INTRODUCTION

An effective complaint procedure helps a college deal with complaints quickly, fairly and consistently and is an important part of an equal opportunity policy. Student's suggestions and complaints help the organisation better understand the requirements and necessity of students. It is important for the organisation to improve from the feed backs provided overtime to provide a better experience to student's wellbeing.

Solving of these problems must be quick and transparent. Often, Students shy away from writing complaints and suggestions into the box, as students are afraid of peer views of their action.

Problem Statement:

A Web Application to allow students to raise complaints and suggestions transparently without exposing their identity.

The Objectives of this application are:

- Allows students to login with E-mail verification before submission of complaints and suggestions
- Allows students to track their submissions from their portal
- Allows teachers to reply and change status of the submissions without exposing student's USN
- Provides a pie chart representing the types of submissions
- Admin can block submissions using foul languages

Back End Design

2.1 Database Design

ER Diagram

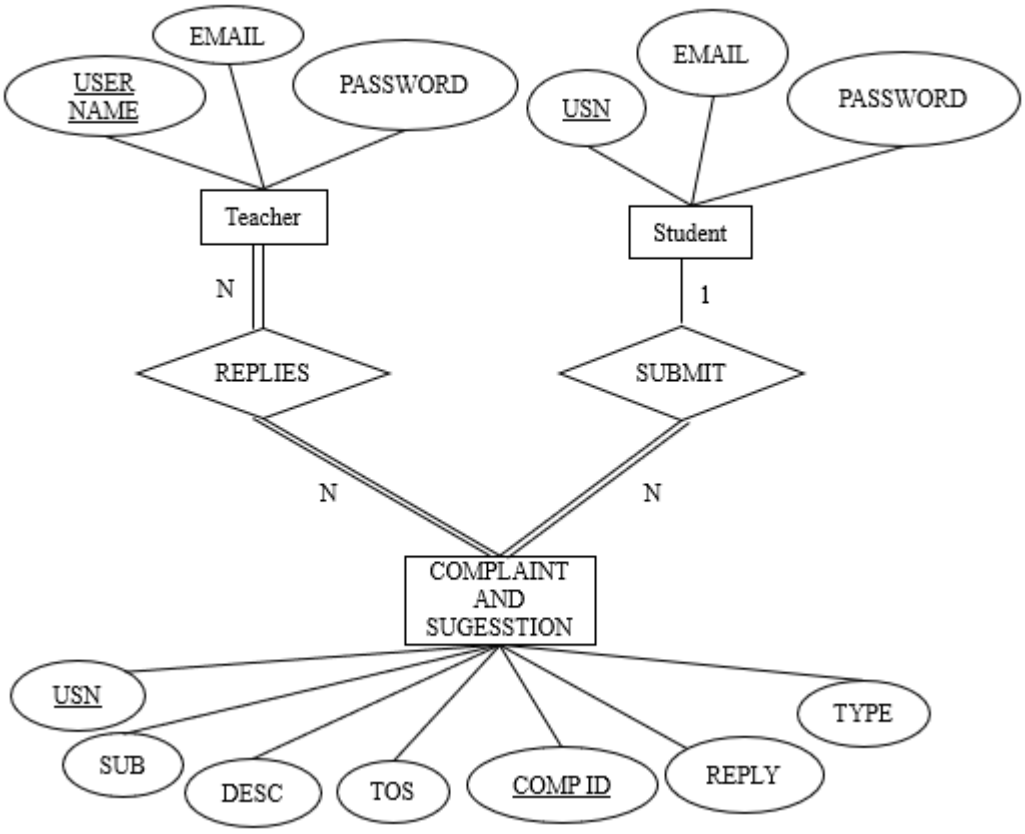


Figure 2.1: ER Diagram

MySQL

MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle Corporation.

PHP with MySQL

Connection to MySQL database is done by mysqli_connect function as shown below:

```
// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
```

```
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
```

To fetch data from database for further processing:

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
      echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"].
"<br>";
   }
} else {
   echo "0 results";
}
// Further Processing

mysqli_close($conn);
```

2.2 PHP for data processing

PHP (Hypertext Pre-processor) is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. It is widely used today as a general-purpose scripting language, particularly useful for web developments and like other scripting language, it can also be embedded directly into the HTML code. Within a short span of time, PHP has gained remarkable popularity and its community is increasing rapidly.

PHP can be used on all major operating systems, including Linux, many Unix variants, also has support for most of the web servers today. This includes Apache, IIS, and many others. Does freedom of choosing an operating system and a web server.

With PHP you are not limited to output HTML. PHP's abilities includes outputting images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. You can also output easily any text, such as XHTML and any other XML file.

Compiling reason to use PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple using one of the database specific extensions (e.g., for mysql), or using an abstraction layer like PDO, or connect to any database supporting the Open Database Connection standard via the ODBC extension.

2.3 JavaScript for Client-Side Processing

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi- paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web.

JavaScript runs on the browser thereby decreasing the load on the server. It is used for making dynamic and interactive webpages. JavaScript allows us to script the HTML content and CSS presentation of documents in web browsers, it also allows us to define behaviour for those documents with event handlers.

Front End Design

3.1 Design for Web Pages

Web design is the process of creating a uniform layout and appearance among all the web pages of the application through right choice of colour scheme, page layout, fonts. Every single web page in a website has different content, but all the pages are using a similar graphic design.

This is usually achieved by using a standard template containing all the basic elements of web design such as CSS style, buttons, backgrounds, borders, etc and modifying it to requirements of our web page. It usually preferable to use the same design language through the web application to enhance the quality of experience and usability.

HTML

HyperText Markup Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web (www). Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation 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 content.

JAVASCRIPT

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. In the implementation of our project, we have made use of jQuery, a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, and animation.

3.2 Framework

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

Bootstrap uses grid system which is built with flexbox allowing html design to broken down to rows and columns. The number of rows in the web page can be infinite. But only up to 12 columns are allowed each row. Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size.

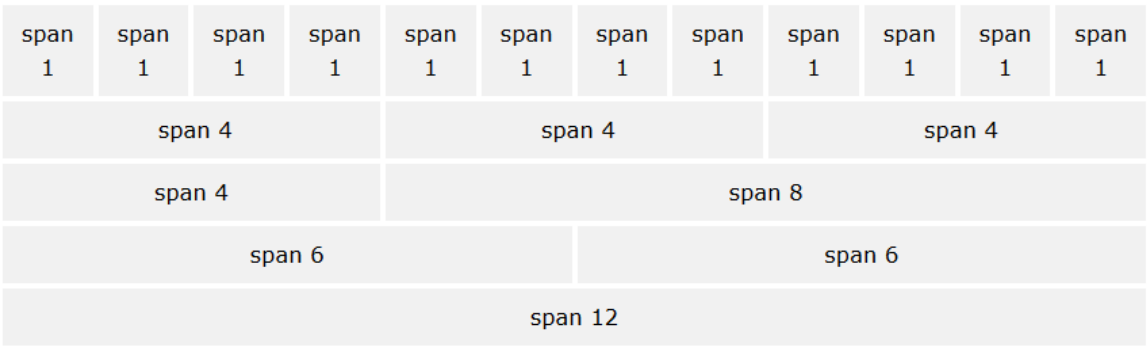


Figure 3.1: Bootstrap Grid

It also contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components to ensure consistency across web pages.
This framework can be added by including bootstraps compiled CSS in html file using:
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

3.3 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer.

3.4 Design

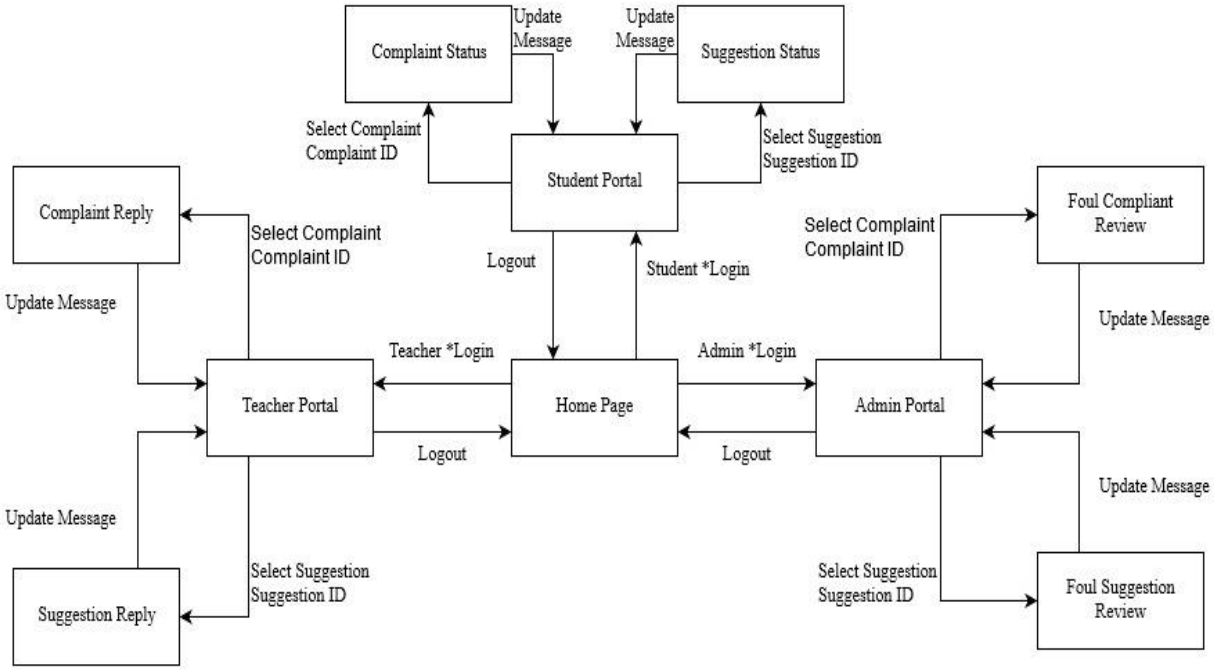


Figure 3.2: Navigation

Steps Involved:

Step 1: The web application loads the home page. With options of: Student, Teacher, and Admin

Step 2: On selecting Student, the user must first login using login page. On successful login, the user is greeted with student portal.

Step 3: Student portal allows user to check for the status of the previous complaints, and suggestions.

Step 4: Student can click to submit complaint and suggestion, after verification page using email verification.

Step 5: Student can click on logout option and is then taken to login page, which has option to return to home page

Step 6: On clicking teacher option in home page, teacher has to login. They are then taken to teacher home page. Teacher can choose to view complaints and suggestions.

Step 7: On clicking on a complaint/suggestion is taken to action page. Where, teacher can change the status of the complaint with option to reply.

Step 8: After replying, the teacher is taken back to teacher home portal. On logging out you are taken to login page, with option to be redirected to home page

Step 9: On clicking Admin option in home page, admin has to login. They are then taken to Admin home page. Admin can choose to view complaints and suggestions which are blocked due to use of foul language. After allowing or disallowing the admin is returned back to Admin home portal. On logging out you are taken to login page, with option to be redirected to home page

Functionalities

Login: This module allows Students, teacher and Admin to log in different access to web pages and controls. The username and hash of password is compared with the Tables stored in Database.

Signup: This module allows Students, teacher and Admin to register their access to the web application. By signing up the user is registered in the Database, so that their login credential is saved for future logins.

Complaint and Suggestion Form: This module allows students to fill up complaints or suggestions. Before uploading to Database. It checks for any foul language violation. It marks the entry to be checked by admin, before allowing it to be viewed by teacher.

Student Submission view: This module allows students to view the status and reply of their complaints or suggestions previous raised. On clicking the user can re-open a submission.

Verification: Student have to enter the One-time password (OTP) sent to their email to verify the authencity of the complaint and suggestions. This module is implemented using PHPMailer Library.

Teacher submission view: This module allows concerned teachers to view the complaints and suggestion without revealing student's USN. On selecting a submission, teacher can update status, reply of the same, and notify the same to student's email.

Admin unblock submission: Display submissions flagged due to misuse of the platform. Allows admin to unblock the submission if it is appropriate, and updates the same in Database.

Implementation

Language used: PHP, JavaScript, HTML, CSS

Login:

```
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
<title>Login Teacher</title>
<link rel="icon" href="stylesheets/bitlogo.png" type="image/x-icon">
<link rel="stylesheet" href="./stylesheets/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body onload="script:color_check();">
<div class="container">
<div class="row">
<div class="col-md-6 fill">
<div class="header">
        <h2>Login</h2>
        <h4>Teacher</h4>
</div>
<form method="post" action="login.php">
<?php include('errors.php'); ?>
<div class="input-group">
        <label>Username</label>
        <input type="text" name="username" >
</div>
<div class="input-group">
        <label>Password</label>
        <input type="password" name="password">
</div>
<div class="input-group">
<button id="color_mode" type="button" onclick="toggle(); return false;"
style="background-color: inherit; border: none;">
<imgsrc="https://content.invisioncic.com/r229491/monthly_2018_10/icon.png.6ea8a7a7f
bcf4c57df7b28ba4e996bb2.png"height="20" width="20"> <p id="color_m"
style="margin-bottom: 0;">Dark Mode</p></button>
<button type="submit" class="btn" name="login_user">Login</button>
        </div>
        <p>Not yet a member? <a href="register.php" class="ref">Sign up</a>
        </p><p>
        Cancel and return to home page.<a href="http://localhost/Build"
class="ref">Portal</a>
```

```
                </p>
        </form>
</div>
<div class="col-md-6 fill-image">
        <img src="pic1.jpg" width="100%" height="550px">
</div></div></div>
</body>
</html>
```

Submission View:

```
<?php
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT sub, descp, ver, sug
   from complaints where tos ='$q' and ver='$r' and usn='$usn'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
   echo "<table style=\"background-color: bisque;\">
   <tr style=\"background-color: black; color: white;\">
   <th>Si_no</th>
   <th>Subject</th>
   <th>Description</th>
   <th>Status</th>
   </tr>";
   while($row = $result->fetch_assoc() and $i<=10) {
      echo "<tr>";
      echo "<td>" . $i . "</td>";
      echo "<td>" . "<a href=\"action_com.php?q='$row[sug]'\">".$row['sub'] ."</a>" .
"</td>";
      echo "<td>" . $row['descp'] . "</td>";
      echo "<td>" . $inVal[$row['ver']] . "</td>";
      echo "</tr>";
      $i = $i+1;
   }
   echo "</table>";
} else {
   echo "0 results";
}
$conn->close();
?>
```

Submission Portal:

```
<?php
   $conn = new mysqli($servername, $username, $password, $dbname);
   // Check connection
   if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
   }
   #for update and calling mailserver
   $sql = "Update complaints SET ver=0 WHERE sug='$sug' ";
   $result = $conn->query($sql);
   if ($result === TRUE) {
      echo "Record updated successfully";
   } else {
      echo "Error updating record: " . $conn->error;
   }
   $conn->close();
?>
```

Verification:

```
<script>
function checkOTP() {
   var otp = parseInt(document.getElementById("otp").value);
   var p_id = document.getElementById("demo");
   p_id.innerHTML = usn + " " + otp + " "+rd + typeof(otp) + typeof(rd);
   if(otp == rd){
      p_id.innerHTML="you have been verified your suggestion is noted we will inform
you shortly of the development"
      document.getElementById("home").innerHTML='you can further browse by going
back home'
   }
   else{
      p_id.innerHTML = "Invalid OTP";
      p_id.setAttribute('style',"color:red;");
   }
   function myFunction() {
      window.open("/Web-Project-master/App");
   }
}
</script>
<body>
   <div class="col-lg-8 text-center">
   <hr>
  <h1>enter the OTP sent to your mail</h1>
  <form  action="javascript:myFunction(); return false;">
   <label for="otp"> OTP verification code:</label>
```

```
    <input type="text" id="otp">
    <input type="button" onclick="checkOTP()" value = "submit">
  </form>
  <hr>
  <h3></h3>
  <p id="demo">  </p>
  <p id="home"></p>
</div>
</body>
```

Teacher home portal:

```
<!DOCTYPE html>
<html lang=en><head>
<title>Teacher Page</title>
<link rel="icon" href="stylesheets/bitlogo.png" type="image/x-icon">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
<link href="https://cdn.bootcss.com/bootstrap/4.0.0-alpha.2/css/bootstrap.min.css"
rel="stylesheet">
<link rel="stylesheet" href="./stylesheets/main.css">
<script>
  function fetch_table1() {
    var h=name;
    var h1 = document.getElementById("complaint").value;
    var temp2 = document.getElementById("complaint_ver").value;
    document.getElementById("demo1").innerHTML = h;
    if (name.length == 0) {
      document.getElementById("demo1").innerHTML = "";
      return;
      } else {
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
        document.getElementById("demo1").innerHTML =  blol + this.responseText;
        }
      };
    xmlhttp.open("GET", "queryengine01.php?q=" + h1 + "&r=" + temp2, true);
    xmlhttp.send();

    }
  }
  function fetch_table2() {
    var h=name;
    var h1 = document.getElementById("suggest").value;
    var temp2 = document.getElementById("suggest_ver").value;
```

```
   document.getElementById("demo1").innerHTML = h;
   if (name.length == 0) {
     document.getElementById("demo1").innerHTML = "";
     return;
     } else {
     var xmlhttp = new XMLHttpRequest();
     xmlhttp.onreadystatechange = function() {
       if (this.readyState == 4 && this.status == 200) {
       document.getElementById("demo1").innerHTML =  blol + this.responseText;
       }
     };
   xmlhttp.open("GET", "queryengine02.php?q=" + h1 + "&r=" + temp2, true);
   xmlhttp.send();
   }
 }
 function next_query1() {
   window.open("queryengine03.php?q=" + name);
 }
 function next_query2() {
   window.open("queryengine04.php?q=" + name);
 }
</script>
</head>
<body onload="script:color_check();">
    <nav class="navbar navbar-fixed-top navbar-dark ">
          <ul>
            <li><a href="http://localhost/Build">Home</a></li>
            <li style="background-color:red;"><a
href="http://localhost/Build/App/teacher">Teacher</a></li>
            <li style="float: right;background-color: aquamarine; border: none; padding:
2px;">
             <button id="color_mode" onclick="toggle()" style="background-color: inherit;
border: none;">
              <img
src="https://content.invisioncic.com/r229491/monthly_2018_10/icon.png.6ea8a7a7fbcf4c
57df7b28ba4e996bb2.png"
              height="20" width="20"> <p id="color_m" style="margin-bottom: 0;">Dark
Mode</p></button>
            </li>
          </ul>
    </nav>
   <div class="container">
     <hr style="background-color:red;">
     <br><br><br><br>
       <div class="row" style="display: flex;">
```

```
        <div class="col-md-5 title-logo"><img src="./stylesheets/student.png" class="img-
responsive"></div>
        <div class="col-md-7 text-right">
          <h3 class="title-super text-uppercase text-thin" style="color:
#686868c9;">Teacher Portal</h3>
          <h4 class="text-uppercase">Information you need.</h4>
        </div></div><div>
        <hr style="background-color:gray;">
      </div>
      <div class="row text-center">
        <h2 class="text-muted">Services</h2>
      </div>
      <div class="row text-center" style="padding-left: 10px;">
        <div class="col-md-6">
          <img src="./stylesheets/support.jpg" class="img-responsive im">
          <h3>Complaints</h3>
          <p>
            <form action="javascript:void(0);">
              <select id="complaint" >
                <option value="classes">Class</option>
                <option value="labs">Labs</option>
                <option value="teachers">Teachers</option>
                <option value="others">Others</option>
              </select>
              <select id="complaint_ver" >
                <option value="0">UnSeen</option>
                <option value="1">Read</option>
                <option value="2">Resolved</option>
              </select>
              <button onclick="fetch_table1()" >Search</button>
              <!-- <p id="demo1"></p> -->
            </form>
          </p></div>
        <div class="col-md-6">
          <img src="./stylesheets/idea.jpg" class="img-responsive im">
          <h3>Suggestions</h3>
          <p>
          <form action="javascript:void(0);">
            <select id="suggest">
              <option value="classes">Class</option>
              <option value="labs">Labs</option>
              <option value="teachers">Teachers</option>
              <option value="others">Others</option>
            </select>
            <select id="suggest_ver">
              <option value="0">UnSeen</option>
```

```
            <option value="1">Read</option>
            <option value="2">Resolved</option>
           </select>
           <button onclick="fetch_table2()">Search</button>
          </form>
          </p></div></div>
       <div class="row text-center">
       <p id="demo1"> </p>
       </div><br><hr>
       <div class="row text-center" style="padding-left:10px;">
         <div class="col-md-6">
          <iframe src="new.php?q=complaints" style="border:none; width:450px;
height:500px;"></iframe>
         </div>
         <div class="col-md-6">
          <iframe src="new.php?q=suggestions" style="border:none; width:450px;
height:500px;" ></iframe>
         </div></div><br>
       <hr style="background-color:gray;">
       <div class="row text-center">
         <div class="col-md-12">
         <p>Images used in this website are not owned by us.<br> Remember to keep the
complaints/suggestions respectful
         </p>
         <hr style="background-color:gray;">
         <a style="text-align: center;">Copyright 2019</a>
         </div>
         <div class="col-md-12">
          <a class="btn btn-social-icon btn-github" href="https://github.com/im-loki/Web-
Project">
              <span class="fa fa-github"></span>
          </a>
</div></div></div>
</body>
</html>
```

Teacher submission view:

```
<?php
$conn = new mysqli($servername, $username, $password, $dbname1);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT EMAIL as Email from users where USERNAME='$db_usn'";
$result = $conn->query($sql);
```

```
if ($result->num_rows == 1) {
// output data of each row
while($row = $result->fetch_assoc()) {
   $email = $row["Email"];
   $email_flag = 1;
}
} else {
echo "0 results";
}
$conn->close();

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO complaints (usn, sub, descp, tos, sug, ver)
VALUES ('$db_usn', '$db_sub', '$db_desc','$db_tos','$db_rd','0')";
if ($conn->query($sql) === TRUE && $email_flag==1 ) {
$mail->isSMTP();
$mail->Host = 'smtp.gmail.com';
$mail->SMTPAuth = true;
$mail->Username = 'projectbit2020@gmail.com';
$mail->Password = '041198@bit';
$mail->SMTPSecure = 'tls';
$mail->Port = 587;
$mail->setFrom('projectbit2020@gmail.com', 'ccwt');
$mail->addAddress($email);
$mail->Subject = 'Sent Test Email';
$mail->isHTML(true);
$mailContent = "<h2>Email from test server</h2>
            <p>bye</p> '$db_rd'";
$mail->Body = $mailContent;

if(!$mail->send()){
   echo "message not sent";
   echo "mail server error:" . $mail->ErrorInfo;
}else{
   echo '<script>';
   echo 'var rd = '.json_encode($db_rd).';';
   echo 'var usn = '.json_encode($db_usn).';';
   echo '</script>';
}
```

```
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}
   echo '<script>';
   echo 'var rd = '.json_encode($db_rd).';';
   echo 'var usn = '.json_encode($db_usn).';';
   echo '</script>';
$conn->close();
?>
```

Admin unblock submission:

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/Exception.php';
require 'PHPMailer/PHPMailer.php';
require 'PHPMailer/SMTP.php';
function mail_service($sug, $serv, $usn, $email) {
   $mail = new PHPMailer;
   $mail->isSMTP();
   $mail->Host = 'smtp.gmail.com';
   $mail->SMTPAuth = true;
   $mail->Username = 'projectbit2020@gmail.com';
   $mail->Password = '041198@bit';
   $mail->SMTPSecure = 'tls';
   $mail->Port = 587;
   $mail->setFrom('projectbit2020@gmail.com', 'ccwt');
   $mail->addAddress($email);
   $mail->Subject = 'Sent Test Email';
   $mail->isHTML(true);
   $mailContent = "<h2>Email from test server</h2>
                <p>Hi '$usn'</p>
                <br>
                <p>Your feedback with id '$sug' </p>
                <p>As been updated to status <strong>'$serv'</strong></p><hr>";
   $mail->Body = $mailContent;
   if(!$mail->send()){
      echo "message not sent";
      echo "mail server error:" . $mail->ErrorInfo;
   }else{
      echo "mail sent";
   }
}
$conn_email = new mysqli($servername, $username, $password, 'registration');
```

```
// Check connection
if ($conn_email->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT email from users where username='$usn' ";
$result = $conn_email->query($sql);
if ($result->num_rows > 0) {
   while($row = $result->fetch_assoc()) {
      $email = $row['email'];
   }
}
$conn_email->close();
#for update and calling mailserver
$sql = "Update complaints SET ver='$serv' WHERE sug='$sug' ";
$result = $conn->query($sql);
if ($result === TRUE) {
   echo "Record updated successfully";
   mail_service($sug, $inVal[$serv], $usn, $email);
} else {
   echo "Error updating record: " . $conn->error;
}
$sql = "Update complaints SET reply='$reply' WHERE sug='$sug' ";
$result = $conn->query($sql);
if ($result === TRUE) {
   echo "Record updated successfully";
   header("Location: http://localhost/Build/App/teacher/index.php");
} else {
   echo "Error updating record: " . $conn->error;
}
$conn->close();
?>
```

Snapshots

Teacher Portal:

Represents the portal layout of teacher after successful login

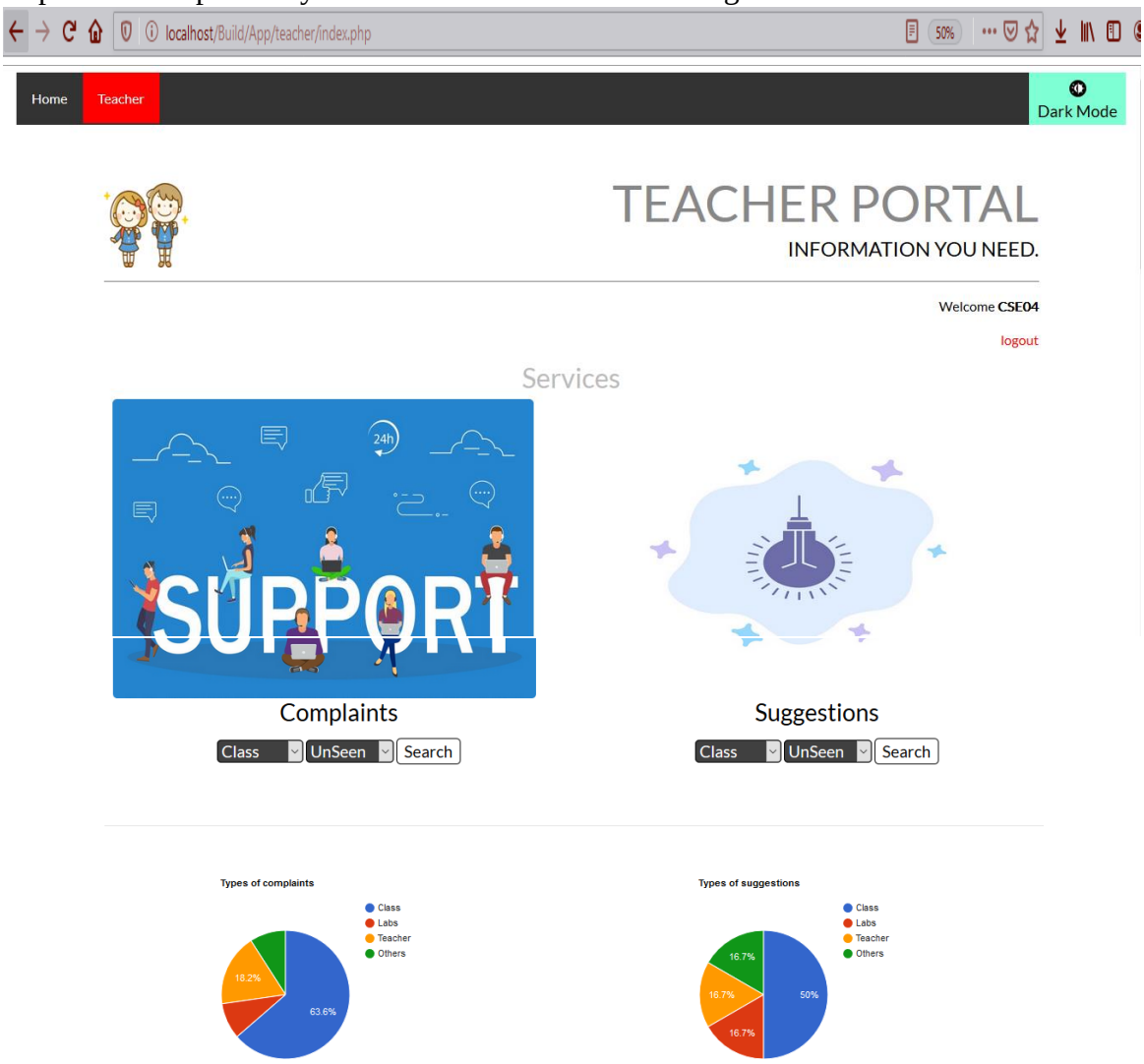


Figure 6.1: Teacher Portal

On clicking the search button, we get a table showing all the relevant submissions

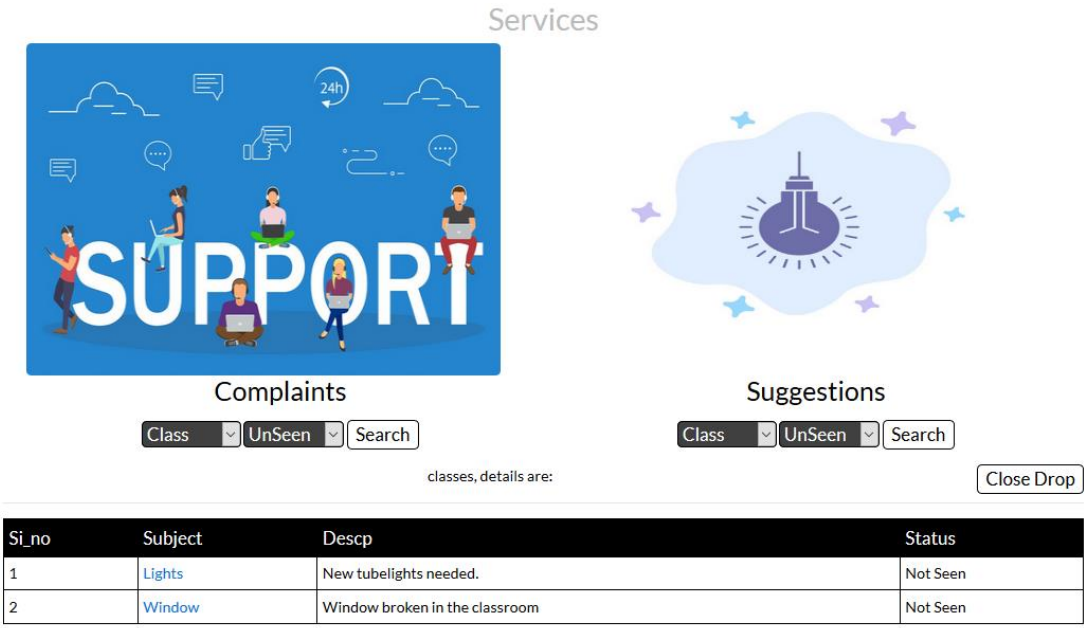


Figure 6.2: Selection view

Teachers can update the status of the submission along with an optional reply

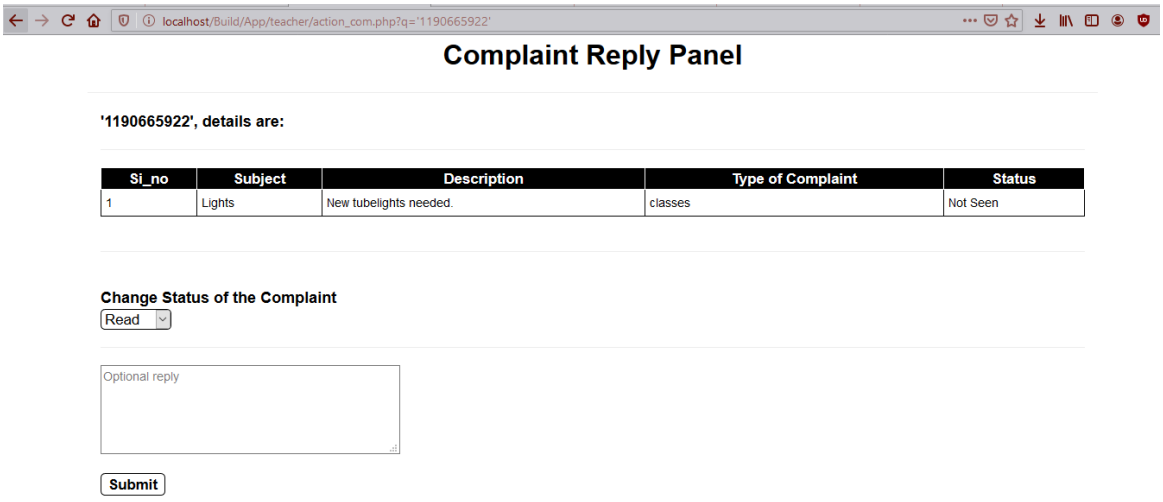


Figure 6.3: Action Panel

Login Portal:

User enters their username and password, then clicks on login

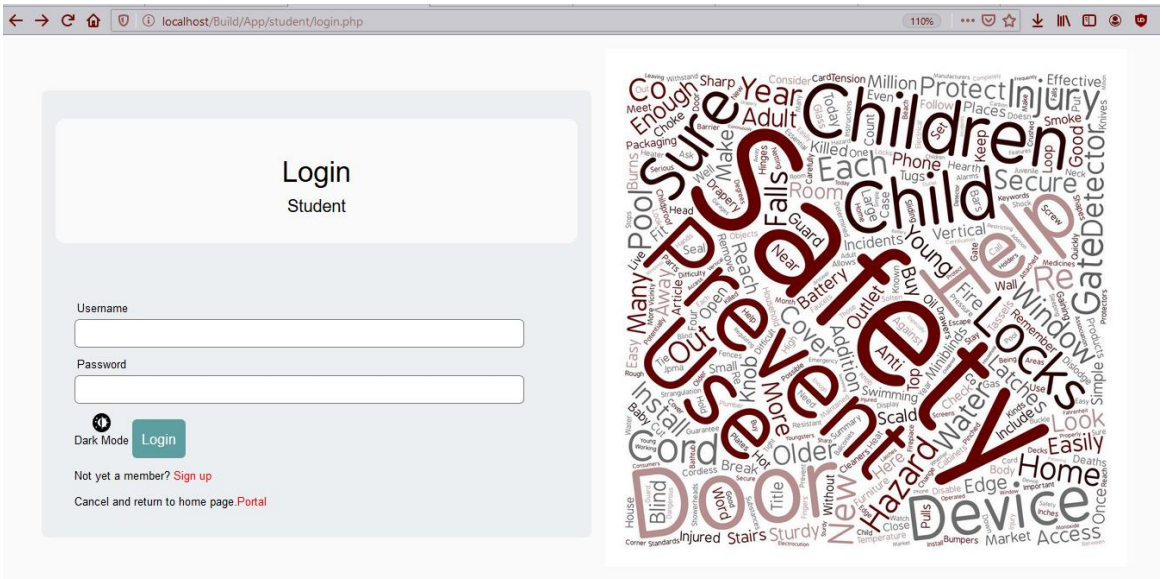


Figure 6.4: Login Page

New Users can register themselves by entering the following details

Figure 6.5: Signup Page

Student Portal:

Represents the portal layout of teacher after successful login

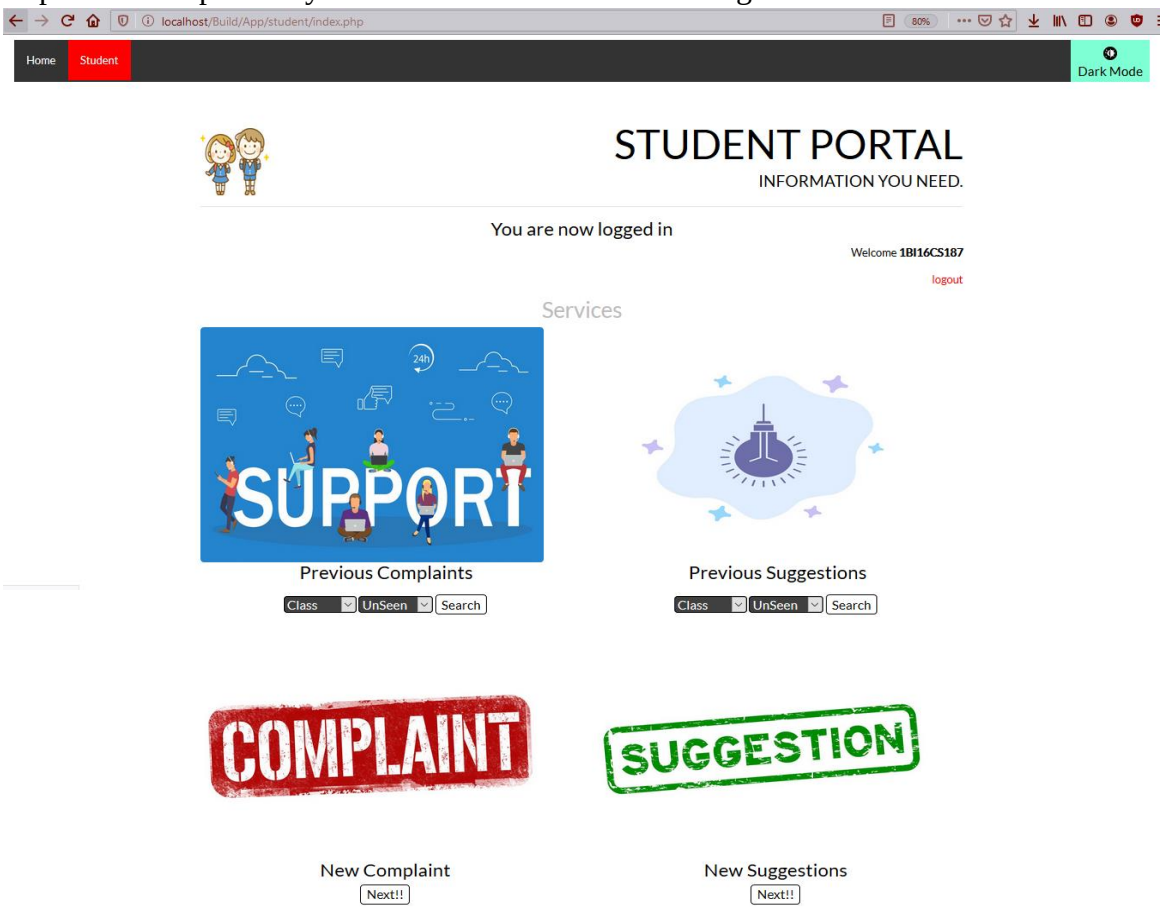
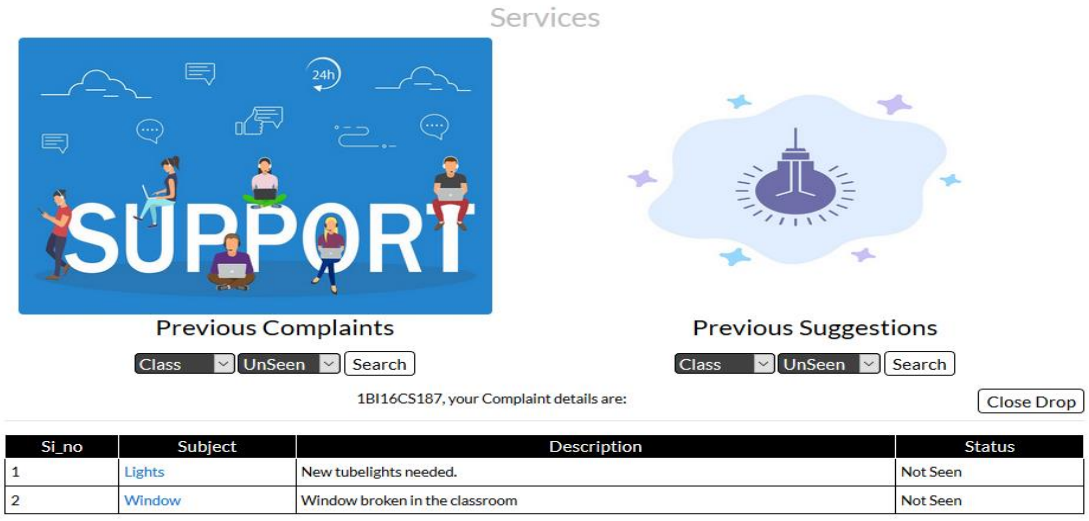


Figure 6.6: Student Portal

On clicking the search button, we get a table showing all the relevant submissions



Si_no	Subject	Description	Status
1	Lights	New tubelights needed.	Not Seen
2	Window	Window broken in the classroom	Not Seen

Figure 6.7: Selection Panel

Student can re-open submission if not satisfied with the response

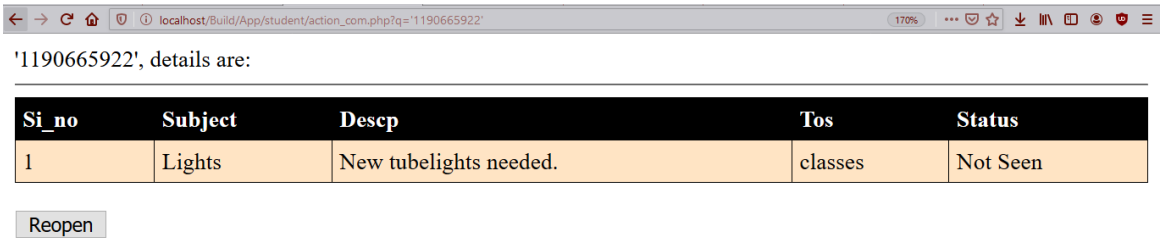


Figure 6.8: Re-open panel

Student can fill up the complaint form

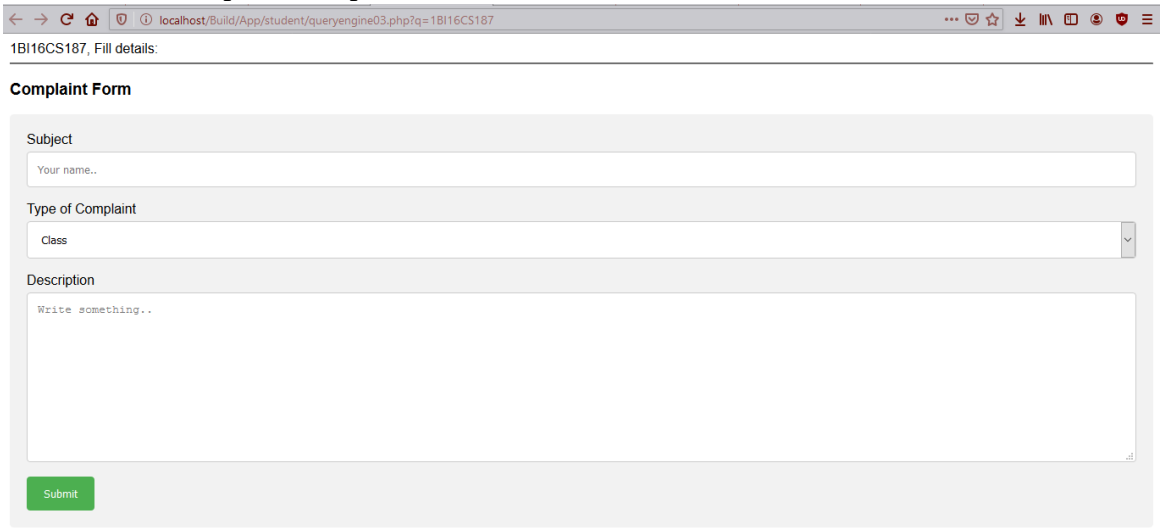


Figure 6.9: Fill-up form

Student must enter the OTP received on their registered mail for authencity

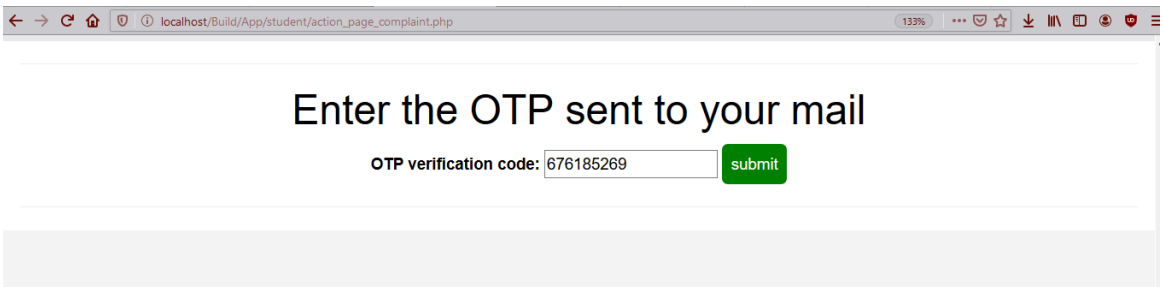


Figure 6.10: Email Verification

Admin Portal:

We get a table showing all the relevant submissions that has used foul language

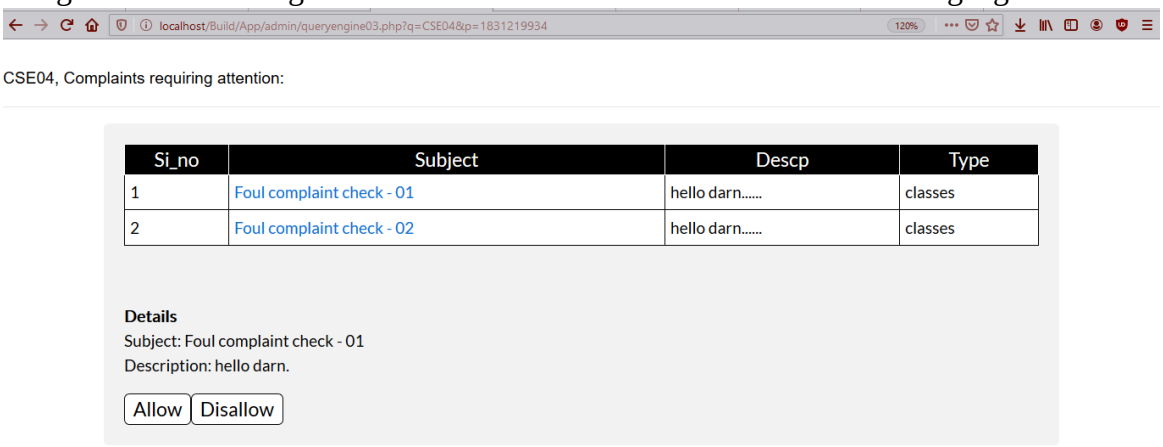


Figure 6.11: Foul Complaint Allow

Provides Option to Allow or Disallow the submission

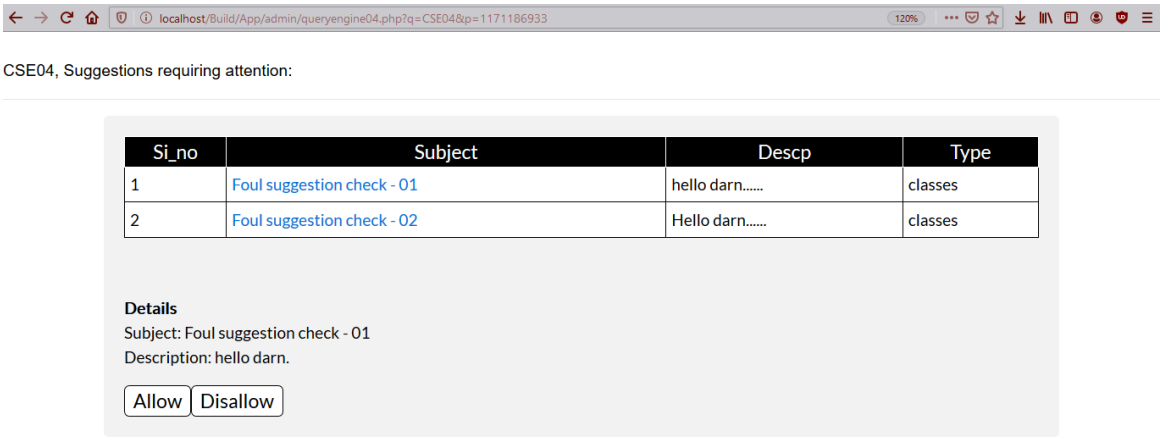


Figure 6.12: Foul Suggestion Allow

Application

- This application can be used in all colleges to collect Complaints and suggestions
- It can be viewed by a teacher without relieving student's usn
- Students get updated by their emails when status of their submission's changes
- Teacher can view a pie chart representing type of submission. So, they can concentrate on a given area
- Submissions using foul language are blocked, and are reviewed by Admin

Conclusion

The internet is very powerful platform for people to share their views in confidence. This project draws further in notion of the same lines, allowing Students to place complaints and suggestions without relieving their identity. Colleges and institutions improve from the useful insights provided through our platform. Thus, allowing them to understand the needs of students wholesomely.

“Online Complaints, Suggestions and Compliances” is a website and through this project we have learnt to design webpages using HTML, CSS, JavaScript and PHP. It has enabled us to have a deeper understanding of how frameworks help in development of websites.

Thus, in this project we have acquired a lot of knowledge about various technologies in web development. We have explored many new concepts on the web, such as JavaScript and AJAX.