**CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **CHAPTER** | **TOPIC** | **PAGE NO.** |
| 1 | Introduction | * 1. Problem statement   2. Objectives   3. Scope | 2 |
| 2 | Literature Study | 2.1 Traditional Database  2.2 Pros and Cons of Traditional Database  2.3 Downfall of Traditional DBMS  2.4 Introduction to DBMS  2.5 Inactive areas for the use of DBMS  2.6 Advantages of DBMS  2.7 Components of DBMS | 3  3  4  4  5  5  6 |
| 3 | Hardware and Software Specification | 3.1 Hardware requirements  3.2 Software requirements | 7  7 |
| 4 | System Design | 4.1 Schema Diagram  4.2 ER Diagram | 8  8 |
| 5 | Implementation | 5.1 HTML  5.2 Bootstrap  5.3 CSS  5.4 MYSQL | 9  10  10  10 |
| 6 | Code Snippets  (HTML, CSS, PHP, Screenshot) | 6.1 Sign In  6.2 Register Page  6.3 Home Page  6.4 Profile Page  6.5 Create Table Commands  6.6 Normalization  6.7 Database Cnnection | 11-22  23  24-26  27  28-29  30  31 |
| 7 | Triggers | 7.1 Trigger Code Snippet | 32 |
| 8 | Conclusion |  | 33 |
| 9 | References |  | 34 |

**CHAPTER 1:**

**INTRODUCTION**

* 1. **Problem Statement**

The concept of **EVENT Database** is vast by itself. The term EVENT is defined as planned public or social occasion. The whole objective of this project is to provide a first-class experience for end users to manage their budget and expenses of their event to take place at a particular venue with an exact date and time.

* 1. **Objectives**

The aim of this project, **EVENT Database** is to meet the following objectives:

1. To provide a great experience to the user.
2. To help the end users easily access, edit, add and delete information.
3. To ensure safety of the information in the database.
4. To ultimately provide an interactive and user-friendly platform for the users to have a good experience with least number of glitches and delays.
5. To help the user to keep track of his/her events.
   1. **Scope**

The scope of this project is to create a database which includes correct information of the customer’s events, budgets and expenses and to provide a robust and error free experience.

**CHAPTER 2:**

**LITERATURE STUDY**

**2.1 Traditional Database**

In the early days of computing, data management and storage was a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g. handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing. This separation of data also led to unnecessary redundancy and a high rate of unreliable and inconsistent data

**2.2 Pros and Cons of Traditional Database**

**Pros:**

* Simple
* Low initial investment

**Cons:**

* Separated ownership
* Unmanaged redundancy
* Data inconsistency
* Lack of data sharing
* Expensive on the long run

**2.3 Downfall of Traditional Database Management System:**

Conceived in a relatively centralized era when software was deployed in static environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere. Today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objectives.

However, legacy database technologies fall short in serving the needs of today’s distributed and cloud environments for the following reasons:

* Inadequate failover capabilities
* Latency issues
* Insufficient provisions during peak demands
* Lack of high availability at all times
* Increasing operational costs
* Inability to meet the demands of global markets

For all of these reasons, traditional databases are. unable to deliver results in a rapidly growing environment where the workload is geographically distributed across heterogeneous data centres. Upgrading to a more distributed data model is costly and complicated and your DBAs can’t just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

**2.4 Introduction to the Database Management System**

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of Data Meaningful data combines to form information. Hence, information is interpreted data-data provided with semantics. MS ACCESS is one of the most common database management software.

Database systems are meant to handle large collections of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crash or attempts at unauthorised access.

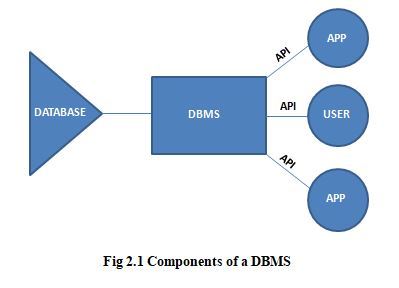
**2.5 Inactive areas for the use of DBMS**

* Airlines: Reservations, Schedules etc.
* Telecom: Calls made, Customer details, Network usage etc.
* Universities: Registration, Results, Grades, etc.
* Sales: Products, Purchases, Customers etc.
* Banking: All transactions, Customer details etc.

**2.6 Advantages of DBMS**

* **Data Independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
* **Efficient data access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored in external devices.
* **Data Integrity and Security:** If data is accessed through DBMS, the DBMS can enforce integrity constraint on the data.
* **Data Administration:** When several users share the data, integrating the administration of the data can offer major improvements. Experienced professionals understand the nature of data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.

**2.7 Components of DBMS**



* **Users**: Users may be of any kind, such as data base administrators, system developers or database users.
* **Database application**: Database application may be Departmental, Personal, Organizational and /or Internal
* **DBMS**: Software that allows users to create and manipulate database access.
* **Database**: Collection of logical data as a single unit.

**CHAPTER 3:**

**HARDWARE & SOFTWARE SPECIFICATION**

**3.1 Hardware Requirements**

* Processor: PENTIUM 4 or above
* RAM: 1GB or above
* Hard Disk: 500MB or above

**3.2 Software Requirements**

Technologies Used:

* Front End: HTML, CSS, JS(java scripts)
* Connection/Controller: Node.js(express js)
* Back-End Database: MySQL

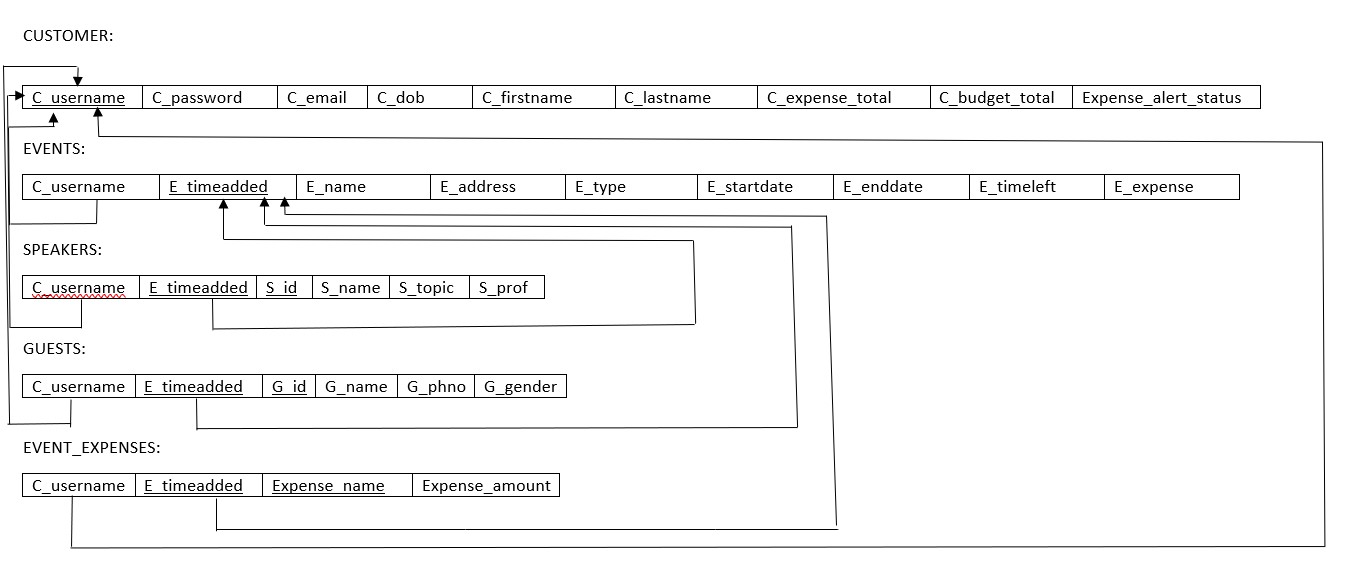
Software:

* Text Editor: Microsoft Visual Studio Code
* Server: MYSQL
* Operating System: Windows XP or above
* Database Support: MySQL
* Back-End: MySQL
* Browser: Any browser that supports html, css and JS.

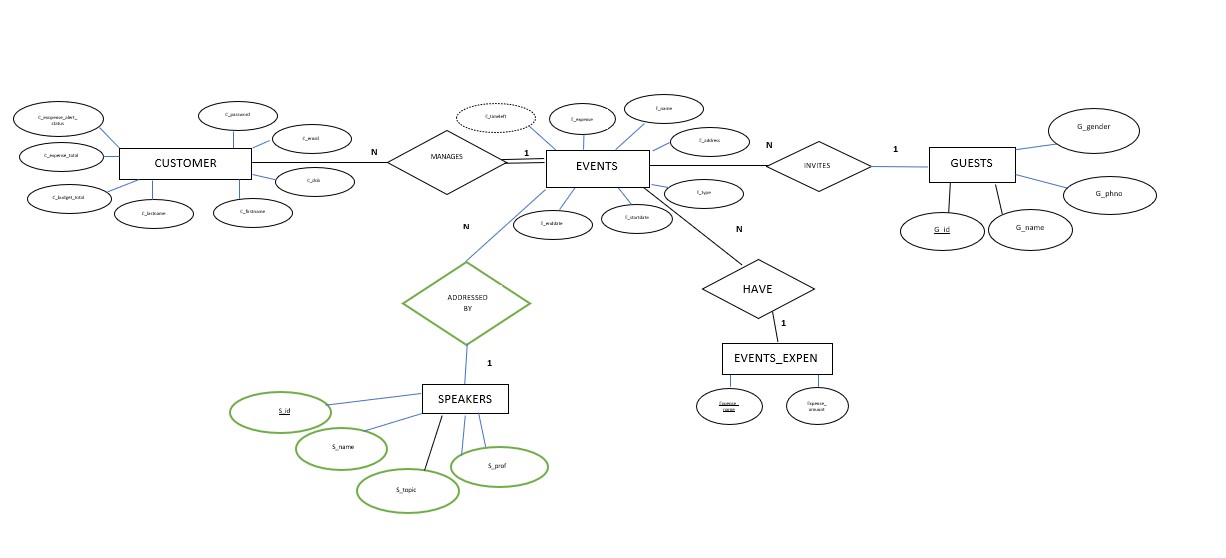
**CHAPTER 4:**

**SYSTEM DESIGN**

**4.1 Schema Diagram:**

****

**4.2 ER Diagram:**

****

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 HTML**

HTML is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the HTML standard.

It was published in October 2014 by the World Wide Web Consortium (W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers, etc. HTML is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML is also a candidate for cross-platform mobile applications, because it includes features designed with low-powered devices in mind.

Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as<main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and<menu> have been changed, redefined or standardized.

The APIs and Document Object Model(DOM) are now fundamental parts of the HTML specification and HTML also better defines the processing for, any invalid documents.

**5.2 Bootstrap**

**Bootstrap** is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and JavaScript based design templates for typography, forms buttons, navigation and other interface components.

Bootstrap is the third-most-starred project on GitHub, with more than 1,35,000 stars, behind only freeCodeCamp and marginally behind Vue.js framework. According to Alexa Rank, Bootstrap getbootstrap.com is in the top 2000 in US while vuejs.org is in top 7000 in US.

**5.3 Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristic, enable multiple web pages to share formatting by specifying the relevant CSS file, and reduce complexity and repetition in the structural content.

* 1. **MYSQL (Structured Query Language)**

**MySQL** is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

**CHAPTER 6**

**CODE SNIPPETS**

**6.1 Sign In**

**6.1.1 HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/css/material-design-iconic-font.min.css">

    <link rel="stylesheet" href="../style.css">

</head>

<body>

    <div class="limiter">

        <div class="container-login">

            <div class="wrap-login">

                <form class="login-form validate-form" action="/auth/login" method="post">

                    <span class="login-form-title">

                        Welcome

                    </span>

                    <div class="wrap-input validate-input" data-validate="Enter valid email">

                        <input type="text" id="email-input" class="input" name="email">

                        <span class="focus-input" data-placeholder="Email"></span>

                    </div>

                    <div class="wrap-input validate-input" data-validate="Enter password">

                        <span class="btn-show-pass">

                            <i class="zmdi zmdi-eye"></i>

                        </span>

                        <input type="password" class="input" name="password">

                        <span class="focus-input" data-placeholder="Password"></span>

                    </div>

                    <div class="container-login-form-btn">

                        <div class="wrap-login-form-btn">

                            <button class="login-form-btn"> Login </button>

                        </div>

                    </div>

                    {{#if message}}

            <h4 id="errormsg" class="alert alert-info mt-4">{{message}}</h4>

            <script>

            setTimeout('$("#errormsg").hide()',3500);

        </script>

        {{/if}}

                    <div class="text-centre">

                        <span class="txt1">

                            Don't have an account?

                        </span>

                        <a class="txt2" href="/register">Sign Up</a>

                    </div>

                </form>

            </div>

            <br>

        </div>

    </div>

    <script src="../jquery.js"></script>

    <script src="../JS\_main.js"></script>

</body>

</html>

**6.1.2 CSS**

@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');

\*

{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body, html

{

    height: 100%;

    font-family:'Poppins' sans-serif;

    animation: transitionIn 0.80s;

}

@keyframes transitionIn {

    from {

        opacity:0;

        transform: rotateX(-10deg);

    }

    to{

        opacity:1;

        transform: rotateX(0);

    }

}

input

{

    outline: none;

    border: none;

}

input:focus::-webkit-input-placeholder

{

    border-color: transparent !important;

}

input::-webkit-input-placeholder

{

    color: #adadad;

}

.limiter

{

    width: 100%;

    height: 100%;

    margin: 0 auto;

}

.container-login

{

    width: 100%;

    min-height: 100%;

    display: flex;

    display: -webkit-box;

    display: -webkit-flex;

    flex-wrap: wrap;

    justify-content: center;

    align-items: center;

    padding: 15px;

    background: #ddd;

}

.wrap-login

{

    width: 350px;

    background: #ddd;

    border-radius: 10px;

    overflow: hidden;

    padding: 66px 55px 33px 55px;

    box-shadow: 0 0 3px 3px #dddddd,

                6px 6px 8px 4px rgba(136,136,136,0.7),

                -6px -6px 8px 4px rgba(244,244,244,0.7);

}

.login-form

{

    width: 100%;

}

.login-form-title

{

    display: block;

    font-size: 30px;

    font-family: 'Poppins';

    font-weight: bold;

    color: #333333;

    line-height: 1.2;

    text-align: center;

    padding-bottom: 60px;

}

.wrap-input

{

    width: 100%;

    position: relative;

    border-bottom: 2px solid #adadad;

    margin-bottom: 38px;

}

.input

{

 font-family: 'Poppins';

 font-size: 15px;

 color: #4caf50;

 line-height: 1.2;

 display: block;

 width: 94%;

 height: 45px;

 background: transparent;

 padding: 0 5px;

}

.focus-input

{

    position: absolute;

    display: block;

    width: 100%;

    height: 100%;

    top: 0;

    left: 0;

    pointer-events: none;

}

.focus-input:before

{

    content: "";

    position: absolute;

    display: block;

    bottom: -2px;

    left: 0;

    width: 0;

    height: 2px;

    background: #4caf50;

    transition: all 0.4s;

}

.focus-input:after

{

    font-family: 'Poppins';

    font-size: 15px;

    color: #999999;

    line-height: 1.2;

    content: attr(data-placeholder);

    display: block;

    width: 100%;

    position: absolute;

    top: 16px;

    left: 0px;

    padding-left: 5px;

    transition: all 0.4s;

    -webkit-transition: all 0.4s;

}

.input:focus + .focus-input:after

{

    top:-15px;

}

.input:focus + .focus-input:before

{

    width: 100%;

}

.has-val.input + .focus-input::after

{

    top:-15px;

}

.has-val.input:focus + .focus-input:before

{

    width: 100%;

}

.btn-show-pass

{

    font-size: 15px;

    color: #999999;

    display:-webkit-flex;

    display:-webkit-box;

    display:flex;

    align-items: center;

    position: absolute;

    height: 100%;

    top: 0;

    right: 0;

    padding-right: 5px;

    cursor: pointer;

    transition: all 0.4s;

}

.btn-show-pass:hover

{

    color: #4caf50

}

.btn-show-pass:active

{

    color: #4caf50

}

button

{

    outline: none !important;

    border: none;

    background: transparent;

}

.txt1

{

    font-family: 'Poppins';

    font-size: 13px;

    color: #666666;

    line-height: 1.5;

}

.txt2

{

    font-family: 'Poppins';

    font-size: 13px;

    color: #333333;

    font-weight: bold;

    line-height: 1.5;

    text-decoration: none;

    transition: all 0.4s;

}

.txt2:hover{

    text-decoration: underline;

}

.container-login-form-btn

{

    display:-webkit-box;

    display:-webkit-flex;

    display:flex;

    flex-wrap: wrap;

    justify-content: center;

    padding-top: 20px;

}

.wrap-login-form-btn

{

    width: 100%;

    display: block;

    position: relative;

    z-index: 1;

    border-radius: 25px;

    overflow: hidden;

    margin: 0 auto;

    box-shadow: 0 0 3px 3px #dddddd,

                4px 4px 6px 2px rgba(136,136,136,0.7),

                -4px -4px 6px 2px rgba(244,244,244,0.7);

}

.login-form-btn

{

    font-family: 'Poppins';

    font-size: 15px;

    color: #4caf50;

    line-height: 1.2;

    font-weight: bold;

    text-transform: uppercase;

    display:-webkit-flex;

    display: -webkit-box;

    display: flex;

    justify-content: center;

    align-items: center;

    letter-spacing: 1px;

    padding:0 20px;

    height: 50px;

    width: 100%;

}

.login-form-btn:hover

{

    background: #4caf50;

    color: #ddd;

}

.login-form-btn:active

{

    background: rgb(255,145,19);

    color: #ddd;

}

.text-centre

{

    text-align: center;

    padding-top: 70px;

}

.validate-input

{

    position: relative;

}

.alert-validate:before

{

    content: attr(data-validate);

    position: absolute;

    max-width: 70%;

    background-color: #fff;

    border-radius: 2px;

    border: 1px solid #c80000;

    padding: 4px 25px 4px 10px;

    top: 50%;

    transform: translateY(-50%);

    right: 0px;

    pointer-events: none;

    font-family: 'Poppins';

    color: #c80000;

    font-size: 13px;

    line-height: 1.4;

    text-align: left;

    visibility: hidden;

    opacity: 0;

    transition: opacity 0.4s;

}

.alert-validate:after

{

    content: "\f1f8";

    font-family: material-design-iconic-font;

    font-size: 21px;

    color: #c80000;

    position: absolute;

    top: 50%;

    transform: translateY(-50%);

    right: 5px;

}

.alert-validate:hover:before

{

    visibility: visible;

    opacity: 1;

}

/\*----------------Responsive----------------\*/

@media (max-width: 576px)

{

    .wrap-login

    {

        padding:77px 15px 33px 15px;

    }

}

@media (max-width: 992px){

.alert-validate:hover:before

{

    visibility: visible;

    opacity: 1;

}

}

**6.1.3 JS**

/\* Focus input \*/

$('input').each(function(){

    $(this).on('blur', function(){

        if($(this).val().trim() !=""){

            $(this).addClass('has-val');

        }

        else

        {

            $(this).removeClass('has-val');

        }

    })

});

if(localStorage.getItem("EMAIL")!=null){

    localStorage.clear();

}

document.querySelector(".login-form-btn").addEventListener('click', function(){

    const email = document.getElementById("email-input").value;

    localStorage.setItem("EMAIL",email);

});

/\*validate\*/

var input = $('.validate-input .input');

    $('.validate-form').on('submit',function(){

        var check = true;

        for(var i=0; i<input.length; i++) {

            if(validate(input[i]) == false){

                showValidate(input[i]);

                check=false;

            }

        }

        return check;

    });

    $('.validate-form .input').each(function(){

        $(this).focus(function(){

           hideValidate(this);

        });

    });

    function validate (input) {

        if($(input).attr('type') == 'email' || $(input).attr('name') == 'email') {

            if($(input).val().trim().match(/^([a-zA-Z0.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-z9\-]+\.)+))([a-zA-Z]{1,5}|[0-9]{1,3})(\]?)$/) == null) {

                return false;

            }

        }

        else {

            if($(input).val().trim() == ''){

                return false;

            }

        }

    }

    function showValidate(input) {

        var thisAlert = $(input).parent();

        $(thisAlert).addClass('alert-validate');

    }

    function hideValidate(input) {

        var thisAlert = $(input).parent();

        $(thisAlert).removeClass('alert-validate');

    }

/\*showpass\*/

var showpass = 0

$('.btn-show-pass').on('click', function(){

    if (showpass == 0)

    {

        $(this).next('input').attr('type','text');

        $(this).find('i').removeClass('zmdi-eye');

        $(this).find('i').addClass('zmdi-eye-off');

        showpass=1;

    }

    else{

        $(this).next('input').attr('type','password');

        $(this).find('i').addClass('zmdi-eye');

        $(this).find('i').removeClass('zmdi-eye-off');

        showpass=0;

    }});

**6.1.4 Screenshots**

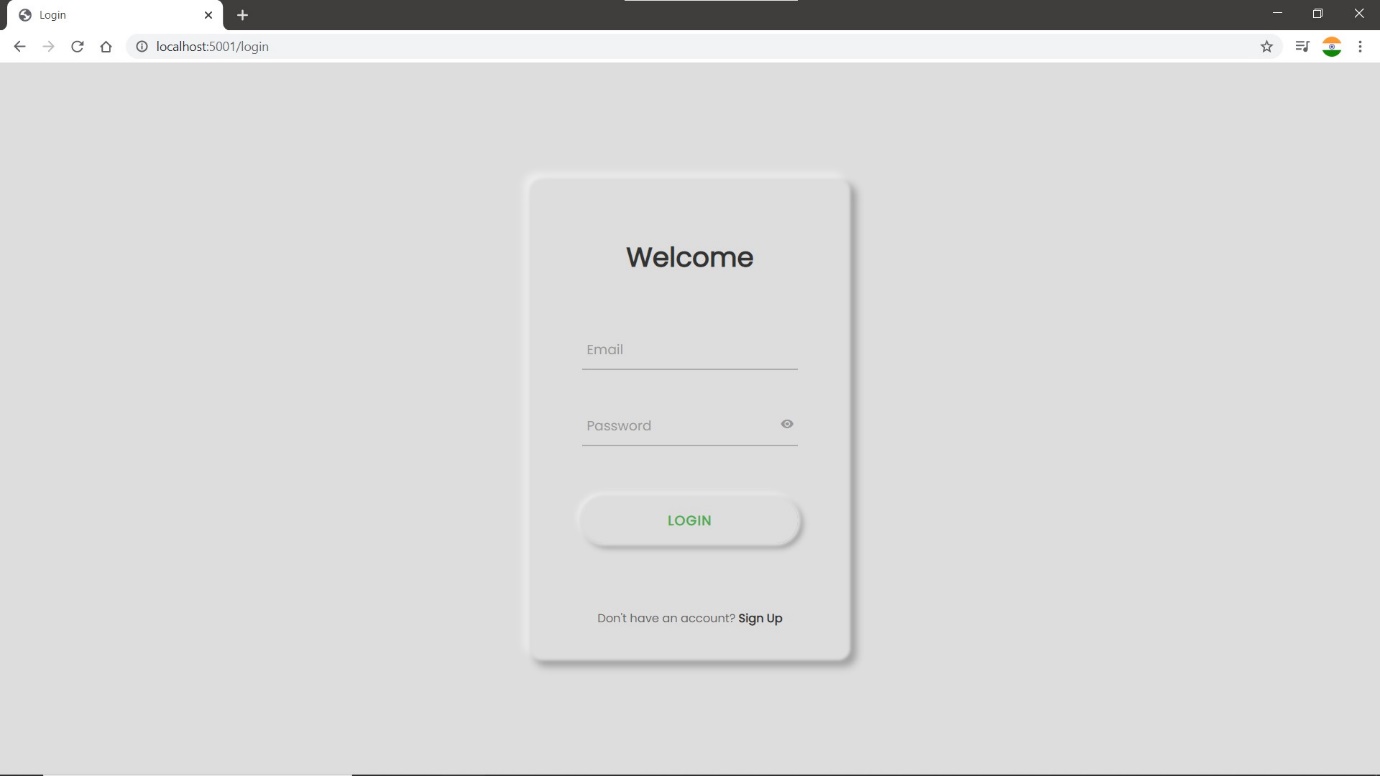
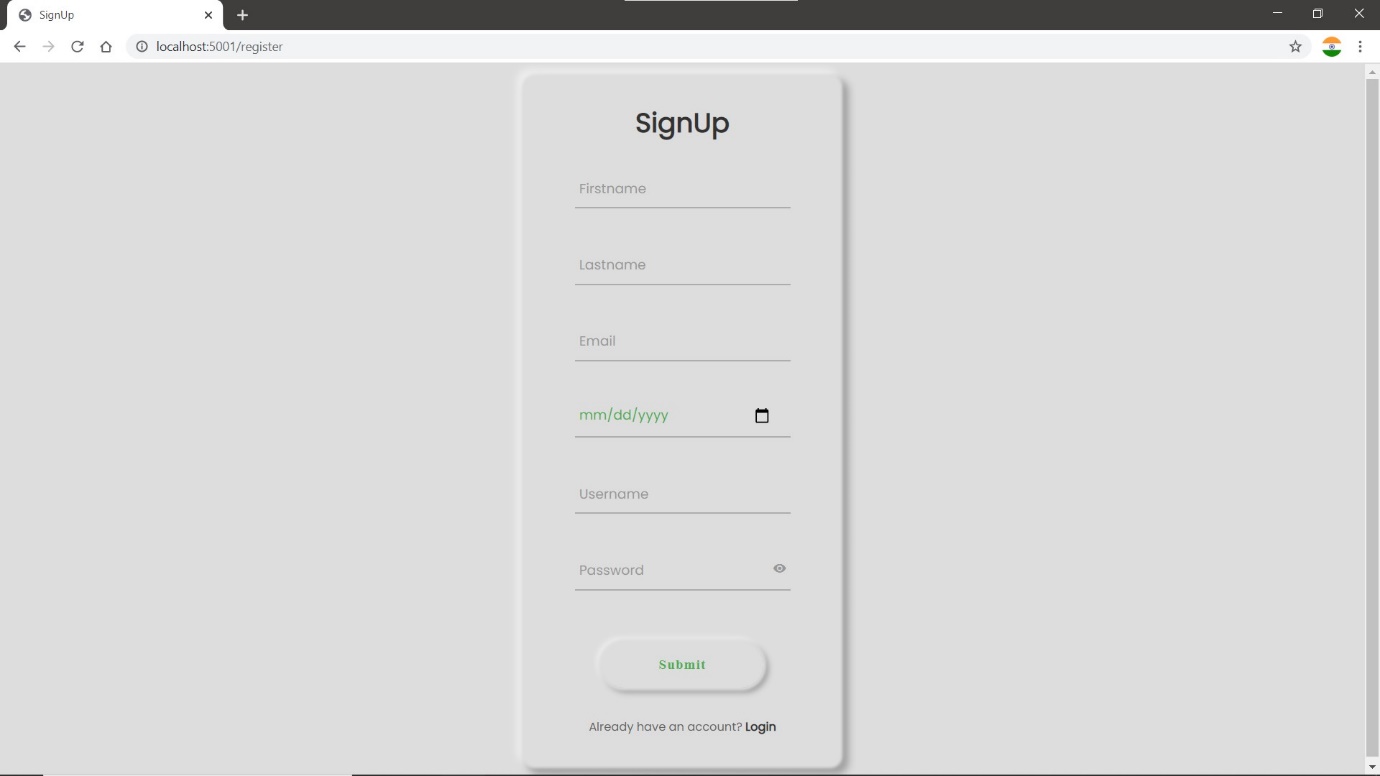
****

Fig: The sign-in page

**6.2 Register Page**

This is a register page where in a new user can register to access the application content. Any user can register and use this application.

****

**Fig: The register page**

**6.3 Home Page**

This is the page to which the user will be navigated before they log in.

**6.3.1 HTML**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

    <link rel="stylesheet" href="../home.css">

    <title>Home</title>

</head>

<body>

    <nav>

        <h4>Node MySQL</h4>

        <ul>

            <li><a href="/">Home</a></li>

            <li><a href="/login">Login</a></li>

            <li><a href="/register">SignUp</a></li>

        </ul>

    </nav>

    <div class="container mt-4">

        <div class="jumbotron">

            <h1 class="display-4">Welcome</h1>

            <p class="lead">This is a simple event management application that is presented as our mini project for the 5th sem DBMS subject.</p>

            <hr class="my-4">

            <p>Here is an example application of individual event management applications.</p>

            <p class="lead">

              <a class="btn btn-primary btn-lg" href="/login" role="button">Login</a>

            </p>

          </div>

    </div>

    <script src="../jquery.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

</body></html>

**6.3.2 CSS**

nav{

    background-color: #563d7c;

    color:#fff;

    padding: 30px 60px;

    display: flex;

    justify-content: space-between;

}

nav ul{

    display: flex;

    justify-content: space-around;

    align-items: center;

}

nav li{

    list-style: none;

}

nav li a{

    color: #fff;

    text-decoration: none;

    font-weight: bold;

    padding: 5px 8px;

}

nav li a:hover{

    color: #fc4d4d;

    text-decoration: none;

}

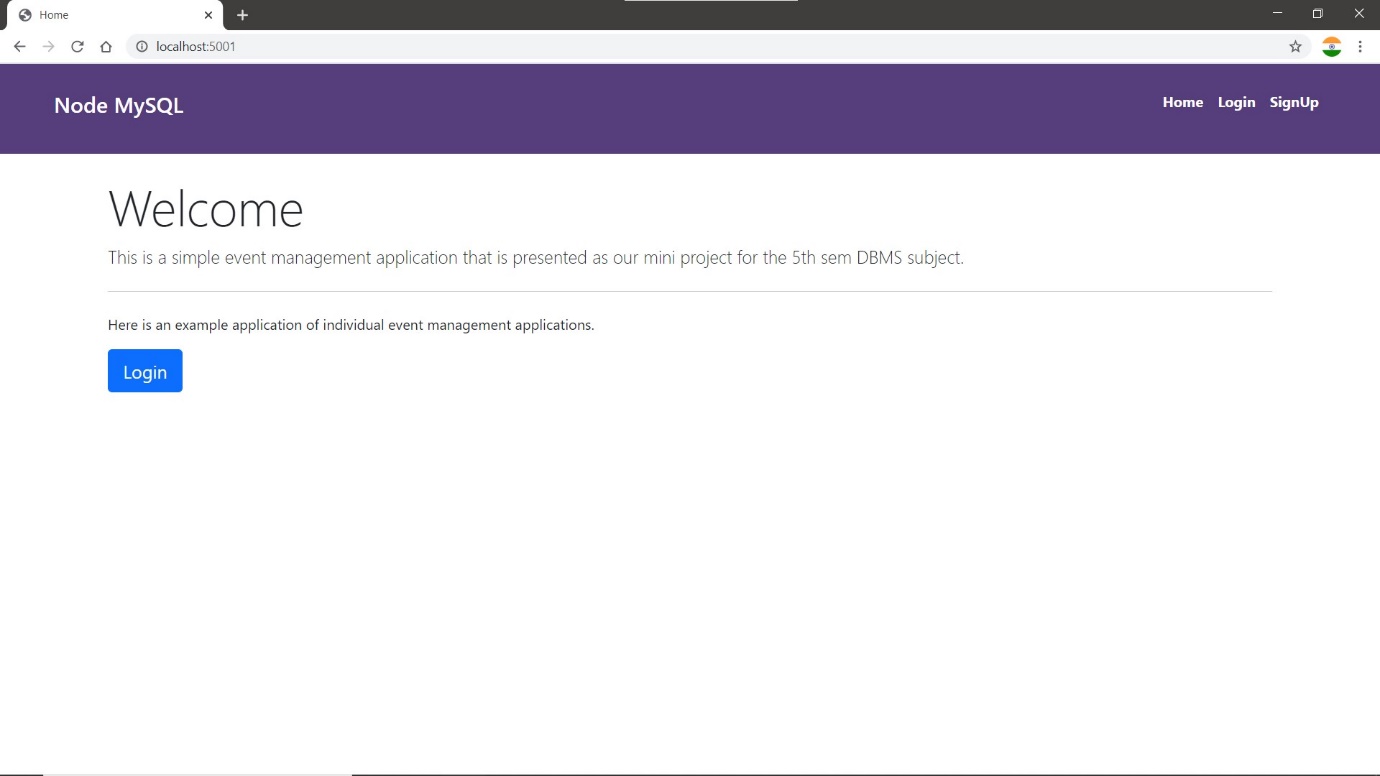
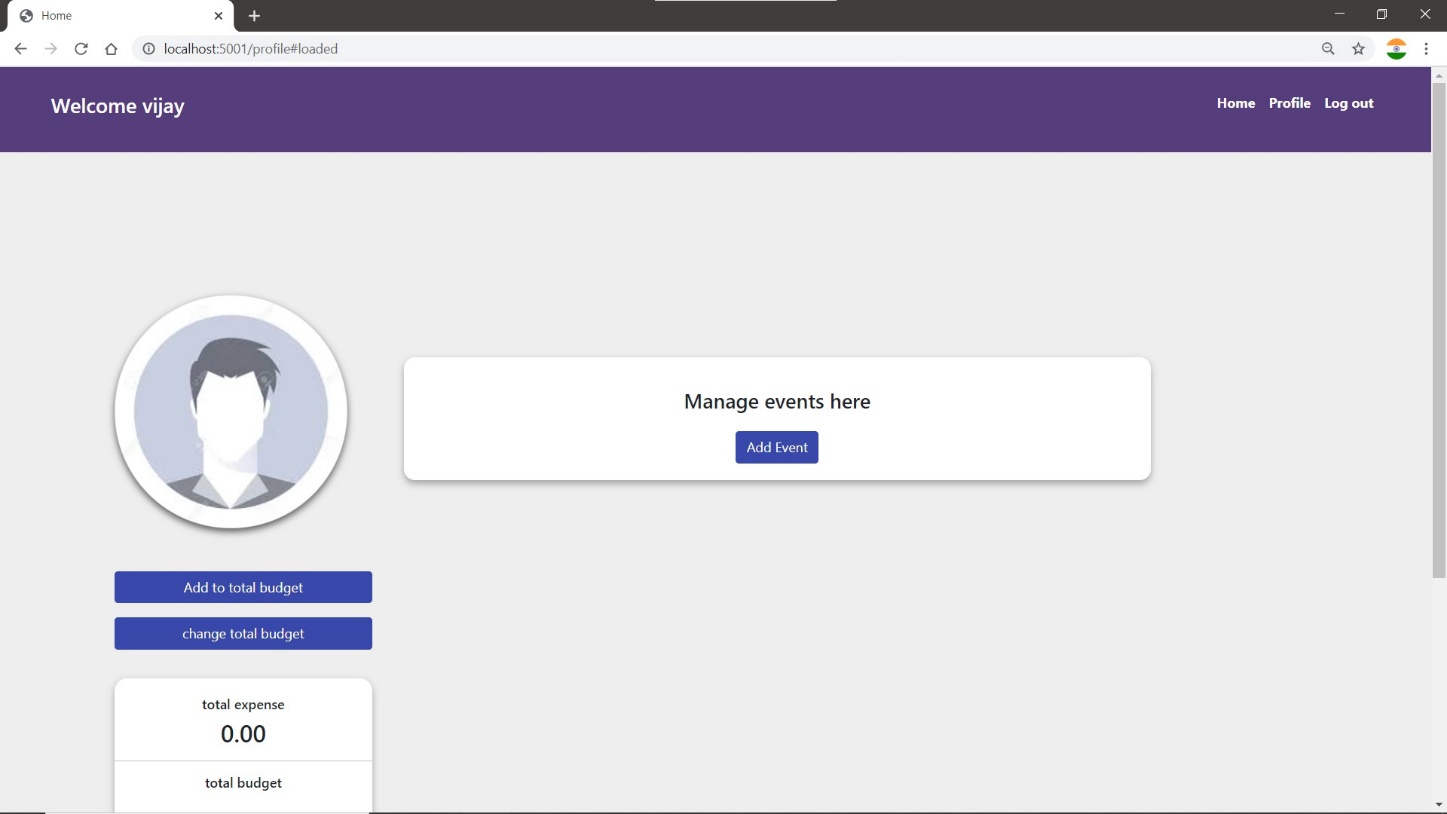
****

Fig: The home page

**6.4 Profile Page**

This is the page to which the user will be navigated after they log in.



**6.5 Table Creation Commands**

CREATE TABLE `customer` (

`C\_username` varchar(30) NOT NULL,

`C\_password` varchar(256) DEFAULT NULL,

`C\_email` varchar(50) NOT NULL,

`C\_dob` date NOT NULL,

`C\_firstname` varchar(50) NOT NULL,

`C\_lastname` varchar(50) NOT NULL,

`C\_budget\_total` decimal(10,2) DEFAULT NULL,

`C\_expense\_total` decimal(10,2) DEFAULT NULL,

PRIMARY KEY (`C\_username`)

);

CREATE TABLE `events` (

`C\_username` varchar(30) NOT NULL,

`E\_timeadded` bigint NOT NULL,

`E\_name` varchar(100) NOT NULL,

`E\_address` varchar(150) DEFAULT NULL,

`E\_type` varchar(45) NOT NULL,

`E\_startdate` date NOT NULL,

`E\_enddate` date DEFAULT NULL,

`E\_timeleft` bigint DEFAULT NULL,

`E\_alert\_status` varchar(45) DEFAULT NULL,

`E\_expense` decimal(10,2) DEFAULT NULL,

`E\_budget` decimal(10,2) DEFAULT NULL,

PRIMARY KEY (`C\_username`,`E\_timeadded`),

KEY `E\_type\_idx` (`E\_type`),

KEY `E\_timadded` (`E\_timeadded`),

KEY `E\_alert\_status\_idx` (`E\_alert\_status`),

CONSTRAINT `C\_username` FOREIGN KEY (`C\_username`) REFERENCES `customer` (`C\_username`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `E\_alert\_status` FOREIGN KEY (`E\_alert\_status`) REFERENCES `e\_alerts` (`E\_alert\_status`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `E\_type` FOREIGN KEY (`E\_type`) REFERENCES `event\_types` (`E\_type`) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE `event\_expenses` (

`C\_username` varchar(30) NOT NULL,

`E\_timeadded` bigint NOT NULL,

`Expense\_name` varchar(100) NOT NULL,

`Expense\_amount` decimal(10,2) DEFAULT NULL,

PRIMARY KEY (`C\_username`,`E\_timeadded`,`Expense\_name`),

KEY `E\_timeadded\_idx` (`E\_timeadded`),

CONSTRAINT `E\_timeadded` FOREIGN KEY (`E\_timeadded`) REFERENCES `events` (`E\_timeadded`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `event\_expenses\_ibfk\_1` FOREIGN KEY (`C\_username`) REFERENCES `customer` (`C\_username`) ON DELETE CASCADE ON UPDATE CASCADE

) ;

CREATE TABLE `speakers` (

`C\_username` varchar(50) NOT NULL,

`E\_timeadded` bigint NOT NULL,

`s\_id` int NOT NULL,

`s\_name` varchar(45) NOT NULL,

`s\_topic` varchar(45) NOT NULL,

`s\_prof` varchar(45) NOT NULL,

PRIMARY KEY (`E\_timeadded`,`s\_id`),

KEY `C\_username\_idx` (`C\_username`),

KEY `C\_username\_speaker` (`C\_username`),

CONSTRAINT `C\_username\_speaker` FOREIGN KEY (`C\_username`) REFERENCES `customer` (`C\_username`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `E\_timeadded\_speaker` FOREIGN KEY (`E\_timeadded`) REFERENCES `events` (`E\_timeadded`) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE `guests` (

`C\_username` varchar(30) NOT NULL,

`E\_timeadded` bigint NOT NULL,

`g\_id` int NOT NULL,

`g\_name` varchar(45) DEFAULT NULL,

`g\_phno` bigint DEFAULT NULL,

`g\_gender` varchar(10) DEFAULT NULL,

PRIMARY KEY (`E\_timeadded`,`g\_id`),

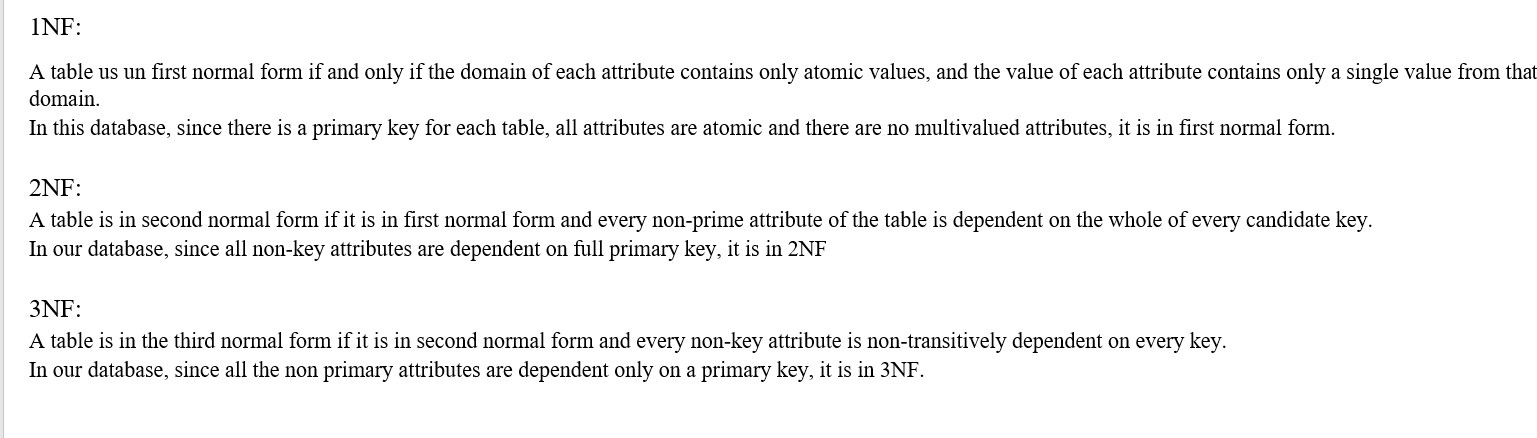
KEY `C\_username\_guest\_idx` (`C\_username`),

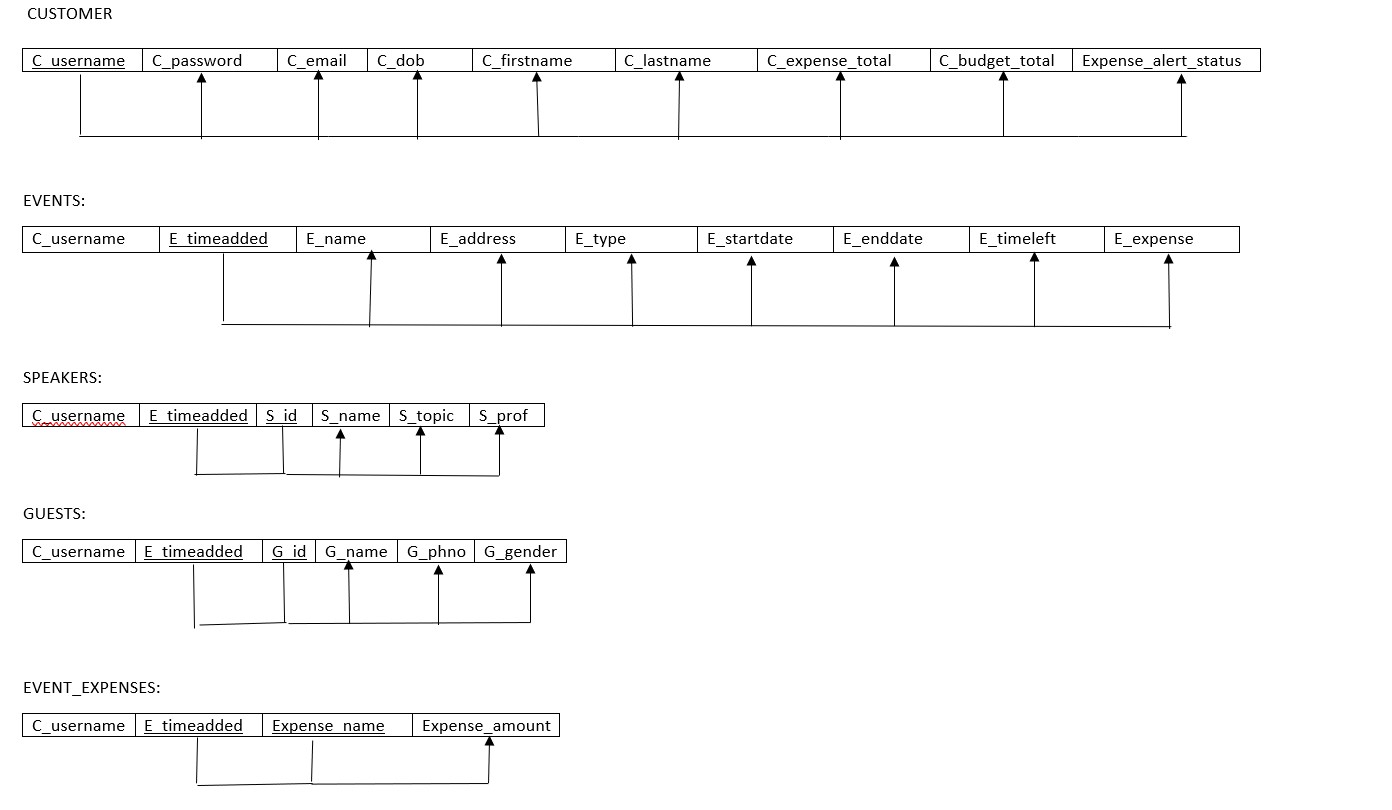
CONSTRAINT `C\_username\_guest` FOREIGN KEY (`C\_username`) REFERENCES `customer` (`C\_username`),

CONSTRAINT `E\_timeadded\_guest` FOREIGN KEY (`E\_timeadded`) REFERENCES `events` (`E\_timeadded`)

);

**6.6 Normalization:**

****

****

**6.7 Database connection**

const express = require('express');

const path = require('path');

const mysql = require('mysql');

const dotenv = require('dotenv');

const cookie = require('cookie-parser');

dotenv.config({path: './.env'});

const app = express();

const db = mysql.createConnection({

    host: process.env.DATABASE\_HOST,

    user: process.env.DATABASE\_USER,

    password: process.env.DATABASE\_PASSWORD,

    database: process.env.DATABASE

});

const cssjslinks = path.join(\_\_dirname,'./');

app.use(express.static(cssjslinks));

app.use(express.urlencoded({ extended: false }));

app.use(express.json());

app.use(cookie());

 app.set('view engine', 'hbs');

/\* app.engine('html', require('ejs').renderFile);

app.set('view engine', 'html');

 \*/

db.connect( (error) => {

    if (error) {

        console.log(error);

    }else{

        console.log('MYSQL connected...');

    }

});

app.use('/', require('./pages'));

app.use('/auth', require('./auth'));

app.listen(5001,()=>{

    console.log('Server started on port 5001'); });

**CHAPTER 7**

**Trigger**

A **database trigger** is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

In this database, when the individual expenses in each event is added, the total expense in the events table is updated by adding the previous value to the new value by the use of trigger.

**7.1 Code Snippet**

CREATE TRIGGER `login`.`event\_expenses\_AFTER\_INSERT` AFTER INSERT ON `event\_expenses` FOR EACH ROW

BEGIN

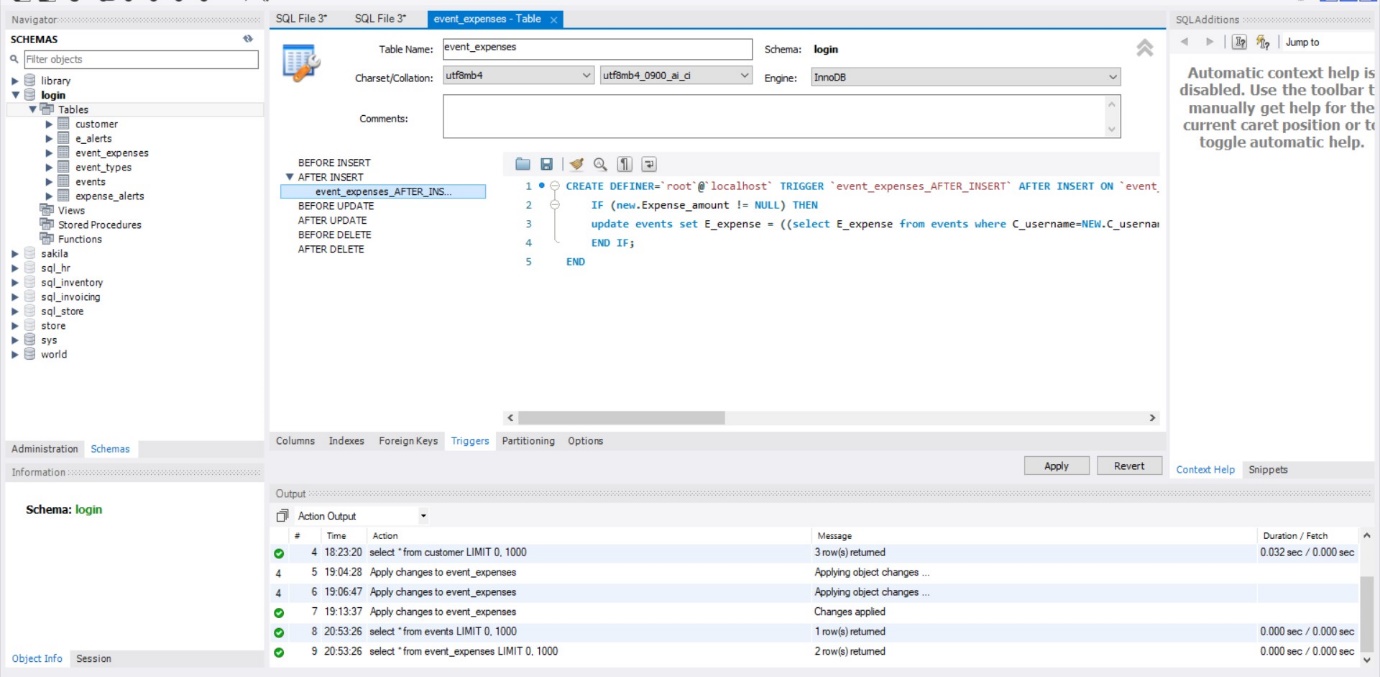
if(new.Expense\_amount != NULL) THEN

update events set E\_expense = ((select E\_expense from events where C\_username=NEW.C\_username AND E\_timeadded=NEW.E\_timeadded)+NEW.Expense\_amount);

END IF;

END

**7.2 Screenshot**

****

**CHAPTER 8**

**CONCLUSION**

With the theoretical knowledge of this subject, we created the “**EVENTS DATABASE**” there by also gained a practical experience about how to create and develop a project which is necessary factor for all the students. It becomes very necessary to take the utmost advantage of any opportunity of gaining practical knowledge that comes along. The construction of this mini project was one of those opportunities.

The whole objective of this project is to provide a streamlined experience for all the end users to access information in a easy and efficient manner as there are hundreds of thousands of satellites launched by dozens of countries and multitude of private organisations. We tried to some extent in order to reach our motive, it can be still increased by doing with future iterations of this project.

**CHAPTER 9**

**References**

1. W3 Schools (HTML and CSS reference) – <https://www.w3schools.com/>
2. The Bootstrap Website (for Design) – <https://www.getbootstrap.com/>
3. FONTS implementation – [fonts.google.com](https://php.net/docs.php/)
4. YouTube website channel (for Design)-<https://www.youtube.com/Animation%20coding>