

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi-590018, Karnataka, INDIA



Mini PROJECT REPORT on

“Database Management System for NGO (Ekk Pahel)”

Submitted in partial fulfillment of the requirements for the V Semester
DBMS LABORATORY WITH MINI PROJECT (15CSL58)

Bachelor of Engineering IN COMPUTER SCIENCE AND ENGINEERING

For the Academic year
2017-2018

BY

Chehak Nayar 1PE15CS042
Divyaksh Shukla 1PE15CS051

Under the Guidance of
Ms. Sangeetha R
Assistant Prof, Dept. of CSE
PESIT-BSC, Bengaluru-560100



PES
INSTITUTIONS

Department of Computer Science and Engineering
PESIT BANGALORE SOUTH CAMPUS
Hosur Road, Bengaluru -560100

PESIT BANGALORE SOUTH CAMPUS

Hosur Road, Bangalore -560100

Department of Computer Science and Engineering



CERTIFICATE

*Certified that the mini project work entitled “**Database Management System for NGO (Ek Pahel)**” is a bonafide work carried out by **Divyaksh Shukla** bearing USN: **1PE15CS051**, student of **PESIT Bangalore South Campus** in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2017-2018. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated and the mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said Degree.*

Signatures:

Project Guide
Ms. Sangeetha R
Assistant Prof, Dept. of CSE
PESIT-BSC, Bengaluru

Head Dept of CSE
Mr. Sandesh B J
Associate Prof., Dept. of CSE,
PESIT-BSC, Bengaluru

External Viva

Name of the Examiners

Signature with date

1. _____

2. _____

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all the lecturers and staff of the Department of Computer Science and Engineering for extending their help and guidance towards our project.

We would like to thank the college management and express our sincere gratitude to Dr. J Surya Prasad, Director/Principal of PESIT (BSC) have given me the opportunity for the completion of this project.

We would like to thank Mr. Sandesh B J, Head of Department Computer Science and Engineering, PESIT (BSC) for giving us the support and encouragement that was necessary for the completion of this report.

We would like to thank our project guides Mrs. Sangeetha R, Assistant Professor and Mrs. Neeta Ann Jacob, Associate Professor for providing us the required assistance, encouragement and constant support which was of a great help to complete this project successfully.

Last but not least; the project would not have been a success without the support of our parents and friends.

Chehak Nayar 1PE15CS042

Divyaksh Shukla 1PE15CS051

ABSTRACT

A nonprofit organization's goals and objectives flow from its stated mission. The mission statement, created during the strategic planning process, describes the organization's overall purpose and is usually concise.

One of the biggest operational challenges faced by nonprofit organizations today is information management. No matter what the organization does, it can't operate without collecting, storing, sharing, and archiving information. In order to deal with more donors (at the top) and more beneficiaries (at the bottom), NGOs increasingly need systems to manage the creation, accessing and deployment information.

In our project we aim at helping out one such organization, "Ekk Pahel" by creating a database management web application to collect and manage its records.

Our project caters to real world organization. This necessitates the need to involve the NGO in the decision-making process to ensure their satisfaction. Our goal is to provide a prototype for an end to end web application to manage its donations and finance, record its past and future events and achievements, keep track of its members, their roles and contributions, calendar of events and schedules. This could help in a smoother and seamless flow in the working process of the NGO, and could help take out the paper and sheet management.

Table of Contents

| Sl.No | Contents | Page No |
|--------------|----------------------|----------------|
| 1 | Introduction | 1 |
| 2 | Project Requirements | 4 |
| 3 | Literature Survey | 5 |
| 4 | Problem Statement | 9 |
| 5 | System Design | 11 |
| 6 | Implementation | 15 |
| 7 | Testing | 21 |
| 8 | Result | 23 |
| 9 | Conclusion | 29 |
| 10 | References | 30 |

List of Tables

| Table No | Name of Table | PageNo |
|-----------------|----------------------|---------------|
| 6.1 | Department Table | 19 |
| 6.2 | Donations Table | 19 |
| 6.3 | Events Table | 19 |
| 6.4 | Event Members Table | 20 |
| 6.5 | Members Table | 20 |

List of Figures

| Fig No | Name of figure | Page No |
|---------------|--------------------------------|----------------|
| 1.1 | Components of DBMS | 2 |
| 5.1 | Schema Diagram | 11 |
| 5.2 | ER Diagram | 12 |
| 5.3 | Dataflow Diagram | 13 |
| 7.1 | Testing for Incomplete Entries | 21 |
| 7.2 | Testing for Complete Entries | 22 |
| 8.1 | Welcome Page | 23 |
| 8.2 | Tables | 23 |
| 8.3 | Statistics | 24 |
| 8.4 | View Members Table | 24 |
| 8.5 | View Members Table | 25 |
| 8.6 | View Donations Table | 25 |
| 8.7 | View Departments Table | 26 |
| 8.8 | View Events Table | 26 |
| 8.9 | View Event Members Table | 27 |
| 8.10 | Add Member Form | 27 |
| 8.11 | Remove Member Form | 28 |
| 8.12 | Add Donations Form | 28 |
| 8.13 | Assign Manager Form | 28 |

Chapter 1**Introduction****1.1 Data**

Computer data is information processed or stored by a computer. This information may be in the form of text documents, images, audio clips, software programs, or other types of data. Computer data may be processed by the computer's CPU and is stored in files and folders on the computer's hard disk.

1.2 Database

A database is a collection of data that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it.

Databases can be classified according to types of content: bibliographic, full-text, numeric, and images. In computing, databases are sometimes classified according to their organizational approaches. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

1.3 Database Management System

A DBMS is software that allows creation, definition and manipulation of database. It is actually a tool used to perform any kind of operation on

data in database. It also provides protection and security to database. It maintains data consistency in case of multiple users.

Database System is a system to achieve an organized, store a large number of dynamical associated data, facilitate for multi-user accessing to computer hardware, software and data, that it is a computer system with database technology

- **Components of DBMS**

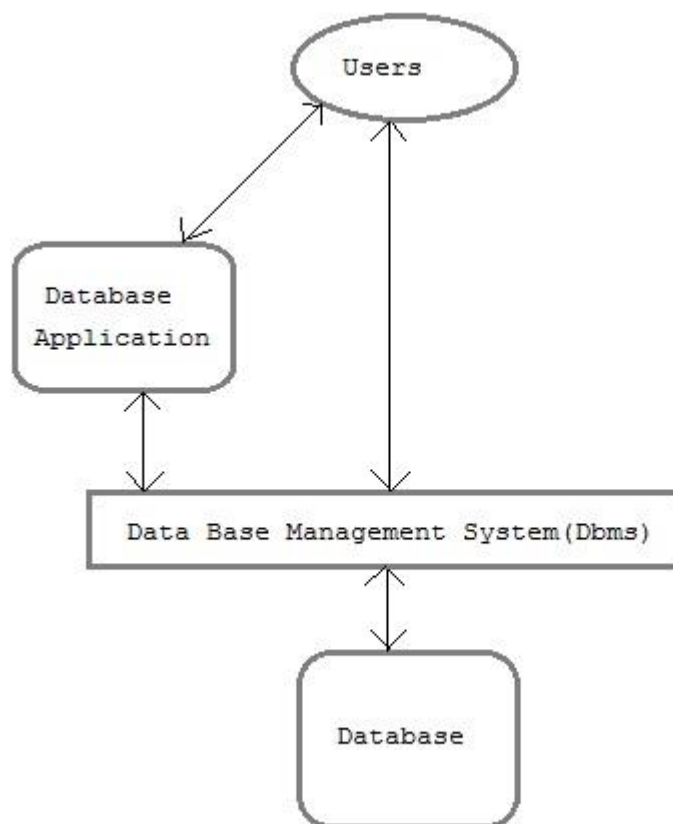


Figure1.1Components of DBMS

- **Users:** Users may be of various type such as DB administrator, System developer and End users.
- **Database application :** Database application may be Personal, Departmental, Enterprise and Internal
- **DBMS:** Software that allow users to define, create and manages database access, Ex: MySql, Oracle etc.
- **Database :** Collection of logical data

- **Functions of DBMS**
 - Provides data Independence
 - Concurrency Control
 - Provides Recovery services
 - Provides Utility services
 - Provides a clear and logical view of the process that manipulates data.

- **Advantages of DBMS**
 - Segregation of application program.
 - Minimal data duplicity.
 - Easy retrieval of data.
 - Reduced development time and maintenance need.

- **Disadvantages of DBMS**
 - Complexity
 - Costly
 - Large in size

Chapter 2

Project Requirements

2.1 Hardware

- Processor :intel i5 2.4GHz, 64bitprocessor
- Ram :4GBRAM
- HardDisk :50GB

2.2 Software

- Operating System : Linux/Windows
- Programming Language : Html, CSS, Javascript
- Technology : NodeJs v8, AngularJS, ExpressJS, PromiseJS (MEAN Stack)
- Database : MySql

2.3 Functional Requirements

- Viewing general characteristics and statistics of the NGO.
- Accessing detailed information of the NGO-Departments, Members, Donations, Events, and Event Members.
- Adding and Removing entries from the databases.

Chapter 3

Literature Survey

3.1 NodeJS

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. It is an open source server framework which runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.) Node.js uses JavaScript on the server. Here is how Node.js handles a file request:

- Sends the task to the computer's file system.
- Ready to handle the next request.
- When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request. It runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient. Node.js can generate dynamic page content, create, open, read, write, delete, and close files on the server. It can collect form data and add, delete, modify data in your database.

Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world. It is a package manager for Node.js packages, or modules if you like. The NPM program is installed on your computer when you install Node.js. A package in Node.js contains all the files you need for a module.

3.2 ExpressJS

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the Node.js foundation.

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love. It is very flexible and pluggable.

3.3 AngularJS

AngularJS is a **JavaScript framework**. It can be added to an HTML page with a `<script>` tag. AngularJS extends HTML attributes with **Directives**, and binds data to HTML with **Expressions**.

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how.

3.4 Html

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a **Markup Language**, we can use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure

of documents like headings, paragraphs, lists, and so forth. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

3.5 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the colour of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the mark up languages HTML or XHTML.

3.6 Javascript

JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language.

It is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites. JavaScript is incredibly versatile. JavaScript itself is fairly compact yet very flexible. Developers have written a large variety of tools on top of the core JavaScript language, unlocking a vast amount of extra functionality with minimum effort. These include:

- Browser Application Programming Interfaces (APIs) — APIs built into web browsers, providing functionality like dynamically creating HTML and setting CSS

- Third-party APIs to allow developers to incorporate functionality in their sites from other content providers, such as Twitter or Facebook.
- Third-party frameworks and libraries you can apply to your HTML to allow you to rapidly build up sites and applications.

3.7 MySQL

MySQL is the most popular Open Source Relational SQL Database Management System. It is one of the best RDBMS being used for developing various web-based software applications. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

MySQL uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. It is very friendly to PHP, the most appreciated language for web development.

It supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Chapter 4

Problem Statement

4.1. Non Governmental Organizations

Non-governmental organizations, nongovernmental organizations, or nongovernment organizations, commonly referred to as NGOs, are usually nonprofit and sometimes international organizations independent of governments and international governmental organizations (though often funded by governments) that are active in humanitarian, educational, healthcare, public policy, social, human rights, environmental, and other areas to effect changes according to their objectives. They are thus a subgroup of all organizations founded by citizens, which include clubs and other associations that provide services, benefits, and premises only to members.

4.2. Importance and Issue of Data management in NGOs

While it may not seem like data management is an obvious ‘must-have’ for NGOs, this is actually one of the most important operating concerns facing the industry today. Often, staff are stuck spending valuable time sorting through file cabinets and tracking down spreadsheets for records. In our increasingly data-driven world, this archaic practice is holding back NGOs from achieving their full potential and best utilizing their resources.

A recent report from Every Action found that 87% of non-profit professionals believe data to be important to operations at their organization, but just a mere 6% feel confident that the data is being used effectively. The non-profit sector realizes the importance of data, but equally recognizes that they are not fully tapping into its potential.

Following are the issues that have been identified:

- Lack of standardized format for collecting information.
- Data collected without time or resources to conduct analysis
- Challenges of data quality control and analysis.

Data management can help non-profits make the world a better place, one data-driven decision at a time.

4.3. Ekk Pahel

Ekk Pahel is a non profit organisation initiated by small group of students to help the underprivileged people in different walks of life.

One statement which often becomes the voice of the NGO is “Each drop make the ocean” is really the base behind the foundation of EKK PAHEL. With the positive attitude of the Founder Ayush Agrawal, Co- founder Aniket Vishal and Abhishek Ranjan towards the change, Ekk Pahel set feet on the ground on 11 April 2016.

You feel eternally happy when your works makes others happy. Realizing the ultimate cause of life and working on one of its way to give smile on the face of others Ekk Pahel is small union of students having clear vision and firm resolution. With a team of diligent and hardworking We organize events at orphanages, old age homes, food distribution to needy in many part of country to get closer to the people of god and give them reasons to laugh. They share their moments and cheer with us which gives us immense happiness that has no substitute in this world.

We would also like to thank every right thinking person who has supported us and it's their support and appreciation which drives us to deliver our best towards the nation.

4.4. Problem Statement

In our project we aim at helping out one such organisation, “Ekk Pahel” by creating a Database management web application to collect and manage its records. Our project caters to real world, operational organisation. This necessitates the need to involve the NGO in the decision-making process to ensure their satisfaction. Our goal is to provide a prototype for an end to end web application to manage its donations and finance, record its past and future events and achievements, keep track of its members, their roles and contributions, calendar of events and schedules. This could help in a smoother and seamless flow in the working process of the NGO, and could help take out the paper and sheet management.

Chapter 5

System Design

5.1. Schema Diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

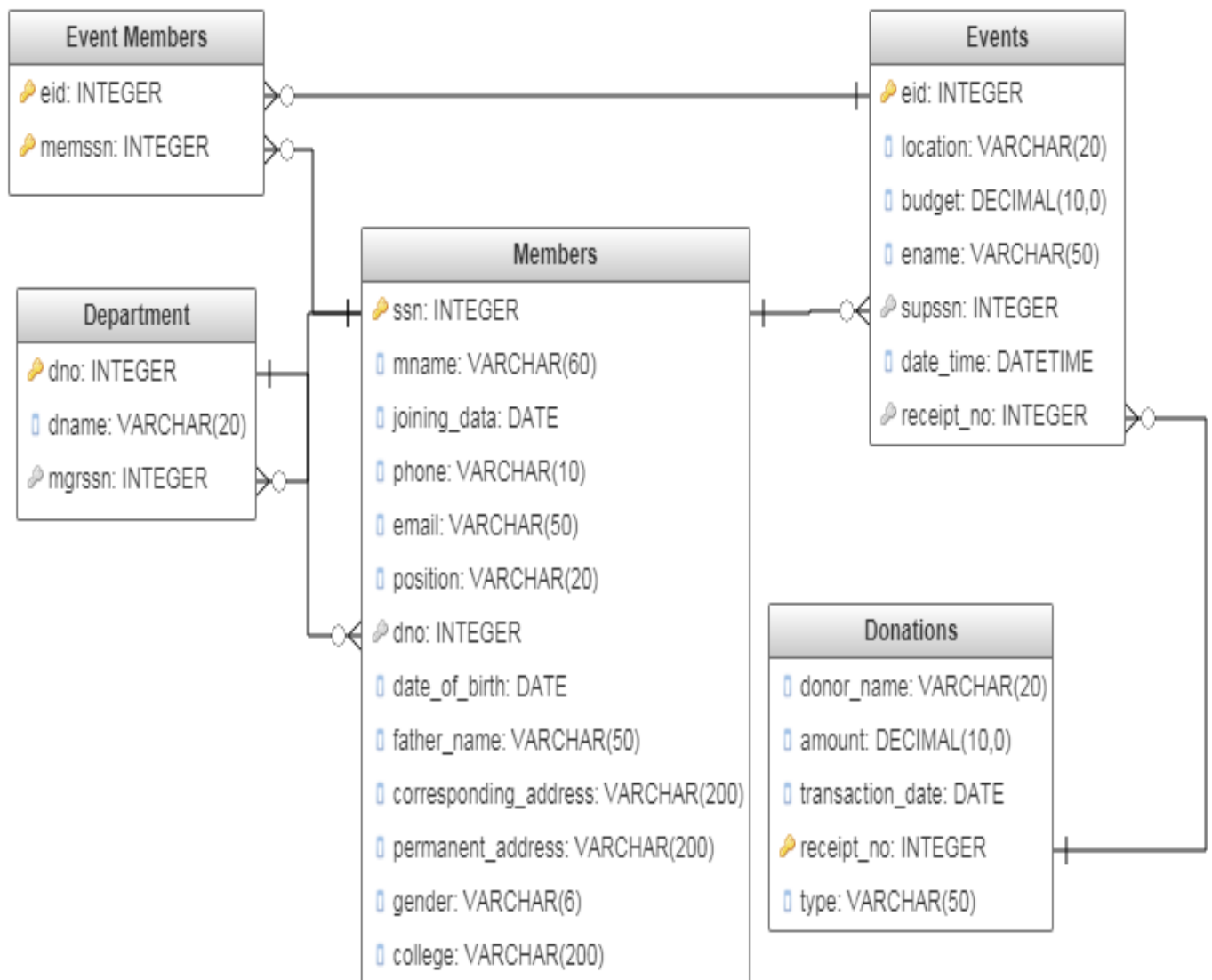


Figure5.1 Schema Diagram

5.2. ER Diagram

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

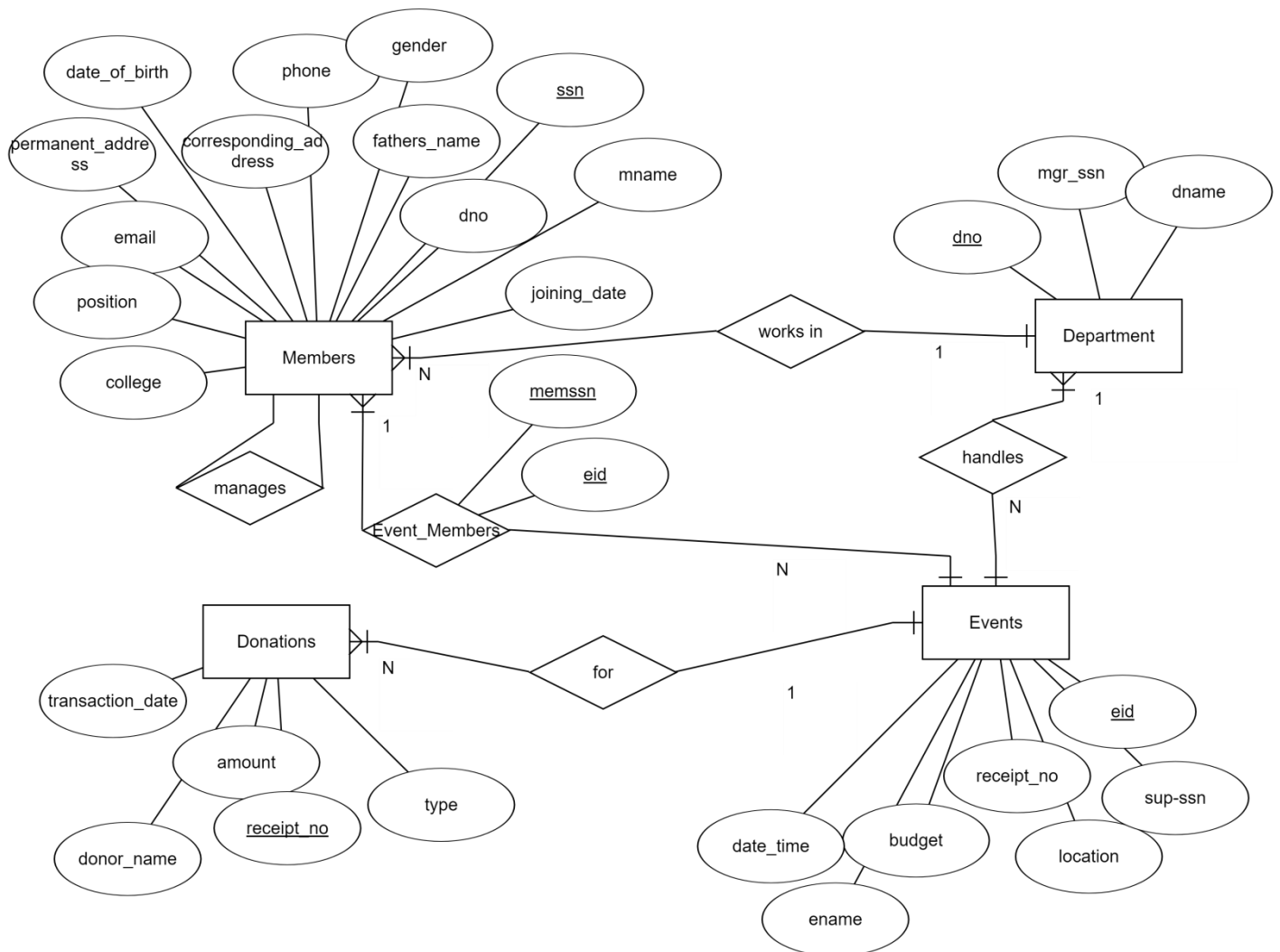


Figure5.2 ER Diagram

5.3. Dataflow Diagram

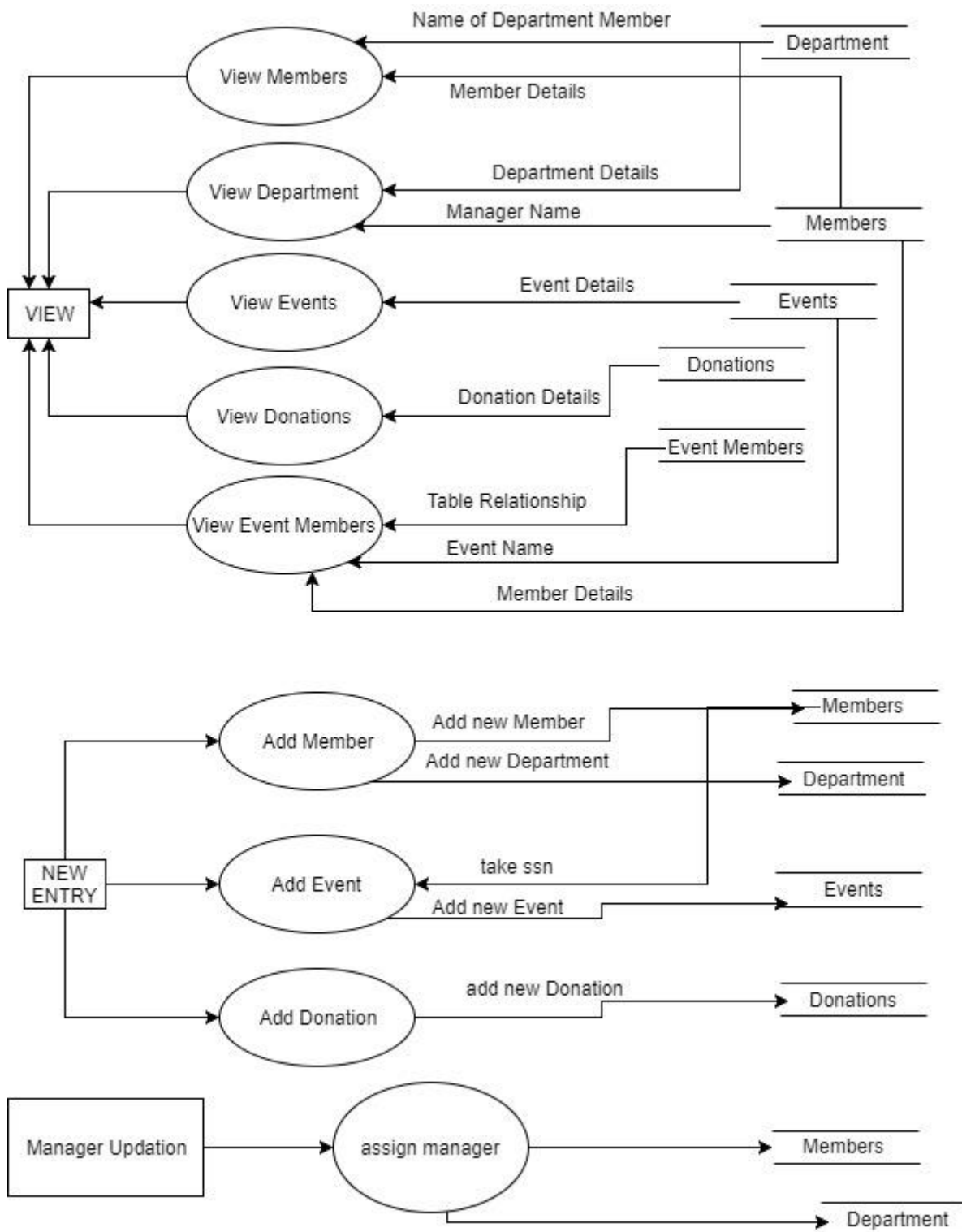


Figure5.3 Dataflow Diagram

5.4. Functional Dependencies

Members:

{ssn} -> Primary Key

{phone, email} -> {mname, father_name, joining_date, corresponding_address, permanent_address}

Department:

{dno} -> Primary Key

{dname} -> {mgrssn}

Events:

{event_id} -> Primary Key

{location} -> {budget, supssn}

Donations:

{receipt_no} -> Primary Key

All of this follows 2NF Normalization level

Chapter 6

Implementation

6.1. Using PromiseJS and MySql Connector

```
connection.query('SELECT m.mname, d.dname FROM Members m, Department d
WHERE d.dno=m.dno ORDER BY m.ssn DESC LIMIT 5;').then((rows) => {
  // Members preview data
  console.log(JSON.stringify(rows));
  members = rows;
  // Query to get latest added departments
  return connection.query('SELECT d.dname, m.mname FROM Department d,
Members m WHERE d.mgrssn=m.ssn ORDER BY d.dno ASC LIMIT 5;')
}).then((rows) => {

  // Departments Preview data
  console.log(JSON.stringify(rows));
  departments = rows;
  flag += 1;

  // Query to get latest added donations
  return connection.query('SELECT donor_name, type FROM Donations ORDER BY
receipt_no DESC LIMIT 5;')
}).then((rows) => {

  // Donations Preview data
  console.log(JSON.stringify(rows));
  donations = rows;
  flag += 1;

  // Query to get latest added Events
  return connection.query('SELECT ename, location FROM Events ORDER BY eid
DESC LIMIT 5;')
}).then((rows) => {

  // Events Preview data
  console.log(JSON.stringify(rows));
  events = rows;
  flag += 1;

  // Query to get the event members
  return connection.query('SELECT ename, mname FROM Events e, Members m,
Event_Members em where e.eid=em.eid and m.ssn=em.memssn LIMIT 5;')
```

```

    // res.render('index', {departments: departments, members: members,
    donations: donations, events: events});

}).then((rows) => {

    // Event Members Preview data
    console.log(JSON.stringify(rows));
    event_members = rows;

    res.render('index', {departments: departments, members: members,
    donations: donations, events: events, event_members: event_members});
});

```

6.2. AngularJS and Form Validation

```

<div class="c-form-bottom" ng-app>

    <form role="form" action="/member/remove" id="contact-form" method="post"
    name="contactForm">
        <div class="form-group" >

            <label for="c-form-name">

                <span class="label-text" style="color:white">NAME:</span>
            </label>
            <input type="text" name="member_name" placeholder="Member's name..."
            class="c-form-subject form-control" required ng-model="name">
            <span class="contact-error" ng-show="contactForm.member_name.$invalid
            && contactForm.member_name.$touched">Give a name to check the
            database...</span>
        </div>

        <div class="form-group">
            <label for="c-form-EmailId">
                <span class="label-text" style="color:white">EMAIL ID:</span>

            </label>
            <input type="email" name="EmailId" placeholder="Members Email
            address..." class="c-form-subject form-control" required ng-
            model="emailID">
            <span class="contact-error" ng-show="contactForm.EmailId.$invalid &&
            contactForm.EmailId.$touched">Email is required</span>
        </div>

        <button ng-show="contactForm.$valid" type="submit" class="btn btn-
        primary btn-md" style="color:white" action="">Remove</button>
    </form>

```

6.3. Making and using Routes on ExpressJS

```
var index = require('./routes/index');
var member = require('./routes/member');
var events = require('./routes/events');
var department = require('./routes/department');
var donation = require('./routes/donation');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use('/', index);
app.use('/member', member);
app.use('/department', department);
app.use('/events', events);
app.use('/donation', donation);
```

6.4. Trigger

A trigger is used here to delete unwanted departments. This trigger deletes the tuple of that department which has no members (the department is deleted after the last member of the department is removed).

```
DROP TRIGGER IF EXISTS deleteDepartment //
```

```
CREATE TRIGGER deleteDepartment AFTER DELETE ON Members FOR EACH ROW
BEGIN
    SET @deptNo = (SELECT dno FROM Department d WHERE NOT EXISTS (SELECT *
FROM Members m WHERE m.dno=d.dno));
    IF (@deptNo IS NOT NULL) THEN
        DELETE FROM Department WHERE dno=@deptNo;
    END IF;
END//
```

6.5. Stored Procedure

Stored procedure helps in storing the data on the database by first checking if the data is present in the database or not and then inserting it. In the “insertNewMember” procedure, a new department is added to the database if the department mentioned is not present.

```
delimiter //
```

```
DROP PROCEDURE IF EXISTS insertNewMember//
CREATE PROCEDURE insertNewMember
(IN name varchar(20), IN join_date date, IN phone_no varchar(10), IN
email_id varchar(100), IN dname varchar(20), IN dateOfBirth date, IN
fatherName varchar(50), IN correspondingAddress varchar(50), IN
permanentAddress varchar(50), IN gender varchar(6), IN college varchar(50))
BEGIN
    IF NOT EXISTS (Select * FROM Department d WHERE d.dname LIKE dname) THEN
        INSERT INTO Department(dname) VALUES(dname);
    END IF;
    IF NOT EXISTS (Select * FROM Members WHERE email_id LIKE email and phone
LIKE phone_no) THEN
        INSERT INTO Members (mname, joining_date, phone, email, dno,
date_of_birth, father_name, corresponding_address, permanent_address,
gender, college) VALUES(name, join_date, phone_no, email_id, (SELECT d.dno
FROM Department d WHERE d.dname LIKE dname), dateOfBirth, fatherName,
correspondingAddress, permanentAddress, gender, college);
    END IF;
END//

delimiter ;
delimiter //
```

```
DROP PROCEDURE IF EXISTS insertNewEvent//

CREATE PROCEDURE insertNewEvent (IN location varchar(20), IN budget
decimal(10,0), IN ename varchar(20), IN supssn varchar(20), IN date_time
datetime, IN receipt_no int)
BEGIN
    IF NOT EXISTS (SELECT * FROM Events e WHERE e.ename = ename and
e.date_time=date_time) THEN
        INSERT INTO Events(location, budget, ename, supssn, receipt_no,
date_time) VALUES (location, budget, ename, supssn, receipt_no, date_time);
    END IF;
END//

delimiter ;
```


6.6. List of Tables

6.6.1. Department

| Field | Type | Null | Key | Default | Extra |
|--------|-------------|------|-----|---------|----------------|
| dno | int(11) | NO | PRI | NULL | auto_increment |
| dname | varchar(20) | YES | | NULL | |
| mgrssn | int(11) | YES | MUL | NULL | |

Table6.1 Deparment

6.6.2. Donations

| Field | Type | Null | Key | Default | Extra |
|------------------|---------------|------|-----|---------|----------------|
| donor_name | varchar(20) | YES | | NULL | |
| amount | decimal(10,0) | YES | | NULL | |
| transaction_date | date | YES | | NULL | |
| receipt_no | int(11) | NO | PRI | NULL | auto_increment |
| type | varchar(50) | YES | | NULL | |

Table6.2 Donations

6.6.3. Event_Members

| Field | Type | Null | Key | Default | Extra |
|--------|---------|------|-----|---------|-------|
| eid | int(11) | NO | PRI | NULL | |
| memssn | int(11) | NO | PRI | NULL | |

Table6.3 Event_Members

6.6.4. Events

| Field | Type | Null | Key | Default | Extra |
|------------|---------------|------|-----|---------|----------------|
| eid | int(11) | NO | PRI | NULL | auto_increment |
| location | varchar(20) | YES | | NULL | |
| budget | decimal(10,0) | YES | | NULL | |
| ename | varchar(50) | YES | | NULL | |
| supssn | int(11) | YES | MUL | NULL | |
| date_time | datetime | YES | | NULL | |
| receipt_no | int(11) | YES | MUL | NULL | |

Table6.4 Events

6.6.5. Members

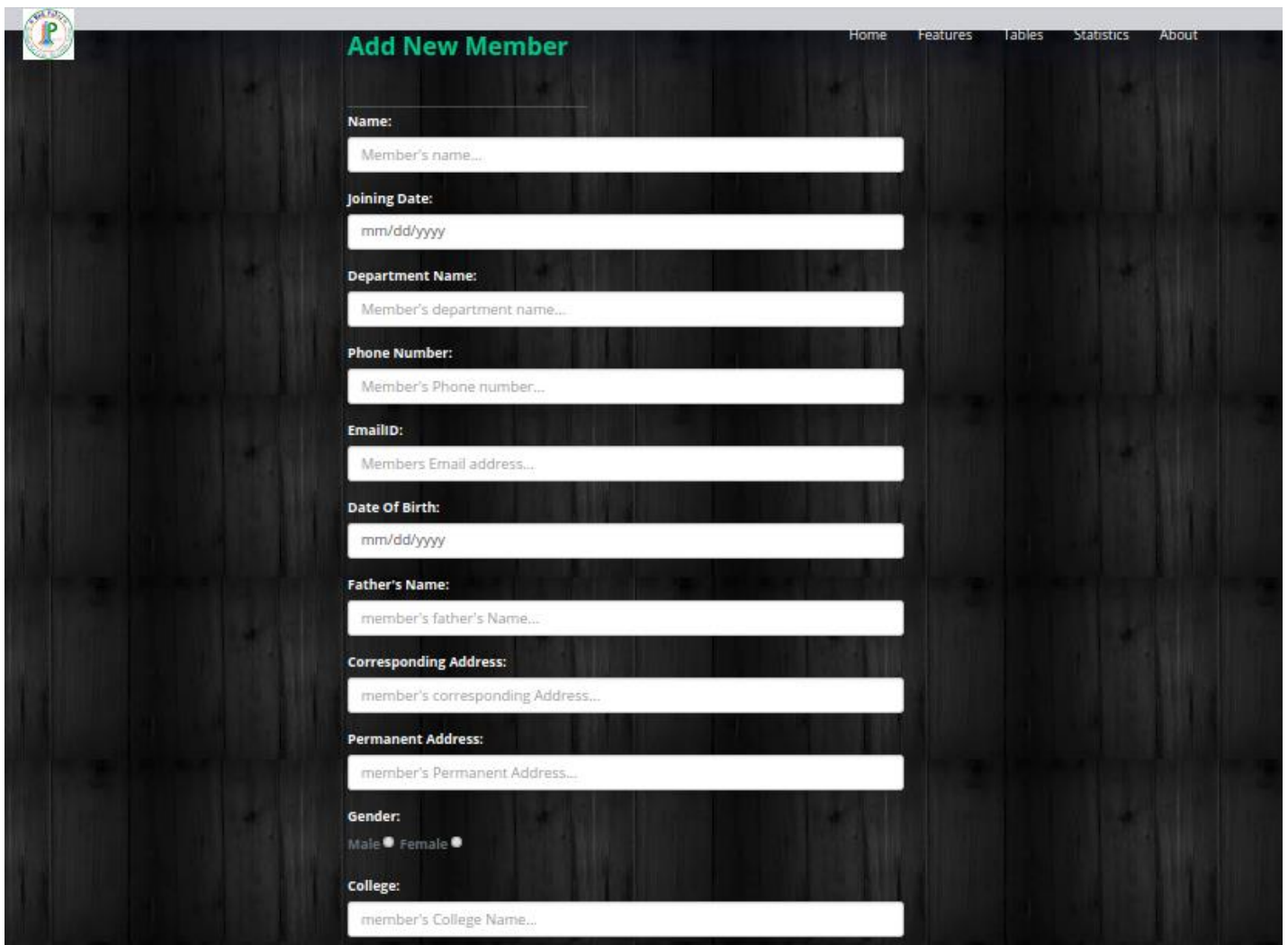
| Field | Type | Null | Key | Default | Extra |
|-----------------------|--------------|------|-----|---------|----------------|
| ssn | int(11) | NO | PRI | NULL | auto_increment |
| mname | varchar(60) | YES | | NULL | |
| joining_date | date | YES | | NULL | |
| phone | varchar(10) | YES | | NULL | |
| email | varchar(50) | YES | | NULL | |
| position | varchar(20) | YES | | NULL | |
| dno | int(11) | YES | MUL | NULL | |
| date_of_birth | date | YES | | NULL | |
| father_name | varchar(50) | YES | | NULL | |
| corresponding_address | varchar(200) | YES | | NULL | |
| permanent_address | varchar(200) | YES | | NULL | |
| gender | varchar(6) | YES | | NULL | |
| college | varchar(200) | YES | | NULL | |

Table6.3 Members

Chapter 7

Testing

It involves the testing of database triggers and logical views which are going to support database refactoring. It performs module testing of database functions, triggers, views, SQL queries etc. It validates database tables, data models, database schema etc. It checks rules of Referential integrity.



Add New Member

Home Features Tables Statistics About

Name:
Member's name...

Joining Date:
mm/dd/yyyy

Department Name:
Member's department name...

Phone Number:
Member's Phone number...

EmailID:
Members Email address...

Date Of Birth:
mm/dd/yyyy

Father's Name:
member's father's Name...

Corresponding Address:
member's corresponding Address...

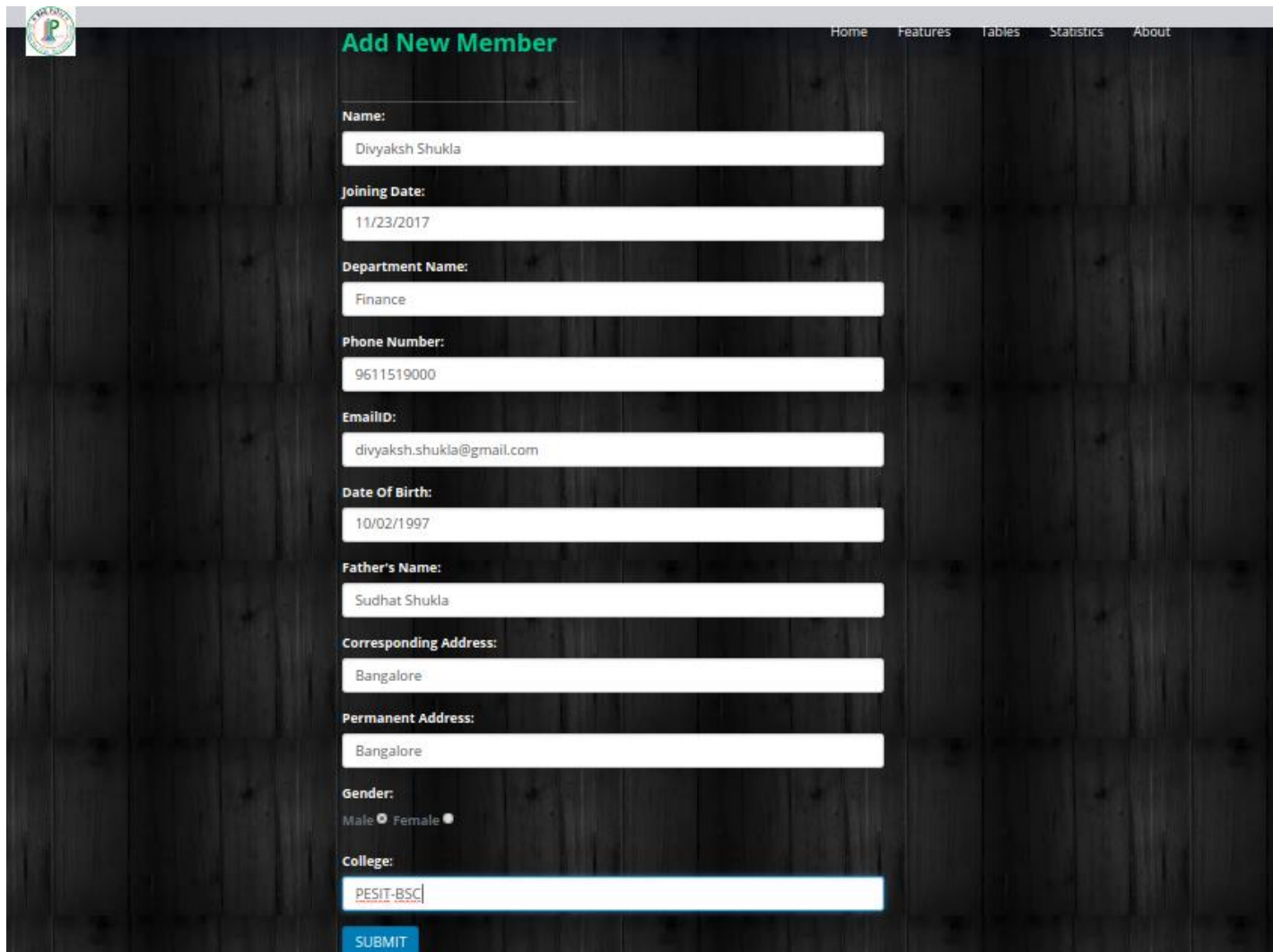
Permanent Address:
member's Permanent Address...

Gender:
Male ☐ Female ☐

College:
member's College Name...

Figure 7.1 Testing for Incomplete Entries

The form is only accepted if the values entered follow the requirements. If an email ID is not valid as per the syntax of the email ID then the “submit” button does not show up. If text is inserted in the phone number tags then again the form is invalid.



Add New Member

Home Features Tables Statistics About

Name:
Divyaksh Shukla

Joining Date:
11/23/2017

Department Name:
Finance

Phone Number:
9611519000

Email ID:
divyaksh.shukla@gmail.com

Date Of Birth:
10/02/1997

Father's Name:
Sudhat Shukla

Corresponding Address:
Bangalore

Permanent Address:
Bangalore

Gender:
Male ☒ Female ☐

College:
PESIT-BSC

SUBMIT

Figure 7.2 Testing for Complete Entries

Chapter 8

Results

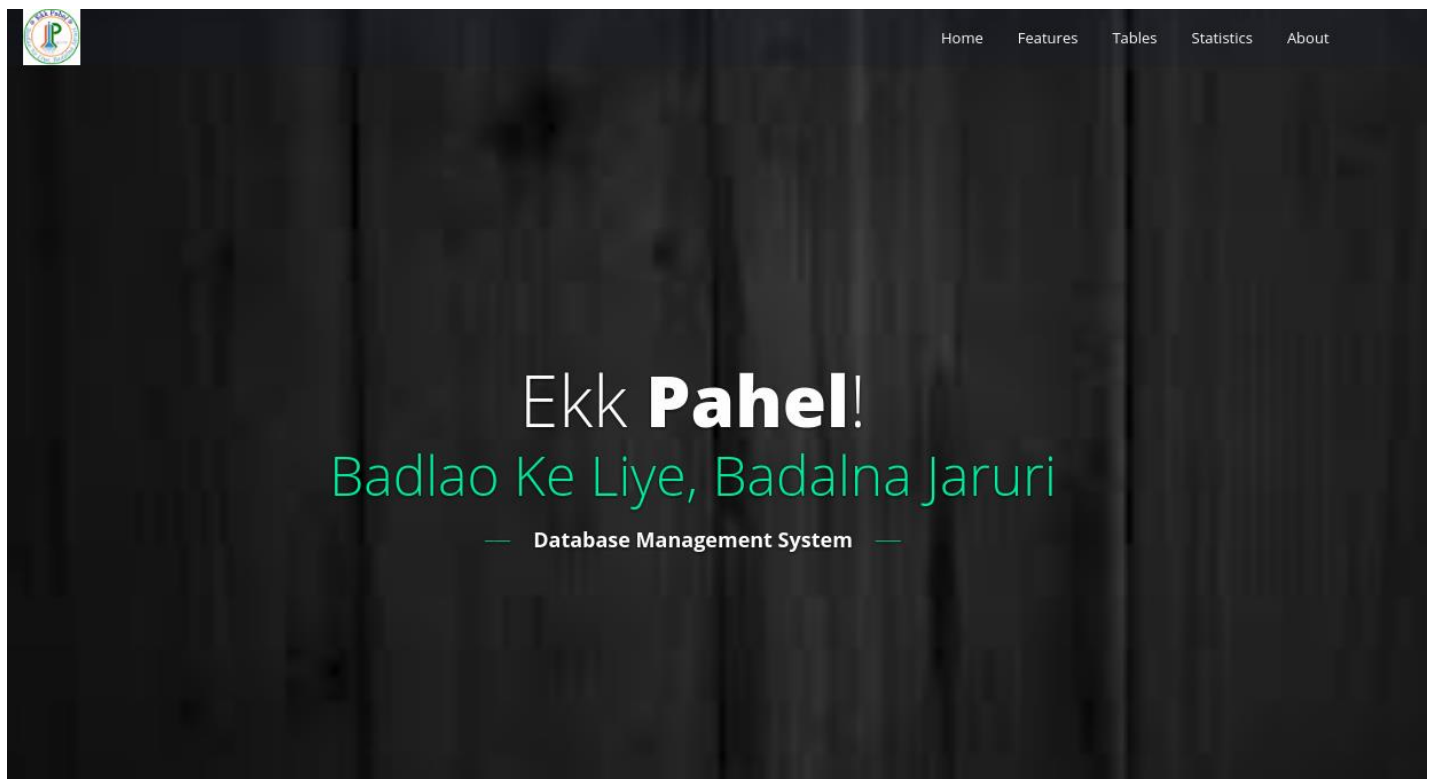


Figure 8.1 Welcome Page

| TABLES | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-----------------|-------------|------------------------|--------------|------------------------|--------------|-------------|-----------|---------------|-----------|-----------------|------------|--|------------|--------------|------------------|-----------------|----------------------|-----------------|------------------------|---------------|-------------------------------|---------------|---------------------------------|---------------|
| The following tables give info about Members, Departments, Donations, Events, Event Members | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div> <div>All</div> <div>MEMBERS</div> <div>DEPARTMENTS</div> <div>DONATIONS</div> <div>EVENTS</div> <div>EVENT MEMBERS</div> </div> | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table> <tr> <th>Member Name</th><th>Department</th></tr> <tr> <td>Divyaksh</td><td>Finance</td></tr> <tr> <td>sumant kumar</td><td>Finance</td></tr> <tr> <td>Md Sanullah</td><td>Marketing</td></tr> <tr> <td>Manisha gupta</td><td>Marketing</td></tr> <tr> <td>Vijendra Pandey</td><td>Organisers</td></tr> </table> | Member Name | Department | Divyaksh | Finance | sumant kumar | Finance | Md Sanullah | Marketing | Manisha gupta | Marketing | Vijendra Pandey | Organisers | <table> <tr> <th>Department</th><th>Manager Name</th></tr> <tr> <td>Finance</td><td>SHIWANI AGRAWAL</td></tr> <tr> <td>Management</td><td>ABHISHEK VISHAL</td></tr> <tr> <td>Donations</td><td>Aniket Vishal</td></tr> <tr> <td>Marketing</td><td>Namita Kumari</td></tr> <tr> <td>Organisers</td><td>Kunal Prateek</td></tr> </table> | Department | Manager Name | Finance | SHIWANI AGRAWAL | Management | ABHISHEK VISHAL | Donations | Aniket Vishal | Marketing | Namita Kumari | Organisers | Kunal Prateek |
| Member Name | Department | | | | | | | | | | | | | | | | | | | | | | | | |
| Divyaksh | Finance | | | | | | | | | | | | | | | | | | | | | | | | |
| sumant kumar | Finance | | | | | | | | | | | | | | | | | | | | | | | | |
| Md Sanullah | Marketing | | | | | | | | | | | | | | | | | | | | | | | | |
| Manisha gupta | Marketing | | | | | | | | | | | | | | | | | | | | | | | | |
| Vijendra Pandey | Organisers | | | | | | | | | | | | | | | | | | | | | | | | |
| Department | Manager Name | | | | | | | | | | | | | | | | | | | | | | | | |
| Finance | SHIWANI AGRAWAL | | | | | | | | | | | | | | | | | | | | | | | | |
| Management | ABHISHEK VISHAL | | | | | | | | | | | | | | | | | | | | | | | | |
| Donations | Aniket Vishal | | | | | | | | | | | | | | | | | | | | | | | | |
| Marketing | Namita Kumari | | | | | | | | | | | | | | | | | | | | | | | | |
| Organisers | Kunal Prateek | | | | | | | | | | | | | | | | | | | | | | | | |
| <table> <tr> <th>Donor name</th><th>Type</th></tr> <tr> <td>ishaan</td><td>cash</td></tr> <tr> <td>Vijendra Pandey</td><td>cash</td></tr> <tr> <td>Mayank Roy</td><td>cash</td></tr> <tr> <td>Ishtyaq Ali</td><td>cash</td></tr> <tr> <td>aakash</td><td>clothes</td></tr> </table> | Donor name | Type | ishaan | cash | Vijendra Pandey | cash | Mayank Roy | cash | Ishtyaq Ali | cash | aakash | clothes | <table> <tr> <th>Event name</th><th>Location</th></tr> <tr> <td>LUNCH WITH NEEDY</td><td>bangalore</td></tr> <tr> <td>NEW YEAR CELEBRATION</td><td>bihar</td></tr> <tr> <td>STUDY KIT DISTRIBUTION</td><td>delhi</td></tr> <tr> <td>GET TOGETHER WITH KIDS OF GOD</td><td>mumbai</td></tr> <tr> <td>Interaction & Skill Development</td><td>kolkata</td></tr> </table> | Event name | Location | LUNCH WITH NEEDY | bangalore | NEW YEAR CELEBRATION | bihar | STUDY KIT DISTRIBUTION | delhi | GET TOGETHER WITH KIDS OF GOD | mumbai | Interaction & Skill Development | kolkata |
| Donor name | Type | | | | | | | | | | | | | | | | | | | | | | | | |
| ishaan | cash | | | | | | | | | | | | | | | | | | | | | | | | |
| Vijendra Pandey | cash | | | | | | | | | | | | | | | | | | | | | | | | |
| Mayank Roy | cash | | | | | | | | | | | | | | | | | | | | | | | | |
| Ishtyaq Ali | cash | | | | | | | | | | | | | | | | | | | | | | | | |
| aakash | clothes | | | | | | | | | | | | | | | | | | | | | | | | |
| Event name | Location | | | | | | | | | | | | | | | | | | | | | | | | |
| LUNCH WITH NEEDY | bangalore | | | | | | | | | | | | | | | | | | | | | | | | |
| NEW YEAR CELEBRATION | bihar | | | | | | | | | | | | | | | | | | | | | | | | |
| STUDY KIT DISTRIBUTION | delhi | | | | | | | | | | | | | | | | | | | | | | | | |
| GET TOGETHER WITH KIDS OF GOD | mumbai | | | | | | | | | | | | | | | | | | | | | | | | |
| Interaction & Skill Development | kolkata | | | | | | | | | | | | | | | | | | | | | | | | |
| <table> <tr> <th>Event name</th><th>Member name</th></tr> <tr> <td>gifting with gratitude</td><td>Ankit Bhagat</td></tr> <tr> <td>gifting with gratitude</td><td>Ankit Bhagat</td></tr> </table> | Event name | Member name | gifting with gratitude | Ankit Bhagat | gifting with gratitude | Ankit Bhagat | | | | | | | | | | | | | | | | | | | |
| Event name | Member name | | | | | | | | | | | | | | | | | | | | | | | | |
| gifting with gratitude | Ankit Bhagat | | | | | | | | | | | | | | | | | | | | | | | | |
| gifting with gratitude | Ankit Bhagat | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 8.2 Tables

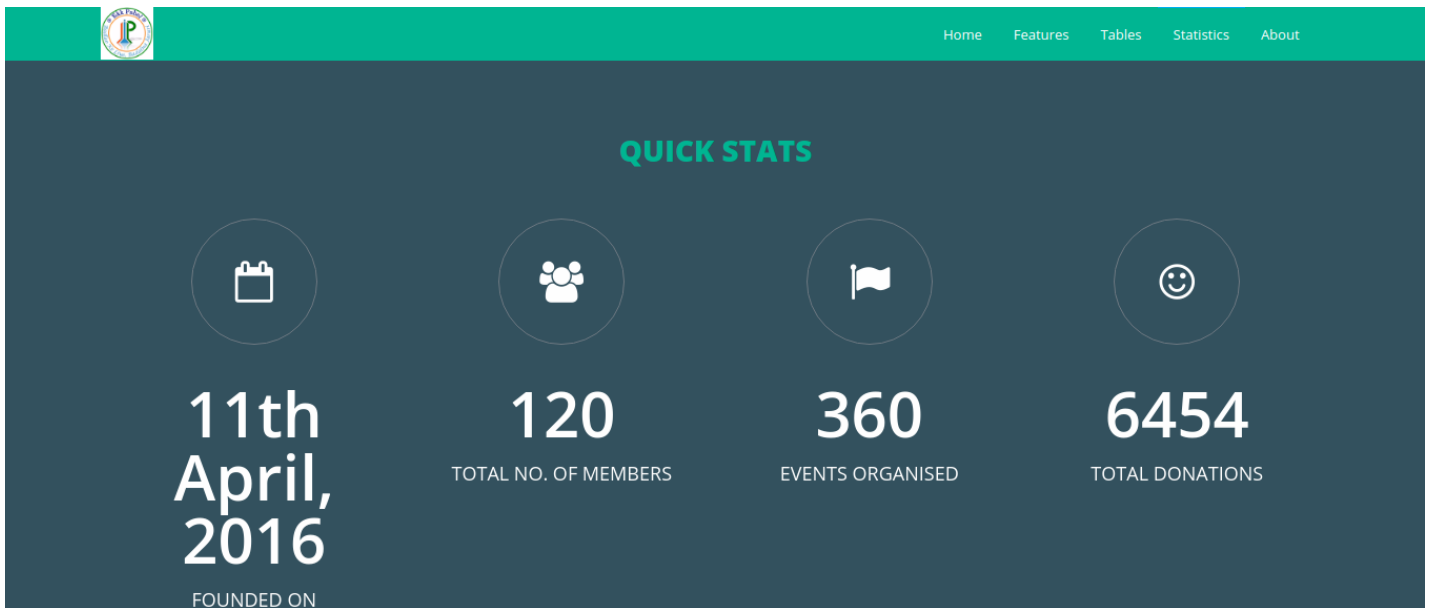


Figure 8.3 Statistics

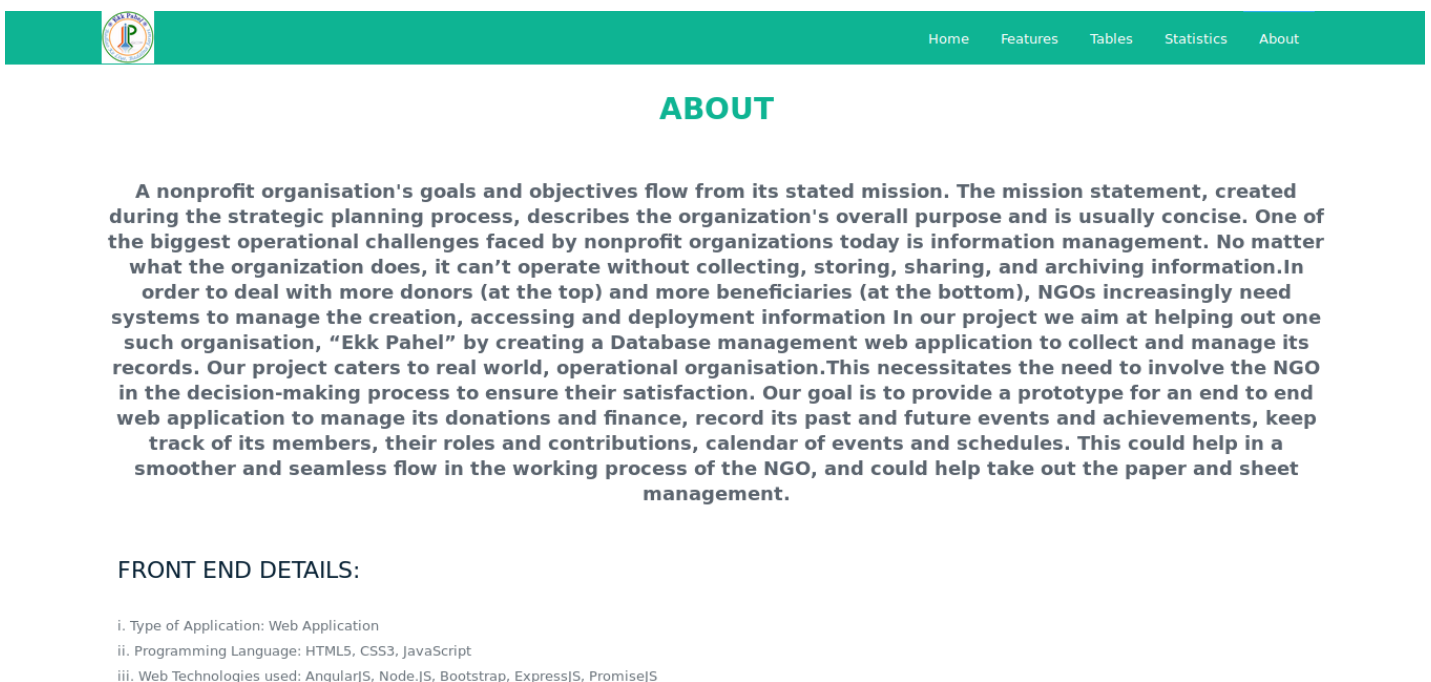



Figure 8.4 About Us



Members Table

[Home](#)
[Features](#)
[Tables](#)
[Statistics](#)
[About](#)

Name:

Joining Date:

Phone:

Email:

Department:

Date of Birth:

Father's name:

Corresponding Address:

Permanent Address:

Gender: ☒ Male ☐ Female

College:

| Name | Joining Date | Phone | Email | Department | DOB | Father's name | Corresponding Address | Gender | College |
|------------------|---|------------|------------------------------|------------|---|-------------------------------|--|--------|---|
| ABHISHEK RANJAN | Fri Jun 03 2016 00:00:00 GMT+0530 (IST) | 9790608434 | abhishek.ranjan@gmail.com | Finance | Thu Jun 11 1998 00:00:00 GMT+0530 (IST) | SATYENDRA THAKUR | vit university, vellore , tamilnadu | Male | vit university |
| Shalin Prasad | Fri Jun 03 2016 00:00:00 GMT+0530 (IST) | 9473139742 | shalinprasad24@gmail.com | Finance | Sat Sep 26 1998 00:00:00 GMT+0530 (IST) | Lalan kumar | Banglore | Male | Dayananda sagar university |
| Nikhil Kumar | Sat Jun 04 2016 00:00:00 GMT+0530 (IST) | 8867638254 | nk27101996@gmail.com | Finance | Thu Oct 10 1996 00:00:00 GMT+0530 (IST) | Pravind Kumar | Svlake view aparments flt no 108 hosur road Bangalore | Male | Pesit south campus |
| ABHISHEK VISHAL | Sat Jun 04 2016 00:00:00 GMT+0530 (IST) | 7760876451 | abhishekvishal3006@gmail.com | Finance | Wed Jun 30 1993 00:00:00 GMT+0530 (IST) | AKHILESHWAR PRASAD SRIVASTAVA | Vacmet India Limited, 110 km stone delhi mathura NH 2, chatta, dist: mathura. | Male | Vacmet India Limited, Mathura |
| sneh kashyap | Sat Jun 04 2016 00:00:00 GMT+0530 (IST) | 9015220042 | skswastika32@gmail.com | Finance | Thu Mar 29 1990 00:00:00 GMT+0530 (IST) | Lalit Kumar | A 501, Tulip Apartments, Sec-69, Gurgaon-122001 | Female | Healthkart |
| Atishey Jain | Sun Jun 05 2016 00:00:00 GMT+0530 (IST) | 7209813863 | atisheyjain97@gmail.com | Finance | Wed Apr 16 1997 00:00:00 GMT+0530 (IST) | Sanjeev Jain | Flat No.513,Vrr Nest, Near Pesit Bsc,Hosur road,Bangalore-560100 | Male | Pesit Bsc |
| Mohit Agrawal | Wed Jul 27 2016 00:00:00 GMT+0530 (IST) | 8408080666 | tatamotorslucknow@gmail.com | Finance | Mon Dec 15 1980 00:00:00 GMT+0530 (IST) | Mr. Shekhar Agrawal | B4/704, Edenn Towers, Opposite Indian Oil Petrol Pump, Wakad, Pune-57 | Male | Senior Manager at MAHINDRA AND MAHINDRA LIMITED, PUNE |
| Surya bhan Kumar | Sat Jul 30 2016 00:00:00 GMT+0530 (IST) | 9905113058 | surya.bhan2749@gmail.com | Finance | Tue Jan 20 1998 00:00:00 GMT+0530 (IST) | Late Arbind kumar singh | LAHANG BHAWAN adarsh colony in the street of Dr ARUN SINGH club road, in front of bdo block,ara,802301 | Male | Delhi university |
| Ayush Baid | Sat Jul 30 2016 00:00:00 GMT+0530 (IST) | 8981145949 | abayushbaid0.ab@gmail.com | Finance | Sat Oct 04 1997 00:00:00 GMT+0530 (IST) | Pradip Baid | 4 Rameshwar Malia Lane,4th Floor,Howrah-711101 | Male | St. Xaviers College, Kolkata |

Figure 8.5 View Members Table



Donations Table

[Home](#)
[Features](#)
[Tables](#)
[Statistics](#)
[About](#)

| Donor name | Type | Amount | Transaction Date | Receipt No |
|-------------------|---------|--------|---|------------|
| Megha Agarwal | cash | 1300 | Thu Nov 02 2017 00:00:00 GMT+0530 (IST) | 123 |
| alankrita | cash | 1000 | Sat Sep 09 2017 00:00:00 GMT+0530 (IST) | 1234 |
| Rishav Raj | clothes | | Mon Dec 12 2016 00:00:00 GMT+0530 (IST) | 1357 |
| Ankit singh | cash | 700 | Thu Feb 23 2017 00:00:00 GMT+0530 (IST) | 2345 |
| Shiksha Rangroo | clothes | | Thu Jun 23 2016 00:00:00 GMT+0530 (IST) | 2468 |
| chepak | books | | Tue Aug 09 2016 00:00:00 GMT+0530 (IST) | 3456 |
| Ashoutosh Kashyap | clothes | | Thu Mar 16 2017 00:00:00 GMT+0530 (IST) | 4567 |
| aakash | clothes | | Fri Apr 07 2017 00:00:00 GMT+0530 (IST) | 5678 |
| Ishtyaq Ali | cash | 500 | Mon May 29 2017 00:00:00 GMT+0530 (IST) | 6789 |
| Mayank Roy | cash | 1200 | Fri Jul 01 2016 00:00:00 GMT+0530 (IST) | 7890 |

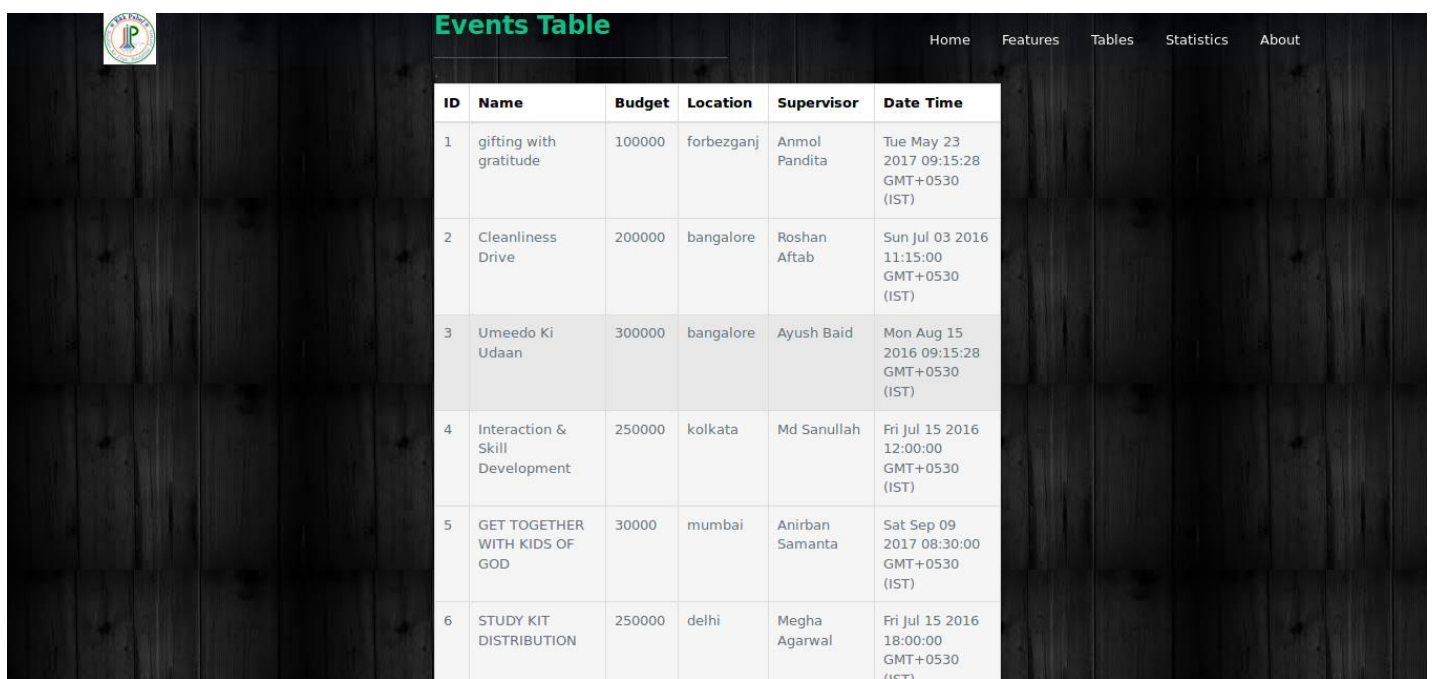
Figure 8.6 View Donations Table



Departments Table

| Name | Manager name |
|------------|-----------------|
| Finance | Divyaksh |
| Management | ABHISHEK VISHAL |
| Donations | Aniket Vishal |
| Marketing | Namita Kumari |
| Organisers | Kunal Prateek |

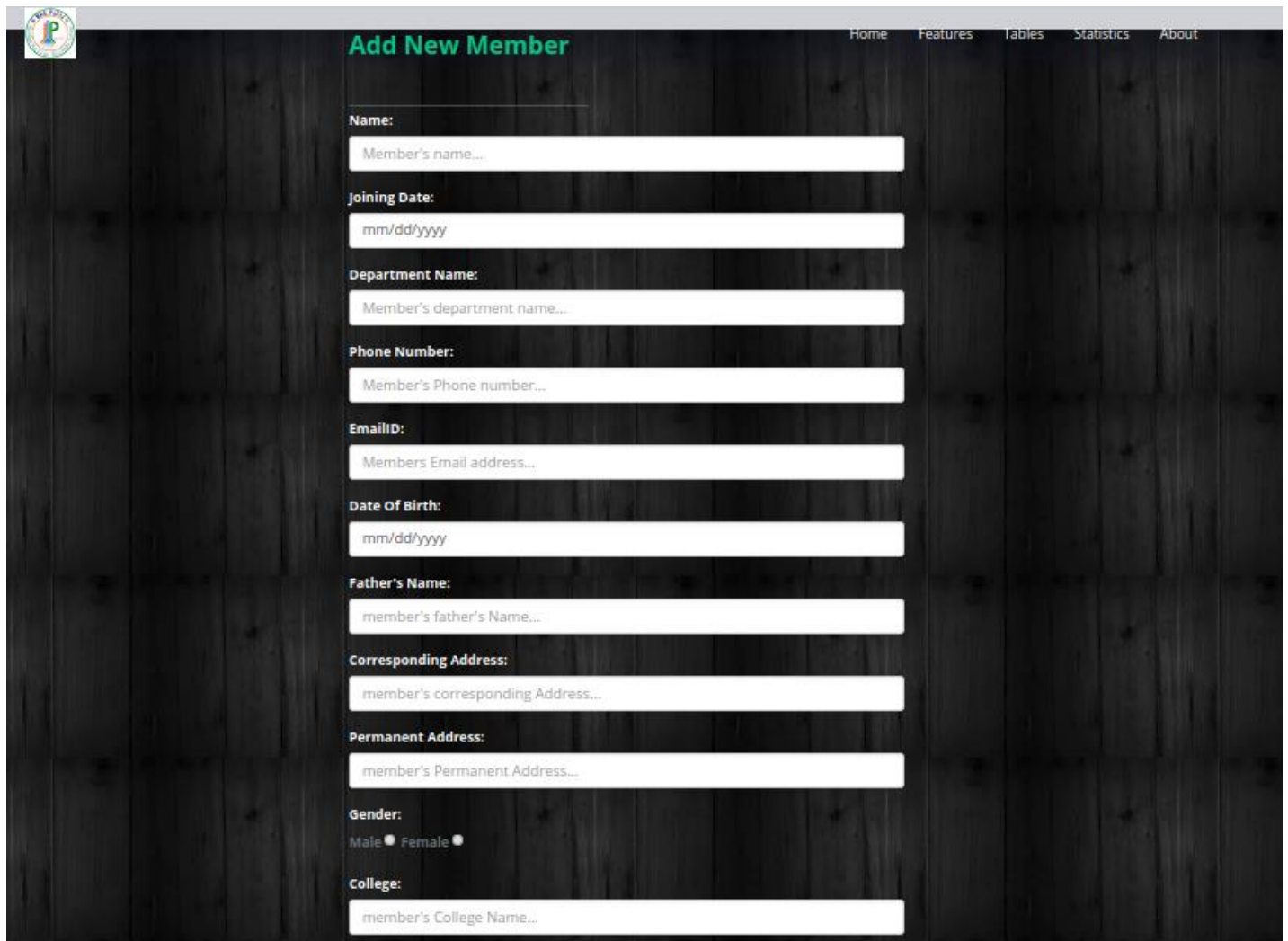
Figure 8.7 View Departments Table



Events Table

| ID | Name | Budget | Location | Supervisor | Date Time |
|----|---------------------------------|--------|------------|-----------------|---|
| 1 | gifting with gratitude | 100000 | forbezganj | Anmol Pandita | Tue May 23 2017 09:15:28 GMT+0530 (IST) |
| 2 | Cleanliness Drive | 200000 | bangalore | Roshan Aftab | Sun Jul 03 2016 11:15:00 GMT+0530 (IST) |
| 3 | Umeedo Ki Udaan | 300000 | bangalore | Ayush Baid | Mon Aug 15 2016 09:15:28 GMT+0530 (IST) |
| 4 | Interaction & Skill Development | 250000 | kolkata | Md Sanullah | Fri Jul 15 2016 12:00:00 GMT+0530 (IST) |
| 5 | GET TOGETHER WITH KIDS OF GOD | 30000 | mumbai | Anirban Samanta | Sat Sep 09 2017 08:30:00 GMT+0530 (IST) |
| 6 | STUDY KIT DISTRIBUTION | 250000 | delhi | Megha Agarwal | Fri Jul 15 2016 18:00:00 GMT+0530 (IST) |

Figure 8.8 View Events Table



Add New Member

Home Features Tables Statistics About

Name:
Member's name...

Joining Date:
mm/dd/yyyy

Department Name:
Member's department name...

Phone Number:
Member's Phone number...

EmailID:
Members Email address...

Date Of Birth:
mm/dd/yyyy

Father's Name:
member's father's Name...

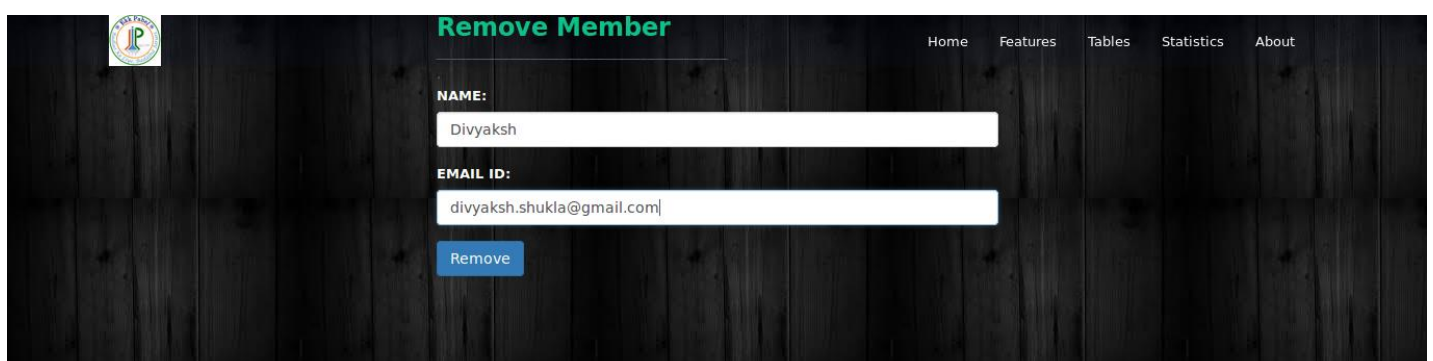
Corresponding Address:
member's corresponding Address...

Permanent Address:
member's Permanent Address...

Gender:
Male ☐ Female ☐

College:
member's College Name...

Figure 8.10 Add Member Form



Remove Member

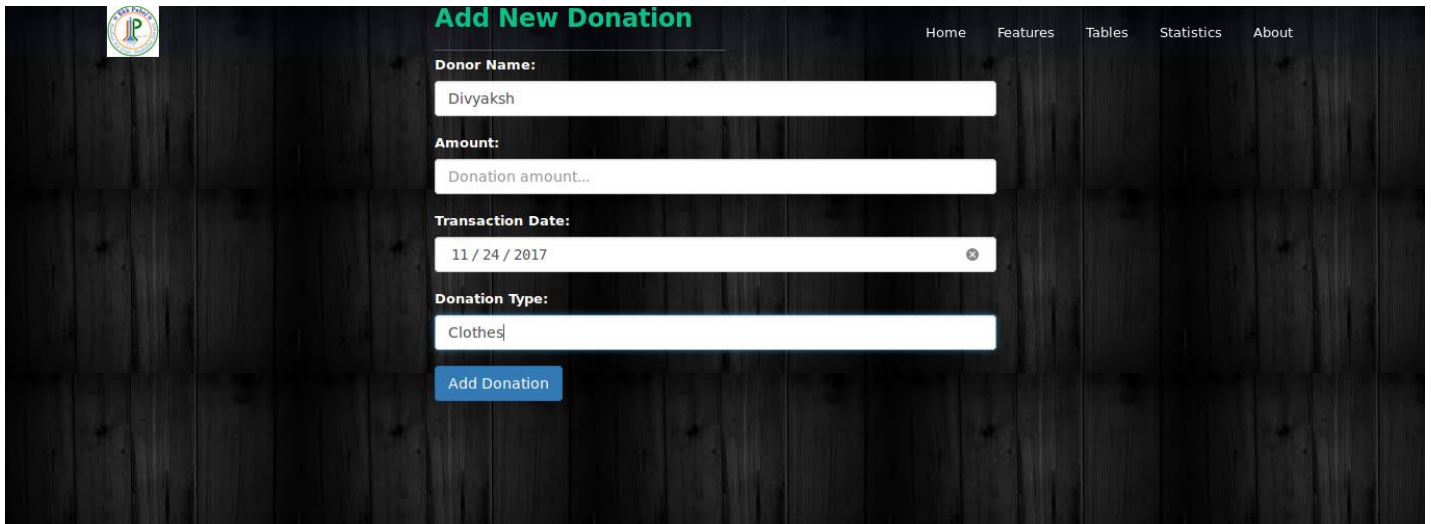
Home Features Tables Statistics About

NAME:
Divyaksh

EMAIL ID:
divyaksh.shukla@gmail.com]

Remove

Figure 8.11 Remove Member Form



Add New Donation

Home Features Tables Statistics About

Donor Name:
Divyaksh

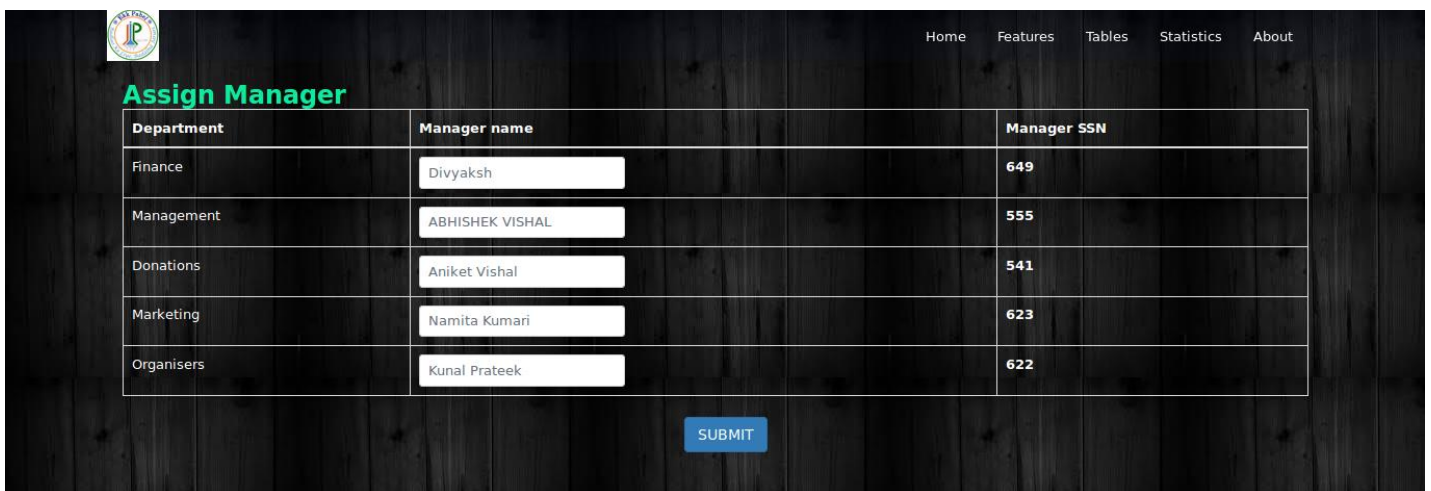
Amount:
Donation amount...

Transaction Date:
11 / 24 / 2017

Donation Type:
Clothes

Add Donation

Figure 8.12 Add Donations Form



Assign Manager

Home Features Tables Statistics About

| Department | Manager name | Manager SSN |
|------------|-----------------|-------------|
| Finance | Divyaksh | 649 |
| Management | ABHISHEK VISHAL | 555 |
| Donations | Aniket Vishal | 541 |
| Marketing | Namita Kumari | 623 |
| Organisers | Kunal Prateek | 622 |

SUBMIT

Figure 8.13 Assign Manager Form

Chapter 9

Conclusion

A database management system for the NGO Ekk Pahel was completed as per planned. The website allows the admin to view database tables, add members and remove members, assign managers to departments, add members to Events, Add donations made by volunteers and hence successfully maintain a database for the NGO.

Further we plan to create graphical representations of the tables for easier access and to view the trend over the past months for better analysis of the data stored in the databases.

Chapter 10

References

- **Literature Survey**
 - <http://ekkpahel.com/>
 - insidebigdata.com
- **Diagrams**
 - <https://erdplus.com/#/standalone>
 - <https://dashboard.genmymodel.com/subscribe>
- **SQL**
 - www.tutorialspoint.com
 - dev.mysql.com
- **Javascript**
 - developer.mozilla.org
- **NodeJs,ExpressJs, AngularJs**
 - www.w3schools.com
- **HTML,CSS**
 - www.tutorialspoint.com