DBMS MINI PROJECT REPORT

Account Management System for a Construction Company

Submitted By:

Name: Bharath D H

SRN: PES2UG20CS086

Section: B

SHORT DESCRIPTION AND SCOPE OF PROJECT

This is an Accounts Management application dedicated to a Construction Company. This helps to maintain records of the Company's transaction within and outside the company.

Employees of the company can login and check the transaction details between them and the company like advance payment received, balance payment and so on.

The Owner of the Company has the admin permissions to access and update the database. The Admin dashboard will have transaction details related to its employees as well as Clients/Architects. Employee details will be sorted based on their roles in the company. This application helps the Company to maintain the data of the company in a structured format.

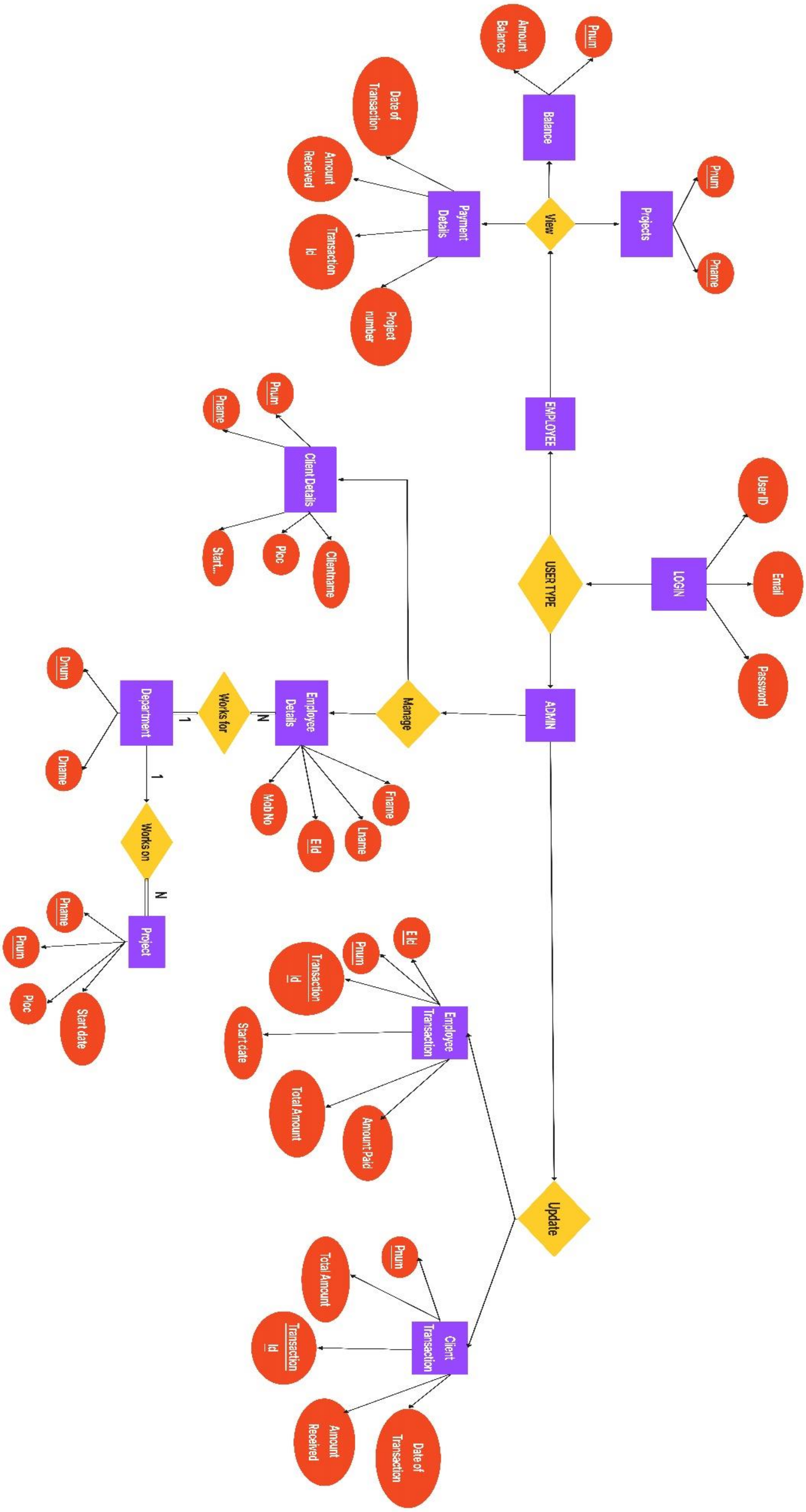
Features:

- Register/Login portal for Employees and Admin
- Manage Transactions of both Clients and Employess
- Employees can view their transactions, balance and projects
- Admin can update transactions and delete employees
- Admin can insert details about projects and departments

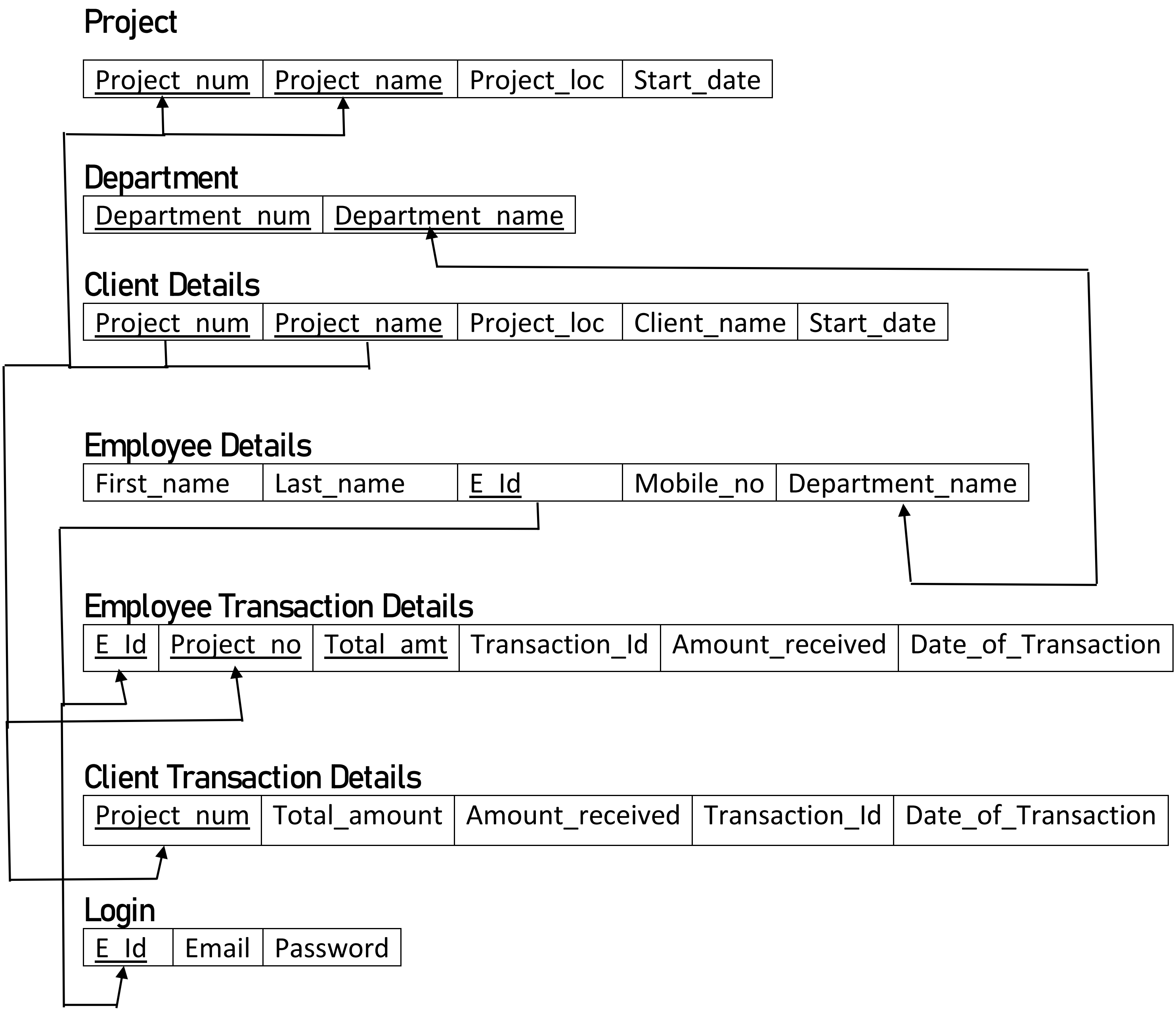
Scope:

This application helps Construction Companies to manage their transactions hassle free. Employees can view all the transactions and balance amount for each project at a single place.

ER DIAGRAM


LOGIN
User ID
Email
Password
USER TYPE
EMPLOYEE
ADMIN
View
Balance
Amount Balance
Pnum
Projects
Pnum
Pname
Payment Details
Date of Transaction
Amount Received
Transaction Id
Project number
Client Details
Pnum
Pname
Start...
Ploc
Clientname
Manage
Employee Details
Fname
Lname
E Id
Mob No
Works for
N
1
Department
Dnum
Dname
1
Works on
N
Project
Pname
Pnum
Ploc
Start date
Update
Employee Transaction
E Id
Pnum
Transaction id
Start date
Total Amount
Amount Paid
Client Transaction
Pnum
Total Amount
Transaction Id
Amount Received
Date of Transaction

RELATIONAL SCHEMA



Project

Project_num	Project_name	Project_loc	Start_date

Department

Department_num	Department_name

Client Details

Project_num	Project_name	Project_loc	Client_name	Start_date

Employee Details

First_name	Last_name	E_Id	Mobile_no	Department_name

Employee Transaction Details

E_Id	Project_no	Total_amt	Transaction_Id	Amount_received	Date_of_Transaction

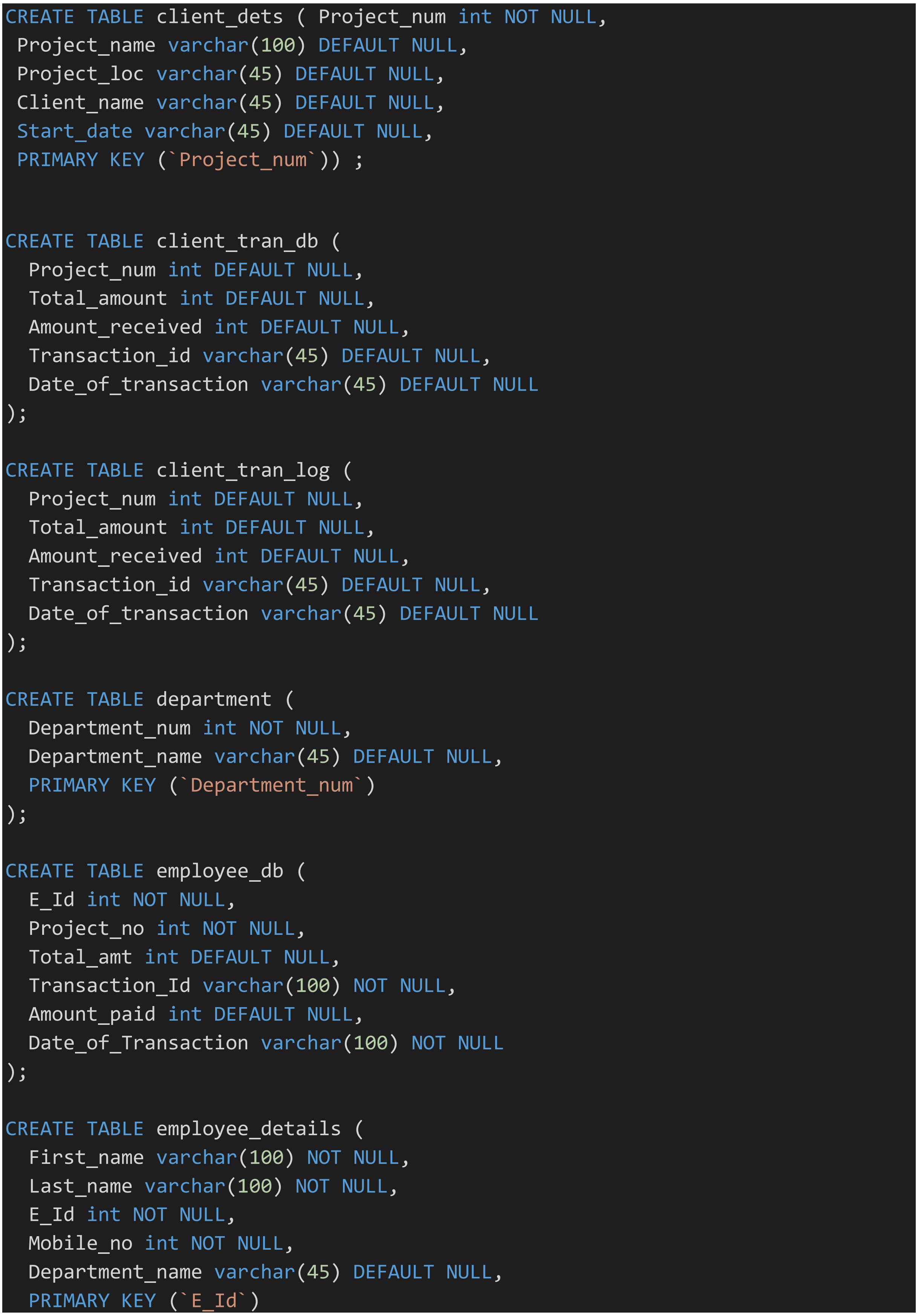
Client Transaction Details

Project_num	Total_amount	Amount_received	Transaction_Id	Date_of_Transaction

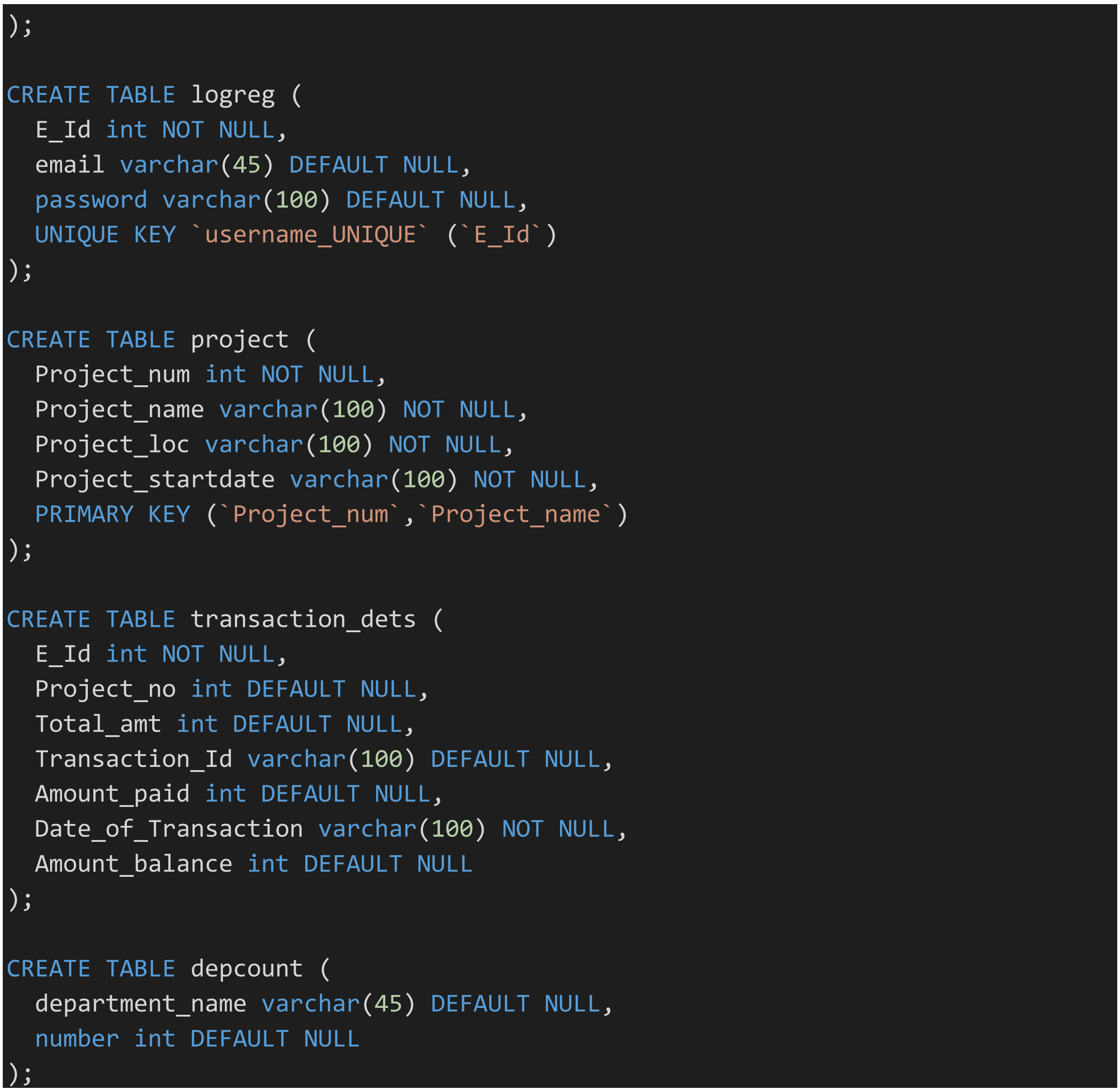
Login

E_Id	Email	Password

DDL STATEMENTS:

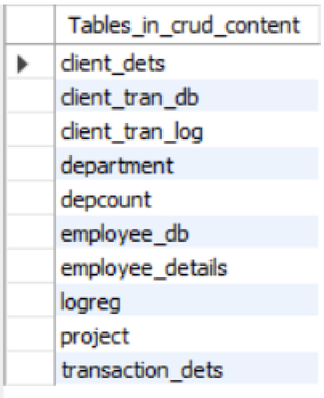


```sql
CREATE TABLE client_dets ( Project_num int NOT NULL,
 Project_name varchar(100) DEFAULT NULL,
 Project_loc varchar(45) DEFAULT NULL,
 Client_name varchar(45) DEFAULT NULL,
 Start_date varchar(45) DEFAULT NULL,
 PRIMARY KEY (`Project_num`)) ;


CREATE TABLE client_tran_db (
  Project_num int DEFAULT NULL,
  Total_amount int DEFAULT NULL,
  Amount_received int DEFAULT NULL,
  Transaction_id varchar(45) DEFAULT NULL,
  Date_of_transaction varchar(45) DEFAULT NULL
);

CREATE TABLE client_tran_log (
  Project_num int DEFAULT NULL,
  Total_amount int DEFAULT NULL,
  Amount_received int DEFAULT NULL,
  Transaction_id varchar(45) DEFAULT NULL,
  Date_of_transaction varchar(45) DEFAULT NULL
);

CREATE TABLE department (
  Department_num int NOT NULL,
  Department_name varchar(45) DEFAULT NULL,
  PRIMARY KEY (`Department_num`)
);

CREATE TABLE employee_db (
  E_Id int NOT NULL,
  Project_no int NOT NULL,
  Total_amt int DEFAULT NULL,
  Transaction_Id varchar(100) NOT NULL,
  Amount_paid int DEFAULT NULL,
  Date_of_Transaction varchar(100) NOT NULL
);

CREATE TABLE employee_details (
  First_name varchar(100) NOT NULL,
  Last_name varchar(100) NOT NULL,
  E_Id int NOT NULL,
  Mobile_no int NOT NULL,
  Department_name varchar(45) DEFAULT NULL,
  PRIMARY KEY (`E_Id`)
```



```sql
);

CREATE TABLE logreg (
  E_Id int NOT NULL,
  email varchar(45) DEFAULT NULL,
  password varchar(100) DEFAULT NULL,
  UNIQUE KEY `username_UNIQUE` (`E_Id`)
);

CREATE TABLE project (
  Project_num int NOT NULL,
  Project_name varchar(100) NOT NULL,
  Project_loc varchar(100) NOT NULL,
  Project_startdate varchar(100) NOT NULL,
  PRIMARY KEY (`Project_num`,`Project_name`)
);

CREATE TABLE transaction_dets (
  E_Id int NOT NULL,
  Project_no int DEFAULT NULL,
  Total_amt int DEFAULT NULL,
  Transaction_Id varchar(100) DEFAULT NULL,
  Amount_paid int DEFAULT NULL,
  Date_of_Transaction varchar(100) NOT NULL,
  Amount_balance int DEFAULT NULL
);

CREATE TABLE depcount (
  department_name varchar(45) DEFAULT NULL,
  number int DEFAULT NULL
);
```

OUTPUT:

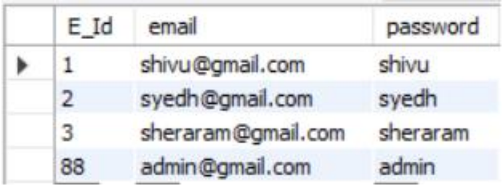


```
Tables_in_crud_content
client_dets
client_tran_db
client_tran_log
department
depcount
employee_db
employee_details
logreg
project
transaction_dets
```

Populating the Database

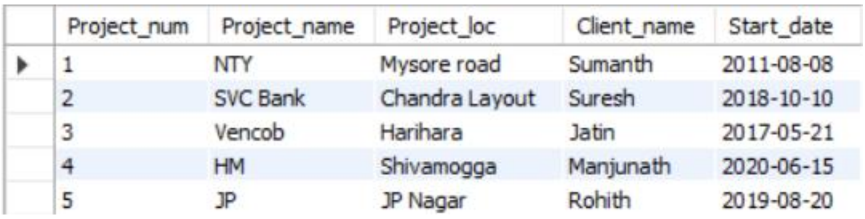
- Admin has the access to add employee details, client details, update transactions. All these operations are done with the help of frontend.

- **Login**



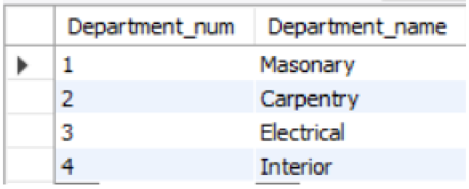
E_Id	email	password
1	shivu@gmail.com	shivu
2	syedh@gmail.com	syedh
3	sheraram@gmail.com	sheraram
88	admin@gmail.com	admin

- **Client Details**



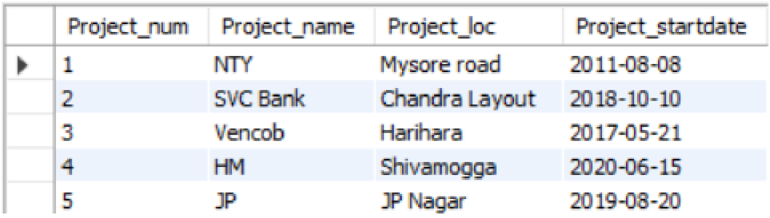
Project_num	Project_name	Project_loc	Client_name	Start_date
1	NTY	Mysore road	Sumanth	2011-08-08
2	SVC Bank	Chandra Layout	Suresh	2018-10-10
3	Vencob	Harihara	Jatin	2017-05-21
4	HM	Shivamogga	Manjunath	2020-06-15
5	JP	JP Nagar	Rohith	2019-08-20

- **Department**



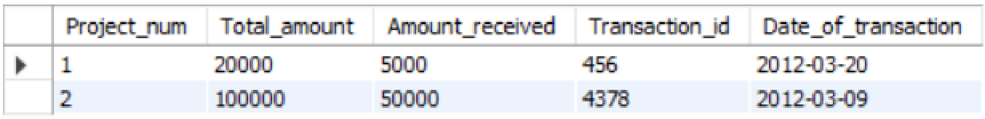
Department_num	Department_name
1	Masonary
2	Carpentry
3	Electrical
4	Interior

- **Project**

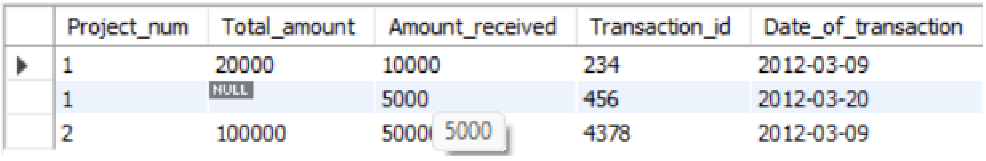


Project_num	Project_name	Project_loc	Project_startdate
1	NTY	Mysore road	2011-08-08
2	SVC Bank	Chandra Layout	2018-10-10
3	Vencob	Harihara	2017-05-21
4	HM	Shivamogga	2020-06-15
5	JP	JP Nagar	2019-08-20

- ## Client Latest Transaction and Transaction Log

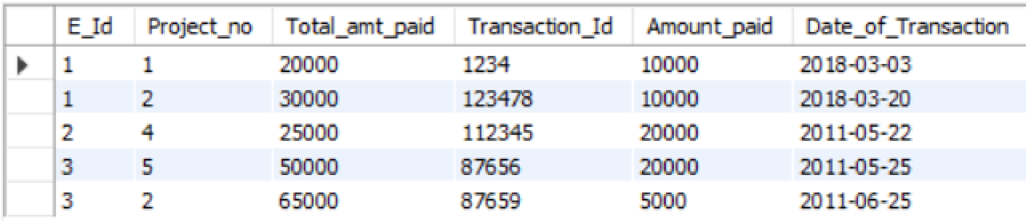


Project_num	Total_amount	Amount_received	Transaction_id	Date_of_transaction
1	20000	5000	456	2012-03-20
2	100000	50000	4378	2012-03-09

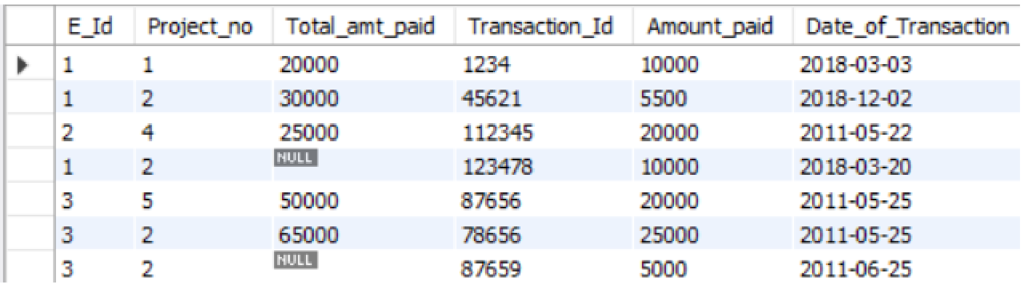


Project_num	Total_amount	Amount_received	Transaction_id	Date_of_transaction
1	20000	10000	234	2012-03-09
1	NULL	5000	456	2012-03-20
2	100000	5000(5000	4378	2012-03-09

- ## Employee Latest Transaction and Transaction Log

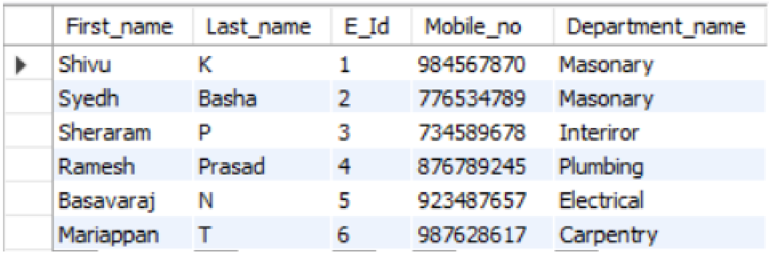


E_Id	Project_no	Total_amt_paid	Transaction_Id	Amount_paid	Date_of_Transaction
1	1	20000	1234	10000	2018-03-03
1	2	30000	123478	10000	2018-03-20
2	4	25000	112345	20000	2011-05-22
3	5	50000	87656	20000	2011-05-25
3	2	65000	87659	5000	2011-06-25



E_Id	Project_no	Total_amt_paid	Transaction_Id	Amount_paid	Date_of_Transaction
1	1	20000	1234	10000	2018-03-03
1	2	30000	45621	5500	2018-12-02
2	4	25000	112345	20000	2011-05-22
1	2	NULL	123478	10000	2018-03-20
3	5	50000	87656	20000	2011-05-25
3	2	65000	78656	25000	2011-05-25
3	2	NULL	87659	5000	2011-06-25

- ## Employee Details



First_name	Last_name	E_Id	Mobile_no	Department_name
Shivu	K	1	984567870	Masonary
Syedh	Basha	2	776534789	Masonary
Sheraram	P	3	734589678	Interiror
Ramesh	Prasad	4	876789245	Plumbing
Basavaraj	N	5	923487657	Electrical
Mariappan	T	6	987628617	Carpentry

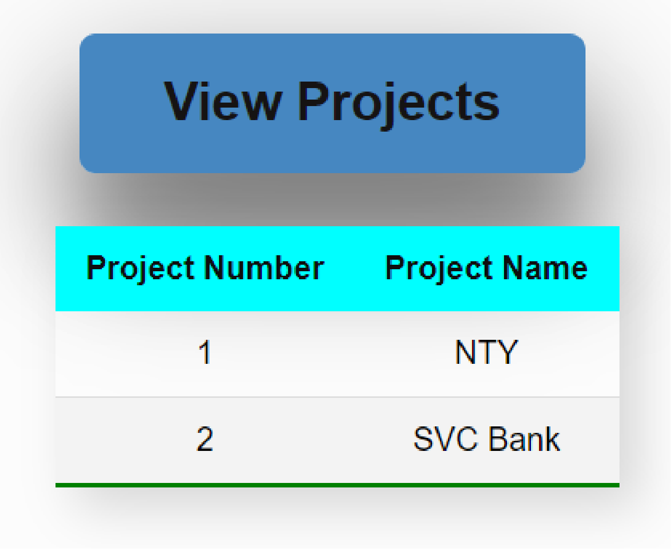
JOIN QUERIES:

```
app.get("/api/usrproj/:E_Id",(req,res)=>{
    const {E_Id} = req.params;
    const sqlGet = "select project_name,project_no from employee_db join
project on Project_no=Project_num where E_Id=?";
    db.query(sqlGet,E_Id,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```

```
app.get("/api/deptran",(req,res)=>{
    var sqlGetdep = "select Department_name,Project_no,sum(Total_amt_paid)as
Totalamt_to_be_paid ,(sum(Total_amt_paid) - Sum(Amount_paid)) as balance from
employee_details natural join transaction_dets group by
Department_name,Project_no;";
    db.query(sqlGetdep,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```

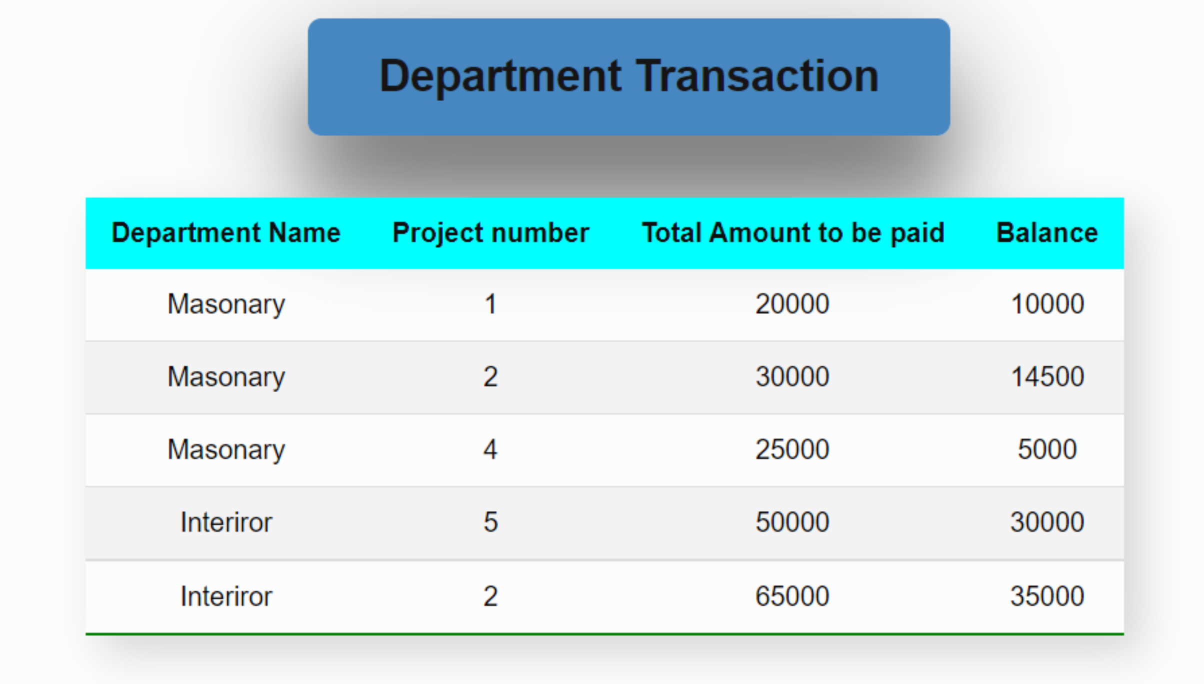
```
app.get("/api/clientdetails",(req,res)=>{
    var sqlGetdep = "select
Project_num,Project_loc,Project_name,Client_name,Start_date,((Total_amount)-
sum(Amount_received)) as balance from client_dets natural join client_tran_log
group by Project_num";
    db.query(sqlGetdep,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```

- View projects on which the Employee is working on



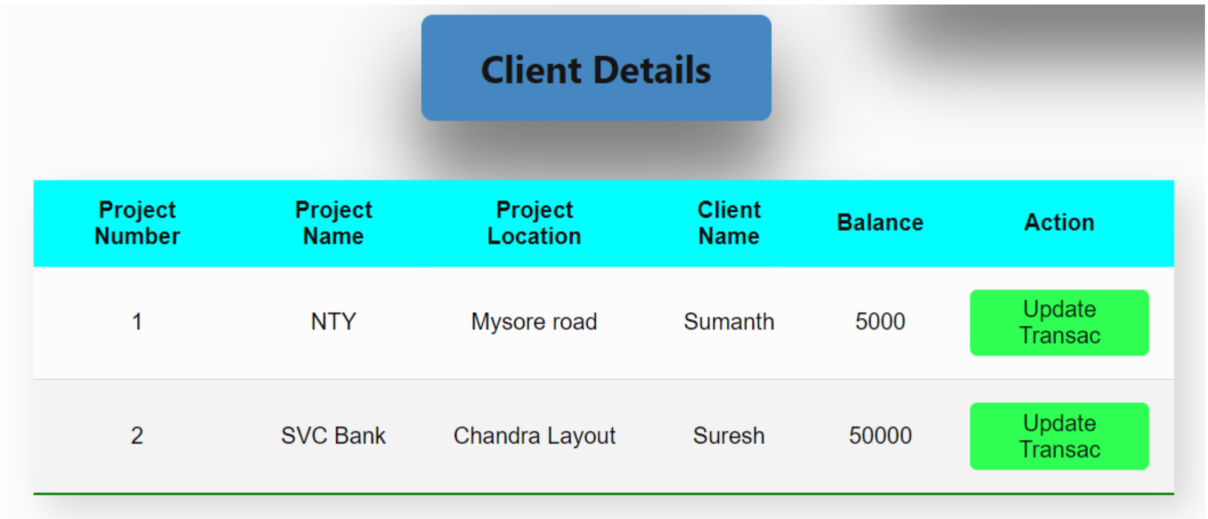
Project Number	Project Name
1	NTY
2	SVC Bank

- View transaction based on the Department



Department Name	Project number	Total Amount to be paid	Balance
Masonary	1	20000	10000
Masonary	2	30000	14500
Masonary	4	25000	5000
Interiror	5	50000	30000
Interiror	2	65000	35000

➢ View Client Details and balance

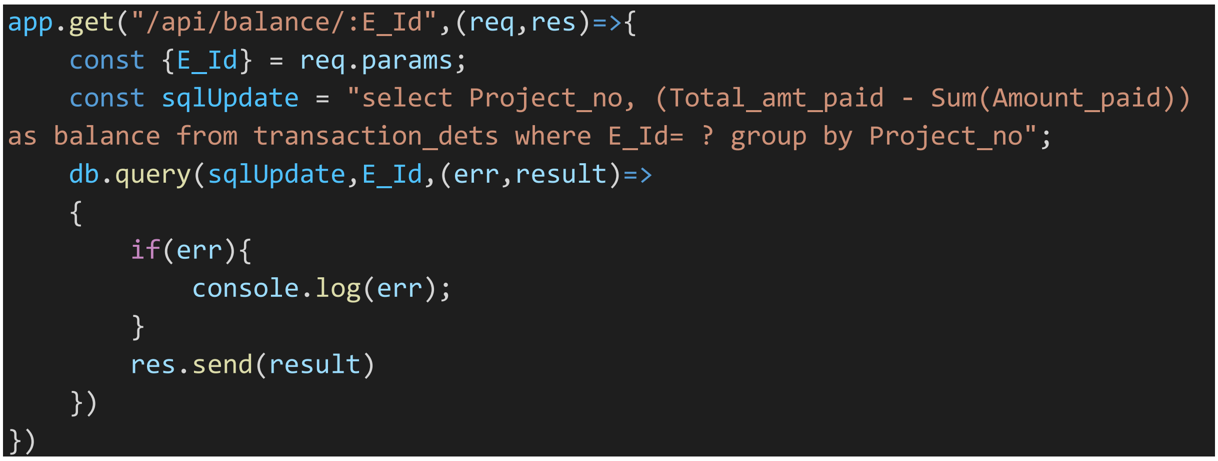


Client Details

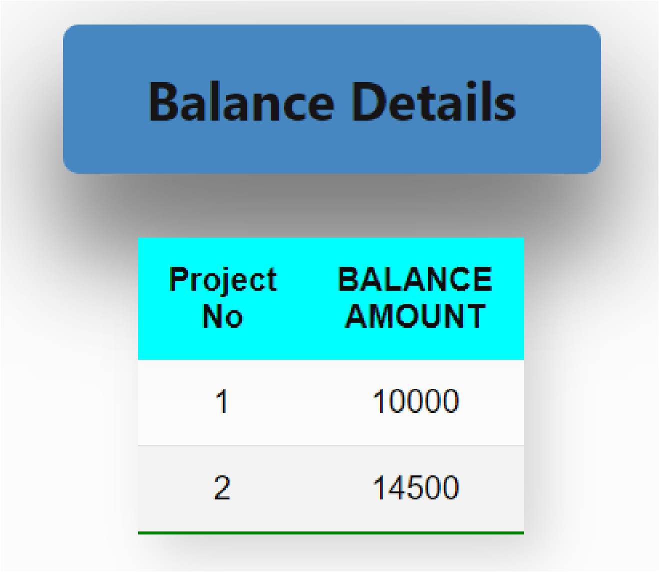
Project Number	Project Name	Project Location	Client Name	Balance	Action
1	NTY	Mysore road	Sumanth	5000	Update Transac
2	SVC Bank	Chandra Layout	Suresh	50000	Update Transac

Aggregate functions:

➢ View balance of employee based on project



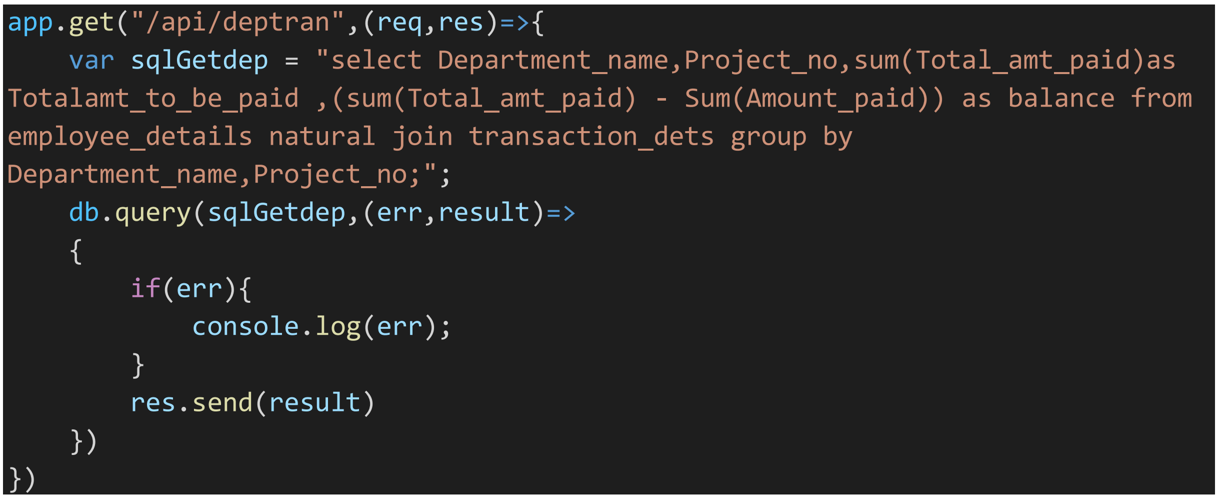
```
app.get("/api/balance/:E_Id",(req,res)=>{
    const {E_Id} = req.params;
    const sqlUpdate = "select Project_no, (Total_amt_paid - Sum(Amount_paid))
as balance from transaction_dets where E_Id= ? group by Project_no";
    db.query(sqlUpdate,E_Id,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```



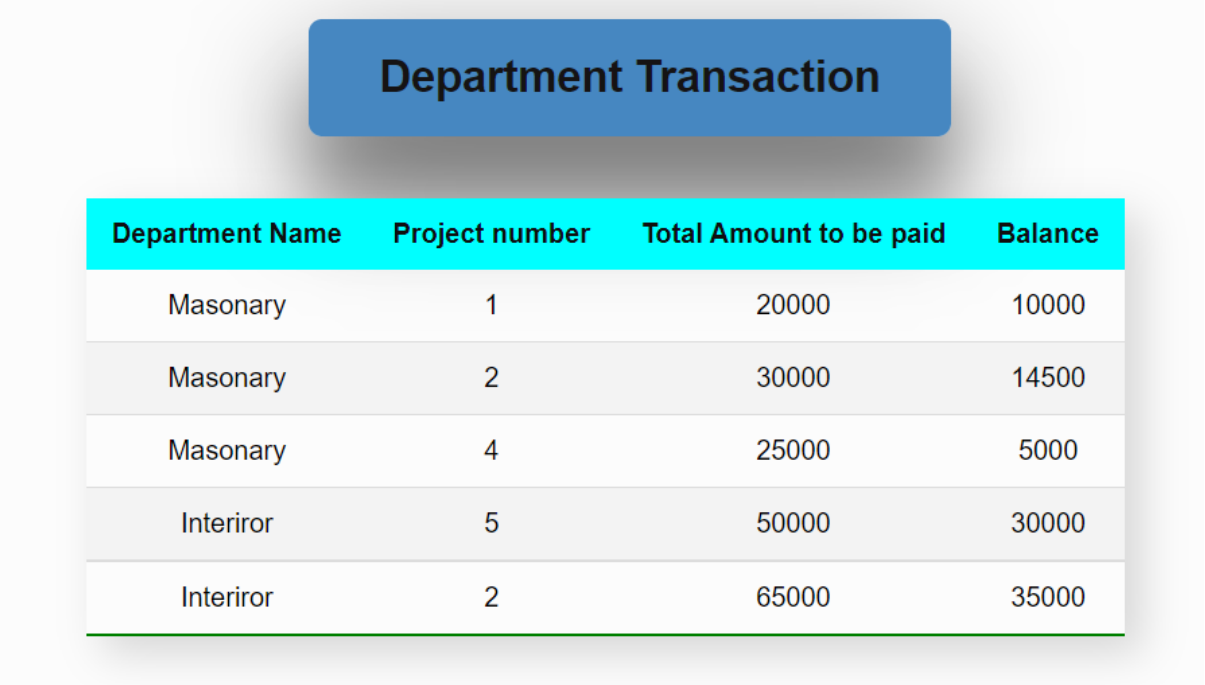
Balance Details

Project No	BALANCE AMOUNT
1	10000
2	14500

- **View transaction based on department**



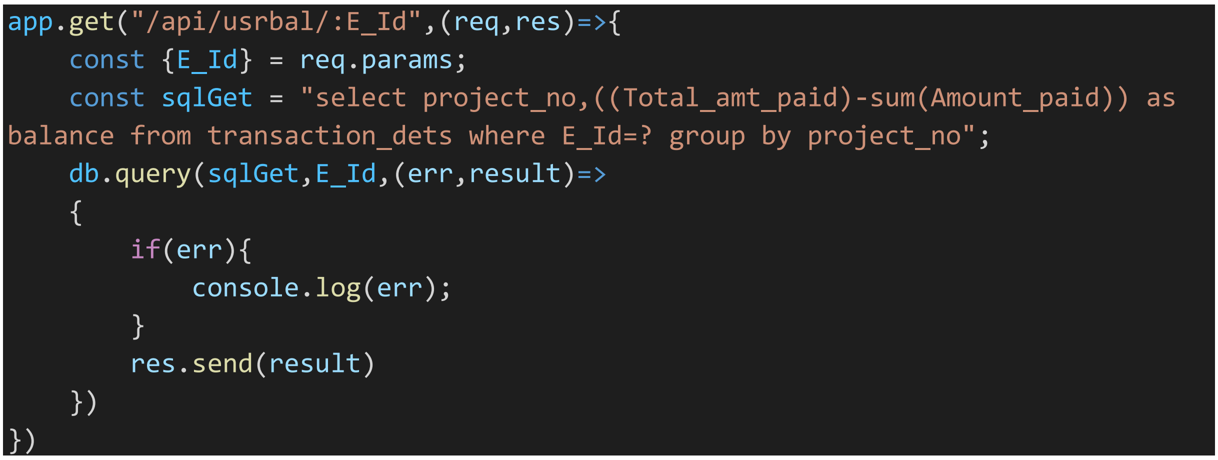
```
app.get("/api/deptran",(req,res)=>{
    var sqlGetdep = "select Department_name,Project_no,sum(Total_amt_paid)as
Totalamt_to_be_paid ,(sum(Total_amt_paid) - Sum(Amount_paid)) as balance from
employee_details natural join transaction_dets group by
Department_name,Project_no;";
    db.query(sqlGetdep,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```



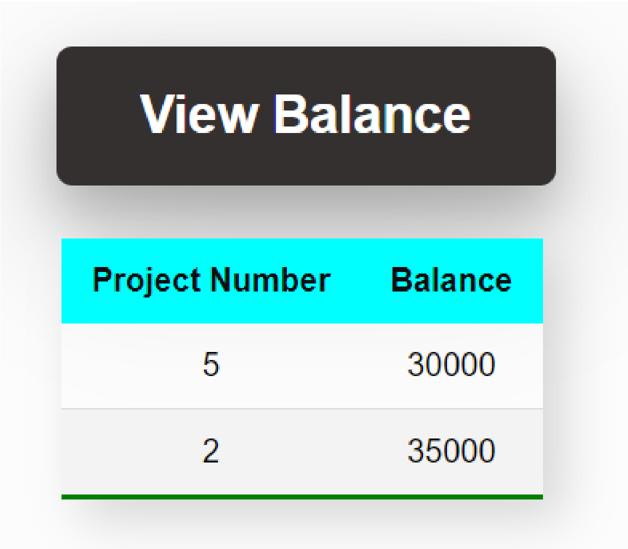
Department Transaction

Department Name	Project number	Total Amount to be paid	Balance
Masonary	1	20000	10000
Masonary	2	30000	14500
Masonary	4	25000	5000
Interiror	5	50000	30000
Interiror	2	65000	35000

- ➢ View Payment balance of user based on project



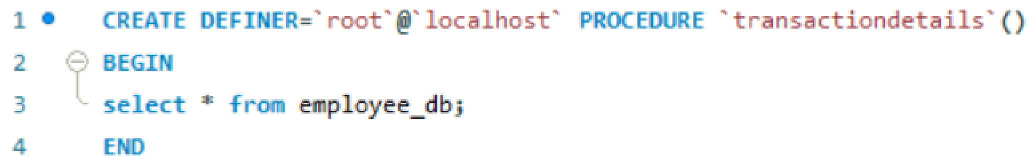
```
app.get("/api/usrbal/:E_Id",(req,res)=>{
    const {E_Id} = req.params;
    const sqlGet = "select project_no,((Total_amt_paid)-sum(Amount_paid)) as 
balance from transaction_dets where E_Id=? group by project_no";
    db.query(sqlGet,E_Id,(err,result)=>
    {
        if(err){
            console.log(err);
        }
        res.send(result)
    })
})
```



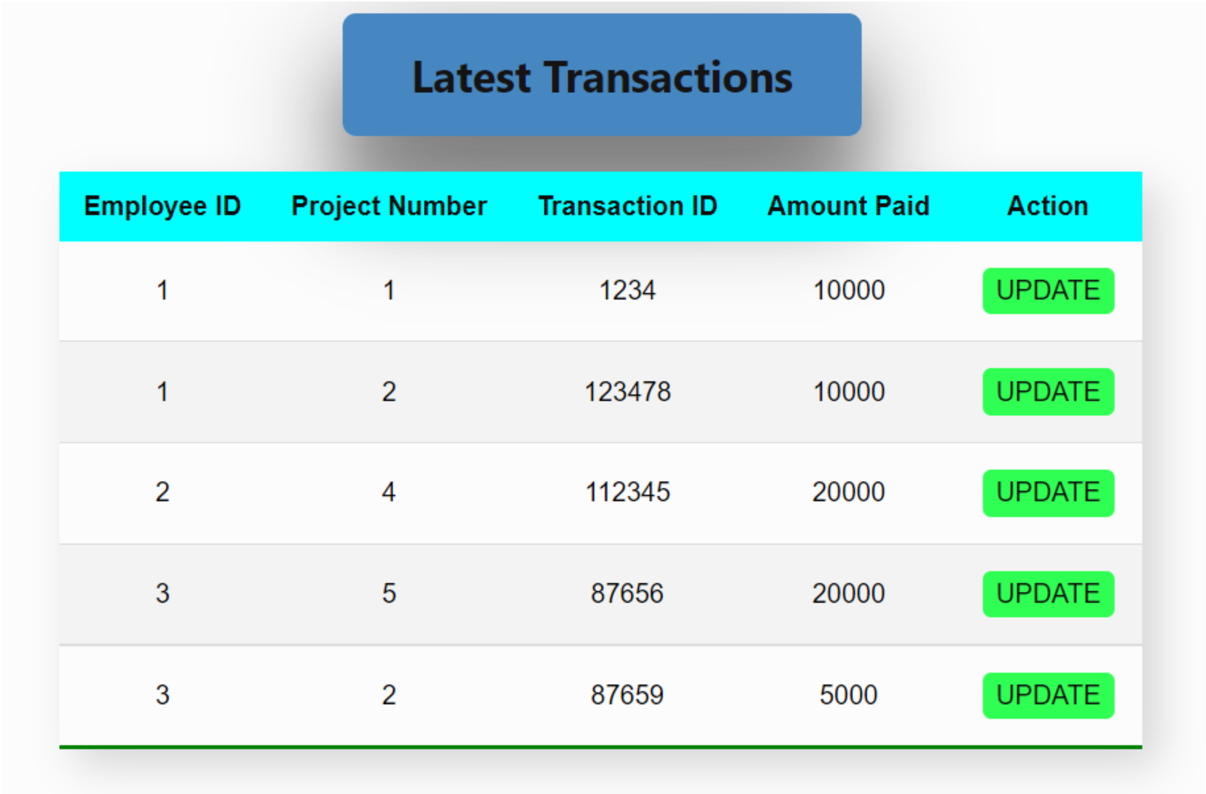
Project Number	Balance
5	30000
2	35000

Functions and Procedures:

- ➢ Procedure to show the latest employee transactions



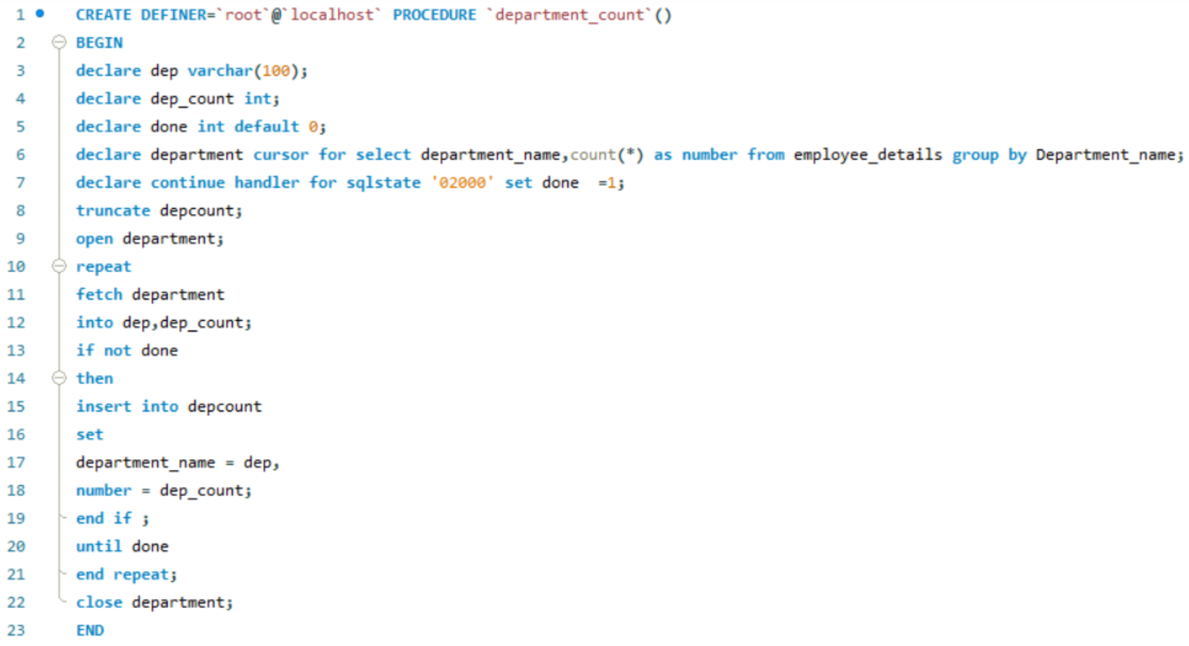
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `transactiondetails`()
BEGIN
select * from employee_db;
END
```



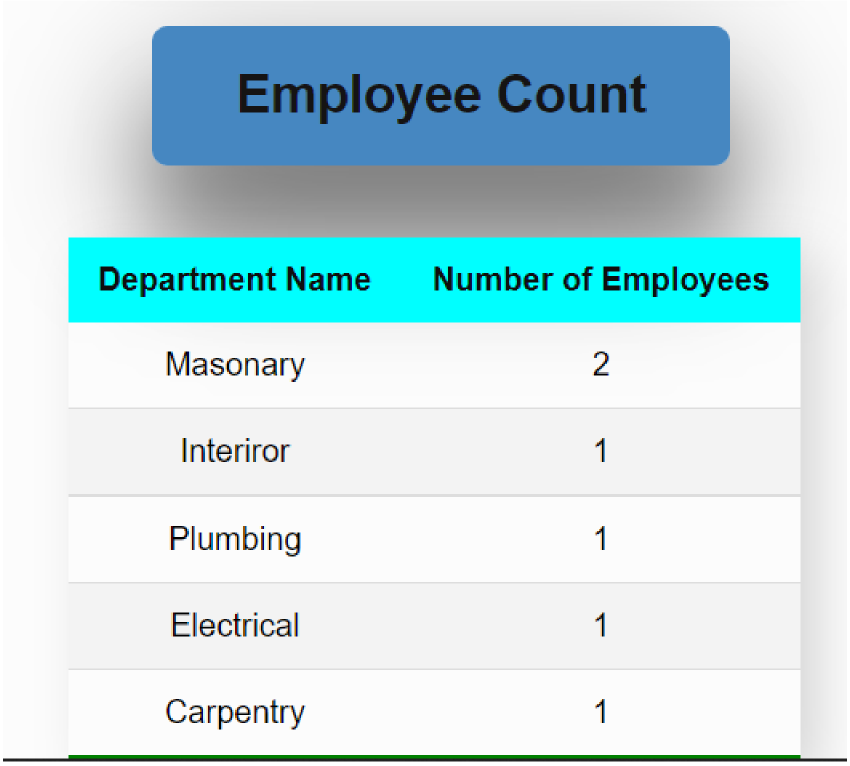
Latest Transactions

Employee ID	Project Number	Transaction ID	Amount Paid	Action
1	1	1234	10000	UPDATE
1	2	123478	10000	UPDATE
2	4	112345	20000	UPDATE
3	5	87656	20000	UPDATE
3	2	87659	5000	UPDATE

- ## Cursor to find the number of employees under each department



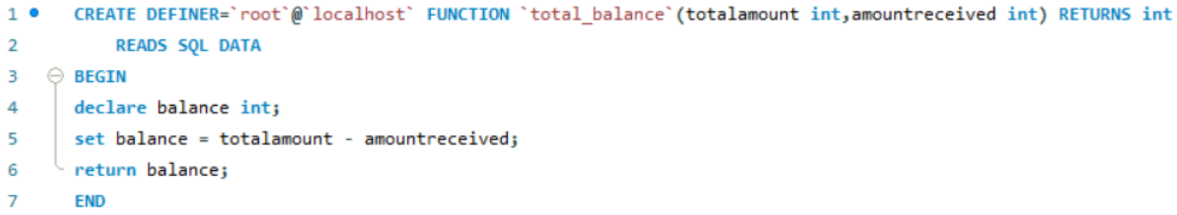
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `department_count`()
BEGIN
declare dep varchar(100);
declare dep_count int;
declare done int default 0;
declare department cursor for select department_name,count(*) as number from employee_details group by Department_name;
declare continue handler for sqlstate '02000' set done  =1;
truncate depcount;
open department;
repeat
fetch department
into dep,dep_count;
if not done
then
insert into depcount
set
department_name = dep,
number = dep_count;
end if ;
until done
end repeat;
close department;
END
```



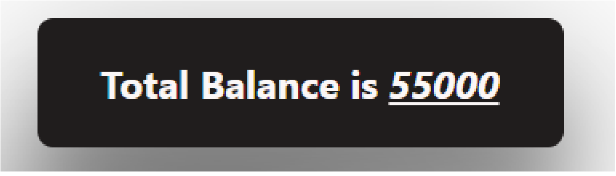
Employee Count

Department Name	Number of Employees
Masonary	2
Interiror	1
Plumbing	1
Electrical	1
Carpentry	1

- ➢ Function to find total balance amount to be received from Clients



```
CREATE DEFINER=`root`@`localhost` FUNCTION `total_balance`(totalamount int,amountreceived int) RETURNS int
    READS SQL DATA
BEGIN
declare balance int;
set balance = totalamount - amountreceived;
return balance;
END
```



Total Balance is *55000*

Triggers:

Inserting the transaction details into another table to maintain a transaction log

- **For maintaining Employee Transaction Log**

```
CREATE DEFINER=`root`@`localhost` TRIGGER `after_employee_db_insert`
AFTER INSERT ON `employee_db`
FOR EACH ROW INSERT
IGNORE into
transaction_dets(E_Id,Project_no,Total_amt_paid,Transaction_Id,Amount_paid,Date_of_Transaction)
values
(new.E_Id,new.Project_no,new.Total_amt_paid,new.Transaction_Id,new.Amount_paid,new.Date_of_Transaction)
```

```
CREATE DEFINER=`root`@`localhost` TRIGGER `after_employee_db_update`
AFTER UPDATE ON `employee_db`
FOR EACH ROW INSERT into
transaction_dets(E_Id,Project_no,Transaction_Id,Amount_paid,Date_of_Transaction)
values
(new.E_Id,new.Project_no,new.Transaction_Id,new.Amount_paid,new.Date_of_Transaction)
```

- **For maintaining Client Transaction Log**

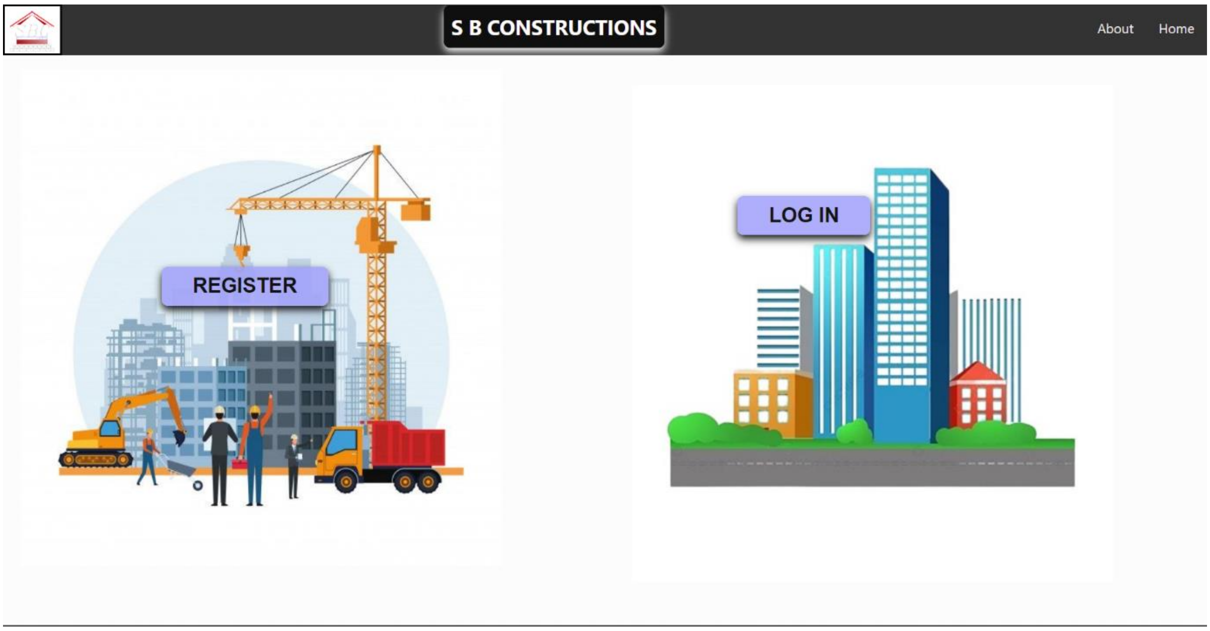
```
CREATE DEFINER=`root`@`localhost` TRIGGER `after_client_dets_insert`
AFTER INSERT ON `client_tran_db`
FOR EACH ROW INSERT
IGNORE into
client_tran_log(Project_num,Total_amount,Amount_received,Transaction_id,Date_of_transaction)
values
(new.Project_num,new.Total_amount,new.Amount_received,new.Transaction_id,new.Date_of_transaction)
```

```
CREATE DEFINER=`root`@`localhost` TRIGGER `after_client_dets_update`
AFTER UPDATE ON `client_tran_db`
FOR EACH ROW insert
```

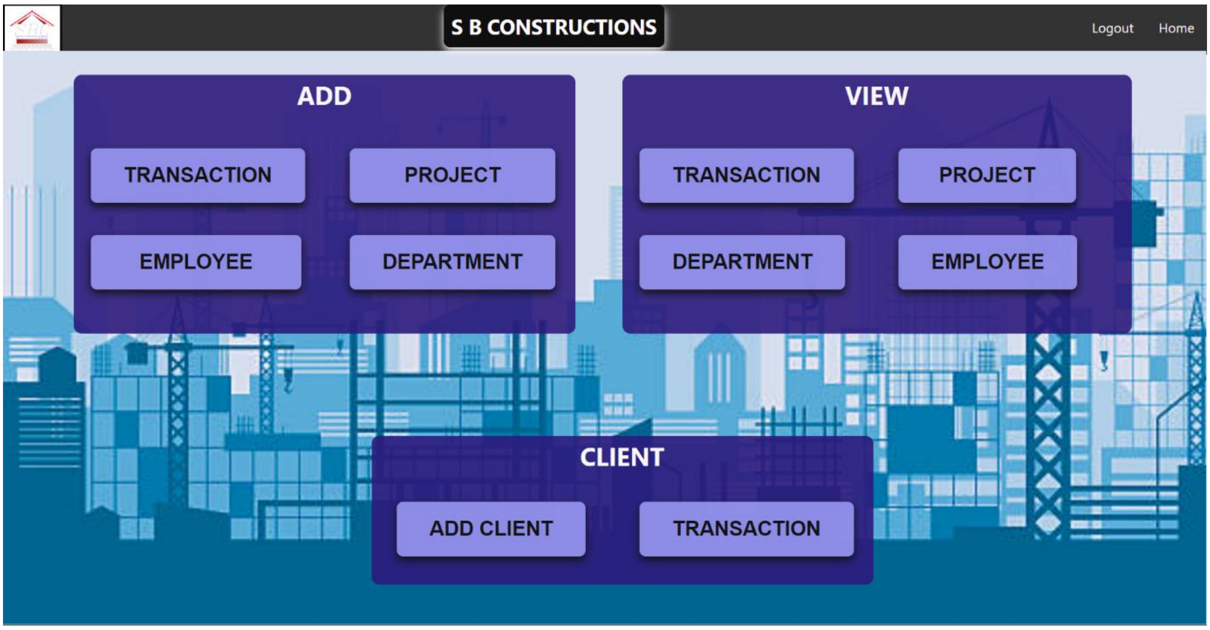
```
ignore into
client_tran_log(Project_num,Amount_received,Transaction_id,Date_of_transaction
)
values
(new.Project_num,new.Amount_received,new.Transaction_id,new.Date_of_transactio
n)
```

FRONTEND

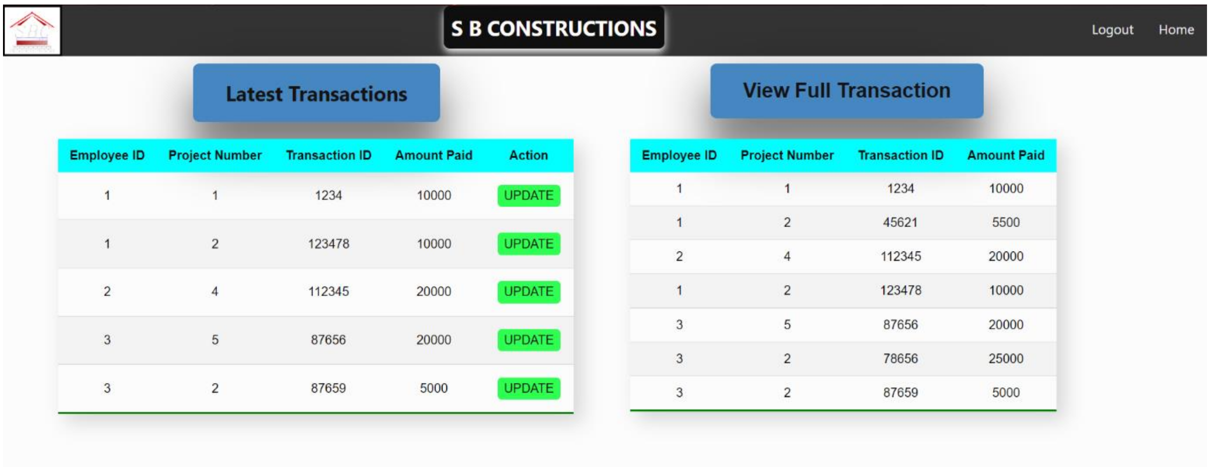
Landing Page



Admin Dashboard



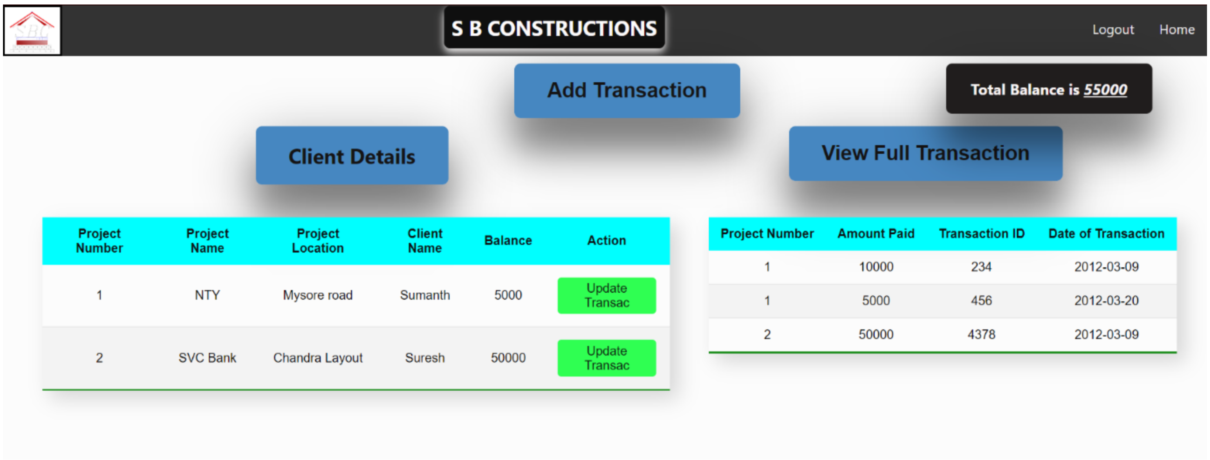
Transaction View



Employee ID	Project Number	Transaction ID	Amount Paid	Action
1	1	1234	10000	UPDATE
1	2	123478	10000	UPDATE
2	4	112345	20000	UPDATE
3	5	87656	20000	UPDATE
3	2	87659	5000	UPDATE

Employee ID	Project Number	Transaction ID	Amount Paid
1	1	1234	10000
1	2	45621	5500
2	4	112345	20000
1	2	123478	10000
3	5	87656	20000
3	2	78656	25000
3	2	87659	5000

Client Transaction

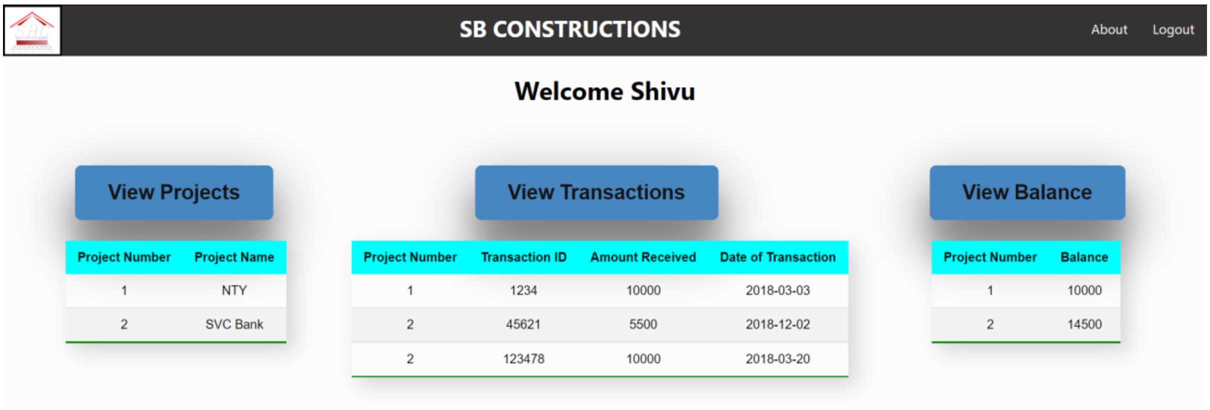


Total Balance is 55000

Project Number	Project Name	Project Location	Client Name	Balance	Action
1	NTY	Mysore road	Sumanth	5000	Update Transac
2	SVC Bank	Chandra Layout	Suresh	50000	Update Transac

Project Number	Amount Paid	Transaction ID	Date of Transaction
1	10000	234	2012-03-09
1	5000	456	2012-03-20
2	50000	4378	2012-03-09

User Dashboard



Welcome Shivu

Project Number	Project Name
1	NTY
2	SVC Bank

Project Number	Transaction ID	Amount Received	Date of Transaction
1	1234	10000	2018-03-03
2	45621	5500	2018-12-02
2	123478	10000	2018-03-20

Project Number	Balance
1	10000
2	14500