

CHAPTER 1

INTRODUCTION

Bug Tracking System is an ideal solution to track the bugs of a product, solution or an application. Bug Tracking System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Tracking System.

The main objective of this system is develop flawless system, which is access real time information from the list of bug bugs given, 24 hours a day 365 days in a year. Another aim is that manage listed of projects . The another main objective of this system is to manage the list of the defects or bugs in the project and make the project user friendly and bugs free system.

This system maintains the products, Bugs and bug Tracking. It has advantage of maintaining bug history it stores all the details of bug and their solutions .

Existing System

In any software development bugs are inevitable. Let it be in any kind of product bugs arise at any phase of development. One has to take a great care in the proper maintenance and resolution of the bugs. In the Existing system the bugs are not properly maintained and they are simply relied on shared lists and email to monitor the bugs.

One has to search the whole database for the details of particular bug which might have occurred sometime earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs. In order to have an efficient product bugs must be maintained properly and should be resolved in time both to reduce time and money spent on the development.

CHAPTER 2

REQUIREMENTS ANALYSIS

The requirement analysis specifies the requirements needed to develop a graphic project. In this phase, we collect the requirements needed for designing the project. The requirements collected are then analyzed and carried to the next phase.

2.1 SOFTWARE REQUIREMENTS:

1. Operating System: Windows 10
2. Scripting Language: HTML,CSS,BOOTSTRAP PHP
3. Front-end Development: HTML,CSS & BOOTSTRAP
4. Back-end Development:PHP & MYSQL

2.2 HARDWARE REQUIREMENTS

1. Processor – Pentium IV or above
2. RAM – 2 GB or more
3. Hard disk – 3 GB or more

CHAPTER 3

DESIGN

3.1 ER DIAGRAM

An **Entity – Relationship model (ER model)** describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

The E-R diagram of our BUG TRACKING SYSTEM contains 7 Entities

(Client Login,Employee,Department, Project,Assign Project,Emp_assignment,Bug report)

In the E-R Diagram:

CLIENT LOGIN:is the entity where we can enter the details about username and Password.

EMPLOYEE: is the entity where we can enter and the details about Employee .

DEPARTMENT: here the above employee belongs to the respective department.

PROJECT: the bug file given by the client.

ASSIGN PROJECT: the assign project is entered in the project.

EMP_ASSIGNMENT: Project is given to the employee.

BUG REPORT: the report of the assigned project by the respective employee.

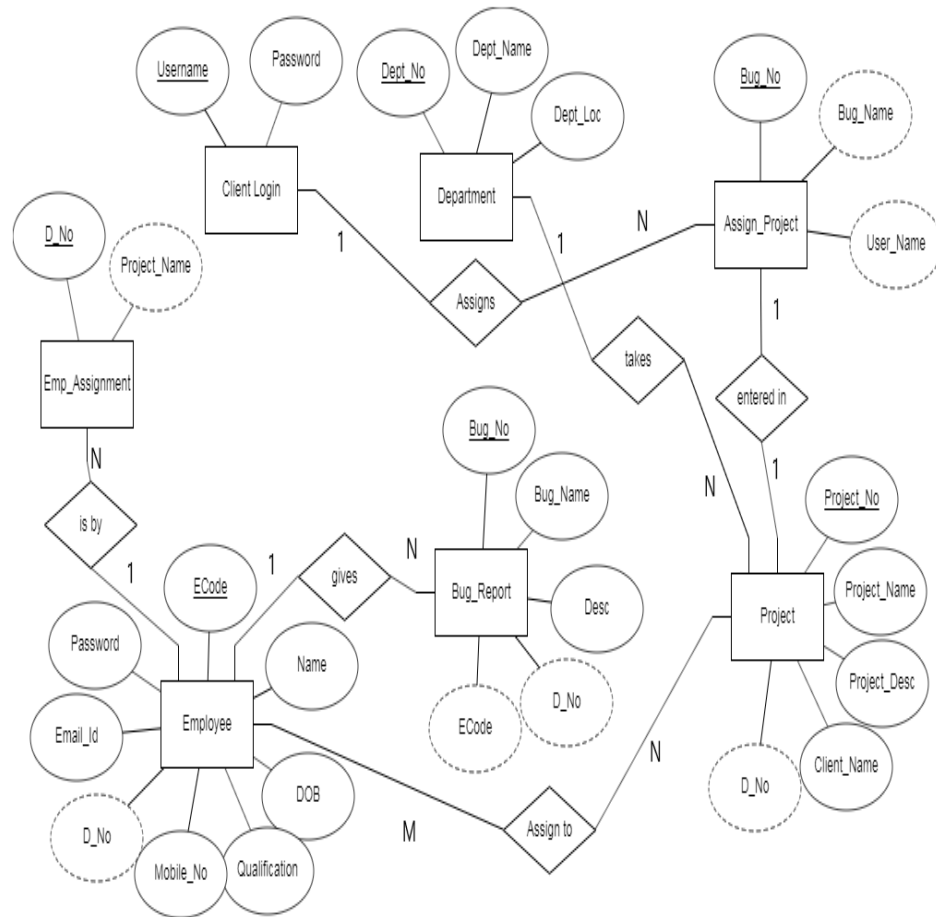


Figure 1: Entity – Relational diagram of Bug Tracking System

BUG TRACKING SYSTEM

SCHEMA:

CLIENT_LOGIN

<u>Username</u>	Password
-----------------	----------

DEPARTMENT

<u>Dept_No</u>	Dept_Name	Dept_Loc
----------------	-----------	----------

PROJECT

<u>Project_No</u>	Project_Name	Project_Desc	Client_Name	D_No	ECode
-------------------	--------------	--------------	-------------	------	-------

ASSIGN_PROJECT

<u>Bug_No</u>	Bug_Name	Username
---------------	----------	----------

EMPLOYEE

<u>ECode</u>	Name	DOB	Qualification	Mobile_No	D_No	Email_Id	Password
--------------	------	-----	---------------	-----------	------	----------	----------

EMP_ASSIGNMENT

<u>D_No</u>	Project_Name
-------------	--------------

BUG_REPORT

<u>Bug_No</u>	Bug_Name	Desc	ECode	D_No
---------------	----------	------	-------	------

Figure 2: Relational Schema Diagram of Bug Tracking System

The term "**schema**" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of **relational** databases).

USECASE DIAGRAM

The boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which are the specific roles played by the actors within and around the system.

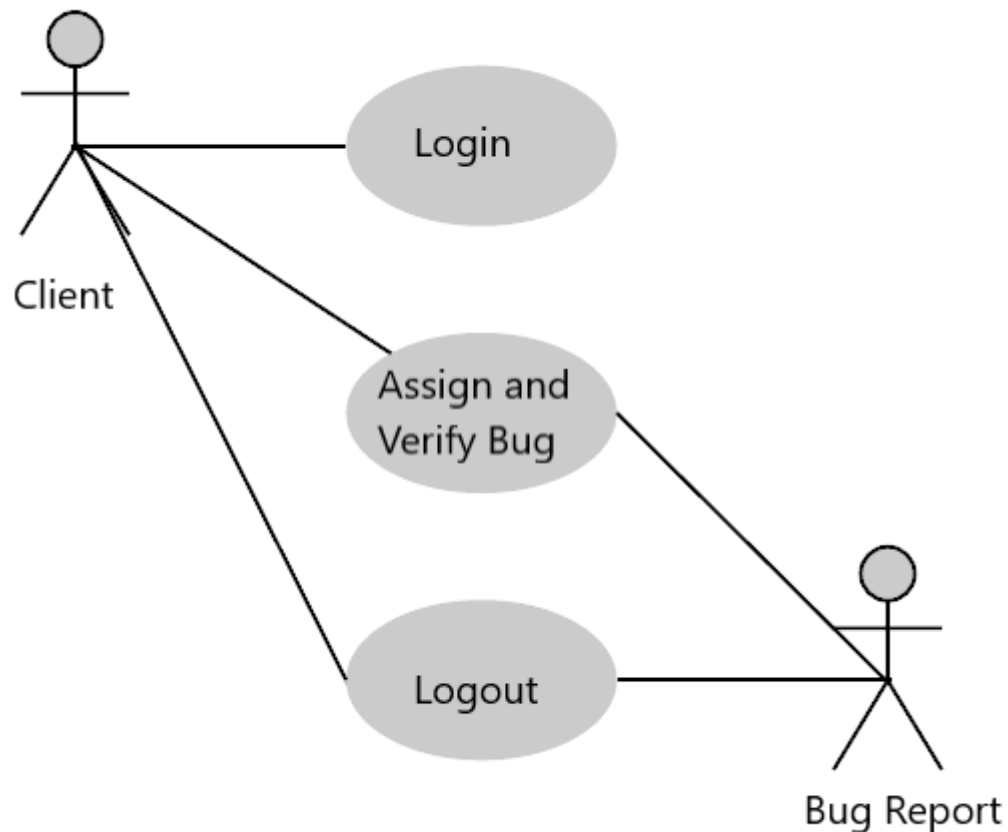


Figure 3: Use case diagram of BUG TRACKING SYSTEM

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION TO FRONT END TOOL

4.1.1 HTML,CSS & BOOTSTRAP

HTML: **HTML** stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text.

HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991. The first ever version of HTML was HTML 1.0 but the first standard version was HTML 2.0 which was published in 1999.

CSS: **Cascading Style Sheets**, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of HTML that makes up each web page. CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

BOOTSTRAP: Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS to facilitate the development of responsive, mobile-first sites and apps. Responsive design makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly; the mobile first approach assumes that smartphones, tablets and task-specific mobile apps are

employees' primary tools for getting work done and addresses the requirements of those technologies in design

.

4.2 INTRODUCTION TO BACK END TOOL

4.2.1 PHP & MYSQL

PHP: The term PHP is an acronym for *PHP: Hypertext Preprocessor*. PHP is a server side scripting language designed specifically for web development. PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc. There are also many hash functions available in PHP to encrypt user's data that makes PHP secure and reliable to be used as a server-side scripting language.

MYSQL: MySQL is a Relational DataBase Management System (RDBMS).

RDBMS means R--DB--MS.

- DB stands for Database, a repository for the information store.

- i. The data in a database is organized into tables, and each table is organized into rows and columns.
- ii. Each row in a table is called a record. A record may contains several pieces (called fields) of information, and each column in a table is known as a field.

-MS stands for Management System, the software that allows you to insert, retrieve, modify, or delete records.

-R stands for Relational, indicates a particular kind of DBMS that is good at relating information stored in one table to information stored in another table by looking for elements common to each of them. Relational DBMS has the advantage of efficient storage, and retrieval mechanisms for data, and uses normalization process during

design of RDBMS. Database normalization process is beyond the scope of this article, and several references are available.

MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. The server operating systems is usually a Linux (like Redhat 9.0 etc.) or Windows 2000 operating system. Typically MySQL is supported on Windows XP, Windows Server 2003, Red Hat Fedora Linux, and Debian Linux, and others. As with any other client/server application, MySQL is a multi-user database system, meaning several users can access the database simultaneously. Here:

- The server (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.

- Clients are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards based SQL (SQL stands for Structured Query Language) language. The client program may contact the server programmatically (meaning a program call the server during execution) or manually. For example, when you are issuing commands over a telnet session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

4.3 CONNECTIVITY OF THE DATABASE

There are three ways of working with MySQL and PHP

1. MySQLi (object-oriented)
2. MySQLi (procedural)
3. PDO

Connecting to MySQL database using PHP

1. **Using MySQLi object-oriented procedure:** We can use the MySQLi object-oriented procedure to establish a connection to MySQL database from a PHP script.

Syntax:

```
?php
$servername = "localhost";
$username = "root";
$password = "";
$conn = mysqli_connect($servername, $username,
$password,"bug_tracking_system") or die("unable to connect to host");
?>
```

Output:

Registration Successful

Explanation: We can create an instance of the mysqli class providing all the necessary details required to establish the connection such as host, username, password etc. If the instance is created successfully then the connection is successful otherwise there is some error in establishing connection.

4.4 MODULES

1. Client registration and login

```
<html>
<head>
<title>Login and Register Form Design</title>
<link rel="stylesheet" type="text/css" href="style1.css">
</head>
<body>
<marquee>
Welcome To Bugtracking System
</marquee>
```

```
<div class="loginbox">
  
  <div class="form">
    <form action="validation.php" class="Login-form" method="post">
      <h1>Login Here</h1>
      <p>Username</p>
      <input type="text" name="user" placeholder="Enter username" required>
      <p>Password</p>
      <input type="password" name="password" placeholder="Enter Password"
required>
      <input type="submit" name="submit" value="Login">
      <p class="message">Don't have an account? <a href="#"> Register</a>
    </p>
  </form>
  <form action="registration.php" class="Register-form" method="post">
    <h1>Register Here</h1>
    <p>Username</p>
    <input type="text" name="user" placeholder="Enter username" required>
    <p>Password</p>
    <input type="password" name="password" placeholder="Enter Password"
required>
    <input type="submit" name="submit" value="Register" >
    <p class="message">Already registered? <a href="#"> Login</a>
  </p>
  </form>
</div>
</div>
<script src='https://code.jquery.com/jquery-3.4.1.min.js'></script>
<script>
$('.message a').click(function(){
$('.form').animate({height: "toggle", opacity: "toggle"}, "slow");
});
</script>
```

```
</body>
```

```
</html>
```

Manipulation Details

Many details such as

Users, product name, product quantity, user_groups, sale_prices, categories and media details can be manipulated by using insert, update and delete queries. The code below is a sneak peak into the execution of queries.

1. Insertion details:-

```
<?php
include "../connection.php";
$ac=$_POST['Dept_No'];
$an=$_POST['Dept_Name'];
$d=$_POST['Dept_Loc'];
$query="insert      into      department(Dept_No,Dept_Name,Dept_Loc)
values($ac,$an,$d)";
mysqli_query($conn,$query) or die($query."Can't Connect to Query...");
?>

<html>
<head>
<title>Registration Form</title>
</head>
<body>
<link href = "registration.css" type = "text/css" rel = "stylesheet" />
<link href = "../style.css" type = "text/css" rel = "stylesheet" />
<ul>
<li style="float:right;"><a href="../index.php">Back to homepage</a></li>
</ul>
<h2>Agent</h2>
<form name = "form1" action='modified.php' method = 'POST' enctype =
"multipart/form-data" >
<div class = "container">
```

```
<div class = "form_group">
<label>Department Number:</label>
<input type = "text" name = "Dept_No" required />
</div>
<div class = "form_group">
<label>Department Name</label>
<input type = "text" name = "Dept_Name" value = "" required />
</div>
<div class = "form_group">
<label>Department Location</label>
<input type = "text" name = "Dept_Loc" value = "" required />
</div>
<div class = "form_group">
<input type = "submit" value = "submit"/>
</div>
<div class = "form_group">
<input type = "reset" value = "reset"/>
</div>
</div>
</form>
</body>
</html>
```

2.Updation of Details:-

```
<?php
include "input.php";

$sql = "select * from department";
$result = mysqli_query($conn,$sql);
?>
<html>
<body>
```

```
<link href = "../style.css" type = "text/css" rel = "stylesheet" />
<link href = "registration.css" type = "text/css" rel = "stylesheet" />
<table width = "100%" border = "1" cellspacing = "1" cellpadding = "1">
<tr>
<td>Department Number</td>
<td>Department Name</td>
<td>Department Location</td>
<td colspan = "2">Action</td>
</tr>
<?php
while($row = mysqli_fetch_object($result)){
    ?>
<tr>
<td>
<?php echo $row->Dept_No;?>
</td>
<td>
<?php echo $row->Dept_Name;?>
</td>
<td>
<?php echo $row->Dept_Loc;?>
</td>
<td> <a href="delete.php?id =
<?php echo $row->Dept_No;?>" onclick="return confirm('Are You
Sure')">Delete
</a>
</td>
</tr>
<?php } ?>
</table>
<?php header('Location: modified1.php')?>;
</body>
</html>
```

```
<html>
<head>
<title>Registration Form</title>
</head>
<body>
<link href = "registration.css" type = "text/css" rel = "stylesheet" />
<link href = "../style.css" type = "text/css" rel = "stylesheet" />
<ul>
<li style="float:right;"><a href="../index.php">Back to homepage</a></li>
</ul>
<h2>Agent</h2>
<form name = "form1" action='modified.php' method = 'POST' enctype =
"multipart/form-data" >
<div class = "container">
<div class = "form_group">
<label>Department Number:</label>
<input type = "text" name = "Dept_No" required />
</div>
<div class = "form_group">
<label>Department Name</label>
<input type = "text" name = "Dept_Name" value = "" required />
</div>
<div class = "form_group">
<label>Department Location</label>
<input type = "text" name = "Dept_Loc" value = "" required />
</div>
<div class = "form_group">
<input type = "submit" value = "submit"/>
</div>
<div class = "form_group">
<input type = "reset" value = "reset"/>
</div>
</div>
```

```
</form>
</body>
</html>
```

3.Deletion of Details:-

```
<?php
include "../connection.php";
if(isset($_GET['id'])){
    $sql = "delete from department where Dept_No = '".$_GET['id']."'";
    $result = mysqli_query($conn,$sql);
}
header('Location:modified1.php');
?>
<html>
<head>
<title>Registration Form</title>
</head>
<body>
<link href = "registration.css" type = "text/css" rel = "stylesheet" />
<link href = "../style.css" type = "text/css" rel = "stylesheet" />
<ul>
<li style="float:right;"><a href="../index.php">Back to homepage</a></li>
</ul>
<h2>Agent</h2>
<form name = "form1" action='modified.php' method = 'POST' enctype =
"multipart/form-data" >
<div class = "container">
<div class = "form_group">
<label>Department Number:</label>
<input type = "text" name = "Dept_No" required />
</div>
```



```
<div class = "form_group">
<label>Department Name</label>
<input type = "text" name = "Dept_Name" value = "" required />
</div>
<div class = "form_group">
<label>Department Location</label>
<input type = "text" name = "Dept_Loc" value = "" required />
</div>
<div class = "form_group">
<input type = "submit" value = "submit"/>
</div>
<div class = "form_group">
<input type = "reset" value = "reset"/>
</div>
</div>
</form>
</body>
</html>
```

CHAPTER 5**TESTING****5.1 TESTING**

Testing is the process of executing a program to find the errors. A good test has the high probability of finding a yet undiscovered error. A test is vital to the success of the system. System test makes a logical assumption that if all parts of the system are correct, then goal will be successfully achieved.

5.2 TYPES OF TESTING

5.2.1 Module Testing.

5.2.2 Integration Testing.

5.2.1 Module Testing

Module testing is the testing of complete code objects as produced by the compiler when built from source.

A library may be composed of a single complied object or several complied objects. There is only a slight difference between unit testing and module testing. Modules are fully formed chunks of coherent source code that can typically be tested by driving a few functions signatures with various stimuli. On the other hand, unit testing (which is considered as part of the implementation phase for this software development process) may involve testing one small part of a function that will never formally implement any function interface.

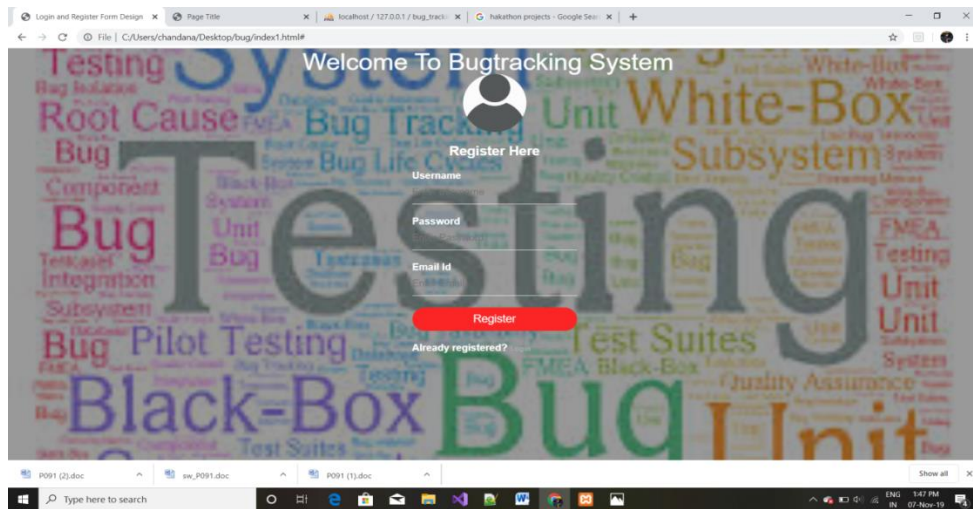
As a result of modules being more self-contained, module testing will likely require less testing infrastructure such as test harness and test stubs. The testing of modules could perhaps even be automated so that they can be included in regression test suites or a acceptance test suites.

5.2.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

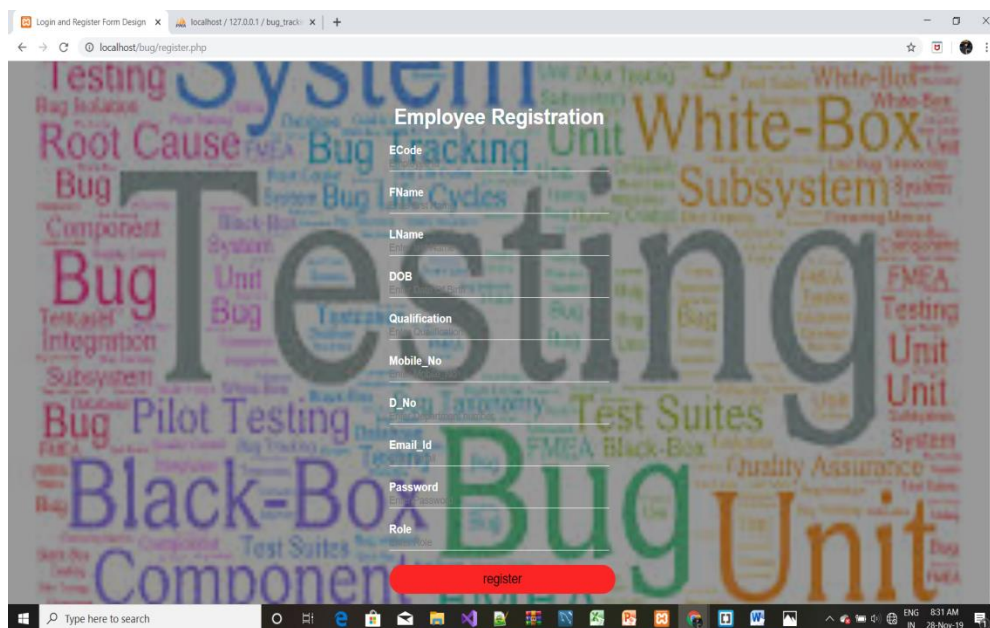
Test Case Id	Description	Input Data	Expected Output	Actual Output	Status
1	Client Login Page	Username Password	Successfully Logged in	Successfully Logged in	Pass
2	Employee Register Page	Username Password	Successfully Registered in	Successfully Registered in	Pass
3	Department	Dept No Dept Name Dept Loc	Display the table of Department	Display the table of Department	Pass
4	Project	Project No Project Name Client Name	Display the project details	Display the project details	Pass
5	Assign Project	Project assign by Client	Gives the bug report	Gives the bug report	Pass
6	Bug Report	Report of the bugs	Display the bug report	Display the bug report	Pass

BUG TRACKING SYSTEM



Snapshot 2: "Register page"

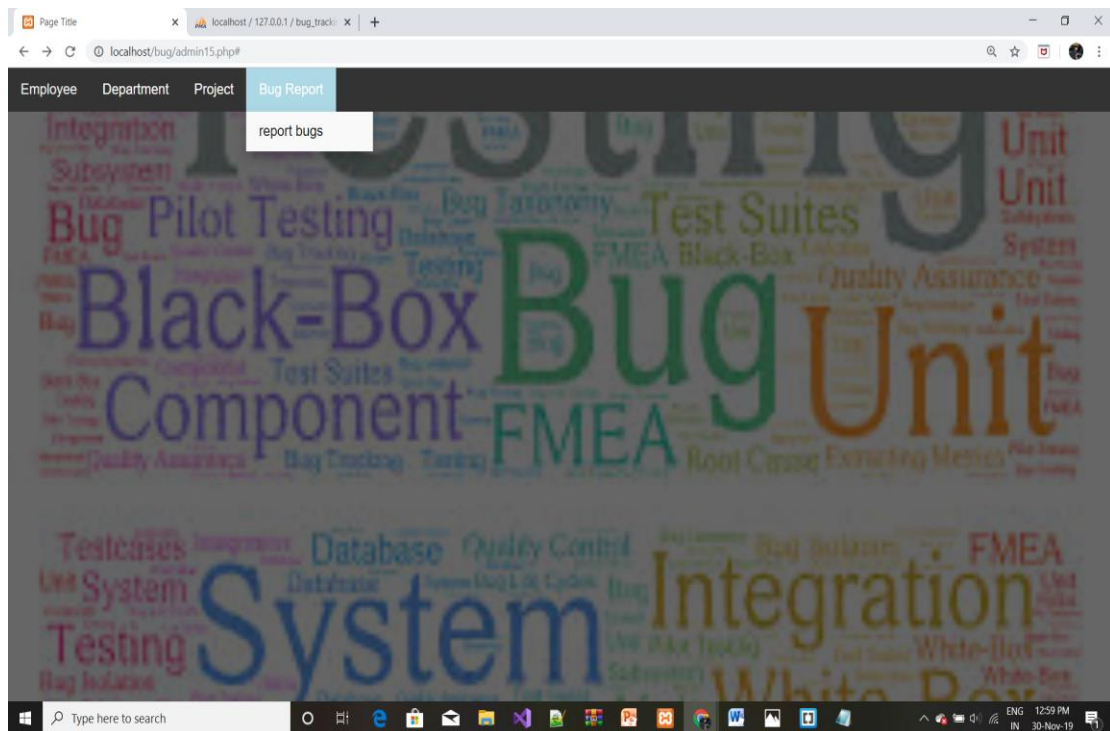
The above snapshot is a Register page which helps the client to register.



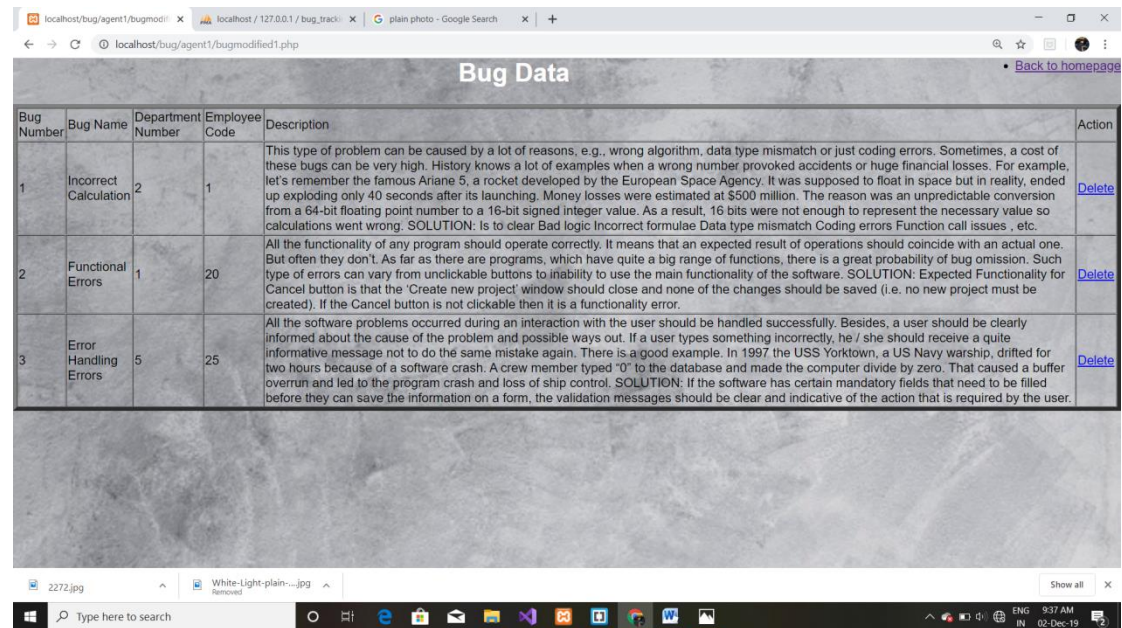
Snapshot 3: "Employee Registration"

The above snapshot is a Register page which helps the employee to register.





BUG TRACKING SYSTEM



Bug Number	Bug Name	Department Number	Employee Code	Description	Action
1	Incorrect Calculation	2	1	This type of problem can be caused by a lot of reasons, e.g., wrong algorithm, data type mismatch or just coding errors. Sometimes, a cost of these bugs can be very high. History knows a lot of examples when a wrong number provoked accidents or huge financial losses. For example, let's remember the famous Ariane 5, a rocket developed by the European Space Agency. It was supposed to float in space but in reality, ended up exploding only 40 seconds after its launching. Money losses were estimated at \$500 million. The reason was an unpredictable conversion from a 64-bit floating point number to a 16-bit signed integer value. As a result, 16 bits were not enough to represent the necessary value so calculations went wrong. SOLUTION: Is to clear Bad logic Incorrect formulae Data type mismatch Coding errors Function call issues , etc.	Delete
2	Functional Errors	1	20	All the functionality of any program should operate correctly. It means that an expected result of operations should coincide with an actual one. But often they don't. As far as there are programs, which have quite a big range of functions, there is a great probability of bug omission. Such type of errors can vary from unclickable buttons to inability to use the main functionality of the software. SOLUTION: Expected Functionality for Cancel button is that the 'Create new project' window should close and none of the changes should be saved (i.e. no new project must be created). If the Cancel button is not clickable then it is a functionality error.	Delete
3	Error Handling Errors	5	25	All the software problems occurred during an interaction with the user should be handled successfully. Besides, a user should be clearly informed about the cause of the problem and possible ways out. If a user types something incorrectly, he / she should receive a quite informative message not to do the same mistake again. There is a good example. In 1997 the USS Yorktown, a US Navy warship, drifted for two hours because of a software crash. A crew member typed "0" to the database and made the computer divide by zero. That caused a buffer overrun and led to the program crash and loss of ship control. SOLUTION: If the software has certain mandatory fields that need to be filled before they can save the information on a form, the validation messages should be clear and indicative of the action that is required by the user.	Delete

Snapshot: "Bug Report"

CONCLUSION AND FUTURE ENHANCEMENT

To conclude the description about the project: The project developed using Visual Studio Code, Brackets, and PhpMyAdmin based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

User comes to the search engine and makes a query, typically by giving key words, the engine looks up the index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text.

Most search engines employ methods to rank the results to provide the “best” results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another.

Search engine is technically the software and algorithms used to perform a search, the term have become synonymous with the website itself.

REFERENCES

- 1) Fundamentals of PHP by W3Schools
- 2) Fundamentals of HTTP and CSS by w3Schools
- 3) <https://en.wikipedia.org/wiki/projects>
- 4) <https://www.w3schools.com>