VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM-590014

A DBMS Mini-Project Report on

"EXPENSE MANAGEMENT SYSTEM"

A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi.

Submitted by:

SHREYANSH KUCHANUR (1DT20CS137)
SMARANYA VIJAYA KRISHNA (1DT20CS140)
VAISHNAVI K S (1DT20CS166)

Under the Guidance of:

Prof .Chaitra Y R

Assistant Professor, Department of CSE

Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY ANDMANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560 082 (Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)CE, CSE, ECE, EEE, ISE, ME Courses Accredited by NBA, New Delhi, NAAC A+

2022-2023

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560 082

(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, NewDelhi) CE, CSE, ECE, EEE, ISE, ME Courses Accredited by NBA, New Delhi, NAAC A+

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project on Database Management System (DBMS) entitled **"EXPENSE MANAGEMENT SYSTEM"** has been successfully carried out by **SHREYANSH KUCHANUR (1DT20CS137), SMARANYA VIJAYA KRISHANA (1DT20CS140) and VAISHNAVI K S (1DT20CS166)** bonafide students of **Dayananda Sagar Academy of Technology and Management** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Dr. M Ravi Shankar	**Guide:**	**Dr. Kavitha C**
Principal , DSATM	**Chaitra Y R**	**HOD , Dept. of CSE**
	Assistant Professor, Dept. of CSE	

ACKNOWLEDGEMENT

It gives us immense pleasure to present before you our project titled " **EXPENSE MANAGEMENT SYSTEM USING HTML, JAVASCRIPT and NODE JS**". The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** for providing us with utmost knowledge,encouragement and the maximum facilities in undertaking this project.

We wish to express a sincere thanks to our respected principal **Dr. M Ravi Shankar**, Principal, DSATM for all his support.

We express our deepest gratitude and special thanks to **Dr. Kavitha C,** H.O.D, Dept. Of Computer Science Engineering, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini-project guide, **Prof. Chaitra Y R,** Assistant Professor, Dept of Computer Science**.**

SHREYANSH KUCHANUR (1DT20CS137)
SMARANYA VIJAYA KRISHNA (1DT20CS140)
VAISHNAVI K S (1DT20CS166)

ABSTRACT

Expense management is a critical skill for anyone looking to thrive in the 21st century. Managing your money can be a source of stress, but it doesn't have to be.

Cash - oriented payments were all any knew about before as online transactions were still pretty risky due to the "hidden middle man fear" that is present. COVID-19 changed all of that. Now we have a new era of digital payments, and the world is changing rapidly. Your ability to manage your expenses will be the difference between staying afloat and sinking.

For example expense management is important for a family to stick on to their budgets and to meet their financial objectives . Tracking expenses can reveal spending issues , instead of recording expenses using pen and paper its easier to use a website. To help you through this, we have a draft prototype of the EMS - Expense Management System.

The motivation that led to the implementation of the proposed system is that there is no proper data consistency , some critical inputs may be missed and manual errors may occur.

We provide an expense management system that allows you to track your expenses , including recurring costs and gains, allow the admin to monitor your expenses, use free rewards for each transaction done by the users.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PURPOSE

- The purpose of this project is to provide a friendly environment to maintain the detailsof expenses in a family.
- The main purpose of this project is to maintain easy communication and record system using computers and to provide different reports.
- The system is an exclusive suit of services for people who seek to handle their earnings and plan their expenses and savings efficiently.
- The system allows the admin to view all the transactions made by the users registered under the admin.
- The User can view the transaction history and also claim the rewards for each transaction.

1.2 SCOPE

- The document only covers the requirements specifications for the EXPENSE MANAGEMENT SYSTEM.
- This document does not provide any references to the other component of the EXPENSE MANAGEMENT SYSTEM.
- All the external interfaces and the dependencies are also identified in this document.
- The overall scope of the feasibility study is to provide sufficient information to allow a decision to be made as to whether the EXPENSE MANAGEMENT SYSTEM project should proceed .

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Hardware Configuration

Client side

RAM: 512MB

Hard disk:10GB **Processor:** 1.0Ghz

Server side

RAM: 1GB

Hard disk: 20GB

Processor: 2.0Ghz

2.2 Software Configuration

Web browser: Chrome or any other equivalent browser

Operating System: Windows or any equivalent OS

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 ANALYSIS

Manual calculations of expenses are very much tedious and time consuming process. Due to the growth of technology in a rapid way people expect everything to be online and easy-going. This project aims at providing an easy access to the users to view their transactions . Allows the admin to monitor the expenses of users registered under the admin. The system allows the user to claim free rewards that comes as an incentive of using the app.

Disadvantages of present system:

- Not user friendly
- Too much clutter
- Time consuming
- Less Efficient

3.2 DESIGN INTRODUCTION

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities-design, coding, implementation and testing that are required to build and verify the software.

3.2.1 CONTROL FLOW DIAGRAM

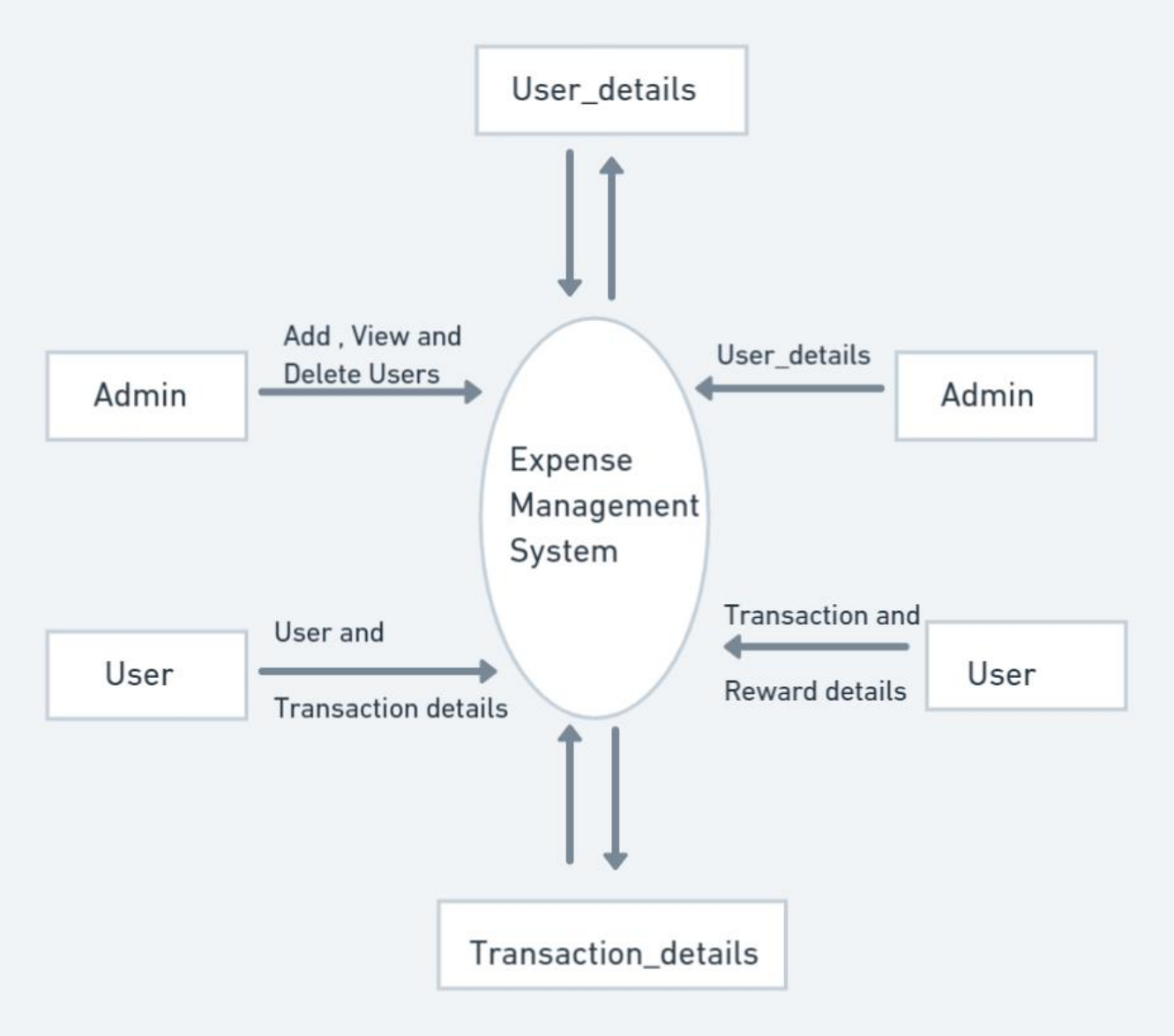


Figure 3.1: shows the flow of control through different entities in the system

3.2.2 SCHEMA DIAGRAM

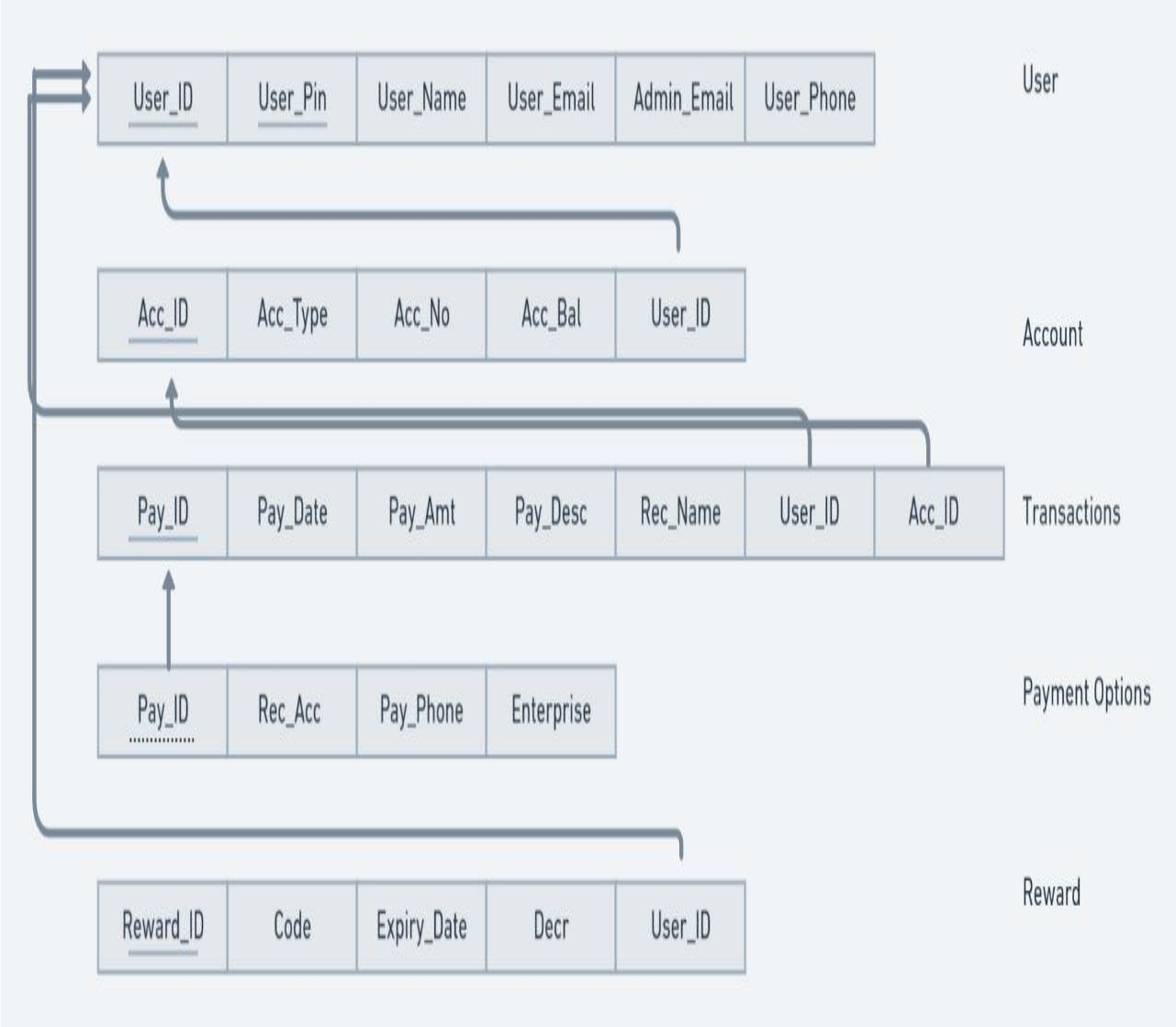


Figure 3.2: skeleton structure that represents the logical view of the entire database

3.2.3 ER DIAGRAM

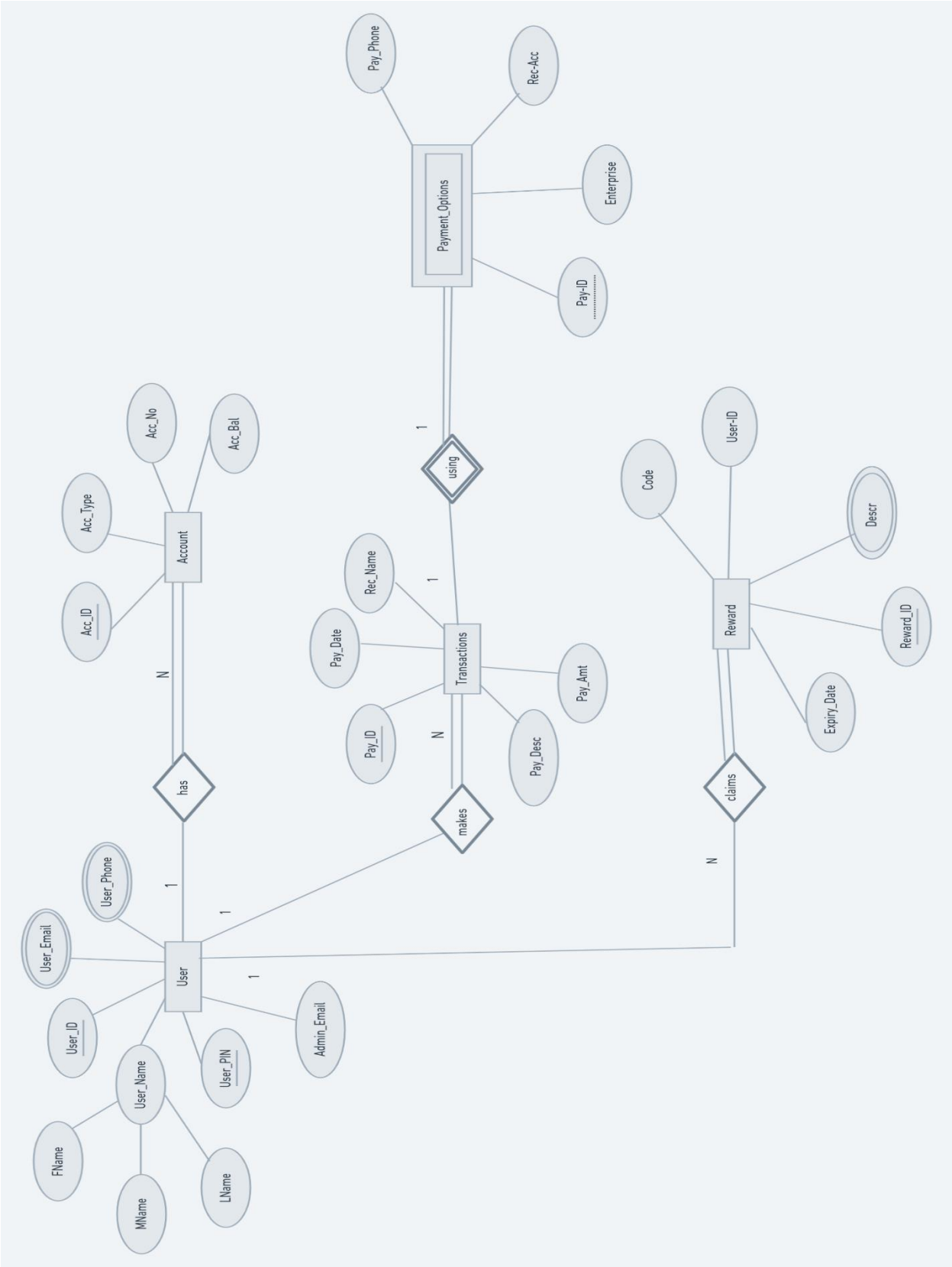


Figure 3.3: shows the relationships of entity sets stored in a database and theirrelationship

3.2.4 DATA TABLES

It contains the description of all the tables in the database

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...
account	InnoDB	10	Dynamic	2	8192	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	100
payment_options	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
rewards	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	5
transactions	InnoDB	10	Dynamic	4	4096	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	55
user	InnoDB	10	Dynamic	2	8192	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	105

Table 3.1 EMS database

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
USER_ID	int	NO	PRI	NULL	auto_increment
USER_PASSWORD	varchar(20)	YES		NULL	
FNAME	varchar(20)	YES		NULL	
MNAME	varchar(20)	YES		NULL	
LNAME	varchar(20)	YES		NULL	
USER_EMAIL	varchar(40)	YES		NULL	
USER_PHONE	decimal(10,0)	YES		NULL	
ADMIN_EMAIL	varchar(40)	YES		NULL	

Table 3.2 Users Table

USER_ID	USER_PASSWORD	FNAME	MNAME	LNAME	USER_EMAIL	USER_PHONE	ADMIN_EMAIL
90	sam@123	Sam		V K	sam@gmail.com	9731248662	sam@gmail.com
103	neel@123	Neelesh		R	neel@gmail.com	8904047282	sam@gmail.com
105	vaish@123	Vaishnavi		K S	vaish@gmail.com	9739981220	sam@gmail.com
106	raman@123	Raman		Singh	raman@gmail.com	8104771721	sam@gmail.com
107	pari@123	Parnika		Aravind	pari@gmail.com	6364308982	sam@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table 3.3 User Registration Table

Field	Type	Null	Key	Default	Extra
ACC_ID	int	NO	PRI	NULL	auto_increment
USER_ID	int	YES	MUL	NULL	
ACC_NO	varchar(20)	YES		NULL	
ACC_TYPE	varchar(10)	YES		NULL	
ACC_BAL	decimal(10,2)	YES		NULL	

Table 3.4 Accounts Table

Field	Type	Null	Key	Default	Extra
PAY_ID	int	NO	PRI	NULL	auto_increment
USER_ID	int	YES	MUL	NULL	
ACC_ID	int	YES	MUL	NULL	
PAY_AMT	decimal(10,2)	YES		NULL	
PAY_DESC	varchar(40)	YES		NULL	
REC_NAME	varchar(20)	YES		NULL	
PAY_DATE	varchar(10)	YES		NULL	

Table 3.5 Transaction

PAY_ID	USER_ID	ACC_ID	PAY_AMT	PAY_DESC	REC_NAME	PAY_DATE
50	103	97	100.00	Metro recharge	Namma Metro	2023-01-20
51	103	97	200.00	Recharge Phone	Mom	2023-01-20
53	103	97	200.00	Metro recharge	Namma Metro	2023-01-21
54	103	97	100.00	Bhel Puri	Sai Ram Chats	2023-01-21
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table 3.6 Transaction history

Field	Type	Null	Key	Default	Extra
PAY_ID	int	NO	PRI	NULL	
ENTERPRISE	tinyint(1)	YES		NULL	
REC_ACC	decimal(20,0)	YES		NULL	
PAY_PHONE	decimal(10,0)	YES		NULL	

Table 3.7 Payment Options

Field	Type	Null	Key	Default	Extra
REWARD_ID	int	NO	PRI	NULL	auto_increment
CODE	varchar(10)	YES		NULL	
DESCR	varchar(20)	YES		NULL	
USER_ID	int	YES	MUL	NULL	
EXPIRY_DATE	varchar(10)	YES		NULL	

Table 3.8 Rewards

CHAPTER 4

IMPLEMENTATION

4.1 MODULES

The PLACEMENT MANAGEMENT SYSTEM contains two modules. They are:

- The Admin module
- The User Module

Admin Module

Home Page: Admin can view and update his/her profile in this page.

View User List: Admin can view the list of users registered.

View And Update User Details: Admin can view the details of each user and delete the users.

Add Users: Admin can register users.

Add Admin: Admin can register a new admin.

User Module

Home Page: Users can view and update their profile details in this page.

Transactions : Users can view all the transactions made and the current available balance.

Add Transaction: Users can add new transactions to the list.

Add Money: Users can add the received money to the list.

See Rewards: Users can view and claim rewards.

4.2 DATABASE CONNECTIVITY

A database connection is a facility in computer science that allows client software to talkto database server software, whether on the same machine or not. A connection is required to send and receive commands.

DB Connectivity

```
const mysql = require('mysql2')
const dotenv = require('dotenv');
dotenv.config();
const con = mysql.createConnection({
    host: process.env.HOST,
    user: process.env.USER,
    password: process.env.PASSWORD,
    database: process.env.DATABASE
});
con.connect((err) => {
    if (err)
        throw err
    console.log("DB connected");

})

module.exports.con = con;
```

- **Localhost:** Localhost is often used in web scripting languages like JavaScript when defining what server the code should run from or where database is located a database is located.
- **Mysql2_connect():** The connect()/mysql_connect() function opens a newconnection to the MySQL server
- **root:** The JavaScript document root is the folder where a JavaScript script is running. When installing a script web developers often need to know the document root. Although many pages scripted with JavaScript run on an Apache server, some run under MicrosoftIIS on Windows
- **dbms_mini_project:** It is the project folder name

4.1 SOURCE CODE

Admin Login Page:

```
app.get('/',
(req, res)
=>{
                    res.render("login");
                });

                app.post('/', (req, res) =>{
                    const admin = req.body.email;
                    const password = req.body.password;
                    // if (admin === "sam@gmail.com" && password === "sam@123") {
                        var query = "select * from USER WHERE ADMIN_EMAIL = ?";
                        data = [admin, admin];
                        var query2 = "SELECT * from TRANSACTIONS WHERE USER_ID = (SELECT USER_ID FROM USER WHERE
                USER_EMAIL = ?)";
                        mysql.query(query, data, (error, result) => {
                            if(result.length === 0){
                                alert('Invalid Admin Credentials');
                                res.redirect('/');
                            }
                            mysql.query(query2, admin, (error, adminpay) =>{
                                res.render("landing", {result, adminpay});
                            })
                        });
                    // }
                })
```

Admin Registration page:

```
app.get('/addAdmin',
(req, res) =>{
                            res.render('newAdmin');
                        })

                        app.post('/addAdmin', (req, res) =>{
                            let fname = req.body.firstName;
                            let mname = req.body.middleName;
                            let lname = req.body.lastName;
                            let password;
                            if(req.body.password == req.body.confirmPassword){
                                password = req.body.password;
                            }
                            let userEmail = req.body.userEmail;
                            let phoneNumber = req.body.phoneNumber;
                            let sql1 = "INSERT INTO USER(USER_PASSWORD, FNAME, MNAME, LNAME,
                        USER_EMAIL, USER_PHONE, ADMIN_EMAIL) VALUES ?";

                                let values1 = [
                                    [password, fname, mname, lname, userEmail, phoneNumber,
                        userEmail]
                                ];

                                mysql.query(sql1, [values1], function(error, result){
                                    if(error) throw error;
                                    res.redirect('/');
                                })

                        })
```

User sign up:

```
app.get('/signup',
(req, res) =>{
                        res.render("signup")
                    });

                    app.post('/signup', (req, res) =>{
                        let fname = req.body.firstName;
                        let mname = req.body.middleName;
                        let lname = req.body.lastName;
                        let admin = req.query.admin;
                        let password;
                        if(req.body.password == req.body.confirmPassword){
                            password = req.body.password;
                        }
                        let userEmail = req.body.userEmail;
                        let phoneNumber = req.body.phoneNumber;
                        let n = req.body.n;
                        let accNumber1 = req.body.accNumber1;
                        let accType1 = req.body.accType1;
                        let currentBal1 = req.body.currentBal1;
                        let accNumber2, accType2, currentBal2;
                        if(req.body.accNumber2 != null){
                            accNumber2 = req.body.accNumber2;
                            accType2 = req.body.accType2;
                            currentBal2 = req.body.currentBal2;
                        }
                        let accNumber3, accType3, currentBal3;
                        if(req.body.accNumber3 != null){
                            accNumber3 = req.body.accNumber3;
                            accType3 = req.body.accType3;
                            currentBal3 = req.body.currentBal3;
                        }

                            let sql1 = "INSERT INTO USER(USER_PASSWORD, FNAME, MNAME, LNAME,
                    USER_EMAIL, USER_PHONE, ADMIN_EMAIL) VALUES ?";

                            let values1 = [
                                [password, fname, mname, lname, userEmail, phoneNumber, admin]
                            ];

                            mysql.query(sql1, [values1], function(error, result){
                                if(error) throw error;

                                let sql2 = "INSERT INTO ACCOUNT(USER_ID, ACC_NO, ACC_TYPE,
                    ACC_BAL) VALUES ?";

                                let values2 = [];
                                if(n==1){
```

```
values2
= [
                            [result.insertId, accNumber1, accType1, currentBal1]
                        ]
                    }
                    else if(n==2){
                    values2 =[
                        [result.insertId, accNumber1, accType1, currentBal1],
                        [result.insertId, accNumber2, accType2, currentBal2],
                    ];
                    }
                    else if(n==3){
                        values2 =[
                            [result.insertId, accNumber1, accType1, currentBal1],
                            [result.insertId, accNumber2, accType2, currentBal2],
                            [result.insertId, accNumber3, accType3, currentBal3],
                        ]
                    }

                    mysql.query(sql2, [values2], function(error, result){
                        if(error) throw error;
                        res.redirect(301, '/userLogin');
                    })
                })
            });

            //Admin Landing Page: Where they can see, add and delete users
            app.get('/landing', (req, res) =>{
                let admin = "sam@gmail.com";
                var query = 'select * from USER WHERE USER_EMAIL != ?';
                var query2 = 'SELECT * from TRANSACTIONS WHERE TRANSACTIONS.USER_ID =
        USER.USER_ID AND USER.USER_EMAIL = ?';
                mysql.query(query, admin, (error, result) => {
                    mysql.query(query2, admin, (error, adminpay) =>{
                        res.render("landing", {result, adminpay});
                    })
                });
            })

            app.get('/delete-user',(req,res)=>{
            const id = req.query.id;
            var query = "delete from USER where USER_ID = ?";
            var query2 = "delete from ACCOUNT where USER_ID = (SELECT USER_ID FROM USER
        WHERE USER_ID = ?)";
            mysql.query(query,[id],(error,result)=>{
                mysql.query(query2, [id], (error, result) =>{
                    if(error) throw error;
                    res.redirect('landing')
                 }) })
            })
```

User Login:

```
app.get('/userLogin',(req,res)=>{
    res.render("userlogin");
    })

    //User Landing Page
    app.get('/userLand',(req,res)=>{
    const {password} = req.query
    var query2 = "select * from USER WHERE
USER_PASSWORD = ? AND USER_EMAIL != ADMIN_EMAIL";
    var query4 = "SELECT * FROM ACCOUNT WHERE
USER_ID = ?";
    var querypay = "select * from TRANSACTIONS
WHERE USER_ID = ?";
    mysql.query(query2, password, (error, result)
=>{
        let uid;
        if(result.length === 0){
            alert('Invalid User Credentials!');
            res.redirect('/userLogin');
        }
        else{
        uid =result[0].USER_ID;
        mysql.query(query4, uid, (error,
balance)=>{
            mysql.query(querypay, uid, (error,
pay) => {
                if(error) throw error;
                res.render('userland', {result,
pay, balance});
            })
        })
        }
    })
})
```

Transaction adding form:

```
app.get('/add',
(req, res) =>{
                        const uid = req.query.uid;
                        let sql1 = "SELECT ACC_NO FROM ACCOUNT WHERE USER_ID = ?"
                        mysql.query(sql1, uid, (err, result)=> {
                            res.render('pay_add', {
                                title: 'Transaction form page',
                                result
                            })
                        })
                    });

                    app.post('/save', (req, res) =>{
                        let sql1 = 'SELECT * FROM USER WHERE USER_EMAIL = ?';
                        let email = req.body.email;
                        let accNo = req.body.accNo;
                        let date = req.body.pay_date;
                        let amt = req.body.pay_amt;
                        let desc = req.body.pay_desc;
                        let rec_name = req.body.rec_name;
                        let pay_method = req.body.paymentMethod;
                        let rec_acc;
                        let rec_phone;
                        let enterprise;
                        if(pay_method == 'enterprise'){
                            enterprise = true,
                            rec_acc = null,
                            rec_phone = null
                        }
                        else if(pay_method == 'phoneNumber'){
                            enterprise = false,
                            rec_acc = null,
                            rec_phone = req.body.phoneNumber
                        }
                        else{
                            enterprise = false,
                            rec_acc = req.body.recAcc,
                            rec_phone = null
                        }
                        mysql.query(sql1, email, (err, result) =>{
                            password = result[0].USER_PASSWORD;
                            uid = result[0].USER_ID;
                            let sql2 = 'SELECT * FROM ACCOUNT WHERE ACC_NO = ?'
                            mysql.query(sql2, accNo, (err, acc) =>{
                                actid = acc[0].ACC_ID;
                                let sql = "INSERT INTO TRANSACTIONS(USER_ID, ACC_ID, PAY_AMT,
                PAY_DATE, PAY_DESC, REC_NAME) VALUES ?";
```

```
let data = [
                            [uid, actid, amt, date, desc, rec_name]
                        ];
                        mysql.query(sql, [data], (err, pay)=> {
                            if(err) throw err;
                            let sqlupdate = "SELECT (ACC_BAL - PAY_AMT) AS REM_BAL
        FROM ACCOUNT, TRANSACTIONS WHERE ACCOUNT.ACC_ID = TRANSACTIONS.ACC_ID AND
        ACCOUNT.ACC_ID = ?"
                            mysql.query(sqlupdate, actid, (err, update) =>{
                                rem = update[0].REM_BAL;
                                let updatefinal = "UPDATE ACCOUNT SET ACC_BAL = ?
        WHERE ACC_ID = ?";
                                data = [rem, actid];
                                mysql.query(updatefinal, data, (err, updating) => {
                                    let sqlpay = "INSERT INTO PAYMENT_OPTIONS(PAY_ID,
        ENTERPRISE, REC_ACC, PAY_PHONE) VALUES ?"
                                    let values = [
                                        [pay.insertId, enterprise, rec_acc,
        rec_phone]
                                    ];
                                    mysql.query(sqlpay, [values], (err, payopt) =>{

        res.redirect('/userland?name='+email+'&password='+password);
                                    })
                                })
                            })
                        })
                    })
                })
            });
```

Adding Money To User Account:

```
app.get('/addmoney',
(req, res)=>{
                                const uid = req.query.uid;
                                let sql1 = "SELECT * FROM ACCOUNT WHERE USER_ID = ?"
                                mysql.query(sql1, uid, (err, account)=> {
                                    res.render('add_money', {
                                        title: 'Add Money To Your Account',
                                        subtitle: '....And Continue making Payments!',
                                        account
                                    });
                                })
                            });

                            app.post('/savebal', (req, res) =>{
                                const uid = req.query.uid;
                                let sql1 = 'SELECT * FROM USER WHERE USER_ID = ?';
                                let {accNo, add_desc, addDate} = req.body;
                                let addAmt = req.body.addAmt;
                                let snd_name = 'FROM '+req.body.snd_name;
                                mysql.query(sql1, uid, (err, user)=>{
                                    email = user[0].USER_EMAIL;
                                    password = user[0].USER_PASSWORD;
                                    let sql2 = "SELECT * FROM ACCOUNT WHERE ACC_NO =
                            ?";
                                    let sql3 = "INSERT INTO TRANSACTIONS(USER_ID,
                            ACC_ID, PAY_AMT, PAY_DATE, PAY_DESC, REC_NAME) VALUES ?"

                                    let sql4 = 'UPDATE ACCOUNT SET ACC_BAL = (ACC_BAL
                            + ?) WHERE ACC_NO = ?';

                                    mysql.query(sql2, accNo, (err, account)=>{
                                        actid = account[0].ACC_ID;
                                        let data1 = [
                                            [uid, actid, addAmt, addDate, add_desc,
                            snd_name]
                                        ]
                                        mysql.query(sql3, [data1], (err, addtran) =>{
                                            let data = [addAmt, accNo];
                                            mysql.query(sql4, data, (err, addbal)=>{

                            res.redirect('/userland?name='+email+'&password='+password);
                                            })
                                        })
                                    })
                                })
                            })
```

Rewards Page:

```
app.get("/rewards",
(req, res) =>{
                                    const email = req.query.email;
                                    var query = "SELECT * FROM REWARDS WHERE USER_ID IS NULL";
                                    var query2 = "SELECT * FROM USER WHERE USER_EMAIL = ?";
                                    mysql.query(query, (error, result) => {
                                        mysql.query(query2, email, (err, user) =>{
                                            uid = user[0].USER_ID;
                                            let query3 = "SELECT * FROM REWARDS WHERE USER_ID = ?";
                                            mysql.query(query3, uid, (err, claim)=>{
                                                if(claim.length == 0){
                                                    alert('No rewards claimed yet!')
                                                    res.render('rewards', {result, user, claim});
                                                }
                                                else{
                                                    res.render('rewards', {result, user, claim})
                                                }
                                            })
                                        })
                                    })
                                })

                                app.get('/claim', (req, res) =>{
                                const code = req.query.code;
                                const email = req.query.email;


                                var query = "select * from USER where USER_EMAIL=?";
                                var query2 = "update rewards set USER_ID = ? where CODE = ?";
                                mysql.query(query,email,(error,result)=>{
                                    let uid;
                                    if(result.length === 0 || code === ''){
                                        alert('Please Enter Code and Email to Claim!')
                                        res.redirect("/rewards?email="+email);
                                    }
```

```
else{
            uid = result[0].USER_ID;
            let data = [uid, code];
            mysql.query(query2, data ,(err,result1)=>{
                if(result1.changedRows === 0){
                    alert('Invalid Code!');
                }
                else{
                    alert('Claimed successfully: '+code);
                }
                res.redirect("/rewards?email="+email);
        })}
    })
})

app.listen(port, ()=>{
    console.log(`Server connected on ${port}`);
});
```

CHAPTER 5

TESTING

Software testing is a process of checking whether the actual software product matches expected requirements and to ensure that software product is defect free.It involves execution of software components using manual or automated tools toevaluate one or more properties of interest.

5.1 SYSTEM TESTING

System testing is a level of testing that validates the complete and fully integratedsoftware product. The purpose of system test is to evaluate the end to end system specifications. Usually, the software is only one element of a larger computer based system.

5.2 MODULE TESTING

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the small building blocks of the program code.

5.3 INTEGRATION TESTING

Testing is a systematic technique or construction the program structure while atthe same time conducting tests to uncover error associated with the interfacing.Scope of testing summarizes the specific functional, performance and internal design characteristics that are to be tested.

5.4 UNIT TESTING

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as a part of unit testing are testing the module interface, examining the local data structureand testing error handling path.

CHAPTER 6

RESULT ANALYSIS AND SCREENSHOTS

Admin Login Page: Where the Admin can login .

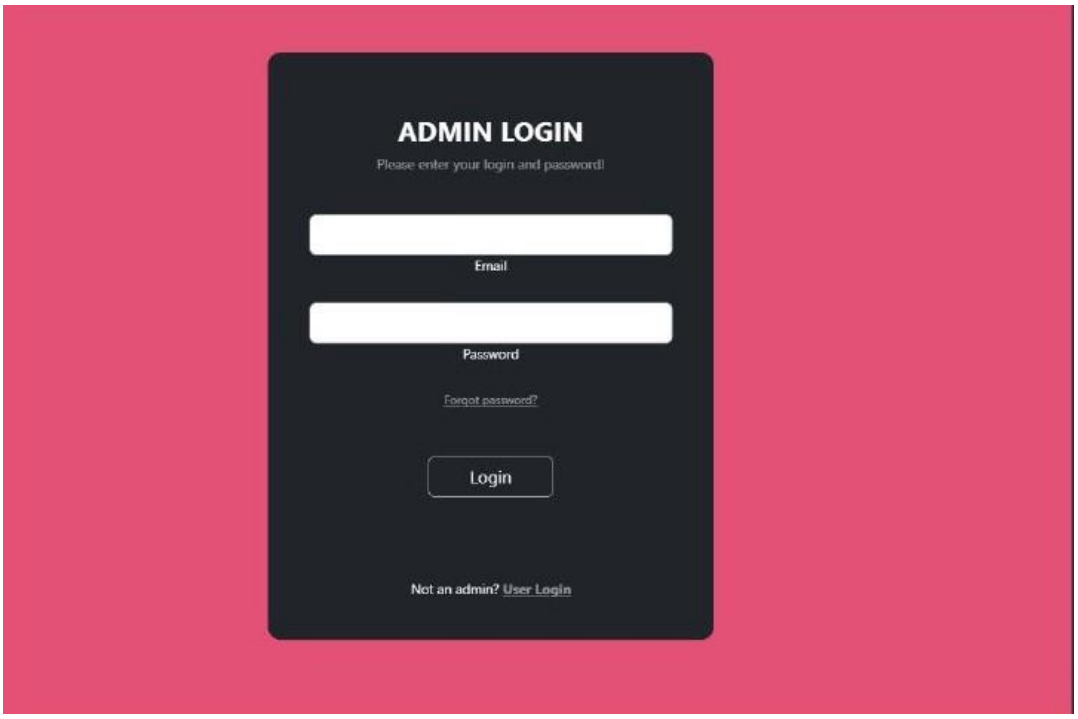


Figure 6.1: Admin Login

Admin Landing Page: Here admin can view all the users who have registered , view each user details and delete the users.

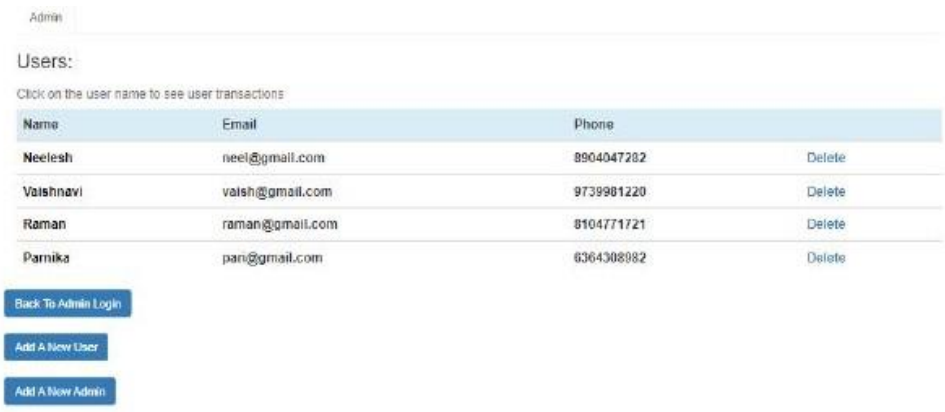


Figure 6.2: Admin Landing Page

Add A New User : Here the admin can register a new user.

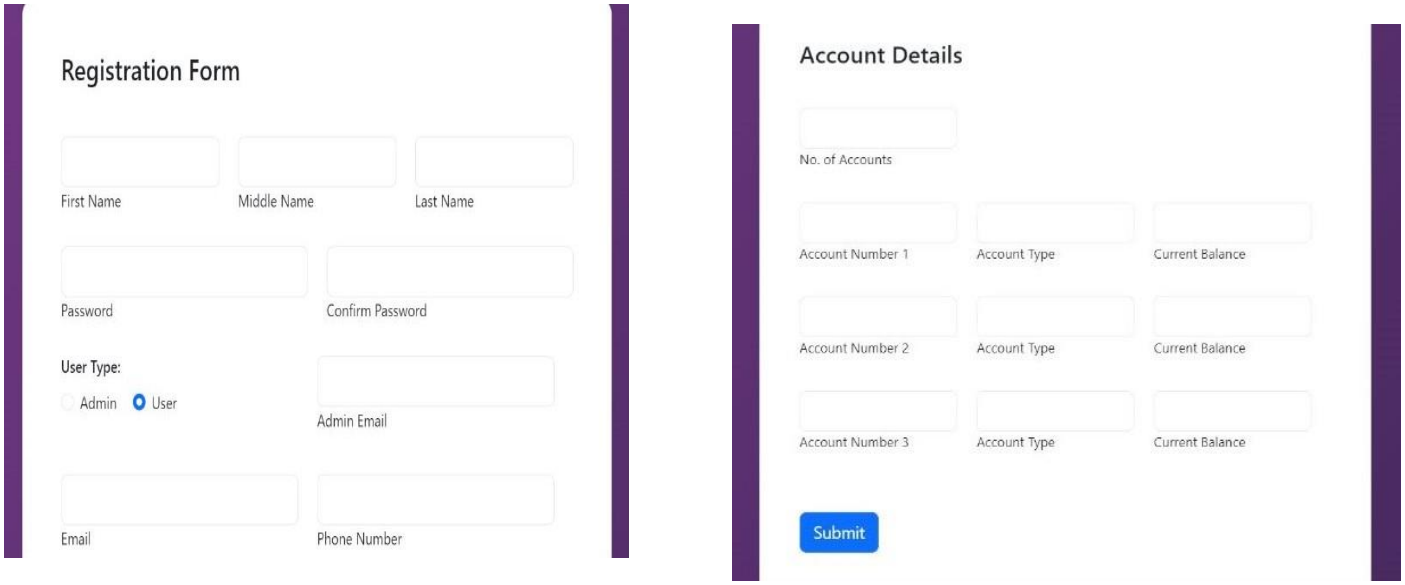


Figure 6.3: Add New User

Add An Admin : Here the admin can add a new admin.

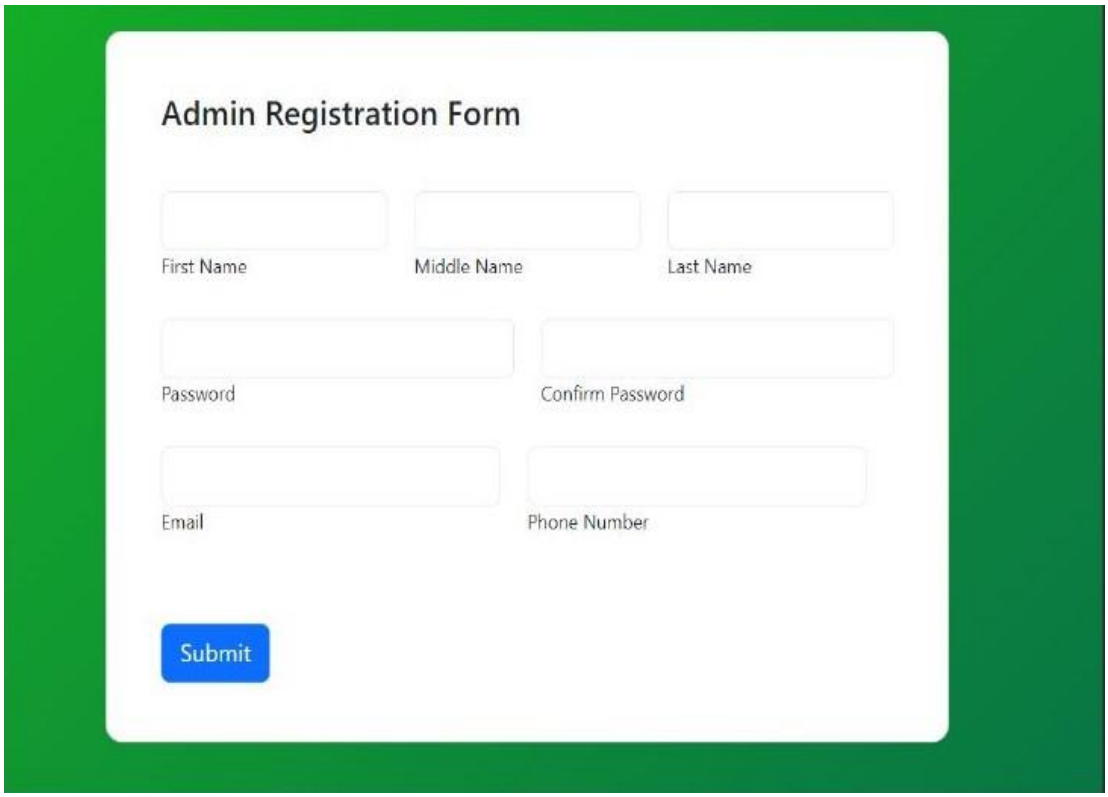


Fig 6.4: Admin Registration Form

User Login : User can login .

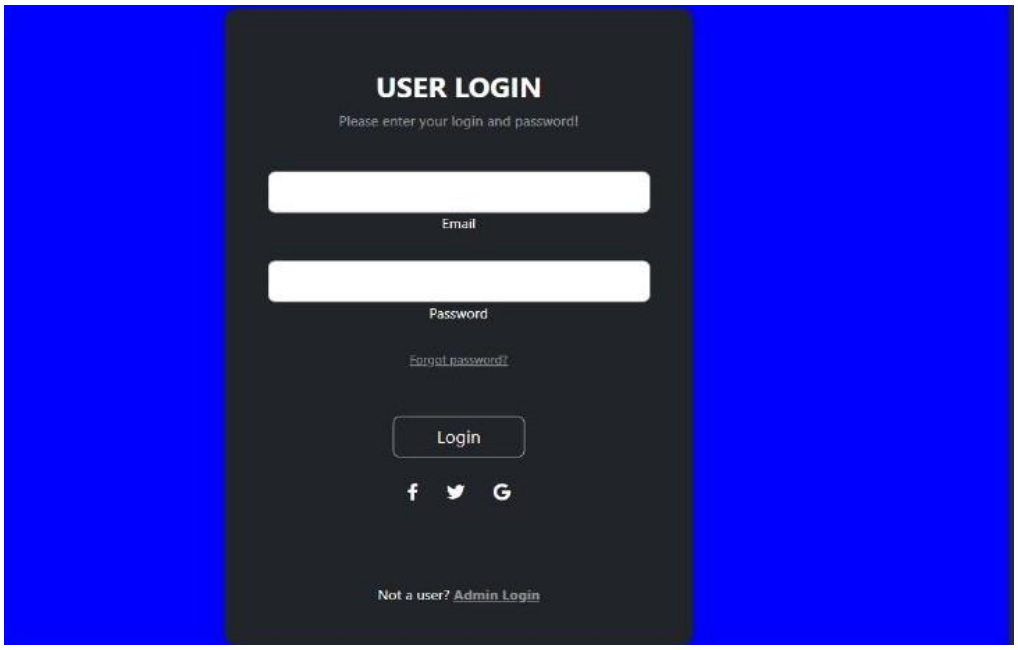


Figure 6.5: User Login

User Landing Page : User can view all the transactions made and the available current balance in each account.

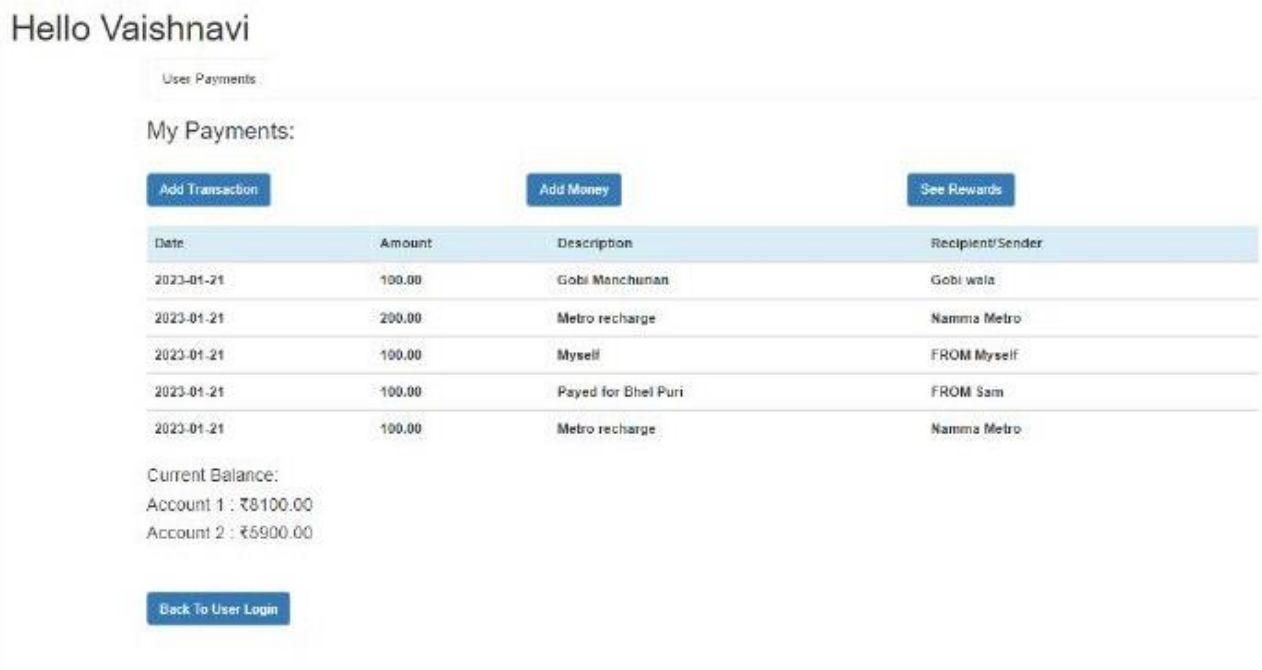
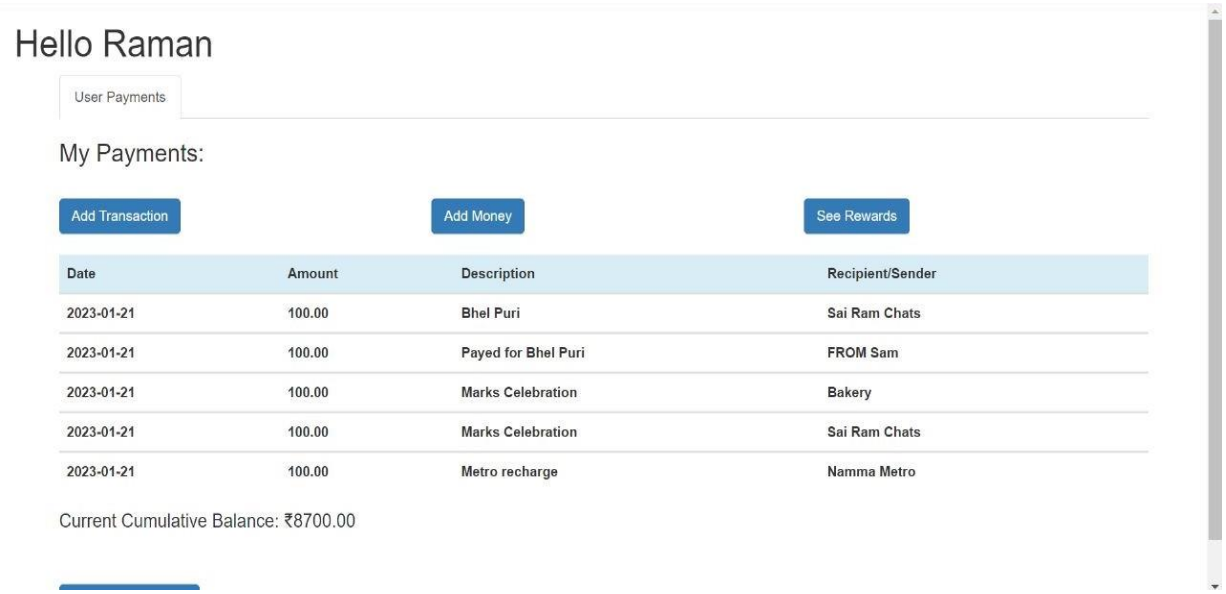


Figure 6.6: Transaction History

Add Transaction : User can add the transaction details to list.

1. Transaction entering form

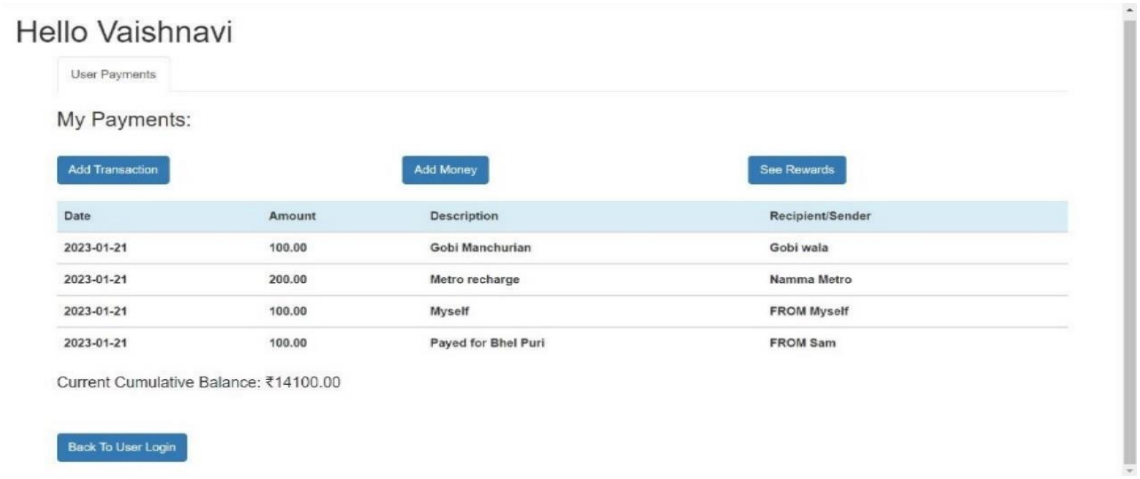


2.Transaction List After Updation

Figure 6.7: Add Transaction

Add Money: User can add money that have been received

1.Adding money details

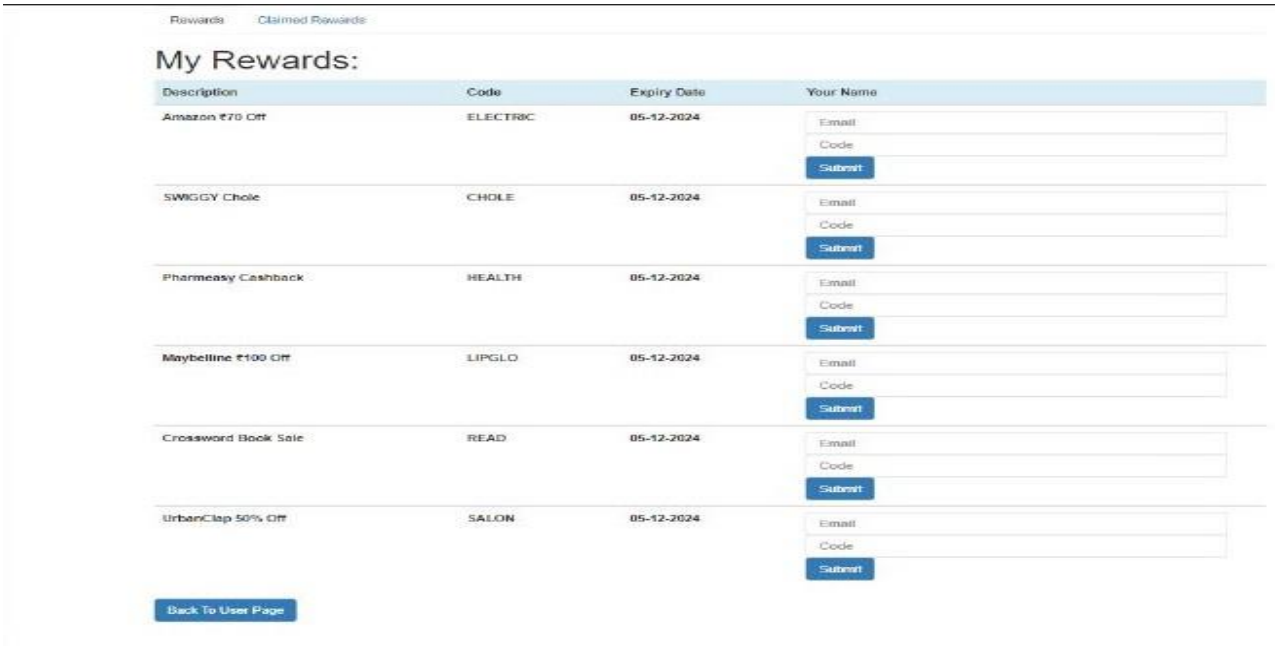


2. User page after adding the details

Figure 6.8: Add Money

See Rewards: Here the user can view the rewards and claim the rewards.

1. My Rewards.



2.Claimed Rewards

Figure 6.9: See Rewards

CONCLUSION AND FUTURE ENHANCEMENTS

The EXPENSE MANAGEMENT SYSTEM is a great improvement over the manualsystem which uses lots of manual work. The computerization of the system speeds up theprocess. This system was thoroughly checked and tested with dummy data and found to be reliable.

MERITS

- The EXPENSE MANAGEMENT SYSTEM is fast, efficient and reliable.
- Avoids data redundancy and inconsistency
- Web-based
- Any number of users can use it
- Provides more security and integrity to data

FUTURE ENHANCEMENTS

As the technology emerges, it is possible to upgrade the system and can be adaptable todesired environment. Based on the future security issues, security can be improved usingemerging technologies. Sub admin module can also be added.

BIBLIOGRAPHY

Book References:

➢ Learn to Code HTML and CSS: Develop and Style Websites (Web Design Courses)

WEBSITE REFERENCES:

For MySQL

- **https://youtu.be/D_wNQR3LeeM**

Node JS

- https://youtu.be/zZa4rWdfkJw
- https://youtu.be/iiy4I_gpVdM

Personal Details:

- **NAME: SHREYANSH KUCHANUR**

  USN: 1DT20CS137

  SEMESTER: 5TH SEM, 'C' SECTION

  COLLEGE: DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

  EMAIL-ID: kuchanurshreyansh@gmail.com

- **NAME: SMARANYA VIJAYA KRISHNA**

  USN: 1DT20CS140

  SEMESTER: 5TH SEM, 'C' SECTION

  COLLEGE: DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

  EMAIL-ID: smaranya.vijayakrishna@gmail.com

- **NAME: VAISHNAVI K S**

  USN: 1DT20CS166

  SEMESTER: 5TH SEM, 'C' SECTION

  COLLEGE: DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

  EMAIL-ID: vaishnaviks01@gmail.com