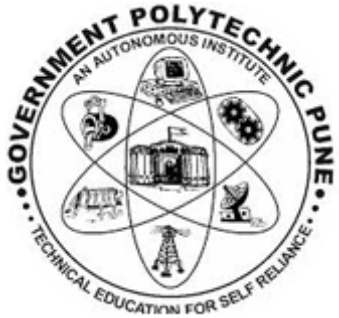
Government Polytechnic, Pune-16

(An Autonomous Institute of Government of Maharashtra)

Department of Information Technology

(Academic Year: 2022-2023)

EVEN – 2022



A

Project Report On

"Computing Infrastructure Complaint Redressal System"

SUBMITTED BY:

1) Piyusha Rajendra Supe	**2007062**
2) Purva Sunil Tonde	**2007065**
3) Anagha Ajay Rakshe	**2007053**

Under the Guidance of

Smt. Mrunal U. Kokate

Government Polytechnic, Pune

(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that

1) Anagha Ajay Rakshe	**2007053**
2) Piyusha Rajendra Supe	**2007062**
3) Purva Sunil Tonde	**2007065**

Has completed the necessary project work and prepared the bonafide on

"Computing Infrastructure Complaint Redressal System"

In a satisfactory manner as a partial fulfillment of requirement of the

THIRD YEAR DIPLOMA IN

INFORMATION TECHNOLOGY

FOR THE ACADEMIC YEAR

2022-2023

(H.O.D)	(Internal Guide)	(External Examiner)	(Principal)
Smt. Mrunal Kokate	Smt. Mrunal Kokate		Dr. Vitthal S. bandal

ACKNOWLEDGEMENT

As our proud privilege and honor with respect we acknowledge the help and guidance received from so many people during this phase of learning through making the project and after completion of it for the things we have learned. The learnings gathered and the increased ability to perform analyze and learn from all the material was very fulfilling. It would not have been possible to successfully complete our project and prepare the presentation and this report devoid of the precious guidance and help that we received from some of the most wonderful people that we were blessed to be with and guided by them.

First and foremost, we express a huge and sincere gratitude to our guide and respected mentor and reverent Head of Department of Information Technology **Smt. Mrunal Kokate** for guiding us throughout the process of learning and proliferating through the development of this project and in triumphantly presenting our work in this report. We are very grateful that we could look up to our mentor and get all our doubts solved. We feel overly fortunate and grateful to be guided by her. Hence, we express sincere gratitude to the respected Head of department of Information technology, Government Polytechnic Pune for support and precious teachings, guidelines, motivation and encouragement throughout the course of project development.

We also thank our parents for raising us the way we are and loving us, teaching us, for encouraging us to follow the right path and being with us throughout. All the valuable guidance by the most precious people we have ever met is gratefully acknowledged and expressed by us.

- **Piyusha Rajendra Supe (Enroll. No. 2007062)**

- **Purva Sunil Tonde (Enroll. No. 2007065)**

- **Anagha Ajay Rakshe (Enroll. No. 2007053)**

Table of contents

ABSTRACT

Complaint management is an essential component of any organisation and has a very important part in carrying out smooth processes and functioning of the organisation. Complaints are generated when the normal things are out of order or are functioning unorderly. A complaint management framework is a structure and set of guidelines to advise and guide an organisation on the best ways to manage complaints and ensure the consistent delivery of high-quality responses to user complaints that fall in line with all relevant regulatory compliance demands. A secure or in-house complaint management solution with centralised complaint management empowers our organisation to resolve user complaints in a timely and efficient manner. The user can view the complaint status through his/her authenticated login.

The Good support software provides comprehensive control over our handling of user queries, and the deployment of our Redressal for complaints. This ultimately works to increase user satisfaction — retaining users and giving a pleasing user experience to all organisation people. User complaints managed by a proper Redressal will give assurance to people of the organisation to work independently and smoothly. With a robust complaint management software solution, we can determine the root cause of computing infrastructure fails and deploy high technician teams that implement effective corrective actions and prevent the recurrence of issues that generate further user complaints. A computing infrastructure user complaint highlights a problem, whether that's a problem with usage, power outages, employees or internal processes of the computing infrastructures, and by hearing these problems directly from our users through our web portal, we can investigate and improve to prevent further complaints in the future.

Here we have designed a system as such where users of different categories under a particular department can launch their complaints and these complaints will be addressed and approved by the head of department. After which the head of department will forward the complaints to the maintenance admin who will further forward the complaints to the maintenance admin. The main task of maintenance admin is to forward the complaints to valid technicians so that they will cater to it and then respond about whether they have resolved it or not. The complaint loop stops when the complaint is resolved and approved by the user. Hence in this way the computing infrastructure complaints will be managed through this web portal.

List of figures

Chapter 1: Introduction and overview

1. Introduction

1.1 Purpose

The purpose of this project is to manage complaints and cater to them by forwarding them to respective authority and take notify technicians to take measure to resolve unorderly things of the computing infrastructure. The purpose of this document is to capture, in natural language and at a functional level, the description and requirements of a computing infrastructure complaint Redressal system. The focus here is the complaints managed for the computing equipment used at Government Polytechnic Pune. This is a functional description of those features required to address current requirements of the system. A short discussion accompanies each requirement, to add the background and framework necessary to explain the functionality. It describes functional requirements and other factors necessary to provide a complete and comprehensive description of the requirements for the software web portal.

1.2 Scope

This software system will be a web portal that allows users of different departments to launch complaints about the problems they are facing while working and using the computing infrastructure at the organization. The complaints will be addressed and will remain the main entity and focus of the complete system. The authorization and approval by the head of department and notifying required technicians will be the processes on a complaint entity.

1.3 Definitions, Acronyms and Abbreviations

1.3.1 Computing infrastructure

The computing infrastructure specifies the things that organization's users use such as routers, computers, cable lines for routers, Internet provision, power supply, switches are all a part of the computing infrastructure.

1.3.2 Administrator

The administrator is the main person who will assign head of department login's initial credentials. He can manage the head of department login's activated and deactivated statuses.

1.3.3 Maintenance administrator

The maintenance admin is the person who will receive the forwarded complaints by the head of department and he will be the one who will act as a connecting link between the complaints and the technicians of various categories.

1.3.4 System

System refers to the complete Complaint Redressal system.

1.3.5 Users

The users refer to particular users under a specific department they can be faculty members, lab assistants or other office support staff

1.3.6 Head of Departments

The head of various departments can add users under their own department and assign an initial login credentials to them

1.3.7 Technicians

The technicians are the people who are responsible for taking preventive and resolution measures for the unorderly things assigned through user complaints. They will be notified by the maintenance admins

1.4 Overview

This is a working document and, as such, is subject to change. In its initial form, it is incomplete by definition, and will require continuing refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of our "Computing Infrastructure and complaint Redressal system."

Thus we have introduced our system through this viable description of the system we wish to make as a web portal.

Chapter 2: Product Description and use cases

2. Overall Description

2.1 Product Perspective

"Computing Infrastructure and complaint Redressal system" is meant to serve as a platform where complaint management of computing infrastructure tasks can be carried out conveniently. Our goal is to develop a system to the academic portal used at Government polytechnic Pune that can have complaint management about computing needs making it more convenient for users who work for the departments. It is supposed to be a real time project that caters to complaint management as the main entity

2.1.1 System Interface

Apache will be used as web server through Xampp. The user inputs data via the web server using HTML forms. The actual program that will perform the operations is written in PHP.

2.1.2 User interface

The system shall provide a very intuitive and simple interface to the user and the administrator, so that the user can easily navigate through pages, launch complaints get approval for them and the administrator can easily manage HOD logins and revoke permissions.

2.1.3 Hardware Interface

a) Server side

The web application will be hosted on a web server which is listening on the web standard port, port 80.

b) Client side

Monitor screen – the software shall display information to the user via the monitor screen

Mouse – the software shall interact with the movement of the mouse and the mouse buttons. The mouse shall activate areas for data input, command buttons and select options from menus.

Keyboard – the software shall interact with the keystrokes of the keyboard. The keyboard will input data into the active area of the database.

2.1.4 Software Interface

a) Server side

An Apache web server will accept all requests from the client and forward it accordingly. A database will be hosted centrally using MySQL. The version of PHP used is as follows:

"*PHP 7.4.29 (cli) (built: Apr 12 2022 20:21:18)*
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies"

b) For mail Functionality

We have used the PHPmailer library which can be easily installed by downloading it from its

official portal. It uses the default SMTP ports in the apache web server that we have used to execute this project on the localhost for development time

c) Client side

An OS which is capable of running a modern web browser which supports JavaScript and HTML5

Here we have used the operating system that was available on our devices that is windows operating system Windows 11

2.1.5 Communication Interfaces

The HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

2.1.6 Memory Constraints

Memory constraints will come into play when the size of MySQL grows to a considerable size.

2.1.7 Operations

The product shall have operations to the database because it is a complete data flow oriented system

2.1.8 Site Adaption Requirements

Not applicable

2.2 Use case descriptions /Introductions

1] Administrator

The system shall provide the administrator ability to create and manage new head of departments or the departments altogether

2] Maintenance admin

The maintenance admin is a person that will act as a middle link between the complaints and the technicians. These complaints at this stage will be approved by head of department and that is why it will be with maintenance admin

3] User

User is the user under a specific department who will be eligible in all ways to be a part of the organization and launch a complaint

4] Technician

The technician is the person who will get the complaints launched after the medieval process and then he will resolve them and provide a status about it to the higher authority

5] Head of department

The HOD will be the person to approve the complaints and check if they are valid after the offline correspondence with his/her department’s users

Use case diagram for the complete system:

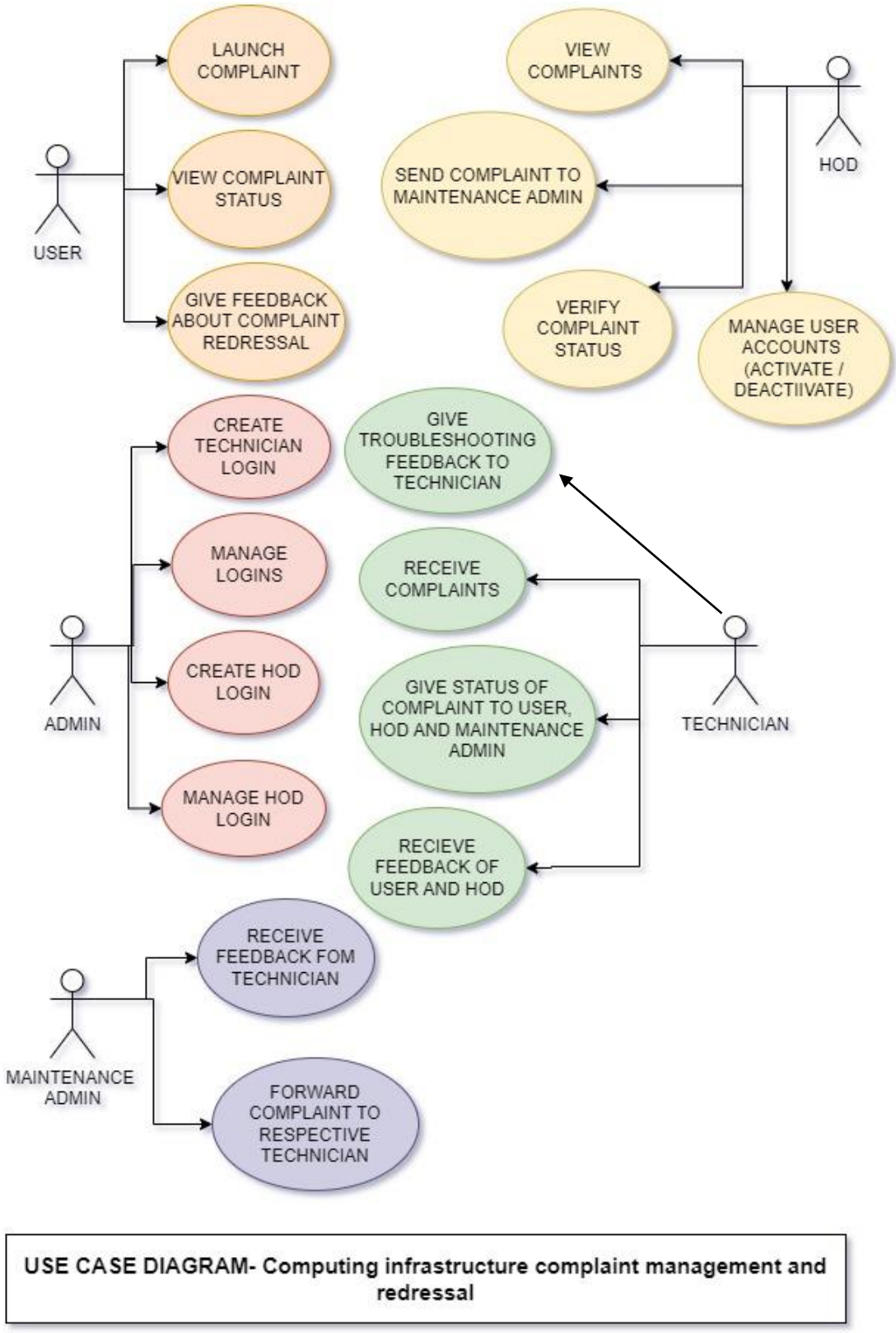


Figure 1 Use case diagram for the project

Chapter 3: Functionalities and sequence diagrams

3.1 These are the functionalities that the system shall integrate:

1] Complaint launching (role: user)

- The user will be able to launch complaint
- View the complaint status
- The user must be able to give the feedback whether the issue is resolved or not
- The user can launch complaints in various categories of complaints that we have identified and shall select it from the dropdown so that at a time only one complaint is managed and launched
- The user will be able to check status description where the explanation of the complaints current status is provided

2] Complaint approval (role: HOD)

- Once the complaint is launched the complaint given by the user is seen by the HOD
- The HOD can approve or disapprove the complaint
- Depending on which the complaint will be further sent to the maintenance admin

3] Technician can resolve the complaints (role: technician)

- The technician can see the complaints only when the maintenance admin has forwarded it
- The complaint will then be worked upon by the technician
- Once the technician feels he/she has resolved the complaint then he can provide the status by his side

4] The complaint forwarding (role: Maintenance admin)

- The maintenance admin is the link between the complaint and the technician so he will forward valid complaints to valid technicians
- The main job of the maintenance admin is to forward complaints

5] Assigning HOD login and technician Logins (role: Admin)

- The Head of departments cannot sign up but the administrator will provide them their login credentials
- The technician login will also be provided by the administrator only

6] Authorization

- The HOD, Technician will be assigned an initial login for which they can change password other than the default password for account safety
- The login credentials will be given to each user, technician, and HOD for the system to be more secure as it will remain on the public domain

8] Sending Email notifications to HOD, USER, TECHNICIAN

- When a complaint is approved by the HOD the user receives email notification regarding it
- When a user gives feedback of complaint resolution the technician receives the email of the user's feedback
- When the technician updates status from his side the user receives the email notification about the status updated by the technician and the status description that is given by the technician

9] Forgot password

- If a user forgets his/her password, then the forgot password link can be used
- The forgot password screen asks for the registered mail ID of user and the account role for which password needs to be reset
- Then user receives an OTP via mail
- The received OTP must then be entered in the reset password page that displays soon after the mail being sent
- The user can then just write the new password and confirm it and then the password will be successfully reset if it matches all the secure password criterion

3.2 Constraints

1] User Interface Constraints

Using this system is fairly simple and intuitive. A user familiar with basic browser navigation skills should be able to understand all functionality provided by the system.

2] Hardware Constraints

The system should work on most home desktop and laptop computers which support JavaScript and HTML5.

3] Software Constraints

The system will be intended to run on Firefox 4 and above, Google Chrome 10 and above and Internet Explorer 8 and above or any other similar configuration browser.

4] Data Management Constraints

System shall be able to interface with other components according to their specifications.

5] Operational Constraints

The system is limited by its operating server in terms of the maximum number of users it can support at a given time.

6] Device constraints

We are at our best trying to make system mobile responsive so that it is conveniently used by users on mobile phones as well as laptop browsers.

3.3 Design Standards Compliance

Specific Requirements for External interface

3.3.1 Web Server

- ✓ Apache will be used as web server
- ✓ The user inputs data via the web server using HTML forms
- ✓ The web server executes the PHP as a module and PHP script retrieves the post data if available.
- ✓ The web server receives information back from the PHP script.
- ✓ The web server displays a HTML page as result to the end-user.

3.3.2 PHP Application

The actual program that will perform the operations is written in PHP. All data will be stored in a database.

3.3.3 MySQL Database

It’s an open source SQL database to store all data which communicates with the

application on the server

3.3 Performance Requirements

The system should support at least 50 concurrent users. This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time

3.4 Logical database requirements

All data will be saved in the database: user accounts and complaints, login credentials

3.5 Design Constraints

1. The communication between the portal software and the database will be in SQL.
2. The portal layout will be produced with HTML/CSS.
3. The product will be written in PHP.
4. The output must be compatible with HTML5
5. The source code must follow the coding conventions of PHP.
6. System administrators must have access to comprehensive documentation.

3.6 Software System Attributes

The software consists of the following elements:

1. The apache web server
2. The PHP application
3. The MySQL database
4. The database should remain consistent at all times in case of an error.

3.6.1 Reliability

The reliability of the overall program depends on the reliability of the separate components.

3.6.2 Availability

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the MySQL server and saved by the administrator.

3.6.3 Maintainability

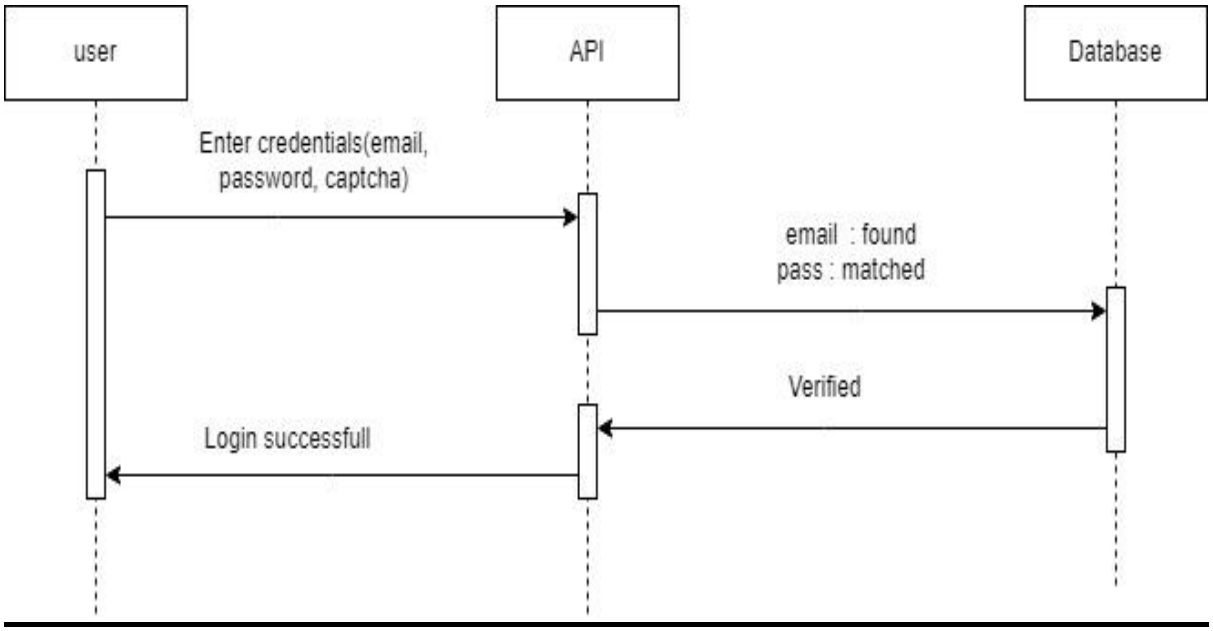
MySQL is used for maintaining the database and the Apache server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

3.6.4 Portability

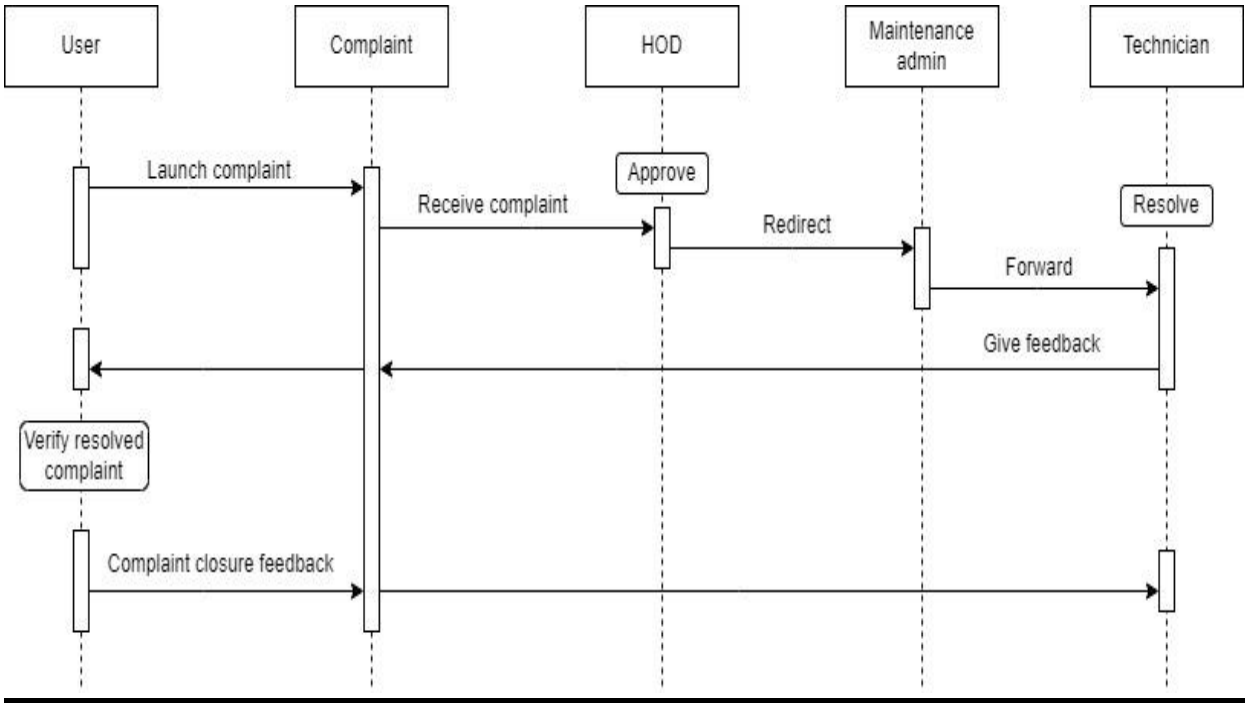
The application is web-based and should be compatible with other systems. Apache, PHP and MySQL programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application

Sequence diagrams

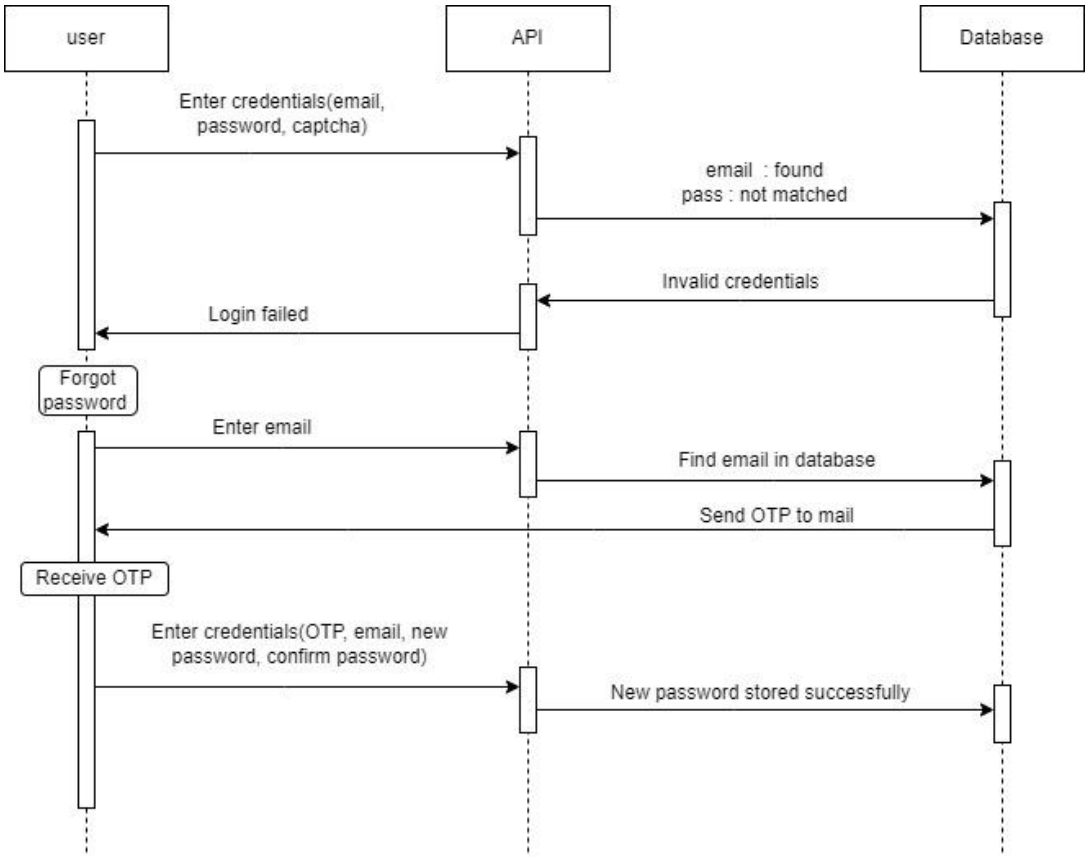
Login Functionality



Launch a complaint



Forgot password



Chapter 04: Cost Estimation and timeline estimation

The project was started with the beginning of the semester. The project began with planning and topic selection and thus proceeded with the diagrams. We are three members with roles of front-end developer, programmer and a tester. So assuming we have three employees and the project we are proceeding with has the duration of 2 months

Cost estimation

With criterion of COCOMO I model (For organic projects)

To complete the project in 2 to three months, the team would need to work at a higher rate of productivity. Assuming a productivity factor of 3.5 lines of code per person-hour, the estimated effort required for the project would be:

Effort = Size of Project / Productivity Factor = 5000 / 3.5 = 1429 person-hours = 3.57 person-months

If lines of code are taken (4.5 per person) this brings us to 5000/4.5 = 1111 person hours = 1.8 person months that means now we can complete this project in less than 2 months if lines of code are increased to 4.5 per person

This would require a team of at least three people working full-time for the duration of the project.

1. **Developer**: Assuming an hourly rate of INR 500 for a developer with minimal experience, and an estimated 160 hours of work over the 2-month period, the cost of the developer would be:

   INR 500/hour x 160 hours = INR 80,000

2. **Programmer:** Assuming an hourly rate of INR 400 for a programmer with minimal experience, and an estimated 160 hours of work over the 2-month period, the cost of the programmer would be:

   INR 400/hour x 160 hours = INR 64,000

3. **Tester**: Assuming an hourly rate of INR 300 for a tester with minimal experience, and an estimated 160 hours of work over the 2-month period, the cost of the tester would be:

   INR 300/hour x 160 hours = INR 48,000

Total Cost for team members = Developer cost + Programmer cost + Tester cost

= INR 80,000 + INR 64,000 + INR 48,000

= INR 1,92,000

The total cost with server and software cost if we host the website will be:

Total cost = Team member cost + Server cost + Software cost

In addition to the cost estimation for the team members, there are two other costs to consider: the server cost and the software required to develop the project.

1. **Server cost:** Assuming you are using a cloud hosting provider such as Amazon Web Services (AWS) or Microsoft Azure, the cost of the server will depend on the specifications of the server you choose.

Assuming a basic server with 1GB of RAM and 30GB of storage, and an estimated cost of INR 818 per month, the cost for 2 months would be:

INR 818/month x 2 months = INR 1634

2. **Software required to develop the project**: The cost of software required to develop the project will depend on the tools and technologies used. Assuming the use of free and open-source software tools such as Apache, the cost of software will be minimal. However, if any proprietary software is required, the cost will vary depending on the specific software and licensing terms.

Assuming a minimal cost of INR 4085 for software required to develop the project, the total cost for the project with three employees of different roles and a duration of 2 months with minimal experience would be:

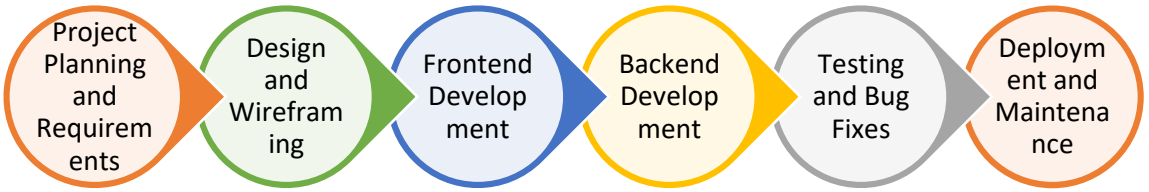
Team member cost + Server cost + Software cost

= INR 1,92,000 + 1634 + 4085

= INR 1,97,719

Therefore, the estimated total cost for the project with Indian salaries would be around INR 1,97,719. However, this is just an estimate, and the actual cost may vary based on specific requirements and other factors. In this distribution assumes that the team will be working full-time on the project for 8 weeks. The specific dates may be adjusted based on the team's availability and other factors. It is important to regularly review and adjust the timeline as needed to ensure successful completion of the project within the desired timeframe.

Timeline estimation:



•Start date: **2023-02-15**	•Start date: **2023-03-08**	•Start date: **2023-03-29**	•Start date: **2023-04-19**	•Start date: **2023-05-10**	•Start date: **2023-05-24**
•End date: 2023-03-07	•End date: 2023-03-28	•End date: 2023-04-18	•End date: 2023-05-09	•End date: 2023-05-23	•End date: 2023-06-06
•Duration: 1 week	•Duration: 1 week	•Duration: 2 weeks	•Duration: 2 weeks	•Duration: 1 weeks	•Duration: 1 weeks

Chapter 05: Flow diagrams DFD

The system is completely data flow oriented and thus the role of managing a database is the most important in this project. The project revolves around the most important entity of its existence that is the 'complaint'

The data flow diagram is as follows:

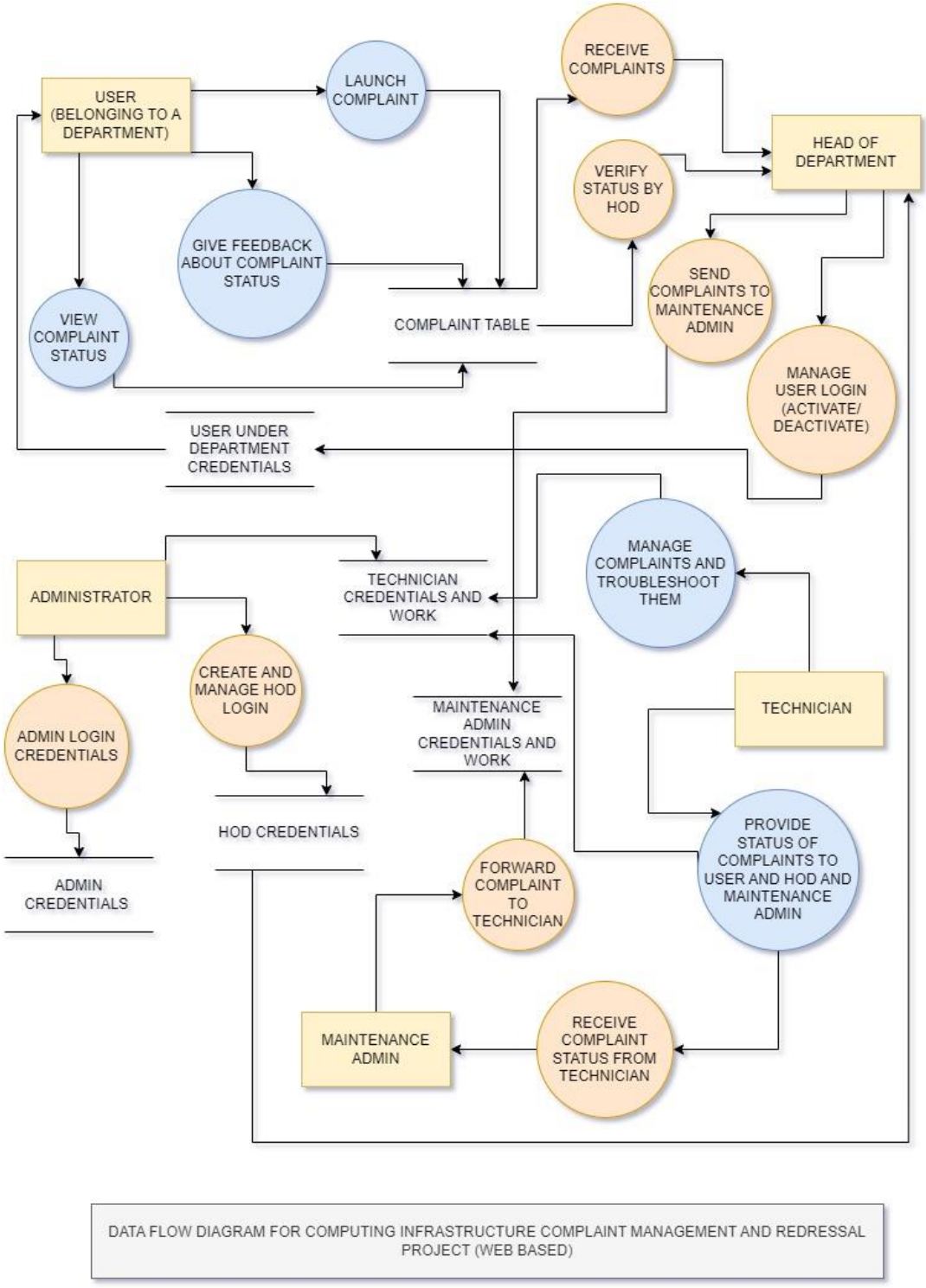


Figure 2 Data flow diagram for the system

Chapter 06: Data base entities and attributes ERD

The entity relationship diagram is as follows:

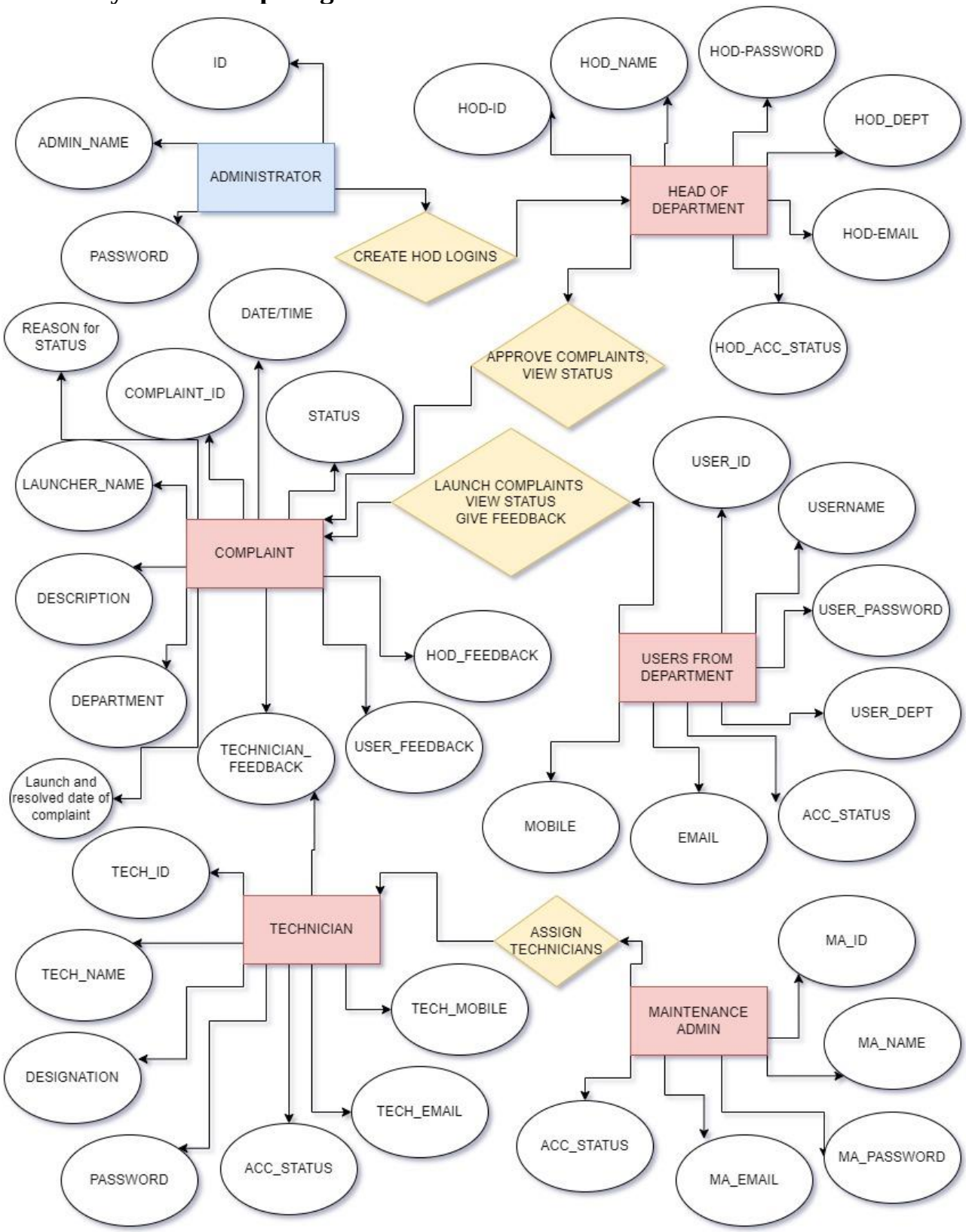


Figure 3 Entity relationship diagram

The data table is as follows:

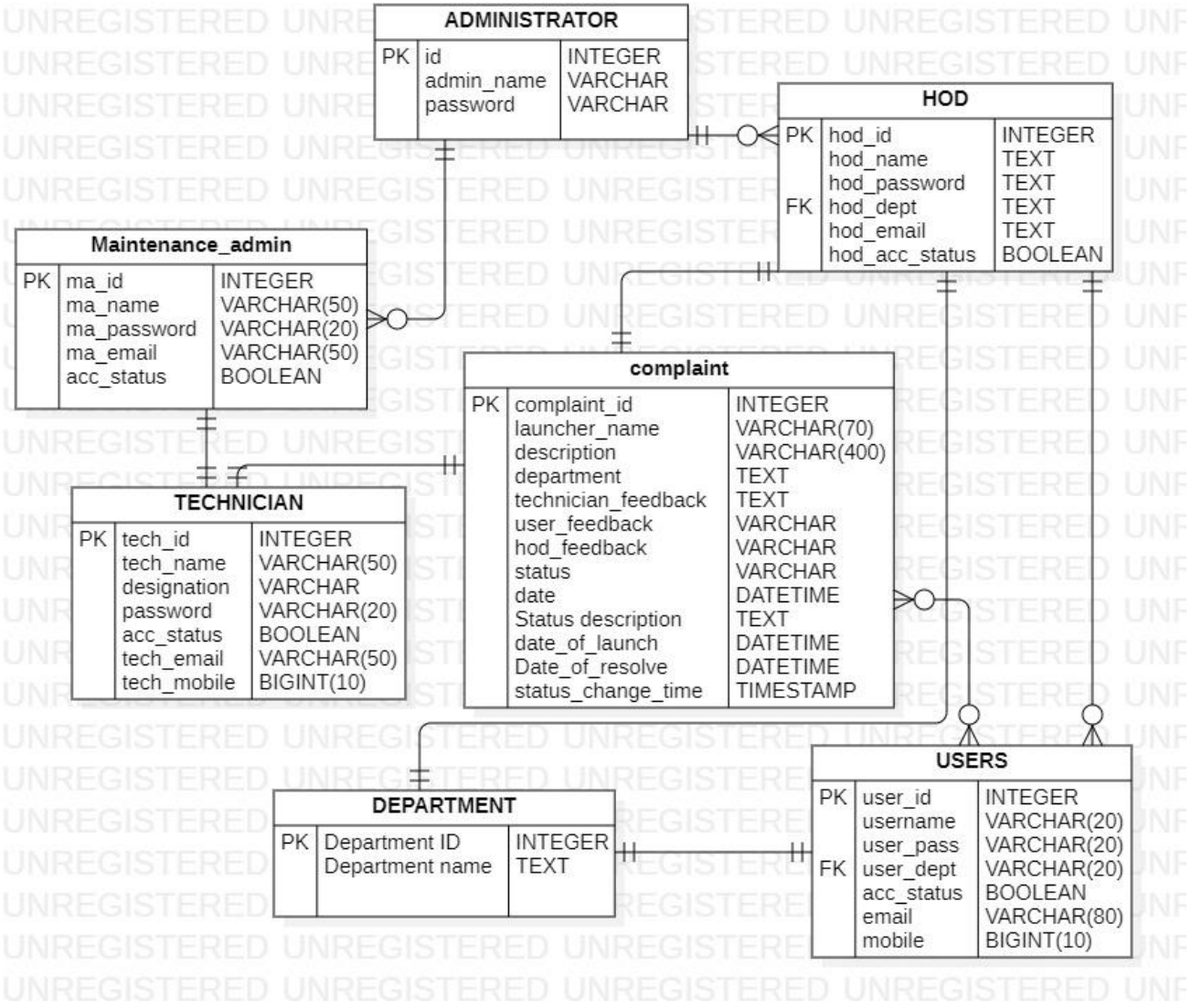


Figure 4 Data table diagram

The tabulated description of each entity is as follows:

Complaint table			
Sr. no	**Attribute**	**Description**	**Constraints**
1	Complaint id	The unique ID of each complaint ever generated	Whole numbers only
2	Launcher name/ID	The name or ID of the launcher who launches the complaint	Text
3	Description	The description of the complaint (it will be category of the complaint)	Text
4	Department ID	The ID of department from which the complaint is launched	Whole numbers only

5	Technician feedback	The revert back of technician that will be given by the technician when the complaint is resolved	Text
6	User feedback	The feedback of user which is given when the user feels that the complaint is now completely resolved	Text
7	HOD feedback	The feedback of HOD whether the complaint is approved or not	Text
8	Status	The status of complaint (pending, in progress, rejected)	Text
9	Date of launch	The date of launching complaint	Formatted date/time only
10	Status description	The current status and the reason why is the current status that way	Text
11	Date of resolved	The date of complaint resolution	Not a string
12	Status change time	The status changed time	Not a string

Department			
Sr. no	**Attribute**	**Description**	**Constraints**
1	Department ID	The unique ID of each department	Whole numbers only
2	Department name	The name of each department	Text

Administrator			
Sr. no	**Attribute**	**Description**	**Constraints**
1	id	The unique ID if there are more than one administrator	Whole numbers only
2	Admin_name	The admin user name for login	Text
3	Password	The admin password for login	Text

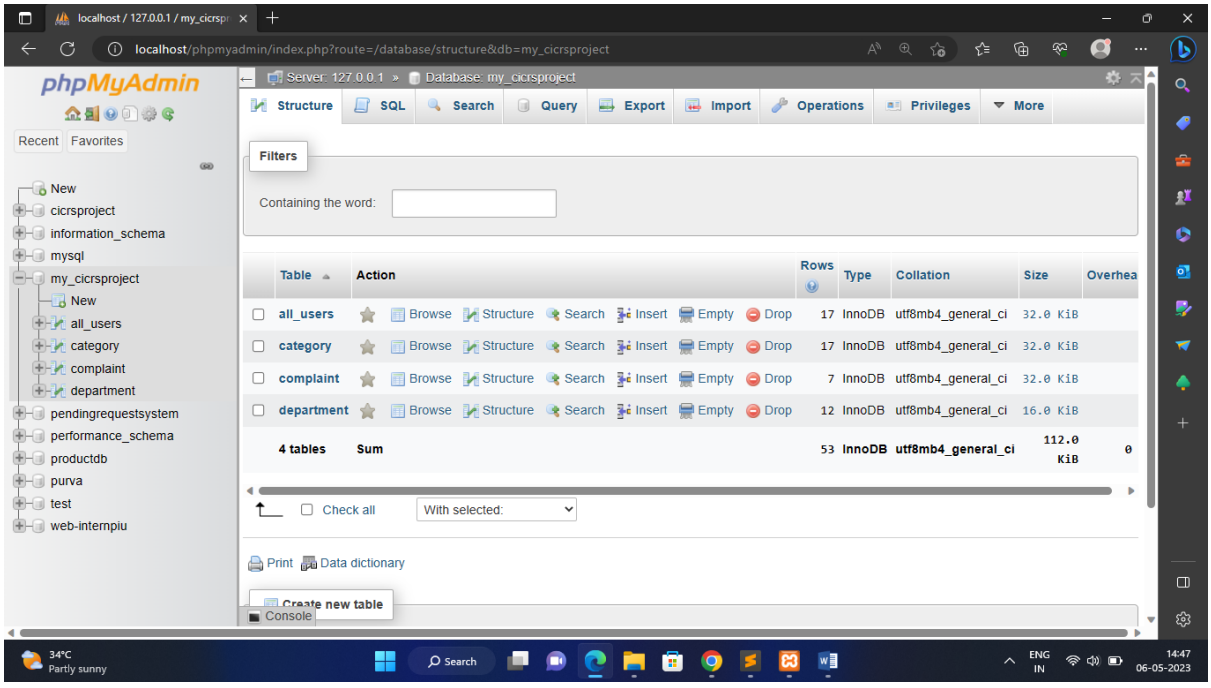
Head of department			
Sr. no	**Attribute**	**Description**	**Constraints**
1	Hod_id	The unique ID of each HOD	Whole numbers only
2	Hod_name	The name of HOD	Text
3	Hod_password	The password for login authorization	Text
4	Hod_dept(Foreign key)	The foreign key from department table	Text
5	Hod_email	The email ID of HOD	Text
6	Hod_acc_status	The account status of the current HOD account activated or deactivated	Bit values (0 or 1)

Technician			
Sr. no	**Attribute**	**Description**	**Constraints**
1	Tech_id	It is unique ID for the technician	Whole numbers only
2	Tech_name	It is name of the technician	Text
3	Designation	It is the work designation of the technician	Text
4	Password	It is the password of the technician's login	Text
5	Acc_status	The account status of the technician account	Bit values (0 or 1)
6	Tech_email	The technician Email for contacting him/her	Text
7	Tech_mobile	The mobile number for contact	Whole numbers only

Maintenance admin			
Sr. no	**Attribute**	**Description**	**Constraints**
1	ID	The unique ID of the maintenance admin	Whole numbers only
2	Name	It is the name for maintenance admin	Text
3	Password	The password required for login	Text
4	Email	The Email ID for contact	Text

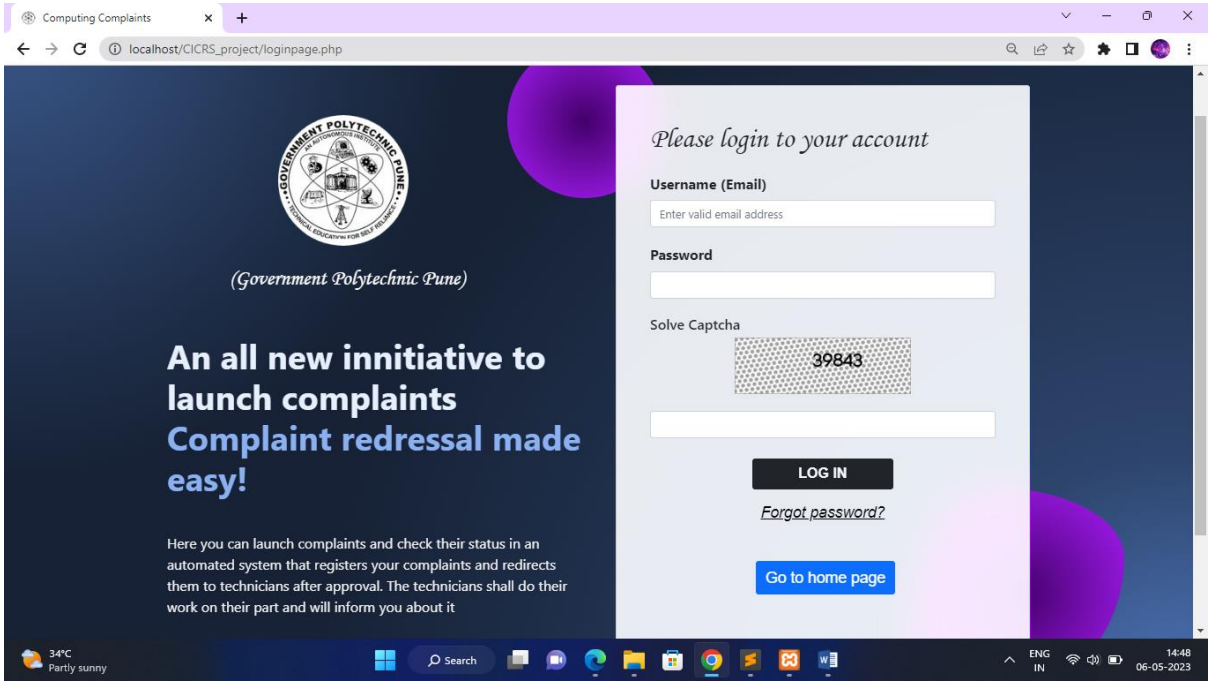
5	Acc_status	Account status of maintenance admin	Bit values only (0 or 1)

Users			
Sr. no	**Attribute**	**Description**	**Constraints**
1	User_id	The unique ID of the user	Whole numbers only
2	Username	The user's name	Text
3	User_password	The password for user	Text
4	User_dept	The department ID of the user (foreign key)	Text
5	Account_status	The account status of the users login	Bit values (0 or 1)
6	Email	It is the email ID of the user	Text
7	Mobile	It is the mobile number of the user	Whole numbers only

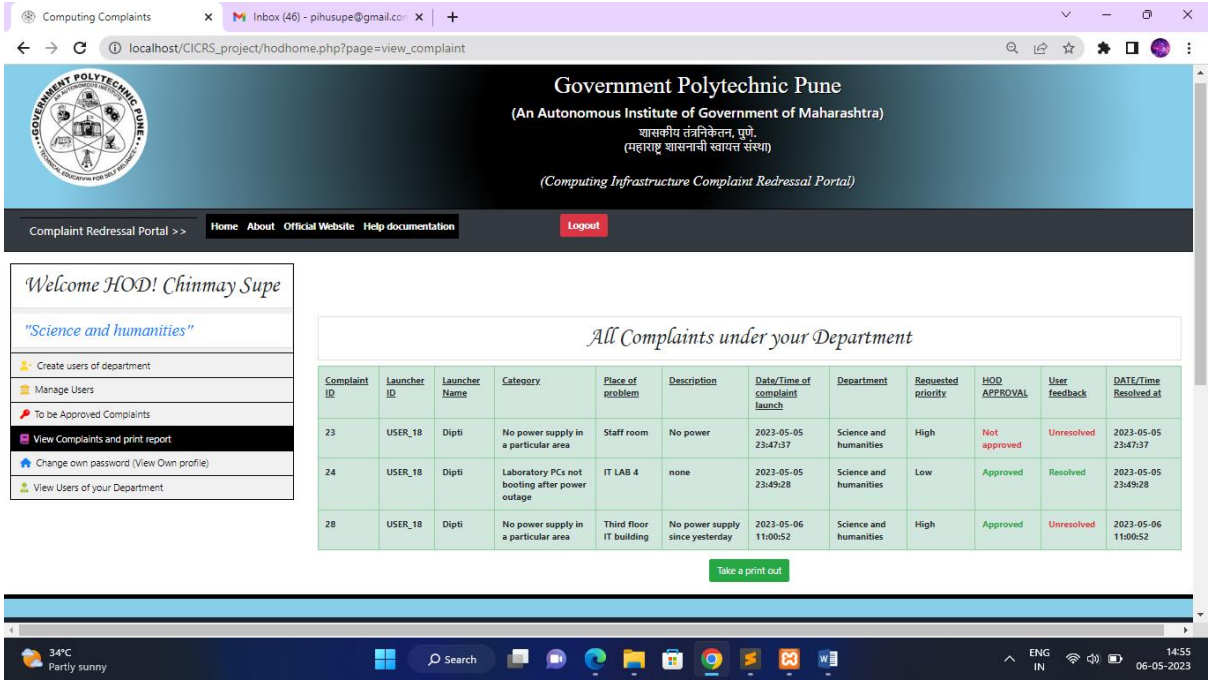


Chapter 07: User Interface Screenshots and Functioning

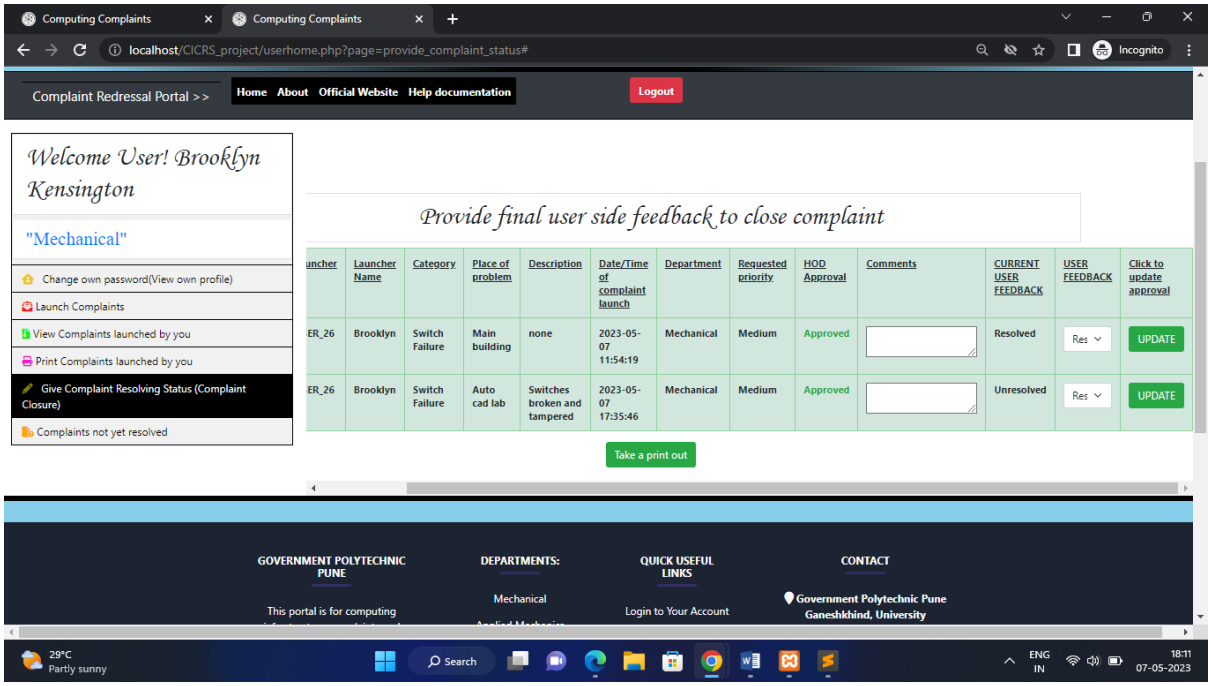
1] Login page



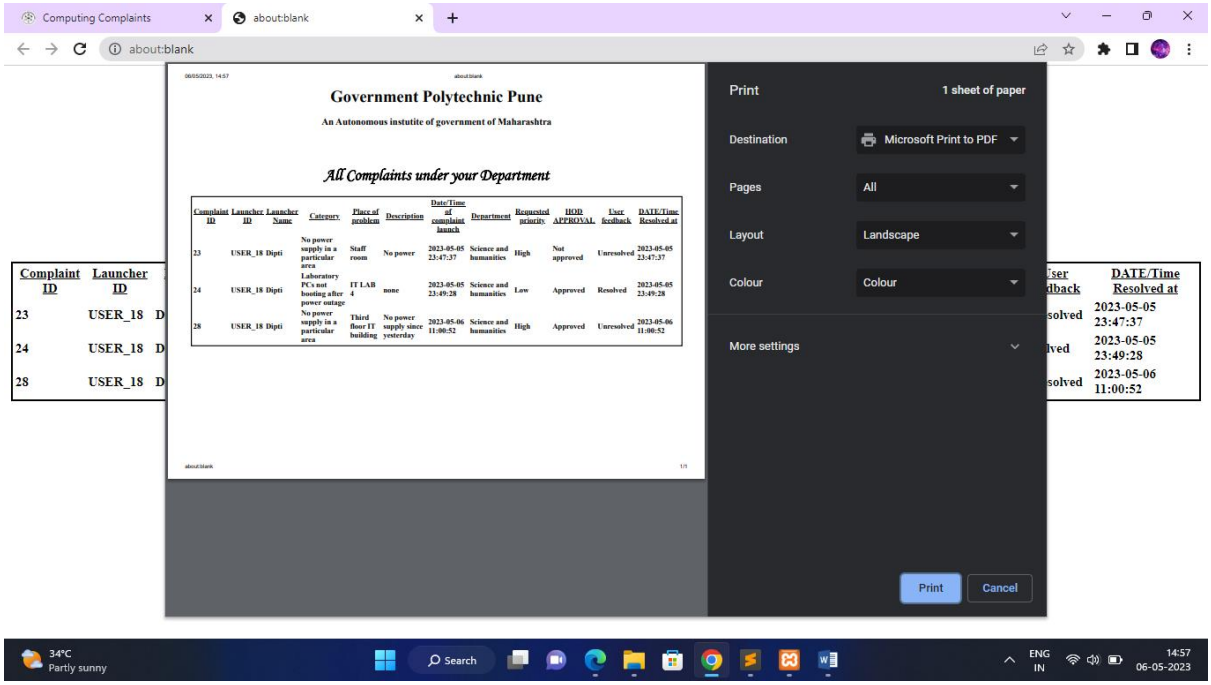
2] Head of Department Panel



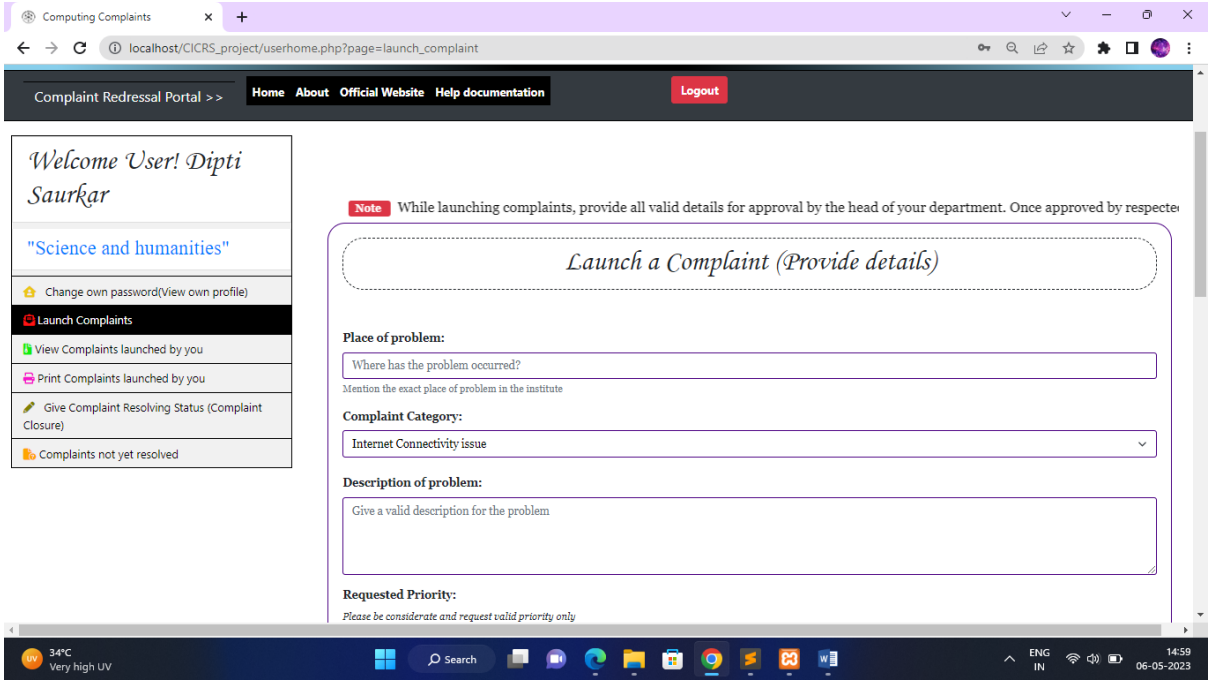
3] User panel



4] Report can be printed

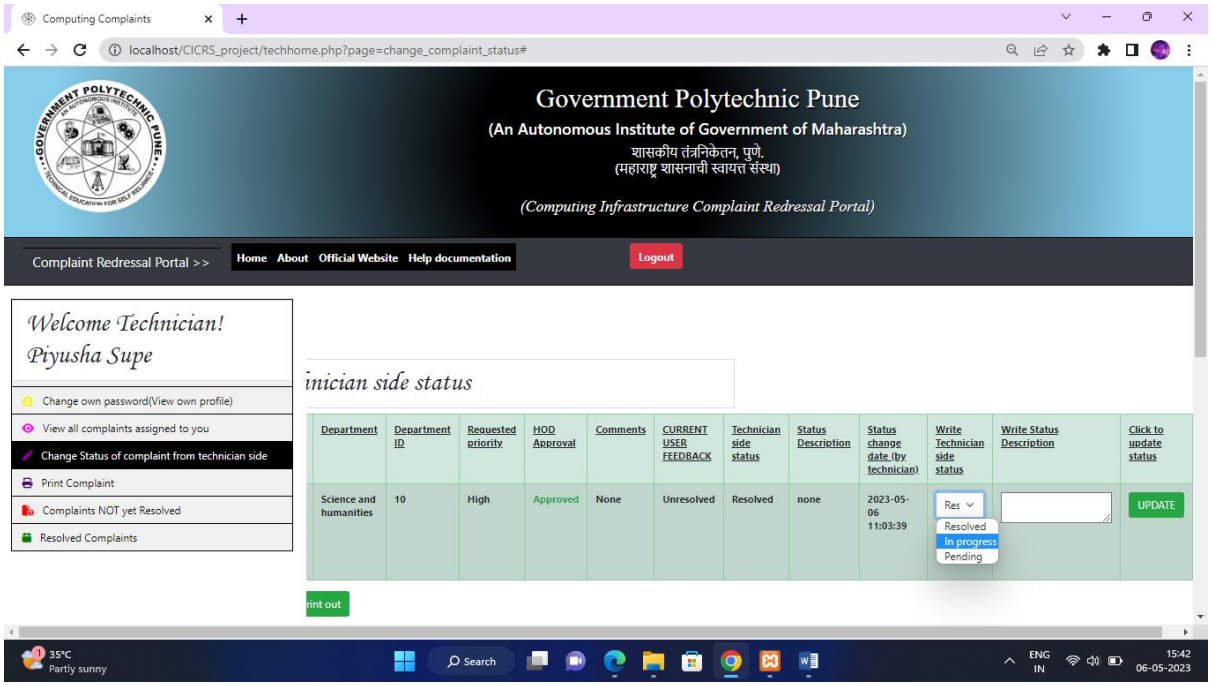


5] User panel (user under a particular department)

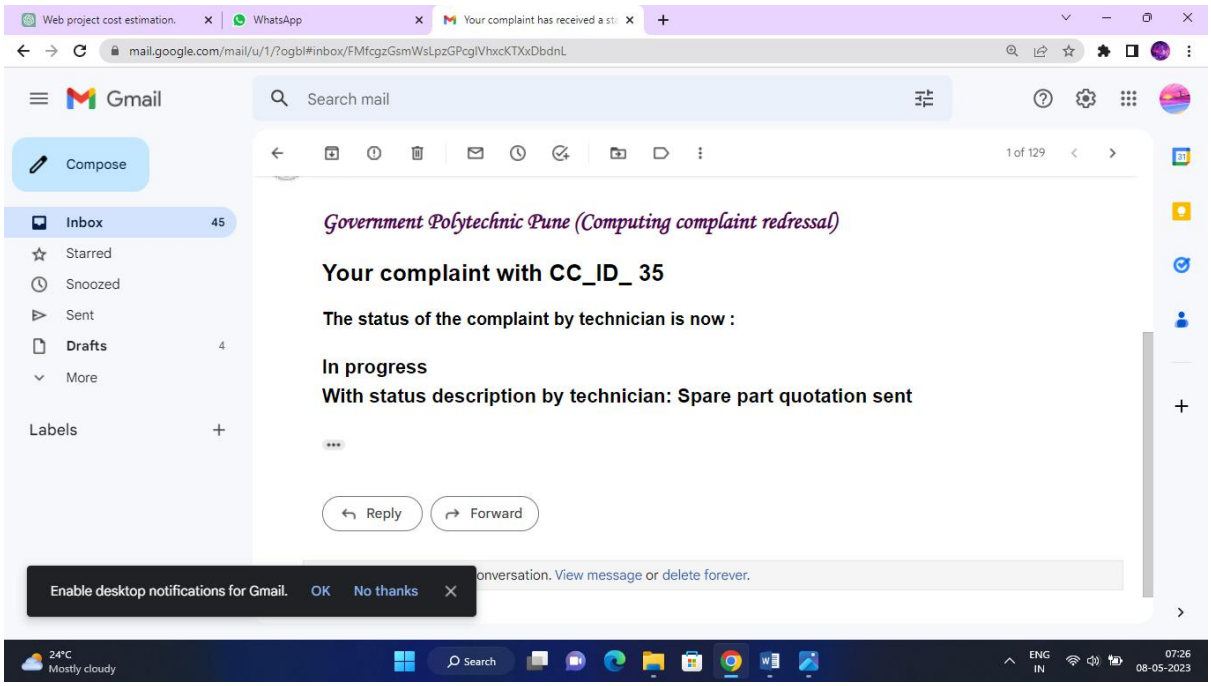


6] Maintenance Admin Panel

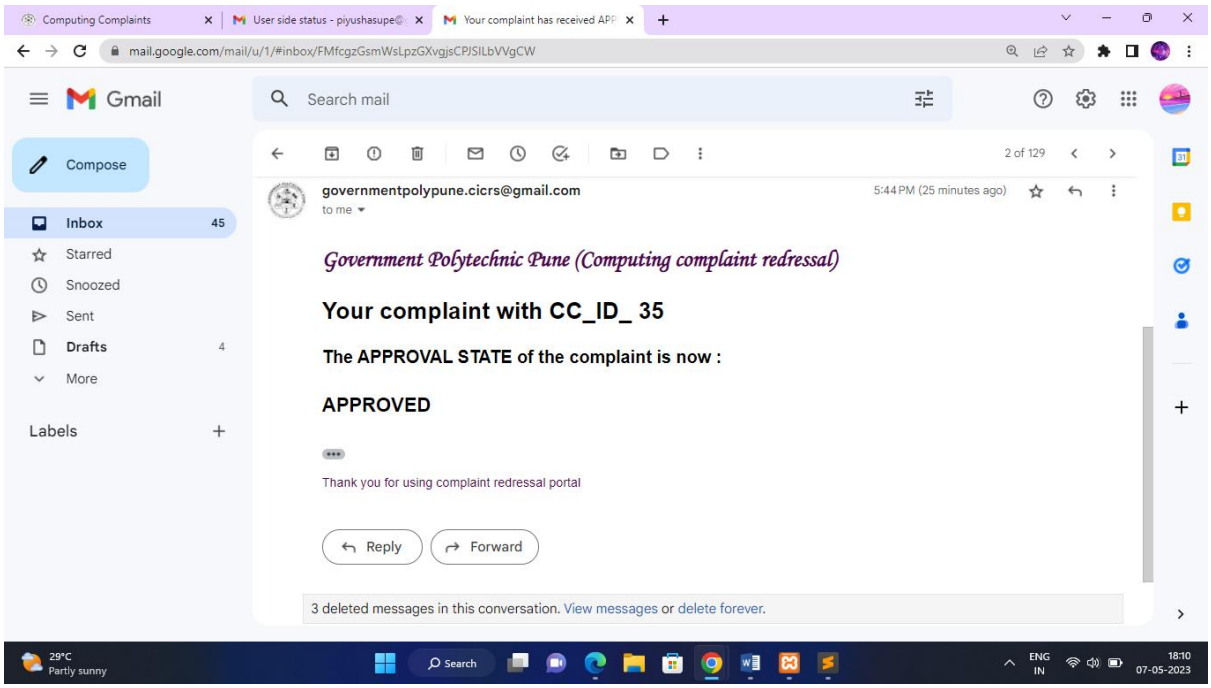
7] Technician



The email notifications will be sent to user on status change



When complaint is approved by HOD



Chapter 8: Significant code snippets

The important code snippets of the project include like the mail notifications being sent, forgot password OTP and the complaint insertion

1] Complaint insertion

```
<?php
if (isset($_POST['launch_complaint'])) {
   include'dbconnect.php';
   $launcher_name=$_SESSION['fname'];
   $uid = $_SESSION['user_id'];
   $category = $_POST['category'];
   $place_of_problem = $_POST['place_of_problem'];
   $description = $_POST['description'];
   $department = $_SESSION['dept'];
   $department_id = $_SESSION['department_id'];
   $launcher_email = $_SESSION['email'];
   $priority  = $_POST['priority'];
$insert_complaint= "insert into complaint (
launcher_name,
uid,
category,
place_of_problem,
description,
date_launch,
time_launch,
date_sort,
department_id,
department,
launcher_email,
priority)
values('$launcher_name','$uid','$category','$place_of_problem','$description',curdate(),now(),curdate(),'$department_id','$department','$launcher_em
ail','$priority')";
$result_launch_complaint  = mysqli_query($conn, $insert_complaint);
if($result_launch_complaint){
   echo "<script>alert('Complaint launched successfully!');window.location.replace('userhome.php');</script>";
}else{ echo "<script>alert('Could not launch complaint due to an error Please try again!');window.location.replace('userhome.php');</script>";}
}
```

2] Mail notifications code (used in forgot password module and mail notifications to various users)

```
if(isset($_POST['update_approval'])){
                    $user_feedback = $_POST['user_feedback'];
                    $cid = $_POST['cid'];
                    $comment = $_POST['comment'];
                    $how_resolved = $_POST['how_resolved'];

$sql = "UPDATE complaint SET date_resolve = curdate(), time_resolve = now(), status = '$user_feedback',how_resolved =
'$how_resolved' , status_desc = '$comment' WHERE cid = $cid";
                        if (mysqli_query($conn, $sql)) {
                            echo "<script> alert('Status given successfully!') </script>";
                            include 'dbconnect.php';
                         // Check if the email already exists in the database
                           $sql = "SELECT * FROM complaint WHERE cid = '$cid'";
                           $result = mysqli_query($conn, $sql);
                           $tomail = mysqli_fetch_assoc($result);
                           $email = $tomail['launcher_email'];

                           if (mysqli_num_rows($result) == 1) {

                             require 'phpmailer/src/Exception.php';
                             require 'phpmailer/src/PHPMailer.php';
                             require 'phpmailer/src/SMTP.php';
                             // require 'vendor/autoload.php';

                                  $message="<!DOCTYPE html>
                                                <html>
```

```
                                $message="<!DOCTYPE html>
                                            <html>
                                            <head>
                                                 <meta charset=\"UTF-8\">
                                                 <title>Notification about your complaint status</title>
                                            </head>
                                            <body>
                                                 <h1 style = \"font-family: monotype corsiva; font-size:24px;\">Government
Polytechnic Pune (Computing complaint redressal)</h1>
                                                 <h1 style=\"font-size: 24px; color: #000000;\">Your complaint with CC_ID_    ".
$cid."</h1>
                                                 <h2 style=\"font-size: 18px; color: #000000;\">The status of the complaint by
technician is now        :</h2>
                                                 <p style=\"font-size: 20px; color: #000000; font-weight: bold;\">".
$_POST['user_feedback']."<br>With status description by technician:  ".$comment."</p>
                                            </body>
                                            </html>"."Thank you for using complaint redressal portal. Please verify from your
login";

                                $mail = new PHPMailer(true);

                                $mail->isSMTP();
                                $mail->Host = 'smtp.gmail.com';
                                $mail->SMTPAuth = true;
                                $mail->Username = 'governmentpolypune.cicrs@gmail.com';
                                $mail->Password = 'ywfbddqgtgrnzlsm';
                                $mail->SMTPSecure = 'tls';
                                $mail->Port = 587;
```

```
                                        $mail->Password = 'ywfbddqgtgrnzlsm';
                                        $mail->SMTPSecure = 'tls';
                                        $mail->Port = 587;

                                        $mail->setFrom('governmentpolypune.cicrs@gmail.com');
                                        $mail->addAddress($email);

                                        // $mail->isHTML(true);
                                        $mail->Subject = "Your complaint has received a status from technician";
                                        $mail->MsgHTML($message);
                                           $mail->IsHTML(true);

                                        // $mail->send();

                                        if ($mail->Send()) {

                                       echo "<script> alert('Mail sent successfully to user');  </script>";
                                    }
                                    else{
                                         echo "<script> alert('Error while sending mail');  </script>";

                                    }
                                }

                            else{
                                 echo"<script>alert('Email does not exist in database');</script>";
                            }
          } else {
               // echo "Error updating status: " . mysqli_error($conn);
               echo "<script> alert('Cannot update status  of  complaint error occured ' </script>";
}
}
               ?>
```

3] Assigning technician to complaints and fetching complaint data

```
<?php
      function provider($cid,$launcher_id,$launcher_name,$category,$place_of_problem,$description,$date_launch,$department,$priority,$hod_approval,$user_feedback){

            include 'dbconnect.php';
      $query = "select * from all_users where role = 'technician' and account_status='1'";
      $res = mysqli_query($conn,$query);
      $row = mysqli_fetch_assoc($res);


                  echo "
                  <tr>
                        <td>$cid </td>
                        <td>USER_$launcher_id </td>
                        <td>$launcher_name </td>
                        <td>$category </td>
                        <td>$place_of_problem </td>
                        <td>$description </td>
                        <td>$date_launch</td>
                        <td>$department</td>
                        <td>$priority</td>
                        <td>$hod_approval</td>
                        <form method = \"post\" action = \"#\">
                        <td>$user_feedback</td>

                        <input type=\"hidden\" name = \"cid\" value=\"$cid\">

                        ";
                        echo "<td>";
                        //echo "<form method='post' action=''>";
                        echo "<input type='hidden' name='id' value='" . $cid . "'>";
                        echo "<select name='assign_tech_id'>";
                        //$tech_name = $row['fname'];
                        while ($row = mysqli_fetch_assoc($res)) {
```

```
                            echo '<option value="' . $row['uid'] . '">' . $row['fname'] . '</option>';
                        }
                        echo "</select>";
                        echo "</td>";

                        echo "<td><button type=\"submit\" name= \"update_approval\" class=\"btn btn-success\">UPDATE</button></td></form>

                  </tr>
                  ";
                  //echo $element;

            }
                  require 'dbconnect.php';
                  $depid = $_SESSION["department_id"];
                  $uid = $_SESSION['user_id'];
                $query = "SELECT * FROM complaint WHERE  hod_approval = 1";
                $result = mysqli_query($conn, $query);
                $row = mysqli_fetch_assoc($result);

                $_SESSION['department']=$row['department'];
                $depmt = $_SESSION['department'];

                  while($row = mysqli_fetch_assoc($result)){
                        if($row['hod_approval']=='1'){$acc_status = "<span class = \"text text-success\">Approved</span>";}
                        if($row['hod_approval']=='0'){$acc_status = "<span class = \"text text-danger\">Not approved</span>";}

                        provider($row['cid'],$row['uid'],$row['launcher_name'],$row['category'],$row['place_of_problem'],$row['description'],$row['time_launch'],
$row['department'],$row['priority'],$acc_status,$row['user_feedback']);
                  }
```

```
                  if(isset($_POST['update_approval'])){
                        //$user_feedback = $_POST['user_feedback'];
                        $cid = $_POST['cid'];
                        //$tech_name = $row['fname'];
                        $tech_id = $_POST['assign_tech_id'];


                        $fetch_tech = "select * from all_users where uid = '$tech_id'";
                        $fetch = mysqli_query($conn, $fetch_tech);
                        $assoc = mysqli_fetch_assoc($fetch);
                        $technician = $assoc['fname'];

                              $sql = "UPDATE complaint SET tech_id = '$tech_id' ,date_status_change = now(),status = 'Assigned to technician', technician =
 '$technician' WHERE cid = $cid";
                            if (mysqli_query($conn, $sql)) {
                                echo "<script> alert('Technician allotted successfully!') </script>";
                            } else {
                                // echo "Error updating status: " . mysqli_error($conn);
                                echo "<script> alert('Cannot allot technician error occured '. mysqli_error($conn)) </script>";
                            }
                  }
            ?>
```

Conclusion

In conclusion, the complaint management system has been successfully implemented in our institute, and it has proven to be an effective tool for resolving issues and improving communication between the admin, users, and head of department. The system has enabled us to streamline the complaint process, making it easier for users to submit their grievances, and for the admin to track and monitor the progress of complaints.

One of the most significant benefits of the system is the increased transparency and accountability that it provides. Users can now track the status of their complaints in real-time, and the admin can quickly identify any bottlenecks or issues that need to be addressed. Additionally, the system has helped to improve communication between the users and the head of department, enabling them to work together to find solutions to complex problems.

The system has also helped to improve the overall quality of service provided by the institute. By providing users with a simple and intuitive platform for submitting complaints, the system has made it easier for them to voice their concerns and provide feedback. This has enabled the institute to identify areas for improvement and make changes to its operations and services to better meet the needs of its users.

In conclusion, the complaint management system has been a valuable addition to our institute, providing a robust and effective platform for managing complaints and improving communication and collaboration between the admin, users, and head of department. The system has helped to improve transparency, accountability, efficiency, and overall service quality, resulting in higher user satisfaction and improved institutional performance. We are confident that the system will continue to be an essential tool in the years to come, enabling us to stay responsive and adaptive to the changing needs of our users and stakeholders.

References

1] References for mail functionality:

- https://github.com/PHPMailer/PHPMailer
- https://www.cloudways.com/blog/send-emails-in-php-using-phpmailer/
- https://www.hostinger.in/tutorials/send-emails-using-php-mail
- https://mailtrap.io/blog/phpmailer/
- https://www.a2hosting.com/kb/developer-corner/php/sending-e-mail-with-phpmailer/
- https://www.hostdime.com/kb/hd/mail/configuring-smtp-mail-settings

2] Document References for making UML diagrams:

- https://www.studocu.com/in/document/lovely-professional-university/software-engineering/online-complaint-management-systemdocx/39043164
- https://www.academia.edu/32923585/Online_complaint_Management_system

3] For studying captcha:

- https://www.studentstutorial.com/php/captcha
- https://www.techtarget.com/searchsecurity/definition/CAPTCHA#:~:text=Advantages%20of%20CAPTCHAs%20include%3A,or%20sign%2Dups%20for%20websites.
- https://www.researchgate.net/publication/336359736_Is_Image-based_CAPTCHA_Secure_Against_Attacks_Based_on_Machine_Learning_An_Experimental_Study
- https://www.baeldung.com/cs/captcha-intro

4] Installing GD library

- https://www.codexworld.com/how-to/install-php-gd-library-windows-server/
- https://www.dmxzone.com/go/5001/how-do-i-install-gd-in-windows/
- https://www.php.net/manual/en/image.installation.php
- http://www.webassist.com/tutorials/Enabling-the-GD-library-setting

5] Database – Fetching data from a database

- https://www.formget.com/read-mysql-data-using-php/
- https://www.stechies.com/error-mysql-shutdown-unexpectedly/