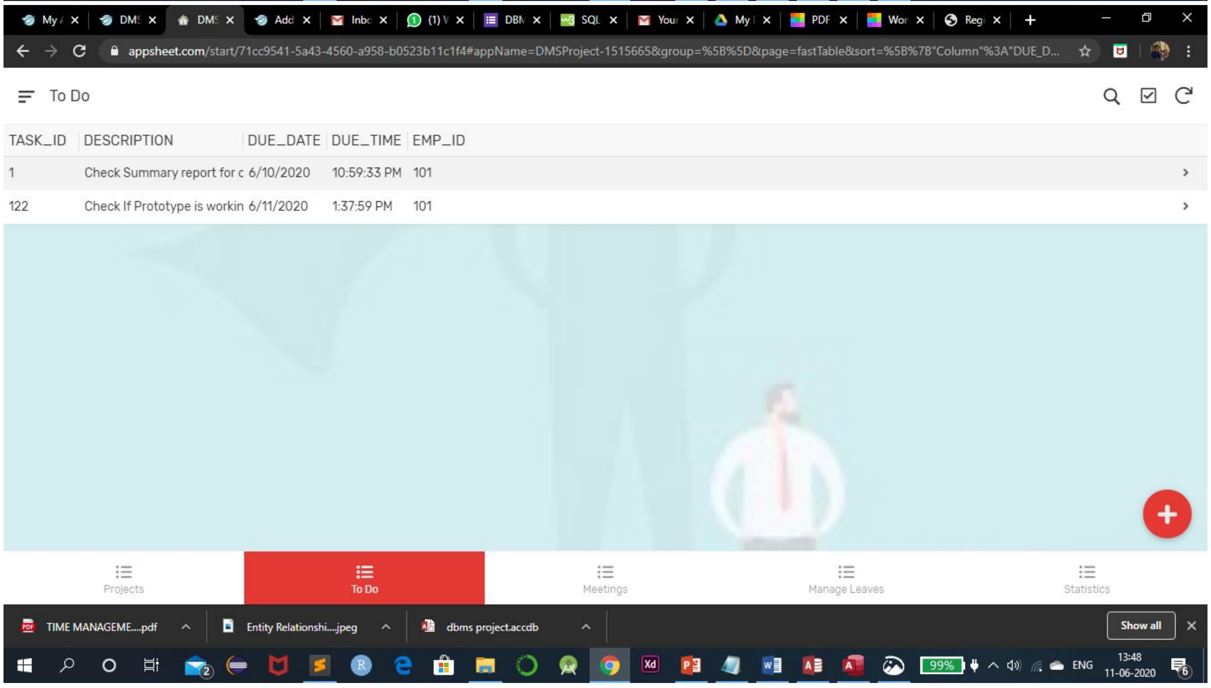
DESCRIPTION
Check If Prototype is working

DUE_DATE
11-06-2020

DUE_TIME
13:37:59

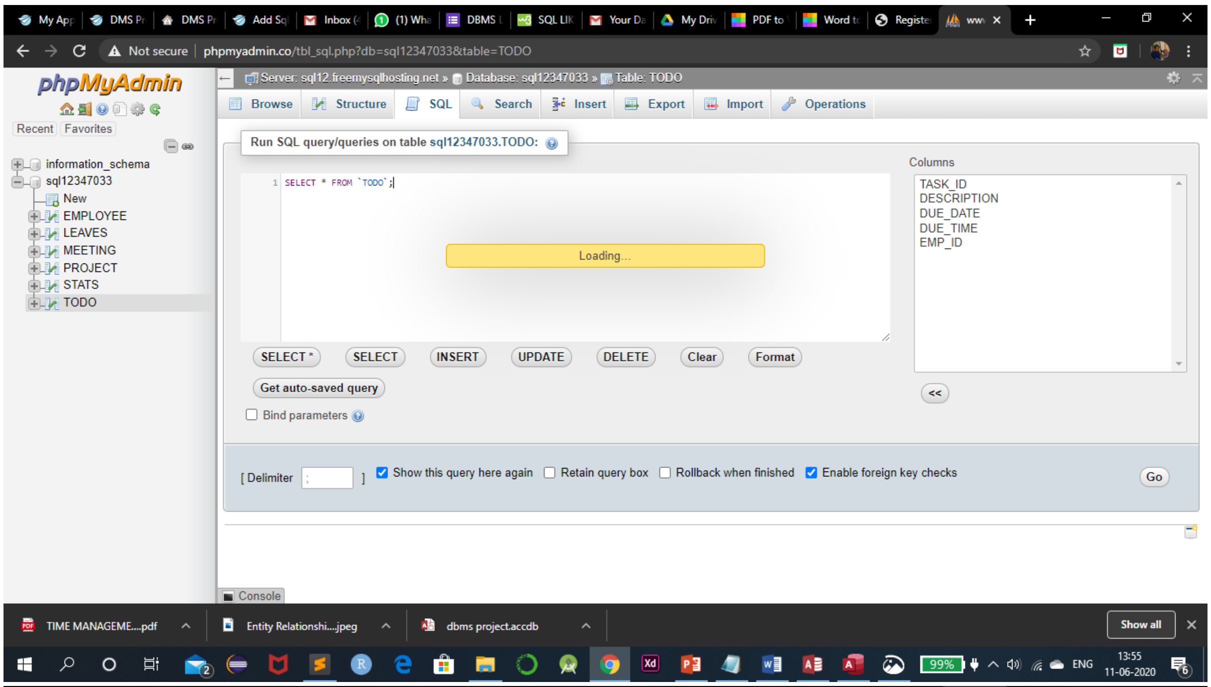
EMP_ID
101

Cancel Save



To Do

TASK_ID	DESCRIPTION	DUE_DATE	DUE_TIME	EMP_ID
1	Check Summary report for c	6/10/2020	10:59:33 PM	101
122	Check If Prototype is workin	6/11/2020	1:37:59 PM	101


Server: sql12.freemysqlhosting.net » Database: sql12347033 » Table: TODO

Run SQL query/queries on table sql12347033.TODO:

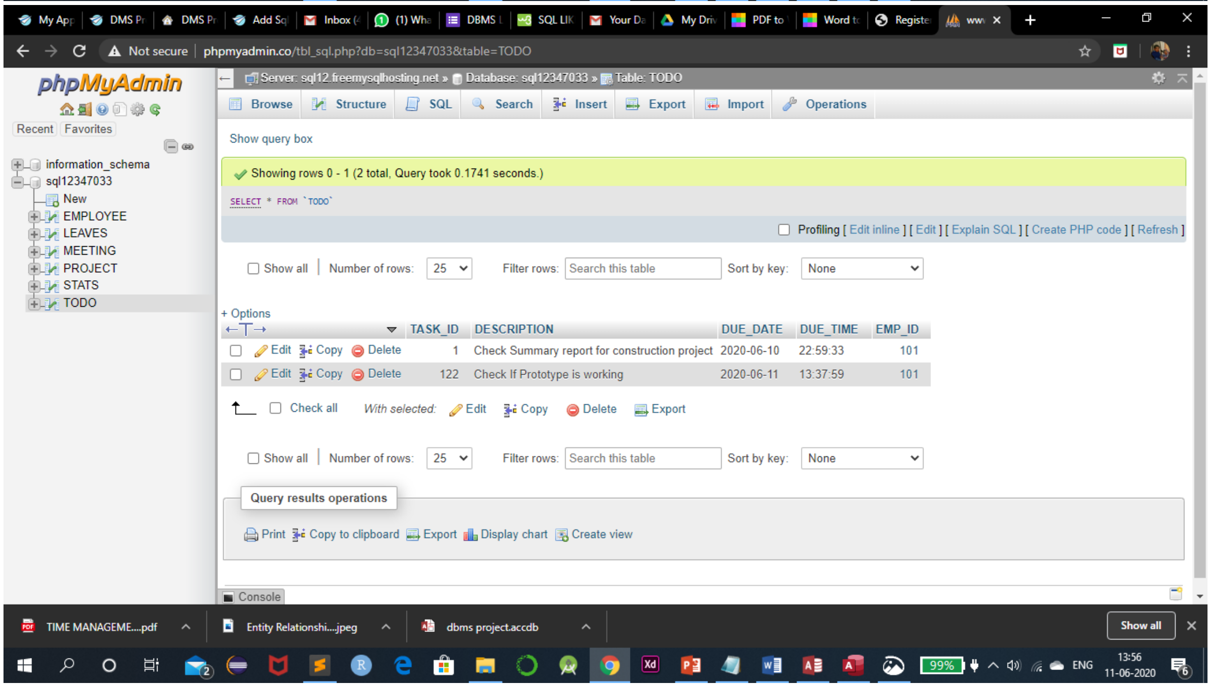
```
SELECT * FROM `TODO`;
```

Loading...

Columns

TASK_ID
DESCRIPTION
DUE_DATE
DUE_TIME
EMP_ID

[Delimiter ;] [✓ Show this query here again] Retain query box, Rollback when finished, [✓ Enable foreign key checks]


Server: sql12.freemysqlhosting.net » Database: sql12347033 » Table: TODO

Showing rows 0 - 1 (2 total, Query took 0.1741 seconds.)

```
SELECT * FROM `TODO`
```

TASK_ID	DESCRIPTION	DUE_DATE	DUE_TIME	EMP_ID
1	Check Summary report for construction project	2020-06-10	22:59:33	101
122	Check If Prototype is working	2020-06-11	13:37:59	101

To install the App on your mobile device click on this link

To open the App on your web browser click this link