ABSTRACT

Is there anyone alive who hasn't ever wanted to let fly at the things, endlessly whirring stuff with a blunt object or thought of swinging a bat at other objects that would soothe the turmoil in their soul? Hoping to fulfill these everyday daydreams of violence as catharsis is The Break Room. Not just any amusement place. But a place where you can pay money to smash stuff. Thus advertising it as a stress reliever; a safe place to blow off some steam without anyone getting hurt.

What if I say, there is now actually a place you can go to vent out all your, frustration on inanimate things and let that inner crooked child of yours go wild. It may seem like a crazy idea. Well it indeed is. But many vouch for its therapeutic value. This new venture called Break Room is now providing a place for people to go absolutely berserk and demolish things to unleash resentment. The Break Room is a way to shatter that wall of stress, one smash at a time. Or maybe more! And we will be providing a Break suit, bats and stuff to hit the articles and one ultimate goal. Break! And relieve stress.

There are few centers across the world. And yes, there is one, first of its kind in India too! So why wait? Head to Gurugram now!

In this, we have provided a platform where users can register or login and book the slots and then visit the place and break! They can view their break stats too!

TABLE OF CONTENTS

Certificate …….. I

Acknowledgment ……. . II

Abstract …… III

Table of Contents …… . IV

List of Figures …… V

List of Tables …… VI

Chapter 1 …… 1

Introduction …….. 1

1.1 The Break Room Management System …… 1

Chapter 2 ……. 3

Literary Survey …… 3

2.1 Traditional File System ……. 3

2.2 Pros and Cons of the Traditional Approach …….. 3

2.3 Downfall of the Traditional Management System …… 4

2.4 Introduction to the Database Management System ……... 4

2.5 Indicative Areas for the use of a DBMS …… 5

2.6 Advantages of a DBMS ……. 5

2.7 Components of a DBMS …… 6

Chapter 3 …….. 7

System Requirements …… 7

3.1 Hardware Requirements …… 7

3.2 Software Requirements ……. 7

Chapter 4 …….. 8

System Design …… 8

4. 1 Schema Design …… 8

4.2 ER Diagram …….. 9

Chapter 5 ……..… 10

Implementation ……..... 10

5.1 HTML5 …….. 10

5.2 CSS ……..... 10

5.3 PHP ……....……..... 11

5.4 SQL ……..……... 12

5.5 Code Snippets …… 14

Chapter 6 …….. 39

Snapshots ……... 39

6.1 The Home Page …….. 39

6.2 The Registration Page ……..... 40

6.3 The Login Page ……... 41

6.4 The Articles ……. 41

6.5 Booking Successful Page …… 42

6.6 The Status Page ……... 42

6.7 Tables …….. 43

Chapter 7 …… 46

Conclusion …… 46

References ……..... 47

LIST OF FIGURES

Fig No.	Description	Page No.
Fig 2.1	Components of a DBMS ……	7
Fig 4.1	Schema Diagram …….....	9
Fig 4.2	ER Diagram for Break Room Management System ……..	10
Fig 6.1	The Home Page ……..	39
Fig 6.2	The Registration Page …….....	40
Fig 6.3	The Login Page ……..	41
Fig 6.4	The Articles …….. …….	41
Fig 6.5	Booking Successful Page …….....	42
Fig 6.6	The Status Page ……..	42

LIST OF TABLES

Table no.	Description	Page No.
Table 6-1	Members Table …….	43
Table 6-2	Location Table ……..	44
Table 6-3	Articles Table ……	44
Table 6-4	Status Table ……...	45
Table 6-5	Bookings Table ……..	46
Table 6-6	Logs Table ……..	46

Chapter 1

INTRODUCTION

1.1 Break Room Management System

Maybe it’s time to put down your chocolates or ice-creams, for we've discovered a break that's even better than snappable chocolate-covered wafers: smashing stuff. To be more specific, we mean smashing stuff with a baseball bat and it's a legitimate, all above-board activity at new Break Room. The Break Room was born in a moment of frustration (and through the efficacy of podcasts) by founder Ed Hunter, after he realised that everybody wants to blow off steam in their own way. And while some might prefer to hit the gym, Hunter is inviting you to hit some crockery off a stool instead.

The Break Room is for those who want to change their typical routine and either live out a destructive fantasy or find catharsis in doing something destructive. It’s not simply about violence, but rather harnessing those destructive tendencies in a controlled and fun way. Stress is a very real health issue for most of us, and while I am definitely not a psychologist, I can attest to the healing powers a good session in the Room can provide. Your time with us is well spent. You’ll come out of the room sweaty and ready to tackle the next challenge life throws at you.

Vent all your anger safely at the Break Room, a venue that openly encourages you Sto destroy objects. Suit up, grab a baseball bat and head into the rooms, where you can throw, bash, smash and break plates and glassware. You can even bring your own items to smash if you’ve always wanted to destroy a certain something (the most popular items to smash in such rooms worldwide are printers, just so you know).

The process of going to town on a bunch of objects isn't so much about being violent, but about releasing our destructive tendencies in a safe and controlled way. You can even play whatever music you like while in the room. Everything from the safety gear to the disposal of the destroyed objects is taken care of, so all you need to do is wear closed-toe shoes and get smashing.

So wait until the shutters go down, don your best Joaquin Phoenix impersonation and swing away at those inner aliens of yours by taking your aggression out in some organised chaos!

This project just gives a visual idea of The Break Room. Not much people know about this. So why not shed some light on it? This project helps you understand what exactly The Break Room is and how it will come handy in your life. It allows the users to register to our domain, or login if they already have and can book the desired slot and view their break stats.

Chapter 2

LITERATURE SURVEY

2.1 Traditional File System

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 a Traditional File System

Pros

- Simple
    - Matched existing business processes and functions.
    - Companies were not as interested in funding complicated information systems.

- Initially low-cost
    - Early computing was not viewed as beneficial for large funding.
    - Systems were designed to be cheap in order to save on cost.

Cons

- Separated ownership
    - Business functions had a high sense of data ownership.
    - Departments unwilling to share data for fear of minimizing their superiority.
- Unmanaged redundancy
    - Multiple instances of the same data appeared throughout various files, systems, and databases.
    - Information updated in one place was not replicated to the other locations.
    - Disk space was very expensive, and redundancy had a big impact on storage.
- Data inconsistency
    - Redundant data stored in various locations was usually never stored the same way.
    - Formatting was not centrally managed.
- Lack of data sharing
    - Same data stored in multiple locations.
    - Caused unnecessary doubling of efforts for processing and managing data.
- High costs in the long run
    - Hiring data processors for each department was very expensive, and each position was typically working on the same thing just for a different area.
    - Doubling of work as well as excessive maintenance costs.

2.3 Downfall of Traditional 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 environment 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 centers. 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 examples of database management software.

Database systems are meant to handle large collection 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 unauthorized access.

2.5 Indicative Areas for the use of a 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 etc.

2.6 Advantages of a DBMS

A Database Management System has many advantages over the traditional file system used in the earlier days:

- 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 utilize 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 on external storage devices.

- Data integrity and Security: If data is accessed through the DBMS , the DBMS can enforce integrity constraint on the data.

- Data administration: When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the 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 a DBMS

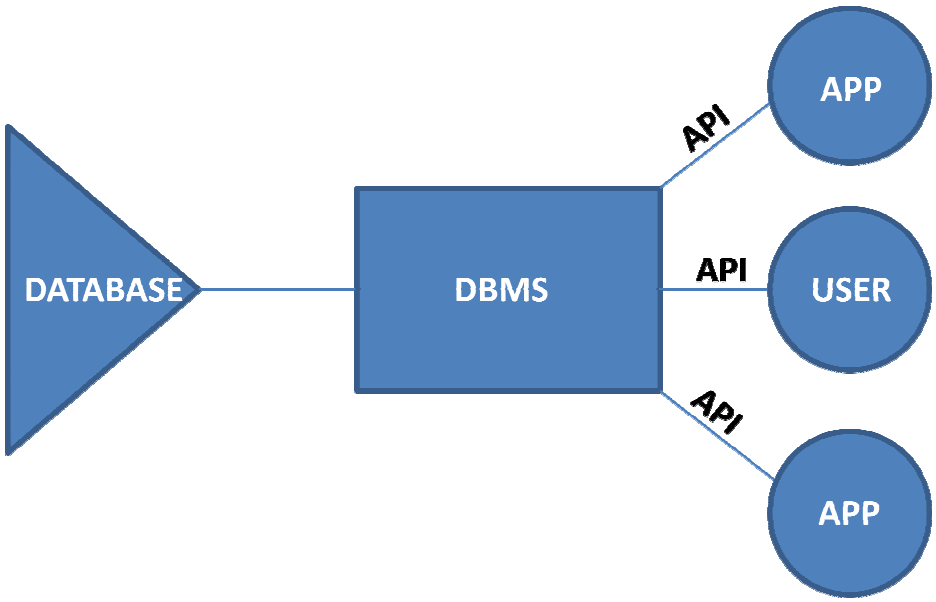


Fig 2.1 Components of a DBMS

- **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

SYSTEM REQUIREMENTS

3.1 Hardware requirements

- Processor: Intel Core 2 Duo or above.
- RAM: 2GB or more.
- Hard Disk: 2GB or more.

3.2 Software requirements

Technologies used:

- Front End: HTML, CSS , PHP
- Connection/Controller: PHP
- Back-End Database: MySQL

Software:

- Text Editor: Sublime Text
- Server: Apache (on XAMPP 7 or WAMP)
- Operating System: Windows 10
- Database Support: MySQL5
- Back-End: PHP 7.2.8

Chapter 4

SYSTEM DESIGN

4.1 Schema Diagram

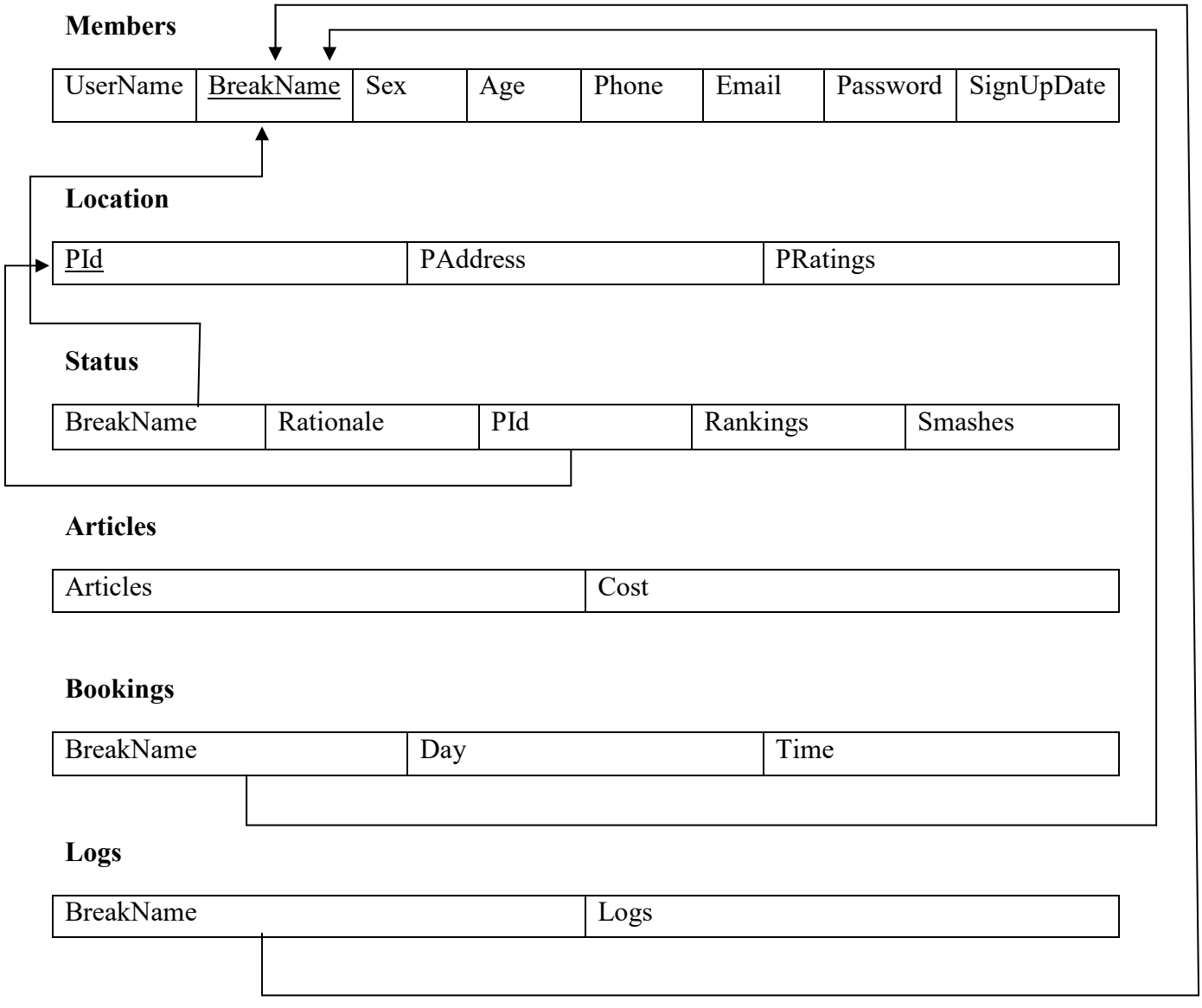


Fig 4.1 Schema diagram for Break Room Management System

4.2 ER Diagram

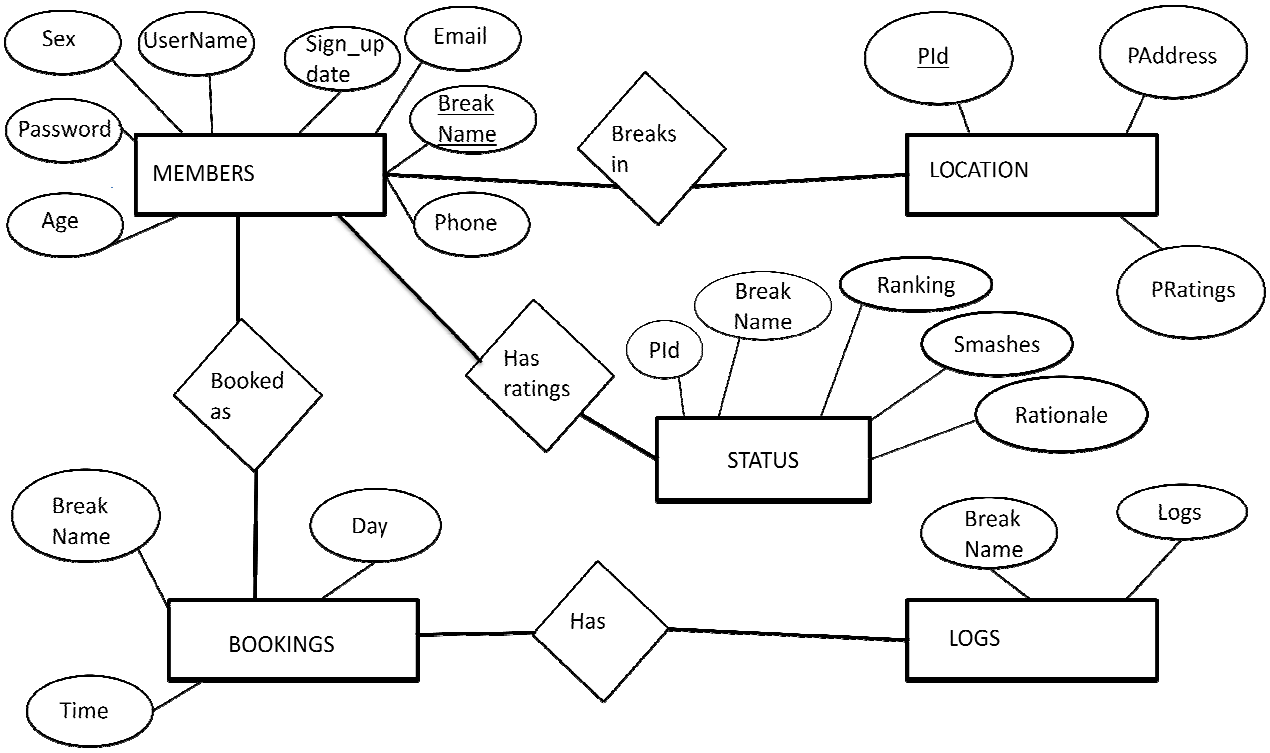


Fig 4.2 ER diagram for Break Room Management System

This ER diagram of Break Room Management System gives a brief idea of its existent entities and the relationships between them and the attributes associated with each entity.

Here we have 5 tables namely Members, Status, Location, Bookings, Logs, and their attributes. We could clearly see how the entities actually behave in the database.

Chapter 5

IMPLEMENTATION

5.1 HTML5

HTML5 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. HTML5 is intended to subsume not only HTML4,but also XHTML1 and DOM Level 2 HTML.

HTML5 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, HTML5 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 HTML5 specification and HTML5 also better defines the processing for, any invalid documents.

5.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML5. 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, colors and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural format.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech based browser or screen reader), and on Braille based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC2318 (march 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other mark-up languages support the use of CSS, including XHTML, plain XML, SVG, and XUL.

5.3 PHP

PHP is a servlet-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in I994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may beany type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

PHP is a scripting language that helps people make web pages more interactive by allowing them to do some more things. A website programmed with PHP can have pages that are password protected. A website with no programming cannot do this without other complex things. Standard PHP file extension are: .php, .php3 or .phtml, but a web server can be set up to any extension.

5.4 SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database.

Management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). In comparison to older read/write APIs like [SAM or VSAM, SQL offers two main advantages: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify how to reach record e.g., with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages for Edgar F Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986 and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different data base systems without adjustments.

5.5 Code snippets

5.5.1 Config – To establish connection

This is a code snippet to show how PHP is used to connect to the local MySQL database using the localhost servlet.

```
<?php
        ob_start();
        session_start();

        $timezone = date_default_timezone_set("Asia/Kolkata");

        $con = mysqli_connect("localhost", "root", "", "BreakRoom");

        if(mysqli_connect_errno()) {
                echo "Failed to connect: " . mysqli_connect_errno();
        }
?>
```

5.5.2 Login Page

```
<?php

        include("Includes/Config.php");
        include("Includes/Classes/Account.php");
        include("Includes/Classes/Constants.php");
         $account = new Account($con);
        include("Includes/Handlers/LoginHandler.php");
?>
```

```
<!DOCTYPE html>
<html>
<head>
	<title> The Break Room </title>
	<style type="text/css">
		body {
			background-image: url("/Project/BR10.jpg");
			background-repeat: no-repeat;
			background-size: cover;
		}
		button {
   background-color: powderblue;
   color: black;
   padding: 14px 20px;
   margin: 8px 0;
   border: none;
   cursor: pointer;
   width: 100%;
}
		button:hover {
   opacity: 0.8;
}
		.cancelbtn {
   width: auto;
   padding: 10px 18px;
   background-color: #f44336;
}
```

```
        </style>
</head>
<body>
        <center>
                <h1 style="color:lavenderblush;font-family:Ariel black;">
                        <b>
                                <i>
                                        Welcome to the House of Shatter!
                                </i>
                        </b>
                </h1>
        </center>
        <form method="POST" action="LoginPage.php">
                <center>
                        <p style="color:lavender;">

                        <label for="breakname"><b> Break-Name: </b></label>
                        <input id="breakname" type="text" name="breakname" size="15" maxlength="30" required />
                        </p>
                        <p style="color:lavender;">

                        <label for="password"><b> Password: </b></label>
                        <input id="password" type="password" name="password" size="15" maxlength="30" required />
                        </p>
```

```
<p style="color:lavender;">
                    <b> Choose a day: </b>
                    <select name="Day" id="Day">
                        <option disabled="disabled" selected="selected"> DAY
                        </option>
                        <option value="Today" name="Day" >Today</option>
                        <option value="Tomorrow" name="Day">Tomorrow</option>
                        <option value="Day after tomorrow" name="Day"> Day after
                          Tomorrow </option>
                    </select>
                    </p>
                <p style="color:lavender;">
                    <b> Choose a slot: </b>
                    <select name="slot" id="slot">
                        <option disabled="disabled" selected="selected"> SLOT
                        </option>
                       <option value="11:00">11:00</option>
                        <option value="12:00">12:00</option>
                        <option value="13:00">13:00</option>
                       <option value="14:00">14:00</option>
                       <option value="15:00">15:00</option>
                        <option value="16:00">16:00</option>
                        <option value="17:00">17:00</option>
                        <option value="18:00">18:00</option>
                        <option value="19:00">19:00</option>
                        <option value="20:00">20:00</option>
```

```
                                <option value="21:00">21:00</option>
                                <option value="22:00">22:00</option>
                                </select>
                            </p>
<div><span style="color: blue"><h1><?php echo $account->getError(Constants::$loginfailed); ?></h1></span><br></div>
                    <button type="submit" name="loginbtn" id="loginbtn">Book Now</button>
            <label>
            <button type="submit" name="profilebtn" id="profilebtn">View Your Profile</button>
            </label>
            <label>
            <b><input type="checkbox" checked="checked" name="remember"> Remember me </b>
            </label>
                    </center>
                    <center>
                    <form action="/Project/HomePage.php">
            <button type="submit" class="cancelbtn" name="cancelbtn">Cancel</button>
                    </form>
            <span><b><a href="#"> Forgot Password?</a></b></span>
            </center>
            <br>

<h2 style="color: blue"> Special note: </h2>
            <table style="color: aqua" class="centered">
                    <tr>
```

```
                <th>Session</th>
                <th>Time*</th>
                <th>Price(Cash only)</th>
            </tr>
            <tr>
                <td>Need a Break</td>
                <td>2 minutes</td>
                <td>Rs. 300</td>
            </tr>
            <tr>
                <td>Lash Out</td>
                <td>3 minutes</td>
                <td>Rs. 400</td>
            </tr>
            <tr>
                <td>Demolition</td>
                <td>5 minutes</td>
                <td>Rs. 500</td>
            </tr>
        </style>
    </table>
```

*Additional time increments are available at additional cost with a maximum of 15 minutes. However, our staff has full authority to deny permission to do so.

*You can purchase more items to break in the room, in addition to the basic Break session (based on availability).

5.5.2.1 Articles Page

```
<fieldset>
<legend style="text-align: right;"><strong><h3 style="color: aqua"> Items and Rates </h3></strong></legend>
<table  style="color: aqua"; align="right";>
<tr>
<th> Article </th>
<th> Cost </th>
</tr>
<?php
$con = mysqli_connect("localhost", "root", "", "BreakRoom");
$sql = "CALL Proc()";
$result = mysqli_query($con,$sql) or die("Error: $sql. ".mysql_error($con));
$row = mysqli_fetch_assoc($result);
do { ?>

<tr>
<td> <?php echo $row['Article']; ?> </td>
<td> <?php echo $row['Cost']; ?> </td>
</tr>
<?php } while($row = mysqli_fetch_assoc($result)) ?>
</table>
</fieldset>
</form>
</body>
</html>
```

5.5.3 Registration Page

```
<?php
    include("Includes/Config.php");
    include("Includes/Classes/Account.php");
    include("Includes/Classes/Constants.php");
    $account = new Account($con);
    include("Includes/Handlers/RegisterHandler.php");
    function getInputValue($name) {
        if(isset($_POST[$name])) {
            echo $_POST[$name];
        }
    }
?>
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
    font-family: Arial, Helvetica, sans-serif;
}
* {
    box-sizing: border-box;
}
.container {
    padding: 16px;
    background-color:                    white;                    }
```

```
input[type=text], input[type=password] {
    width: 100%;
    padding: 15px;
    margin: 5px 0 22px 0;
    display: inline-block;
    border: none;
    background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
    background-color: #ddd;
    outline: none;
}
hr {
    border: 1px solid #f1f1f1;
    margin-bottom: 25px;
}
.registerbtn {
    background-color: #4CAF50;
    color: white;
    padding: 16px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
}
```

```
.registerbtn:hover {
    opacity: 1;
}
a {
    color: dodgerblue;
}
.signin {
    background-color: #f1f1f1;
    text-align: center;
}
</style>
</head>
<body>
<div>
    <hr>
    <table bgcolor="lightcyan" border="1" width="100%">
    <tr>
        <td width="150"><img src="/Project/TheBreakRoom.jpg" alt="Image not found"
            width="240" height="150"/></td>
    <th><center><h1 style="color:black">Everything inside starts shattering from here!</h1></center></th>
    </tr>
    </table>
</div>
<div>
        <h1><center> Join our elite group! </center></h1>
      <p><center> Fill this once. Book and break as many times as you want. </center></p>
```

```
<hr>

<form id="Registrationform" action="RegistrationPage.php" method="POST">

<span style="color: red"><?php echo $account->getError(Constants::$usernameerror); ?></span><br>

<label for="username"><b>Name: </b></label>

<input id="username" name="username" type="text" placeholder="Enter your name" value="<?php getInputValue('username') ?>" required>

<span style="color: red"><?php echo $account->getError(Constants::$breaknameerror); ?></span><br>

<span style="color: red"><?php echo $account->getError(Constants::$breaknameused); ?></span><br>

<label for="breakname"><b>Break-Name: </b></label>

<input id="breakname" name="breakname" type="text" placeholder="Enter a funky name for a swing" value="<?php getInputValue('breakname') ?>" required>

<label for="sex"><b> Sex: </b></label>

<input id="sex" name="sex" type="radio"  value="M"> M

<input id="sex" name="sex" type="radio" value="F"> F

<input id="sex" name="sex" type="radio" value="O"> T

</br>

</br>

<label for="number"><b>Age: </b></label>

<input id="number" name="age" type="text" placeholder="Enter your age" value="<?php getInputValue('number') ?>" required>

<label for="phnumber"><b>Phone: </b></label>

<input id="phnumber" name="phnumber" type="text" placeholder="Enter your contact number" value="<?php getInputValue('phnumber') ?>" required>

<span style="color: red"><?php echo $account->getError(Constants::$emailerror); ?></span><br>

<label for="email"><b>Email ID: </b></label>
```

```
<input id="email" name="email" type="text" placeholder="Enter Email" value="<?php
getInputValue('email') ?>" required>

    <span style="color: red"><?php echo $account->getError(Constants::$passworderror1);
?></span><br>

    <span style="color: red"><?php echo $account->getError(Constants::$passworderror2);
?></span><br>

    <span style="color: red"><?php echo $account->getError(Constants::$passworderror3);
?></span><br>

    <label for="psw"><b>Password: </b></label>

    <input id="psw" name="psw" type="password" placeholder="Enter Password"
value="<?php getInputValue('psw') ?>" required/>

    <label for="pswrepeat"><b>Repeat Password</b></label>

    <input id="pswrepeat" name="pswrepeat" type="password" placeholder="Repeat
Password" value="<?php getInputValue('pswrepeat') ?>" required>

  <hr>

  <h2> Instructions: </h2>

  </br>

  <ul>

      <li> Those who are under 14, sufferers of severe to moderate Health conditions, or
are a danger to themselves or others are NOT permitted to participate in the Break Room.
</li>

      <li> The Break Room does not claim to be a mental help or medical facility; we do

        not treat, give diagnosis or provide medical therapy of any kind. </li>

      <li> Wear the costume that we give that cover your feet like boots, sneakers etc. </li>

       <li> Bring an ID PROOF (Driver's license, passport, Adhaar card, School or College

       ID). </li>

</ul>

  <p> Reach to us at 455413553, between 11:00 AM and 10:00 PM. </p>

  </br>
```

```
    <button type="submit" class="registerbtn" name="registerbtn"> Let me in </button>
    </form>
  </div>
</body>
</html>
```

5.5.4 Home Page

```
<!DOCTYPE html>
<html>
<head>
        <title> Home Page </title>
        <style type="text/css">
        body {
                        background-image: url("/Project/BR4.jpg");
                        background-repeat: no-repeat;
                        background-size: cover;
                }
        button {
    background-color: purple;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
}
```

```
button:hover {
   opacity: 0.8;
}
        </style>
</head>
<body>
        <center>
         <figure>
                <img src="/Project/BREAKROOM_logo.jpg" width="400" height="200">
                <br/>
                <figcaption style="color: rgb(255,0,0);"> <h4> <i> Ctrl.Shift.Esc </i> </h4>
</figcaption>
         </figure>
        </center>
        <h1>
                <b>
                        <center>
                         <p style="color: lavender;"> Wanna feel good? Come. <strike> Break
things! </strike> </p>
                        </center>
                </b>
        </h1>
        <h2>
                <b>
                        <center>
                                <p style="color: lavender;"> You know you need this. </p>
                         </center>
                        </b>
```

```
</h2>
        <center>
            <video width="800" height="400" controls><source src="/Project/The Break Room Atlanta_ The Experience.mp4" type="video/mp4">
            </video>
        </center>
        <br/>
        <center>
            <h1 style="color:orange;font-family:Comic Sans MS;">
                So what exactly is a Break Room?
            </h1>
        </center>
        <center>
            <p style=" font-family: Courier; font-size: 14;color: lightblue; ">
                If your boss is giving you a hard time. Or your significant other is stressing you out. Or you had a break-up.
                </br>
            Or you were stuck in B'lore traffic for hours! Or you are a student whose energy is getting drained uselessly.
            </br>
            Or God forbid, your life is stressing you out badly. Whatever your reason or maybe no reason at all, The Break Room
            </br>
            provides you with a place to let loose. You've tried Yoga and Meditation or Gym, try something different.
            </br>
            Break something or everything at the Break Room!
        </p>
```

```
<p style=" font-family: Courier; font-size: 14; color: lightblue; ">

                    Everybody wants to blow off steam in their own way. That's only natural. Some people go for a drive, others go for a run.

                    <br/>

                    At The Break Room we hit things. With bats and anything you like. It's a little more liberating than the other stuff you do.

                </br>

                So come to the Break Room, pick up a bat, put on a Break Suit and feel good.

                </br>

                You know you want to :')

            </p>

        </br>

            <p style="font-family:Ariel black;font-size:60;text-align:center;color:red;font-weight:54">

                    <marquee direction="left">

                     DISCLAIMER: Seems like a dream stuff to do. Let us make that a reality!

                    </marquee>

            </p> </center>

        </br>

  <form action="/Project/LoginPage.php">

      <button type="submit" name="loginbtn" id="loginbtn"> Already have an account? LogIn </button>  </form>

  <form action="/Project/RegistrationPage.php">

      <button type="submit" name="registerbtn" id="registerbtn"> Don't have an account yet? SignUp </button> </form>

</body>

</html>
```

5.5.5 Account.php

```
<?php
	class Account {
		private $errorArray;
		private $sex;
		public function __construct($con) {
			$this->con = $con;
			$this->errorArray = array();
		}
		public function login($bn, $pw) {
		$query = mysqli_query($this->con, "SELECT * FROM members WHERE breakname='$bn' AND password='$pw'");
			if (mysqli_num_rows($query) == 1) {
				return true;
			}
			else {
				array_push($this->errorArray, Constants::$loginfailed);
				return;
			}
		}
		public function register($un, $bn, $sex, $age, $phnumber, $em, $pw1, $pw2) {
		$this->validateUsername($un);
		$this->validateBreakname($bn);
		$this->$sex=$sex; $this->validateAge($age);
		$this->validatePhone($phnumber);
		$this->validateEmail($em);
		$this->validatePasswords($pw1,$pw2);
```

```
            if(empty($this->errorArray) == true) {
                return $this->insertMemberDetails($un, $bn, $sex, $age, $phnumber, $em, $pw1);
            }
            else
                return false;
        }
        public function getError($error) {
                if (!in_array($error, $this->errorArray)) {
                    $error = "";
                }
                return "<span class='errorMessage'>$error</span>";
            }
            private function insertMemberDetails($un, $bn, $sex, $age, $phone, $em, $pw) {
            $date = date("Y-m-d");
            echo "INSERT INTO members VALUES( '$un', '$bn', '$sex', '$age', '$phone', '$em', '$pw', '$date')";
            $result = mysqli_query($this->con, "INSERT INTO members VALUES('$un', '$bn', '$sex', '$age', '$phone','$em', '$pw', '$date')");
            return $result;
        }
        private function validateUsername($un)
                {
                    if (strlen($un) > 38 || strlen($un) < 4) {
                    array_push($this->errorArray, Constants::$usernameerror);
                        return;
                }
```

```
}
private function validateBreakname($bn) {
                        if (strlen($bn) > 19 || strlen($bn) < 3) {
                        array_push($this->errorArray, Constants::$breaknameerror);
                        return;
                        }
private function validateAge($age) {
                        if (!filter_var($age, FILTER_VALIDATE_INT)) {
                        array_push($this->errorArray, Constants::$ageerror);
                        return;
                        }
                }
private function validatePhone($phnumber) {
                        if (!filter_var($phnumber, FILTER_VALIDATE_INT)) {
                                array_push($this->errorArray, Constants::$ageerror);
                                return;
                        }
                        if (strlen($phnumber) > 10) {
                        array_push($this->errorArray, Constants::$ageerror);
                        return;
}
}
                private function validateEmail($em)
                {
                        if (!filter_var($em, FILTER_VALIDATE_EMAIL)) {
                        array_push($this->errorArray, Constants::$emailerror);
                        return;
```

```
                }
                private function validatePasswords($pw1,$pw2)
                {
                        if ($pw1 != $pw2) {
                                array_push($this->errorArray,
Constants::$passworderror1);
                                return;
                        }
                        if (preg_match('/[^A-Za-z0-9]/', $pw1)) {
                                array_push($this->errorArray,
Constants::$passworderror2);
                                return;
                        }
                        if (strlen($pw1) > 10 || strlen($pw1) < 4) {
                                array_push($this->errorArray,
Constants::$passworderror3);
                                return;
                        }
                }

        }

    ?>
```

5.5.6 Status Page

```
<?php
include("Includes/Config.php");
   $Break=$_SESSION['breakname'];
   $sql="SELECT * FROM status WHERE breakname='$Break'";
   $rsl=mysqli_query($con,$sql);
   if(mysqli_num_rows($rsl)>0) {
         while ($row = mysqli_fetch_array($rsl)) {
                        $na=$row["breakname"];
                        $ra=$row["Rationale"];
                        $pid=$row["PId"];
                        $ran=$row["Rankings"];
                        $sma=$row["Smashes"];
                     }
         }
?>
<!DOCTYPE html>
<html>
<head>
   <title>Your Break Status</title>
   <link href="//netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
   <script src="//netdna.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
   <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
<style type="text/css">
body                                                                    {
```

```
    background-image: url("/Project/BR4.jpg");
            background-repeat: no-repeat;
            background-size: cover;
        }
    </style>
</head>
<body>
<link href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.min.css" rel="stylesheet">
<div class="container" >
  <center>
  <div class="row">
      <div class="col-lg-3 col-md-3 col-sm-12 col-xs-12">
          <div class="row" style="position: center">
              <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12" style="border-radius: 16px; background-position: center;" > <center>
                  <div class="well profile col-lg-12 col-md-12 col-sm-12 col-xs-12">
                      <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 text-center" style="background-color: yellow" >
                          <figure>
                              <img src="http://www.localcrimenews.com/wp-content/uploads/2013/07/default-user-icon-profile.png" alt="" class="img-circle" style="width:75px;" id="user-img">
                          </figure>
<h5 style="text-align:center;"><strong id="user-name">Break Name: <?php echo $na; ?></strong></h5>
<p style="text-align:center;font-size: smaller;overflow-wrap: break-word;" id="user-email"></p>
<p style="text-align:center;font-size: smaller;"><strong></strong><span class="tags" id="user-status">Rationale: <?php echo $ra; ?></span></p>
```

```
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 divider text-center"></div>

                    <p style="text-align:center;font-size: smaller;"><strong>PID: <?php echo $pid; ?></strong></p>

                    <p style="text-align:center;font-size: smaller;" id="user-role">Software Engineer</p>

                    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 divider text-center"></div>

                      <div class="col-lg-6 left" style="text-align:center;overflow-wrap: break-word;">

                        <h4><p style="text-align: center;"><strong id="user-globe-rank"><?php  echo $ran; ?></strong></p></h4>

                        <p><small class="label label-success">Global Ranking</small></p>

                      </div>

                      <div class=" col-lg-6 left" style="text-align:center;overflow-wrap: break-word;">

                        <h4><p style="text-align: center;"><strong id="user-college-rank"><?php echo $sma; ?> </strong></p></h4>

                        <p> <small class="label label-warning">Smashes</small></p>

                        <!-- <button class="btn btn-info btn-block"><span class="fa fa-user"></span> View Profile </button>-->

                      </div>

                    </div>

                  </div>

                  </center>

                </div>

              </div>

            </div>

          </center>
```

```
        </div>
    </div>
</div>
</body>
</html>
```

5.5.6 Stored Procedure snippet

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Proc`()
    NO SQL
SELECT * FROM articles$$
DELIMITER ;


BEGIN
Proc();
END
```

The stored procedure created here is named as Proc(). And it retrieves the entire contents of the Articles table when Proc() is called.

5.5.7 Trigger snippet

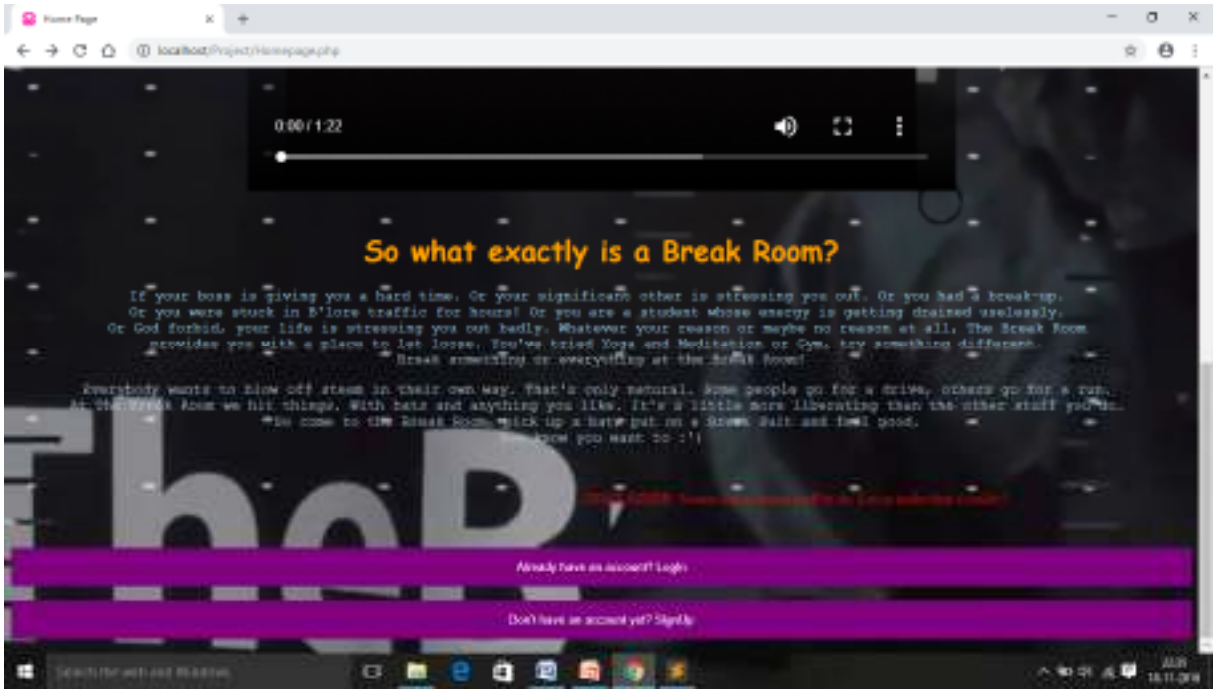
```
CREATE TRIGGER `logh` AFTER INSERT ON `bookings`
FOR EACH ROW INSERT into logs VALUES (New.BreakName,CURRENT_TIMESTAMP);
```

'logh' trigger is created on the 'logs' table. When we enter the proper values in the login page and submit, the values 'breakname' and 'time' are updated in the 'bookings' table. While this triggers and the subsequent 'logs' value is updated in the 'logs' table.

Chapter 6

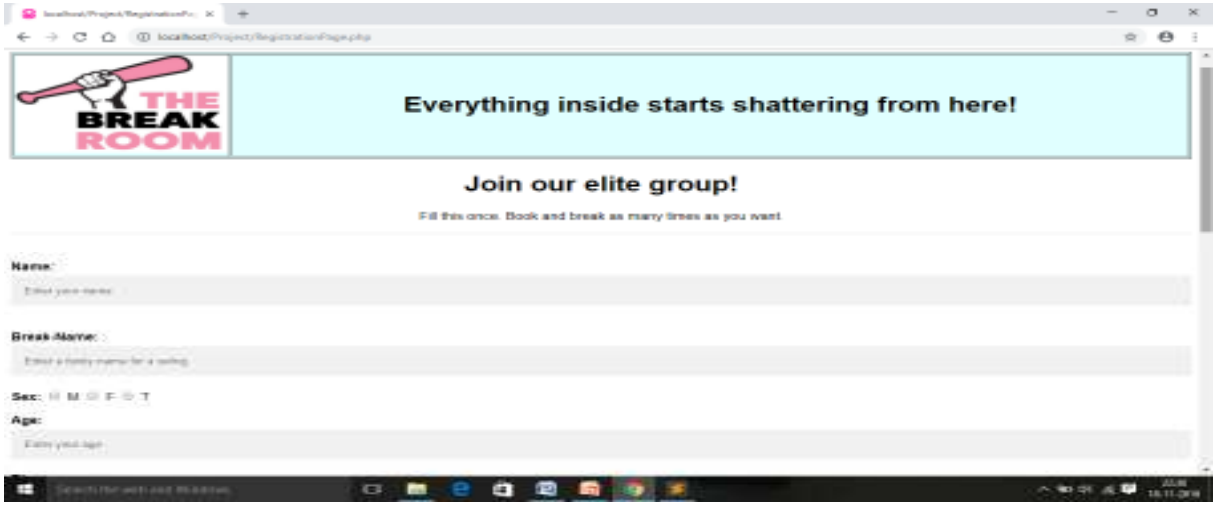
Snapshots

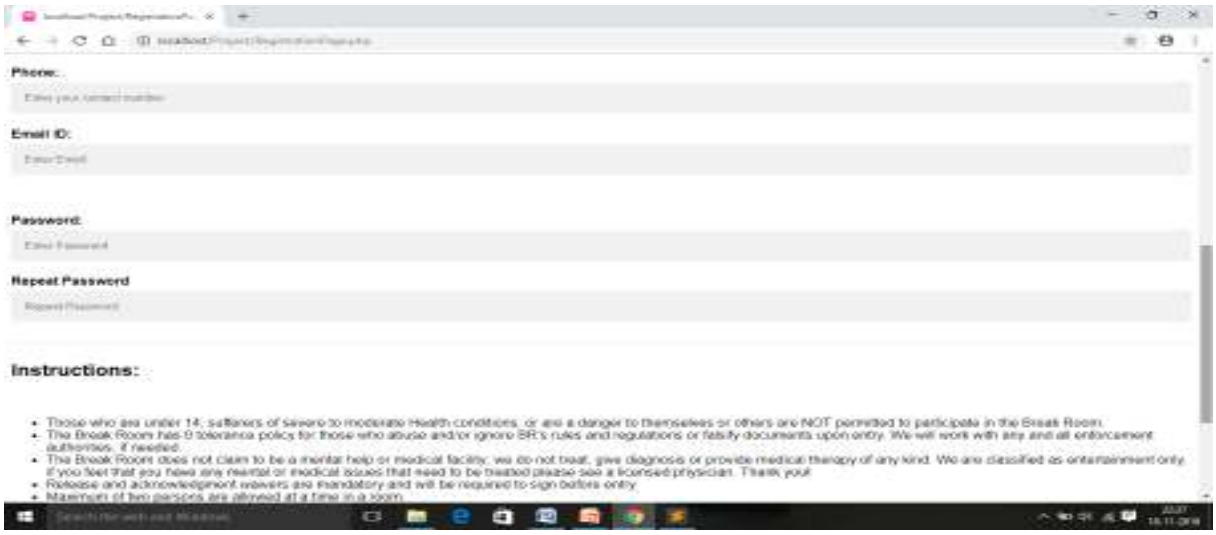
6.1 Home Page

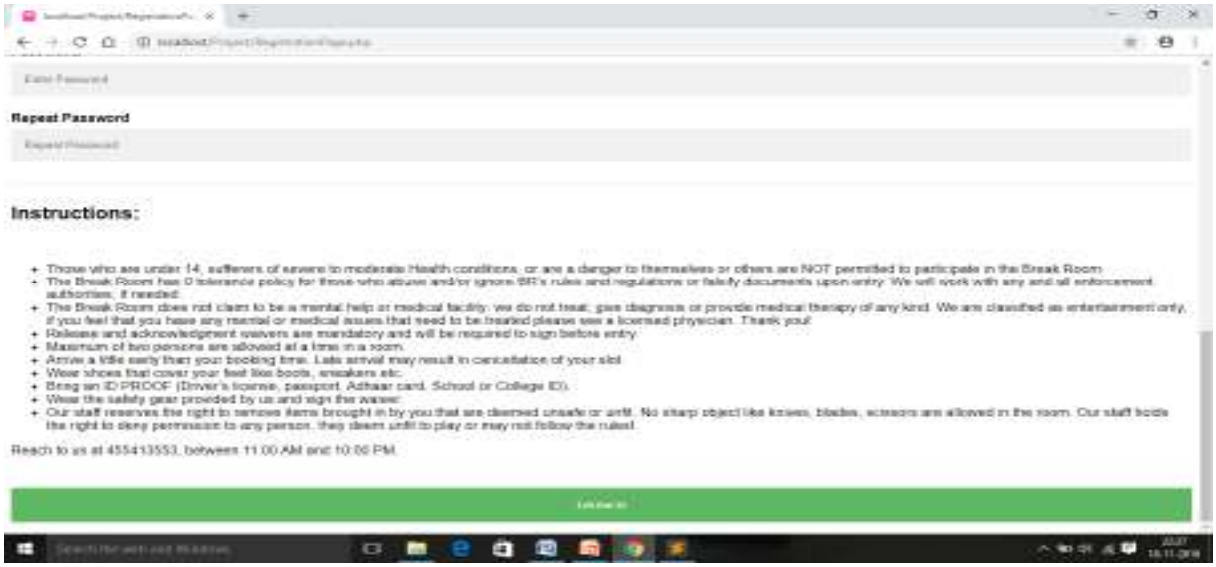


This page provides the face for the Break Room Management System. It gives a brief description of The Break Room and allows users to go to login or registration page.

6.2 Registration Page

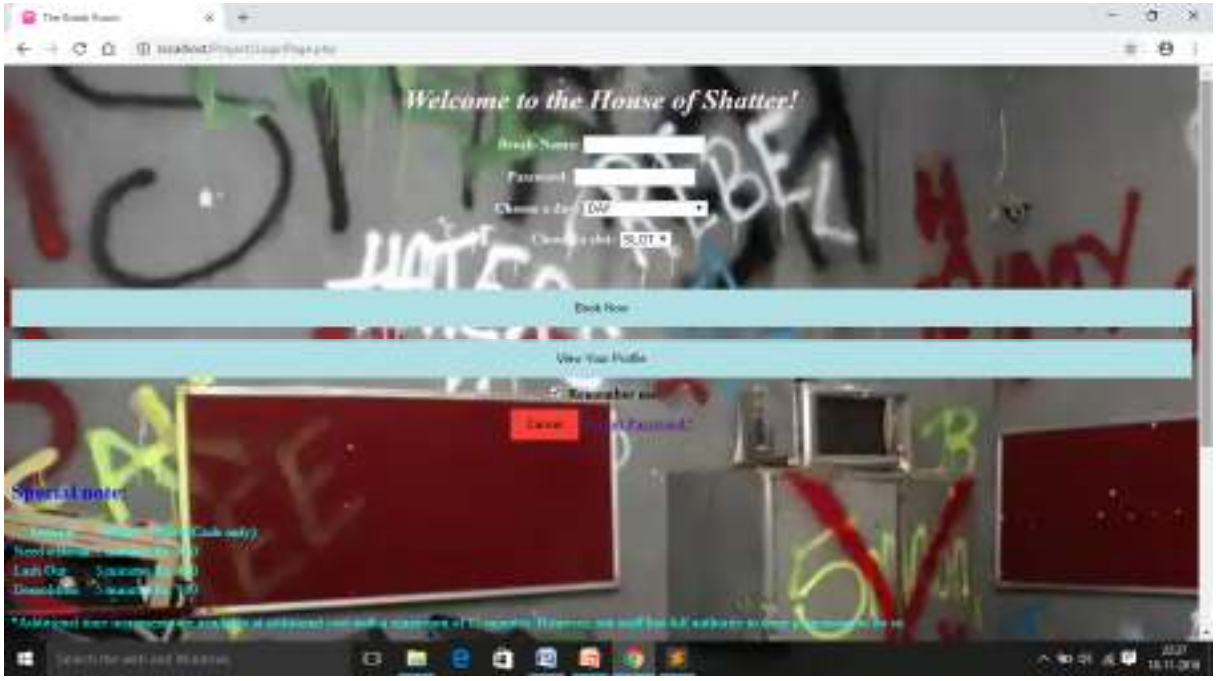






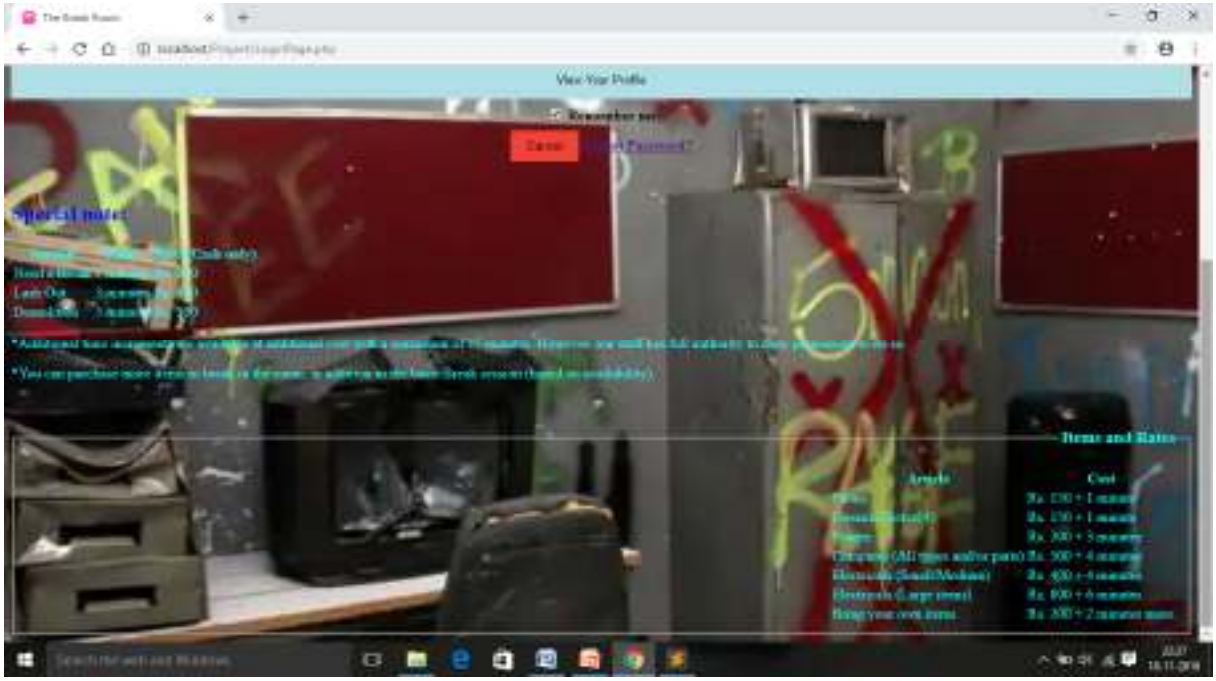
This is where the users can register to our platform, and join our therapeutic community!

6.3 Login Page



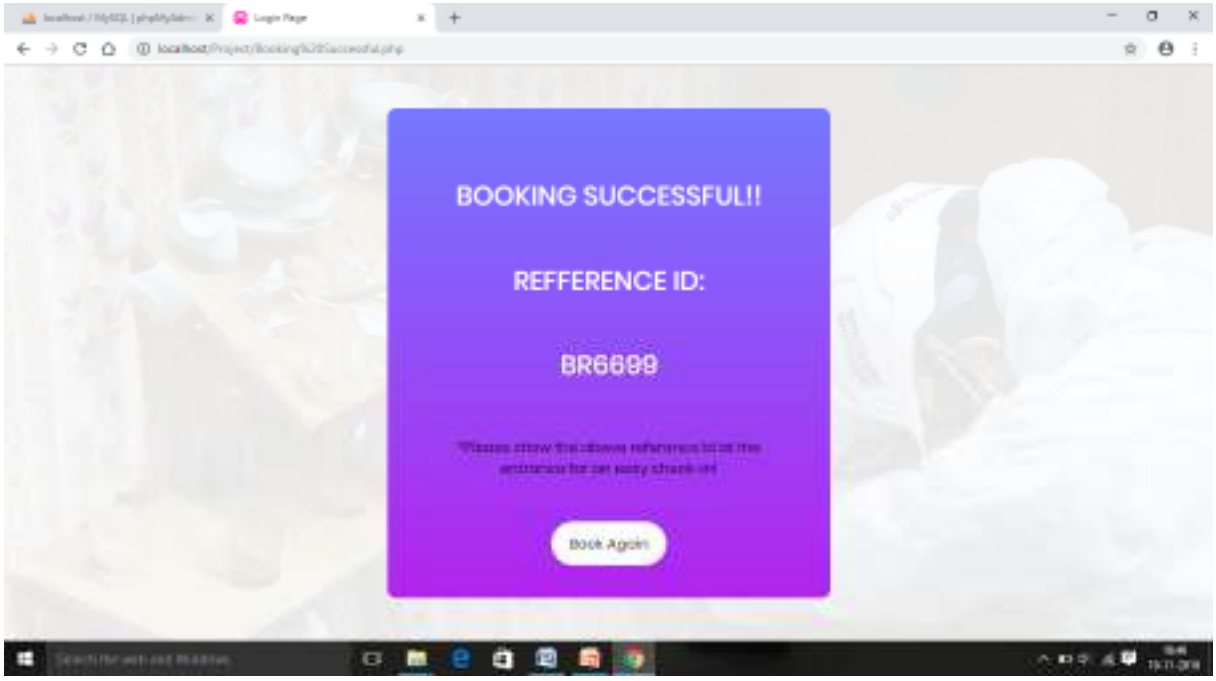
This provides the users to login either to visit their profile and view their Break stats or to book a slot.

6.4 The Articles



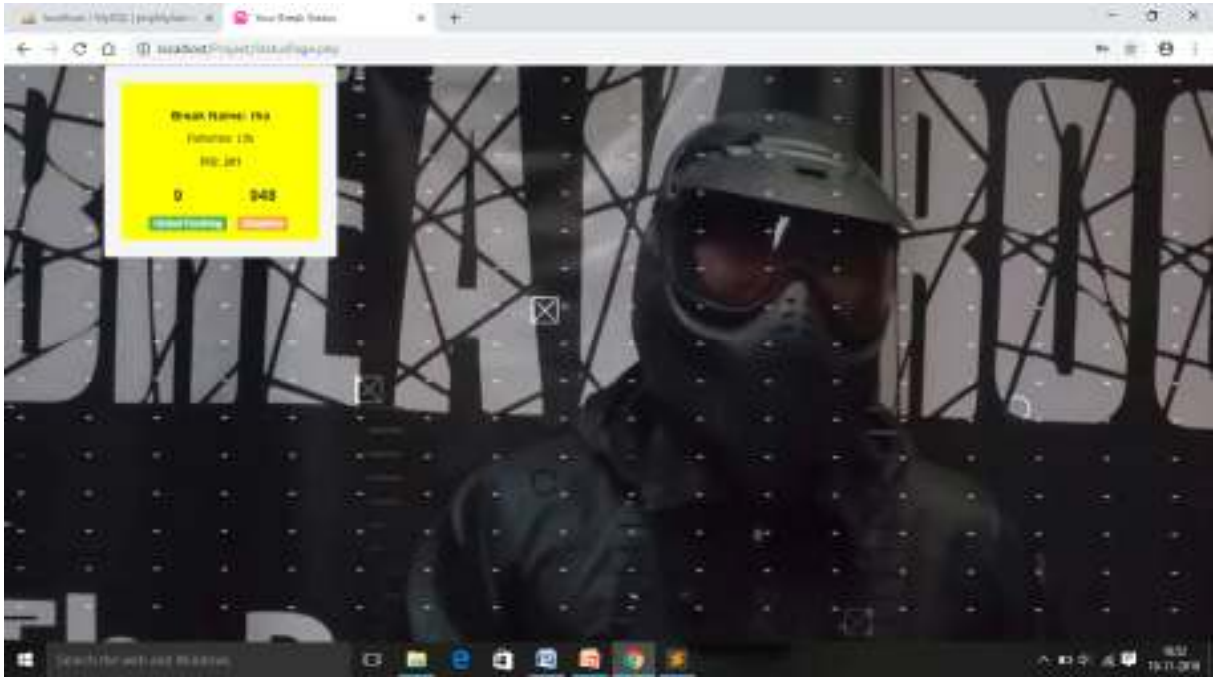
This describes the type-of-Break one could choose from. Also this shows some of the articles available and related time and cost.

6.5 Booking Successful Page



Once the user logs successfully and books a slot without any hassle, their respective reference id is shown for future reference.

6.6 Status Page (Statistics)



Once the user successfully logs and wants to view his/her statistics, this profile page shows him the same.

6.7 Tables

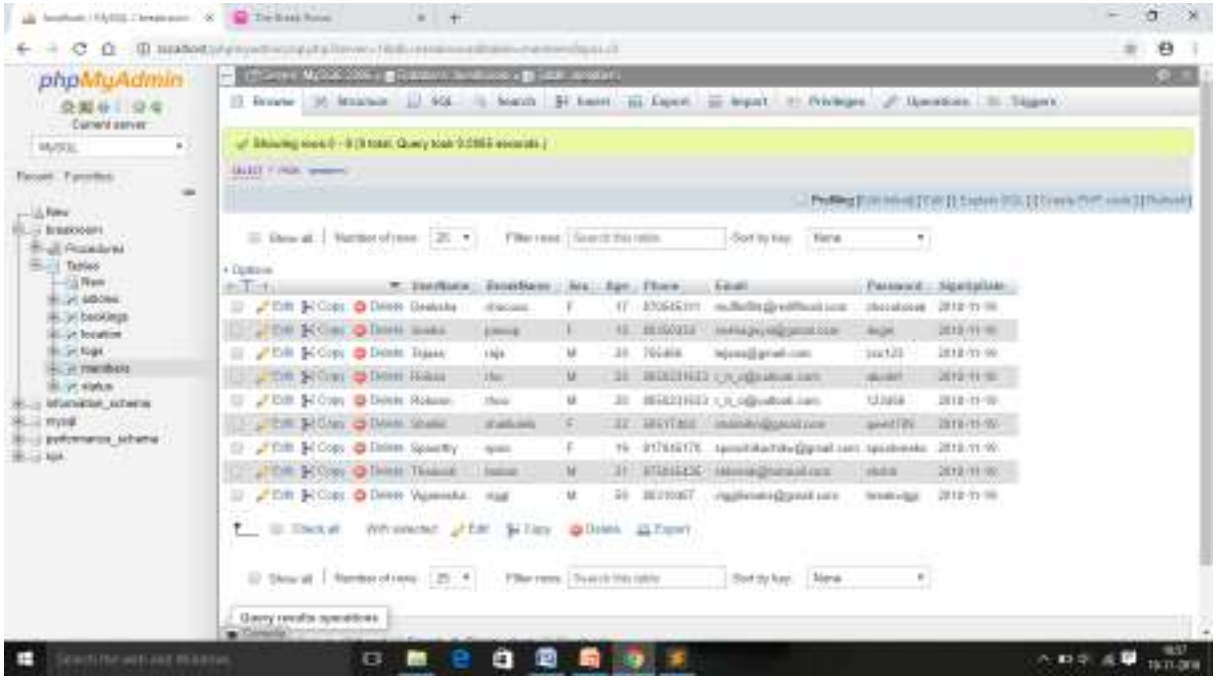


Table 6-1 Members Table

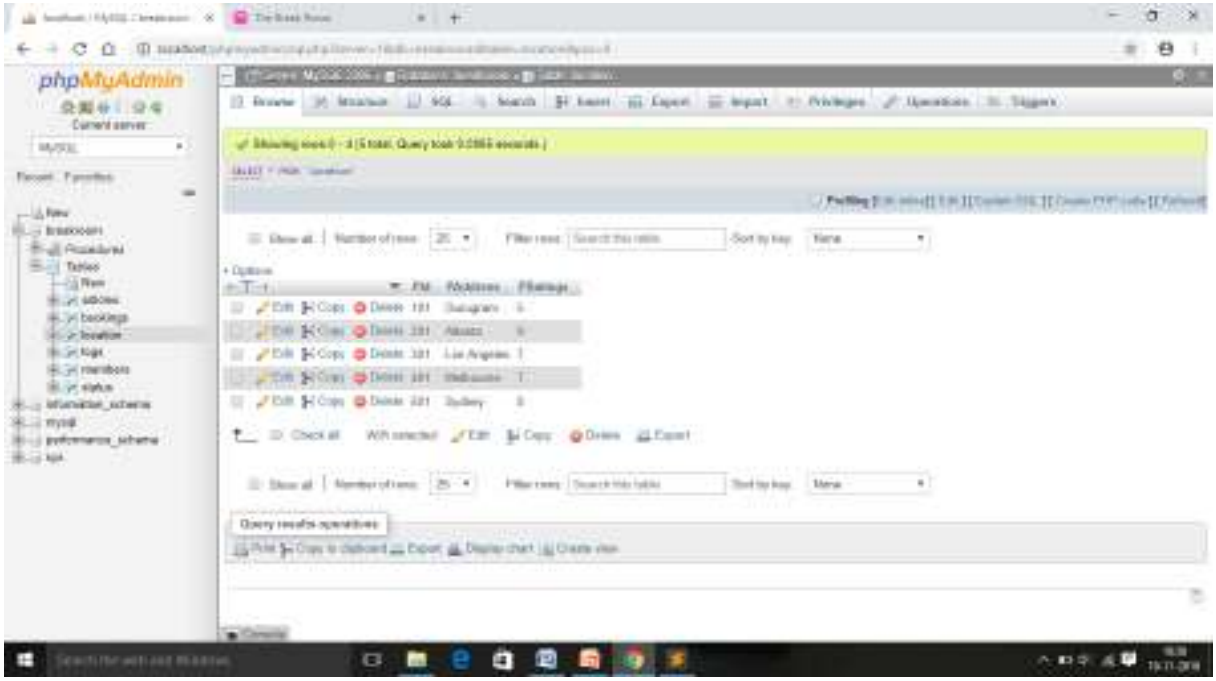


Table 6-2 Location Table

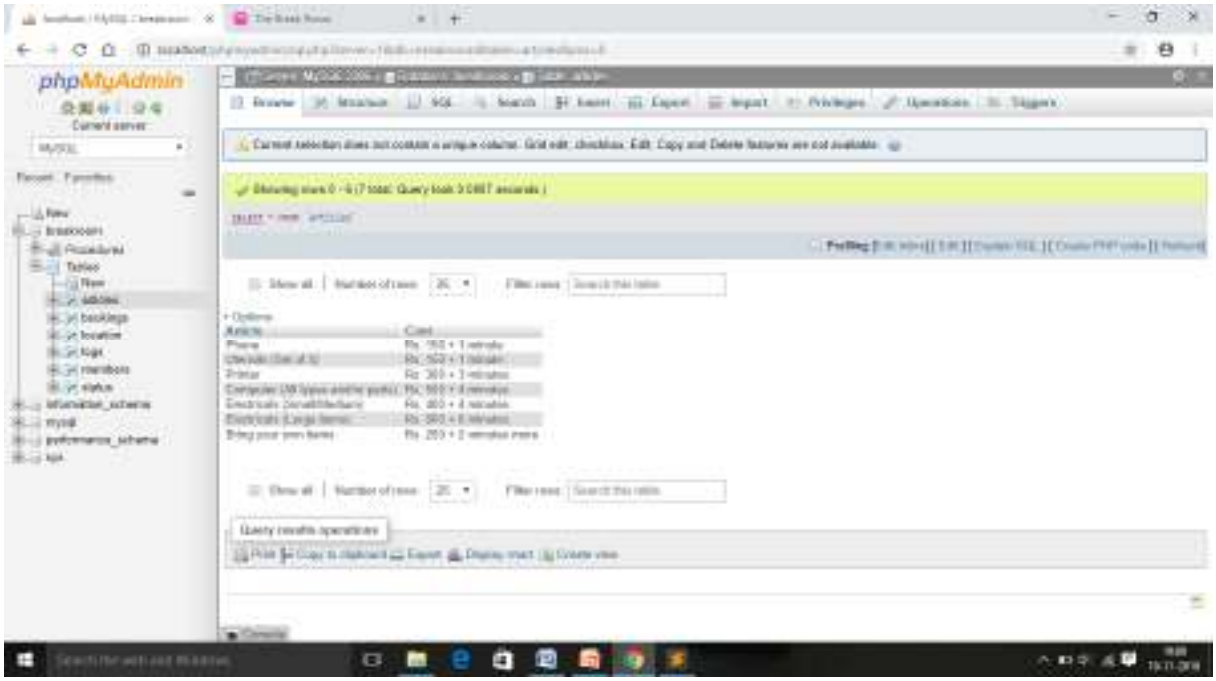


Table 6-3 Articles Table

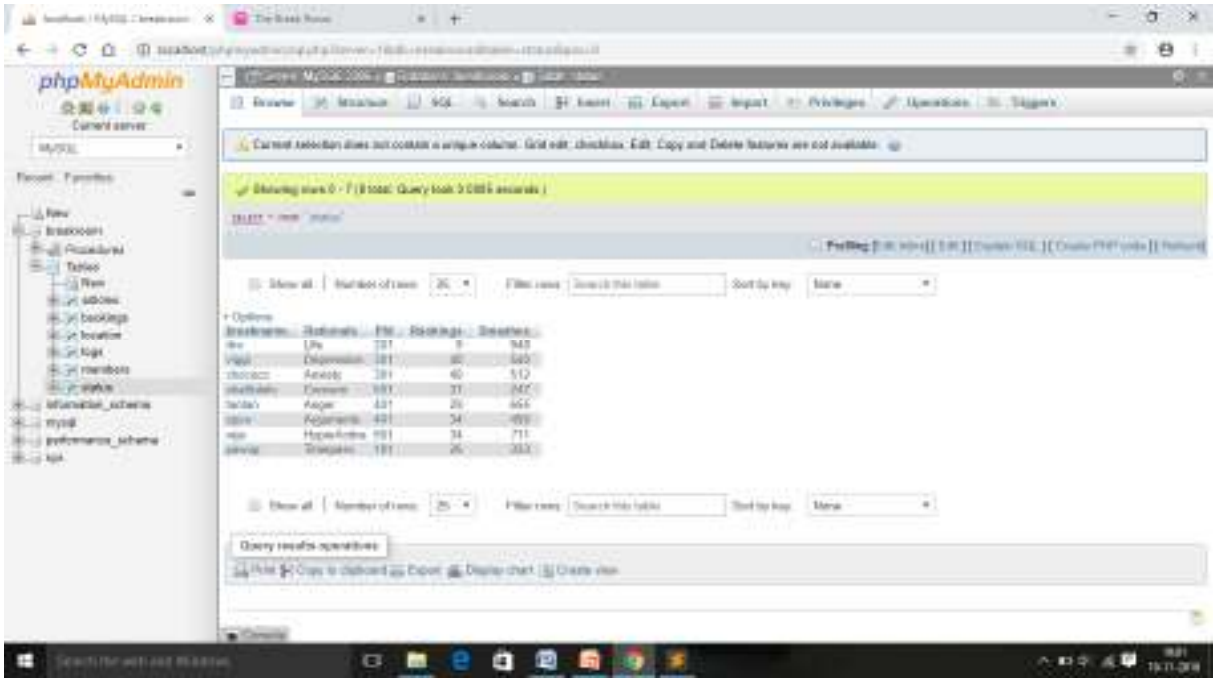


Table 6-4 Status Table

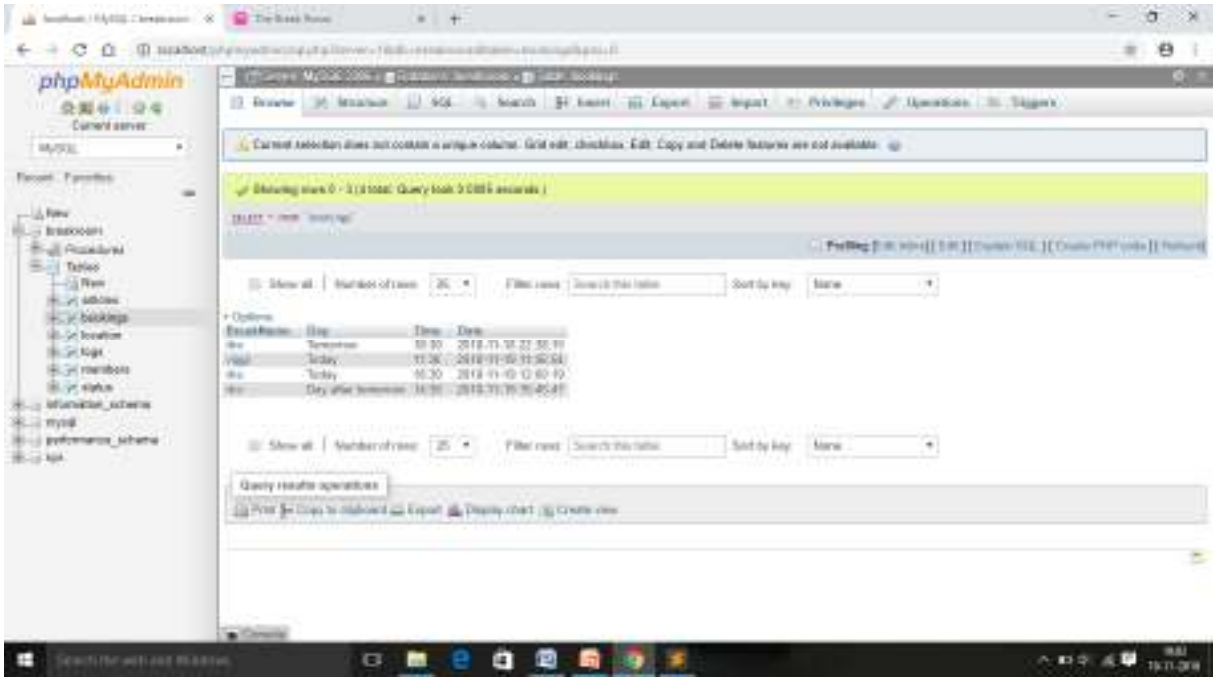


Table 6-5 Bookings Table

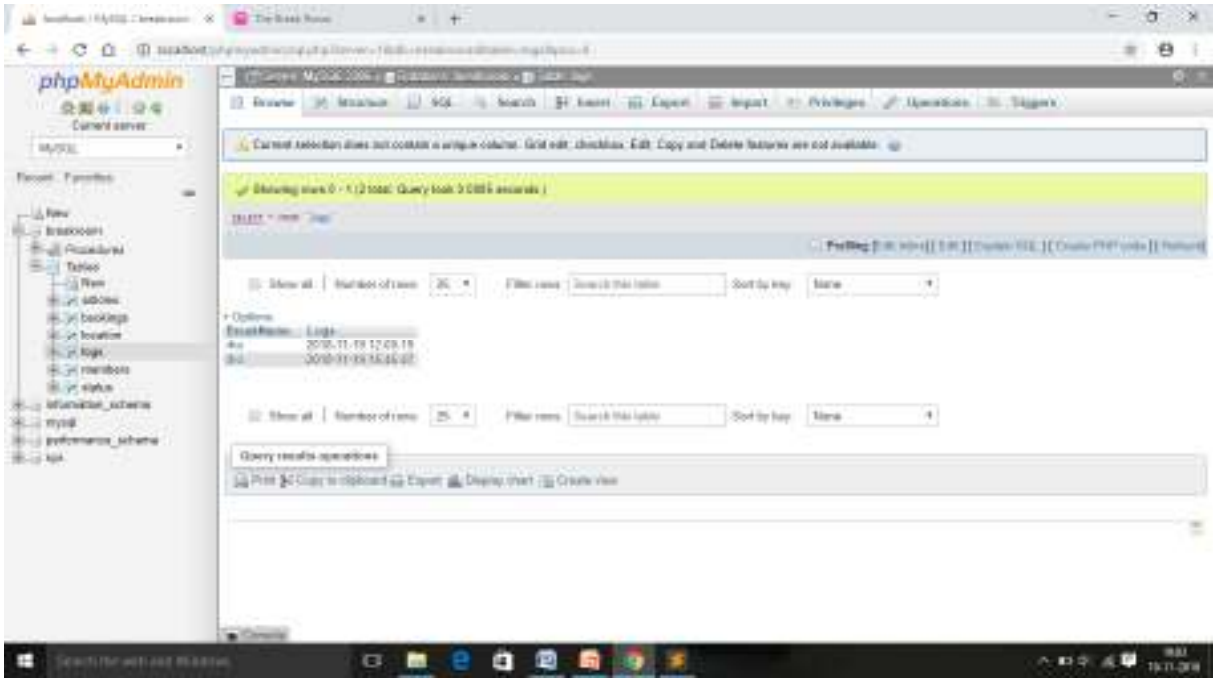


Table 6-6 Logs Table

Chapter 7

CONCLUSION

With the present theoretical inclinations of our syllabus, it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The construction of this mini project was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts, thus making us more competent.

This project 'Break Room Management System' is about The Break Room. The bottom line of which is that it is one of those friends in your life who relieves your stress without doing anything. This project is just to give an insight about the unsung concept of Break Room. While this might just be the most chic production design you'll see, this will also be highly useful if and when implemented. You'll absolutely love it here.

BIBLIOGRAPHY

- W3Schools (HTML, CSS and PHP references) – https://www.w3schools.com/
- PHP Official Documentation – http://php.net/docs.php
- The MySQL Documentation – https://dev.mysql.com/doc
- Stack Overflow – https://stackoverflow.com
- Stack Exchange – https://stackexchange.com/
- Udemy - https://udemy.com/
- Wikipedia - https://wikipedia.org
- YouTube – https://www.youtube.com