|  |  |  |
| --- | --- | --- |
| **Tech Saksham**  Final Project Report |  |  |

**“Hostel Records and Complaint Management System”**

**“RGUKT-SRIKAKULAM”**

|  |  |
| --- | --- |
| **ROLL NO** | **NAME** |
| S170497 | Pithani Satya Sri |
| S170505 | Nayana Lalitha |
| S170672 | Sk. Mehaboobunisa |
| S170567 | Sanaboina Prema Jyothi |

|  |  |
| --- | --- |
|  |  |
|  | Mr. S.S. Ahmed Ali |
|  | Master Trainer |

**ABSTRACT**

Many students had troubled about raising the complaints in the hostel . There are few drawbacks in keeping and maintaining a hostel. However, the Hostel records Complaint management system is Web based software to provide college students raise the problems in hostel more efficiently. Additionally, Students raise concerns in their rooms. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. It is headed by warden.

It is a software developed for managing hostel records. The software helps hostel warden in managing complaints of the hostel in an efficient manner. This project helps manage hostel rooms problems , hostel stocks and other things related to hostel problems.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table of Contents** | **Page No.** |
| 1 | Chapter 1: Introduction | 1-2 |
| 2 | Chapter 2: Services and Tools Required | 3 |
| 3 | Chapter 3: Project Architecture | 4 |
| 4 | Chapter 4: Architecture Blocks Detail Working | 5 |
| 5 | Conclusion | 6 |
| 6 | References | 7 |
| 7 | Code | 8-26 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**

This system is designed in favors of the rooms record management of our college which helps the authorities to save the records of the student about our room and other things. It helps us from the manual work from which it is very difficult to find the student problems. This system gives an idea about digitalizing in a better way.

The HRCMS will also contain special features like the student’s problems. Provides the searching facilities based on various factor. Such as room problems, hostel problems. It tracks all the information of allotted beds, fans, cots etc. Manage the information of allotted rooms and increase the efficiency of managing the rooms.

* 1. **Feature**
* Less human error and manual labor can be reduced
* High Security and avoid the corruption of the data
* Easy to handle the data and data updating
* Data redundancy can be avoided to some extent
* Backup data can be easily generated
* Data consistency
* Easy to retrieve and modify the data
* User friendly and flexible
  1. **Advantages**

Less human error and manual labor can be reduced, High Security and avoid the corruption of the data, Easy to handle the data and data updating, Data redundancy can be avoided to some extent, Backup data can be easily generated, Data consistency, Easy to retrieve and modify the data, User friendly and flexible**.**

* 1. **Scope**

The proposed project Rooms Record management is a computerized system. The use of this project in the hostel can reduce all the problems discussed above in the current manual hostel management system. Will make the monitoring of student moments and stock details easy. This project enable user to store all the data in a database, it is very easy to fetch the data from here.

* 1. **Future Work**

With a goal to provide the students complaint and problems occur in hostel. Web Application is also developed with using react JS and the other thing is it can be developed with basics of web development. To fulfil the above requirement, using Html, CSS and bootstrap along with php a Web Application can be designed. It is designed primarily for students, providing a simple ability to raise the complaint. The major design goal was to keep the system digital, while providing an (almost) effective system.

**CHAPTER 2**

**SERVICES AND TOOLS REQUIRED**

**2.1 Tools and Softwares used**

**Software Requirements:**

* HTML
* CSS
* Bootstrap
* JavaScript
* PHP My Admin
* MySQL
* Google chrome (other web browsers which supporting latest versions of above s/w)

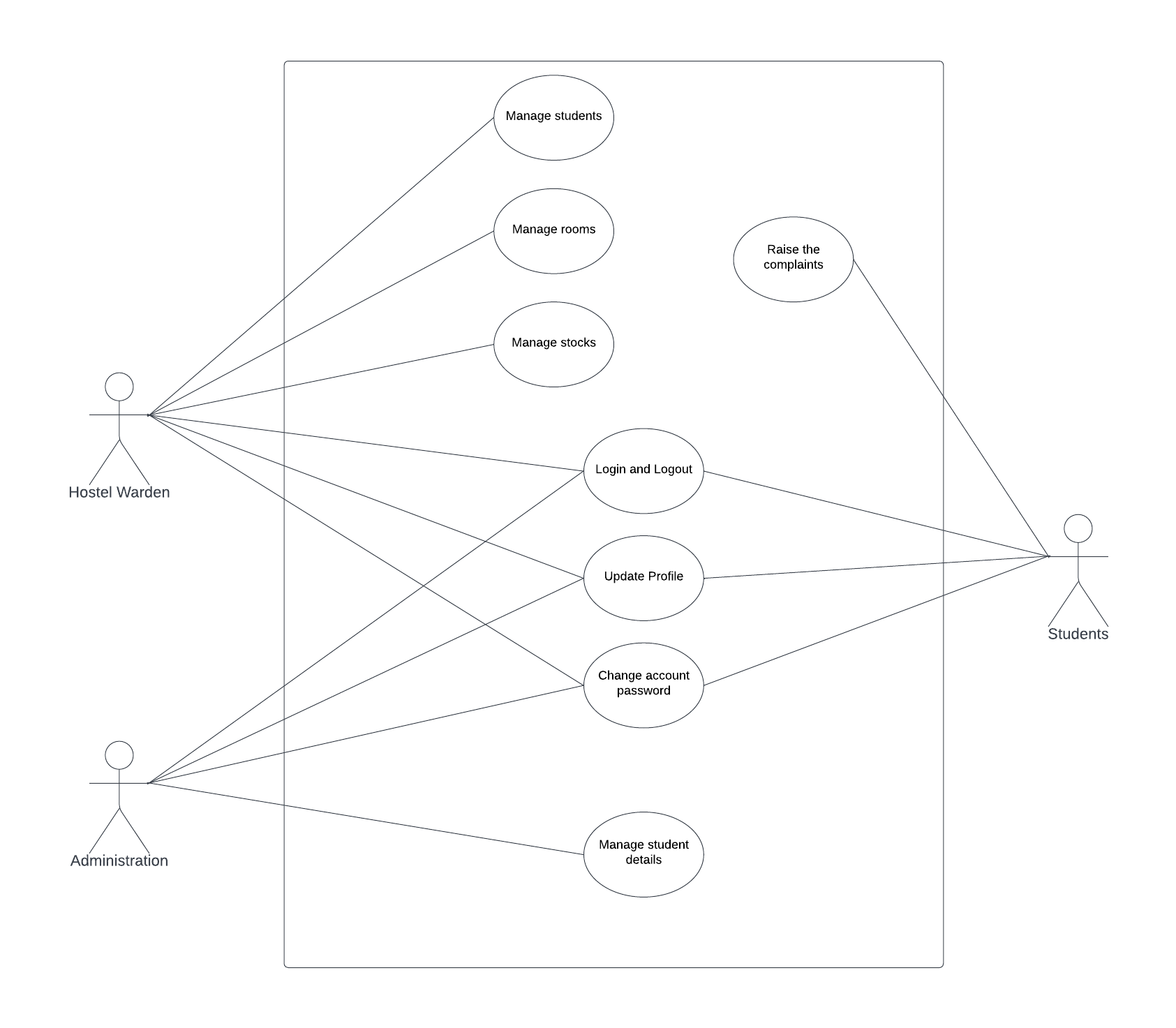
**Hardware Requirements:**

* RAM: 4GB above
* Hard disk: 500 GB above

**CHAPTER 3**

**PROJECT ARCHITECTURE**

**3.1 Architecture**



**CHAPTER 4**

**ARCHITECTURE BLOCKS DETAIL WORKING**

**4.1 Blocks**

Unified Modelling Language (UML) was created in 1995 by using merging diagramming conventions used by three application development methodologies OMT by James Rumbaugh, Objector y by Invar Jacobson and the Brooch procedure by using Grady Brooch. Previous to this time, these three amigos, together with a few dozen other practitioners had promoted competing methodologies for systematic program development, each and every with its possess system of diagramming conventions. The methodologies adopted a sort of cookbook sort of pushing a application task via a succession of life cycle stages, culminating with a delivered and documented software. One purpose of UML was once to slash the proliferation of diagramming techniques by way of standardizing on a original modelling language, as a result facilitating verbal exchange between builders. It performed that goal in 1997 when the (international) Object administration team(OMG) adopted it as a commonplace. Some critics don’t forget that UML is a bloated diagramming language written by means of a committee. That said, I do not forget it to be the nice manner to be had today for documenting object-oriented program progress. It has been and is fitting more and more utilized in industry and academia. Rational Rose is a pc Aided program Engineering (CASE) software developed by way of the Rational organization underneath the course of Brooch, Jacobson and Rumbaugh to support application progress using UML. Rational Rose is always complex due to its mission of wholly supporting UML. Furthermore, Rational Rose has countless language extensions to Ada, C++, VB, Java, J2EE, and many others. Rational Rose supports ahead and reverse engineering to and from these langue ages. However, Rational Rose does now not aid some usual design tactics as knowledge drift diagrams and CRC cards, due to the fact that these will not be a part of UML. Considering that Rational Rose has so many capabilities it isa daunting task to master it. Happily, loads can be executed making use of only asmall subset of these capabilities. These notes are designed to introduce beginner builders into making productive use of the sort of subset.

**CONCLUSION**

To conclude the description of the project is to very useful for hostel allotment. It deals with the problems on managing a hostel and avoids problems which occur when carried manually. Identification of the drawbacks of the existing system leads to designing of computerized system. It will be compatible to the existing system with the system which is more user friendly and works very efficiently.

**REFERENCES**

[**https://ieeexplore.ieee.org/document/9077688**](https://ieeexplore.ieee.org/document/9077688)

[**http://dspace.cusat.ac.in/xmlui/handle/123456789/1182**](http://dspace.cusat.ac.in/xmlui/handle/123456789/1182)

**CODE**

**6.1 HOME PAGE**

<!doctype html>

<html>

<head>

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

<style type="text/css">

.header1 {

color:#b3974b;

font-size: 16px;

font-family: Tahoma;

/\*font-weight: bold;\*/

text-align: center;

}

.header2 {font-size: 16px; color:#AA0000; font-family: Tahoma;text-align: center;}

</style>

<body style="background-color: white;">

<div class="main">

<img src="logo.jpg" alt="logo" align="left" class="title-logo" width="75">

<nav class="nav-menu d-none d-lg-block">

<label class="logo">RGUKT-AP</label>

<ul>

<li ><a class="active" href="index.php">Home</a></li>

<li><a class="active"href="admin/admin.php">Admin</a></li>

<li><a class="active" href="student/login.php">Student Login</a></li>

</ul>

</nav><!-- END NAV-->

</div><br>

&nbsp;&nbsp;

<div class="header2" style="font-size: 30px;">Hostel Records of Compliant Management System</div>

</div><hr color="#AA0000">

<div class="Scrolling">

<!-- <marquee onmouseover="stop()" onmouseout="start()" behavior="scroll" scrollamount="4" direction="right"color="greenyellow"> WELCOME TO HCMS

</marquee> -->

</div>

<section></section>

<footer id="main-footer">

<p>COPY RIGHT &copy; 2022 #CSE STUDENTS</p>

</footer>

</body>

**6.2 LOGIN PAGE**

<?php include('server.php') ?>

<!DOCTYPE html>

<html>

<head>

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

</head>

<body background-color:black;>

<div class="header">

<h2>Login</h2>

</div>

<form method="post" action="">

<?php include('error.php'); ?>

<div class="input-group">

<label>Student Id</label>

<input type="text" name="Student\_Id">

</div>

<div class="input-group">

<label>Password</label>

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

</div>

<div class="input-group">

<button type="submit" class="btn" name="login\_user">Login</button>

</div>

<p>

Not yet a member? <a href="register.php">Sign up</a>

</p>

</form>

</body>

</html>

**6.3 REGISTER PAGE**

<?php include('server.php') ?>

<!DOCTYPE html>

<html>

<head>

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

</hea>

<body>

<div class="header">

<h2>Register</h2>

</div>

<form method="post" action="register.php">

<?php include('errors.php'); ?>

<div class="input-group">

<label>Student Id</label>

<input type="text" name="Student\_Id" value="<?php echo $Student\_Id;?>">

</div>

<div class="input-group">

<label>Username</label>

<input type="text" name="username" value="<?php echo $username;?>">

</div>

<div class="input-group">

<label>Email</label>

<input type="email" name="email" value="<?php echo $email;?>">

</div>

<div class="input-group">

<label for="gender\_type">Gender</label>

<select id="gender\_type" name="gender\_type">

<option value="male">MALE</option>

<option value="female">FEMALE</option>

</select>

</div>

</div>

<div class="input-group">

<label for="hostel\_type">HOSTEL</label>

<select id="hostel\_type" name="hostel\_type">

<option value="K4">K4</option>

<option value="I1">I1</option>

</select>

</div>

<div class="input-group">

<label>Room No.</label>

<input type="text" name="roomno">

</div>

<div class="input-group">

<label>Password</label>

<input type="password" name="password\_1">

</div>

<div class="input-group">

<label>Confirm password</label>

<input type="password" name="password\_2">

</div>

<div class="input-group">

<button type="submit" class="btn" name="reg\_user">Register</button>

</div>

<p>

Already a member? <a href="login.php">Sign in</a>

</p>

</form>

</body>

</html>

**6.4 COMPLAINTS PAGE**

<?php

session\_start();

// variable declaration

$Student\_Id = "";

$roomno = "";

$errors = array();

$\_SESSION['success'] = "";

// connect to database

$db1 = mysqli\_connect('localhost', 'root', '', 'minipro');

// REGISTER USER

if (isset($\_POST['sub\_user'])) {

// receive all input values from the form

$complaint\_date = mysqli\_real\_escape\_string($db1, $\_POST['complaint\_date']);

$Student\_Id = mysqli\_real\_escape\_string($db1, $\_POST['Student\_Id']);

$phoneno = mysqli\_real\_escape\_string($db1, $\_POST['phoneno']);

$roomno = mysqli\_real\_escape\_string($db1, $\_POST['roomno']);

$complaint\_type = mysqli\_real\_escape\_string($db1, $\_POST['complaint\_type']);

$description = mysqli\_real\_escape\_string($db1, $\_POST['description']);

$sql=mysqli\_query($db1, "select complaintid from complaints order by complaintid desc limit 1");

// form validation: ensure that the form is correctly filled

if (empty($Student\_Id)) { array\_push($errors, "Student\_Id is required"); }

if (empty($complaint\_type)) { array\_push($errors, "complaint\_type is required"); }

if (empty($roomno)) { array\_push($errors, "Email is required"); }

if (empty($complaint\_date)) { array\_push($errors, "date is required"); }

// register user if there are no errors in the form

if (count($errors) == 0) {

$query1 = "INSERT INTO complaints (complaint\_date, Student\_Id, phoneno, roomno, complaint\_type, description)

VALUES ('$complaint\_date','$Student\_Id','$phoneno', '$roomno', '$complaint\_type','$description')";

$sql=mysqli\_query($db1, "select complaintid from complaints order by complaintid desc limit 1");

$\_SESSION['success'] = "Your complaint is registered ";

echo '<script> alert("Your complaint has been successfully filled")</script>';

$result = mysqli\_query($db1, $query1);

//$rowcount=mysqli\_num\_rows($result);

}

else {

array\_push($errors, "Wrong input");

}

}

?>

**ADMIN PAGE**

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<link rel="stylesheet" type="text/css" href="css/comp.css">

</head>

<body>

<?php include("sidebar.php");?>

<div class="mo">

<div class="header">

<h2 >Student Complaints Page</h2>

</div>

<div>

<form method="post">

<label for="complaint\_date">Date of Complaint:</label>

<input type="date" id="complaint\_date" name="complaint\_date">

<div class="input-group">

<label>Student Id</label>

<input type="text" name="Student\_Id" value="<?php echo $Student\_Id; ?>">

</div>

<div class="input-group">

<label>Phone No.</label>

<input type="text" name="phoneno">

</div>

<div class="input-group">

<label>Room No.</label>

<input type="text" name="roomno">

</div>

<div class="input-group">

<label for="complaint\_type">Complaint Type</label>

<select id="complaint\_type" name="complaint\_type">

<option value="Electricity issue">Electricity issue</option>

<option value="Carpentry issue">Carpentry issue</option>

<option value="leakage issue">leakage issue</option>

<option value="housekeeping issue">Housekeeping issue</option>

<option value="Mess food issue">Mess food issue</option>

<option value="Other issue">Other issue</option>

</select>

</div>

<div class="input-group">

<label for="description">Problem Description</label>

<textarea id="description" name="description" placeholder="Write something.." rows="5" cols="43"></textarea>

</div>

<div class="input-group">

<button type="submit" class="btn" name="sub\_user">Submit</button>

</div>

<!-- <p> <a href="a.php?logout='1'" style="color: red;">logout</a> </p> -->

</form>

</div>

</div>

</body>

</html>

**6.5 STYLE CSS**

.main{

background-color:#AA0000 ;

height: 100px;

color: white;

padding-top: 0px;

padding-left: 10px;

font-size: 20px;

}

.Scrolloing{

color: greenyellow;

}

label.logo{

color: white;

font-size: 35px;

line-height: 80px;

padding: 0 100px;

font-weight: bold;

}

body{

background: url(cloud.png) no-repeat;

background-size: cover;

height: calc(100vh - 80px);

}

section{

background: url(hostell.jpg) no-repeat;

background-size: cover;

height: calc(100vh - 80px);

}

#main-footer{

background-color: #AA0000;

height: 50px;

text-align: center;

padding-top: 10px;

color: white;

}

form, .content {

width: 30%;

margin: 0px auto;

padding: 20px;

border: 1px solid #B0C4DE;

background: white;

border-radius: 0px 0px 10px 10px;

}

.header {

width: 30%;

margin: 50px auto 0px;

color: white;

background: #5F9EA0;

text-align: center;

border: 1px solid #B0C4DE;

border-bottom: none;

border-radius: 10px 10px 0px 0px;

padding: 20px;

}

.input-group {

margin: 10px 0px 10px 0px;

}

.input-group label {

display: block;

text-align: left;

margin: 3px;

}

.input-group input {

height: 30px;

width: 93%;

padding: 5px 10px;

font-size: 16px;

border-radius: 5px;

border: 1px solid gray;

}

.btn { padding: 10px; font-size: 15px; color: white; background: #5F9EA0;