Project Title: Philadelphia University Forms

Submitted By:

Munes Mutasem Zaki Bani Fawaz - 201710379

Ibrahim Abdulhakim Mustafa Hammad – 201720212

Supervisor:

Dr. Muhammad Al-Tayee

January – 2021

Approval

We certify that we have read the project titled “Philadelphia University Forms”, and as a members of project evaluation committee we had examined the students in the content of this document and knowledge related to it, and we certify that it is adequate with standings as a project for partial fulfillment of the requirements of B.Sc. in Software Engineering department.

Chairman:

Name:

Date:

Signature:

Member:

Name:

Date:

Signature:

Certificate

It is certified that this project has been prepared and written under my direct supervision and guidance. I also would like to certify that this document is approved for submission and evaluation.

Supervisor:

Signature:

Date:

Dedication

THIS PROJECT IS DEDICATED TO ALL OUR FAMILY MEMBERS OF

MUNES BANI FAWAZ & IBRAHIM AL HAMMAD ESPECIALLY

OUR PARENTS, BROTHERS AND SISTERS. WE LOVE YOU ALL.

ALSO, WE WOULD LIKE TO DEDICATE THIS PROJECT TO OUR TEACHERS FOR ALL THE GOOD TIMES WE HAD WITH THEM THROUGH OUR TIME IN THE UNIVERSITY AND FOR ALL THE GOOD AND HARD WORK THEY DID BY TEACHING US, MAY THEIR KNOWLEDGE CONTINUE THROUGHOUT ALL GENERATIONS.

SO, WE DEDICATE THIS PROJECT TO:

DR. LAMIS AL-QORA'N

DR. ABDULRAHMAN OBIDAT

DR. ENAS AL-NAFFAR

DR. MUHAMMAD AL-TAYEE

DR. RAWAN ABU AL-LAIL

Table of Contents

List of Figures

List of Tables

Glossary of Terms

Term	Definition
HOD	Head of Department

Table (1): Glossary of Terms

Chapter 1 - System Conception

Here we are going to answer 6 questions in regards to understand the application and its usage and what it is intended for.

1.1 - Who is the application for?

- Student: The Student will fill an online form on the website and send it to the advisor.
- Advisor: The Advisor receive the form from student to review the form and approve it and give reason.
- Head of department: The Head of department receive the form from Advisor to review the form and approve it and give reason.

1.2 - What problems will it solve?

The biggest problems that will get solved is that this project gives the advantage of not needing to go to the University to apply for a certain form and rather gives an easy, comfortable, time/effort saving way of applying for the form online without the need of the person to reach the university and it also gives the advantage of Approving/Rejection forms online with showing the status of the form without the need to contact the University or another party involved.

1.3 - Why is it needed?

The Student need this website to save time by filling the form online and by doing that you will reduce the loss of data and lower the cost, that will also lead to reduce the paperwork.

1.4 - Where will it be used?

It will be used on any device (Desktop-based, Phone-based) and it will also be available from homes or within the university so basically anywhere with a device that can open websites.

1.5 - How will it work?

First you must enter the URL website on any browser you want on any device the website should be opened and log in page must appear, fill your username and password provided by the University, then the main page will appear this page will allow you choose a form that you need to fill and send it to the advisor and head of department. From the Architecture point of view the website uses n tier Architecture the first tier called Presentation layer in this layer we present the view and the design of the website. Second layer called Business Logic layer. Business logic acts as an interface between Client layer and Data Access Layer. All business logic – like validation of data, calculations, data insertion/modification are written under business logic layer. Third layer called Data Access Layer: This is the data layer function, which receives the data from the business layer and performs the necessary operation into the database.

1.6 - When is it needed?

It is needed when a student needs to apply for a certain University Form that he/she needs to request or apply for something that involves his study in the university. All that is when the project is finished and that is at approximately the 10th of February 2020.

Chapter 2 - Domain Analysis

Here we are going to explain the Classes and what they are needed for in order to understand the project furthermore by analyzing the classes and to explain what each class is.

2.1 - Classes (Brainstorm)

(admins) (accounts) (login) (logout) (authentication) (notifications) (info) (details) (forms) (form send) (users) (privileges) (redirect) (deny) (login) (approval) (students).

2.2 - Bad Classes

(forms) (form send) (deny) (logout) (notifications) (info) (details) (privileges) (redirect) (accounts) (users).

2.3 - Good Classes

(System) (Students) (Admins) (Forms)

2.4 - Data Dictionary

System: Will contain each of the following classes that will control the user's login session based by type:

1-Students. (Will have handle all the credentials for the students that will use the system).

2-Admins. (Will have handle all the credentials for the advisors and head of departments that will use the system).

Forms: Will handle the user and admin side of the system for the forms than are/can be applied on the system. Including three forms (Overload, Closed Division, Call Up).

2.5 - Association's & Attributes

Class Diagram:

Here we will see the class diagram for the project that is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling thus it can be used for data modeling. All that is demonstrated in Figure (1) on the next page.

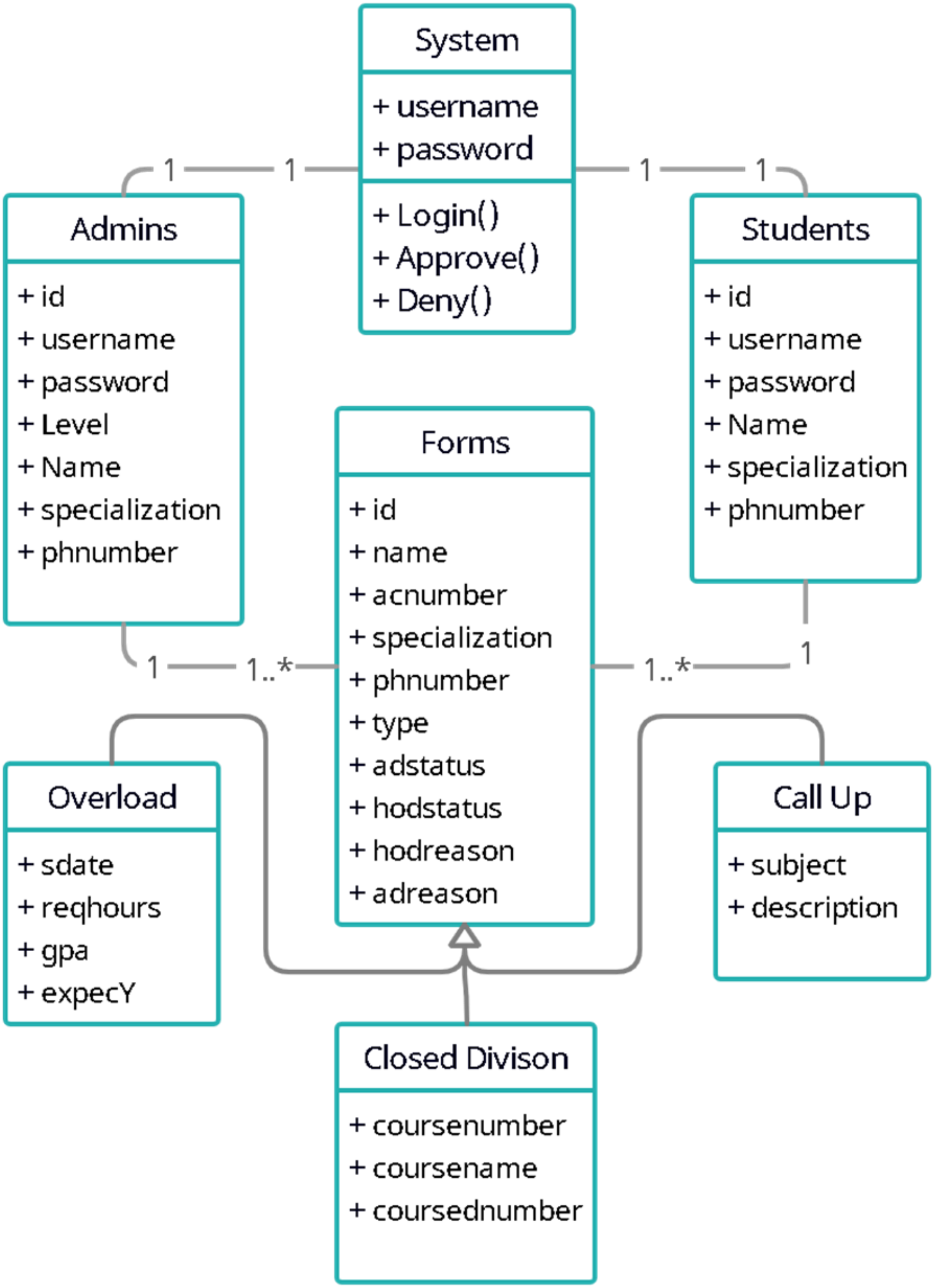


Figure (1): Class Diagram

Chapter 3 - Application Analysis

3.1 - Introduction about application analysis

Requirement is a condition or capability possessed by the software or system component in order to solve a real-world problem. The problems can be to automate a part of a system, to correct shortcomings of an existing system, to control a device, and so on. **IEEE** defines requirement as a condition or capability needed by a user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. Requirements describe how a system should act, appear or perform. For this, when users request for software, they provide an approximation of what the new system should be capable of doing. Requirements differ from one user to another and from one business process to another.

Different Types of Software Requirements:

a) Functional Requirements.
b) Non-Functional Requirements.

Find Actors:

a) Student.
b) Advisor.
c) Head of Department.

3.2 - Functional requirement

a) Login: All Users (Students, Advisors, Head of Departments) can use their username and password to log in to the website.
b) Logout: All Users (Students, Advisors, Head of Departments) can log out from the website using a button.
c) Fill Out A Form: The Student after Signing in will fill the form he needs.
d) Send Form: After filling a form the student sends the form to the advisor and head of department.
e) View Form Details: To view all the information in the forms.
f) View All Forms: The Student, advisor and HOD can view all the forms.
g) Approve: The advisor and HOD can approve the form.
h) Deny: The advisor and HOD can deny the form.
i) View Profile Info: To show all information corresponding to the user.
j) View Submitted Forms: The advisor and HOD can view the forms that students applied.

3.3 - Non-Functional requirement

a) Reliability: This quality attribute specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions.
b) Performance: defines how fast a software system or its particular piece responds to certain users’ actions under certain workload.

3.4 - Use Case Diagram

Here we will see the use case that represents a written description of how users will perform tasks on the website. It outlines from a user's point of view the system's behavior as it responds to a request. All that demonstrated in Figure (2).

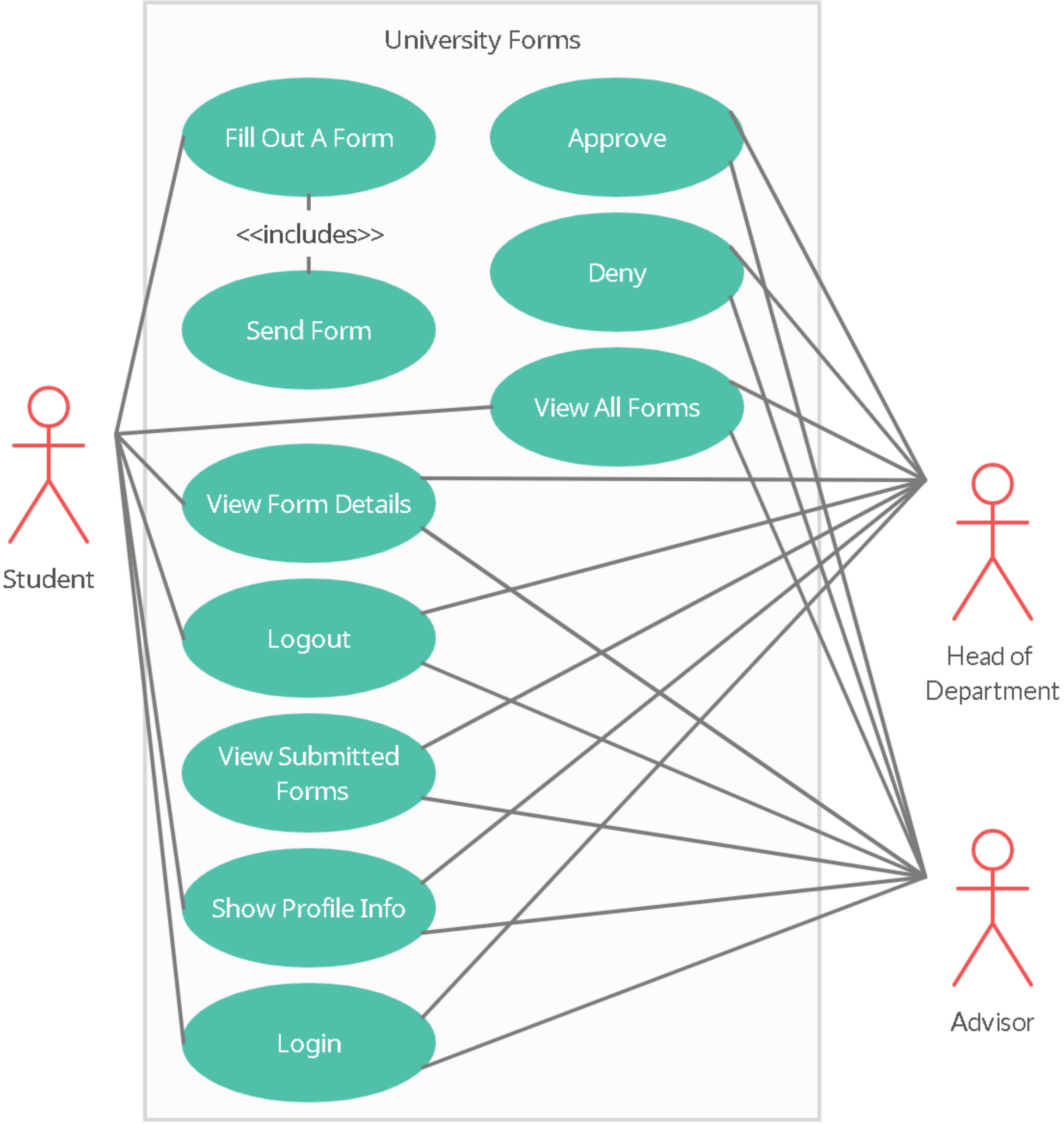


Figure (2): Use Case Diagram

3.5 - Use Case Requirements Specification

Here we will talk about the use case specifications that is typically created in the analysis and design phase in an iterative manner. At first, only a brief description of the steps needed to carry out the normal flow of the use case (what functionality is provided by the use case) that is written.

Overview	
Title	Login
Description	Use username and password to enter the website
Actor	Student, Advisor, Head of Department
Initial Status	None
Basic Flow	
Step1: Enter the URL of the website. Step2: Choose Student Portal or Admin Portal (Advisor, Head of Department). Step3: Enter username and password. Step4: Click login.	
Post Condition	
After clicking login the portal will open.	
Alternative Flow	
In Step 3 when you enter an invalid password an alert will appear.	

Overview	
Title	Logout
Description	Used to exit the portal in the website
Actor	Student, Advisor, Head of Department
Initial Status	Logged in
Basic Flow	
Step1: Click the logout button at the top of the website.	
Post Condition	
After clicking logout the portal will close.	
Alternative Flow	
None.	

Overview	
Title	Send Form
Description	Used after filling a form to send it on the system
Actor	Student
Initial Status	Logged in, Form Filled
Basic Flow	
Step1: Click the submit button after filling the form.	
Post Condition	
After clicking submit the form will be sent.	
Alternative Flow	
In Step1 when clicking submit if a form field is not filled it will prompt to be filled.	

Overview	
Title	Fill Out A Form
Description	The Student after he Signed in, he will fill the form he needed
Actor	Student
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: Choose the form needed to fill. Step3: After filling the form click submit.	
Post Condition	
When you submit you will get a notification counter in the top screen of the website.	
Alternative Flow	
Enter non-valid data cause alert to appear.	

Overview	
Title	View All Forms
Description	The Student, advisor and HOD can view all the form.
Actor	Student, advisor and HOD
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: Click My Form in the top section of the website.	
Post Condition	
After clicking my form all of your forms will appear to you.	
Alternative Flow	
None.	

Overview	
Title	View Form Details
Description	The Student, advisor and HOD can view form details
Actor	Student, advisor and HOD
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: Click details beside your forms to view details.	
Post Condition	
After clicking details all form information’s will appear to you.	
Alternative Flow	
None.	

Overview	
Title	Approve / Deny
Description	After review the form advisor and HOD can approve it or deny it
Actor	Advisor and HOD
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: After reviewing a form the advisor and HOD can approve it or deny it.	
Post Condition	
Notification will appear.	
Alternative Flow	
None.	

Overview	
Title	View Profile Info
Description	Show all information about profile
Actor	Student, advisor and HOD
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: After clicking Profile all the information of the users will appear.	
Post Condition	
Information will appear.	
Alternative Flow	
None.	

Overview	
Title	View Submitted Forms
Description	Show received form by enter student id
Actor	Advisor and HOD
Initial Status	Logged in
Basic Flow	
Step1: Log in. Step2: Choose Student id from the drop-down list. Step2: After clicking view all the forms will appear to you.	
Post Condition	
Submitted forms will appear.	
Alternative Flow	
None.	

Chapter 4 - System Design

4.1 - Introduction about software Architecture

4.1.1 - What is software architecture?

The software architecture of a system depicts the system’s organization or structure, and provides an explanation of how it behaves. A system represents the collection of components that accomplish a specific function or set of functions. In other words, the software architecture provides a sturdy foundation on which software can be built. A series of architecture decisions and trade-offs impact quality, performance, maintainability, and overall success of the system. Failing to consider common problems and long-term consequences can put your system at risk. There are multiple high-level architecture patterns and principles commonly used in modern systems. These are often referred to as architectural styles. The architecture of a software system is rarely limited to a single architectural style. Instead, a combination of styles often makes up the complete system.

4.1.2 - What is your software Architecture?

It’s a three tier Architecture.

4.1.3 - Why did you choose it?

- Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.
- The chief benefit of three-tier architecture is that because each tier runs on its own infrastructure, each tier can be developed simultaneously by a separate development team, and can be updated or scaled as needed without impacting the other tiers.

4.2 - Benefits of three-tier architecture

- Faster development: Because each tier can be developed simultaneously by different teams, an organization can bring the application to market faster, and programmers can use the latest and best languages and tools for each tier.
- Improved scalability: Any tier can be scaled independently of the others as needed.
- Improved reliability: An outage in one tier is less likely to impact the availability or performance of the other tiers.
- Improved security: Because the presentation tier and data tier can't communicate directly, a well-designed application tier can function as a sort of internal firewall, preventing SQL injections and other malicious exploits.

4.3 - Description of the three tiers in detail

1. Presentation Tier: (Tier 0):
    - The presentation tier is the user interface and communication layer of the application, where the end user interacts with the application. Its main purpose is to display information to and collect information from the user. This top-level tier runs on a web browser developed using HTML, PHP, CSS, Ajax, jQuery, Bootstrap and JavaScript.

2. Application Tier: (Tier 1):
    - The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed sometimes against other information in the data tier using logic, a specific set of rules. The application tier can also add, delete or modify data in the data tier.
    - The application tier is typically developed using PHP and SQL and communicates with the data.
3. Data Tier: (Tier 2):
    - The data tier, sometimes called database tier, data access tier or back-end, is where the information processed by the application is stored and managed. This Project uses a relational database management system called MySQL Server.

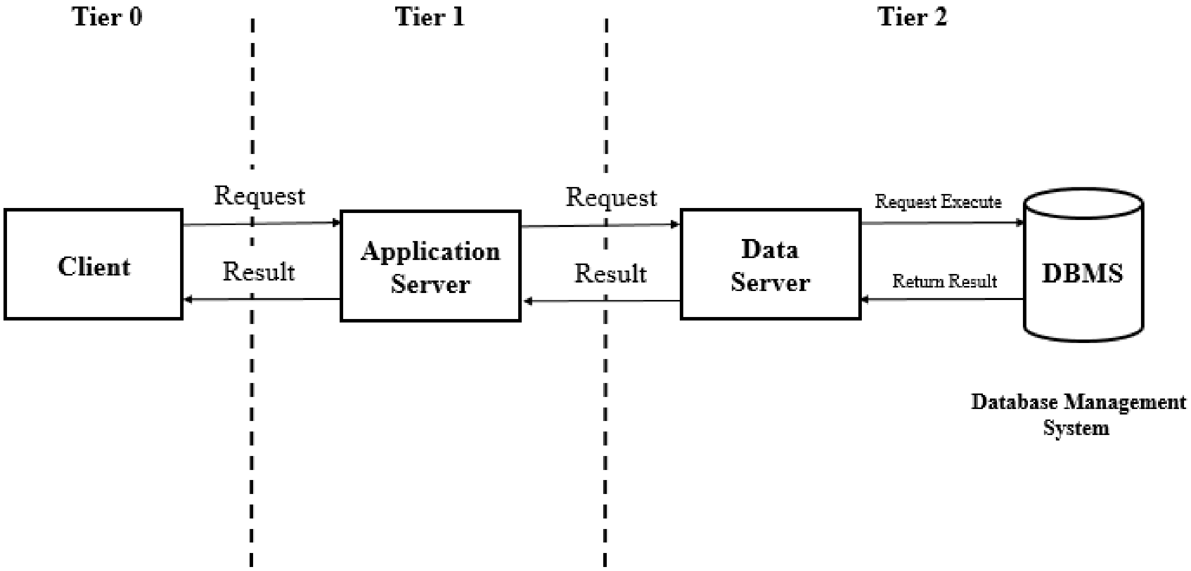


Figure (3): Three Tier Architecture

Chapter 5 - System Design

5.1 - Introduction about System Design

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. So here we are going to talk about the class diagram, database table, user interface design.

5.2 - Class diagram

As before here will see the class diagram for the project that is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling thus it can be used for data modeling, as demonstrated in Figure (4).

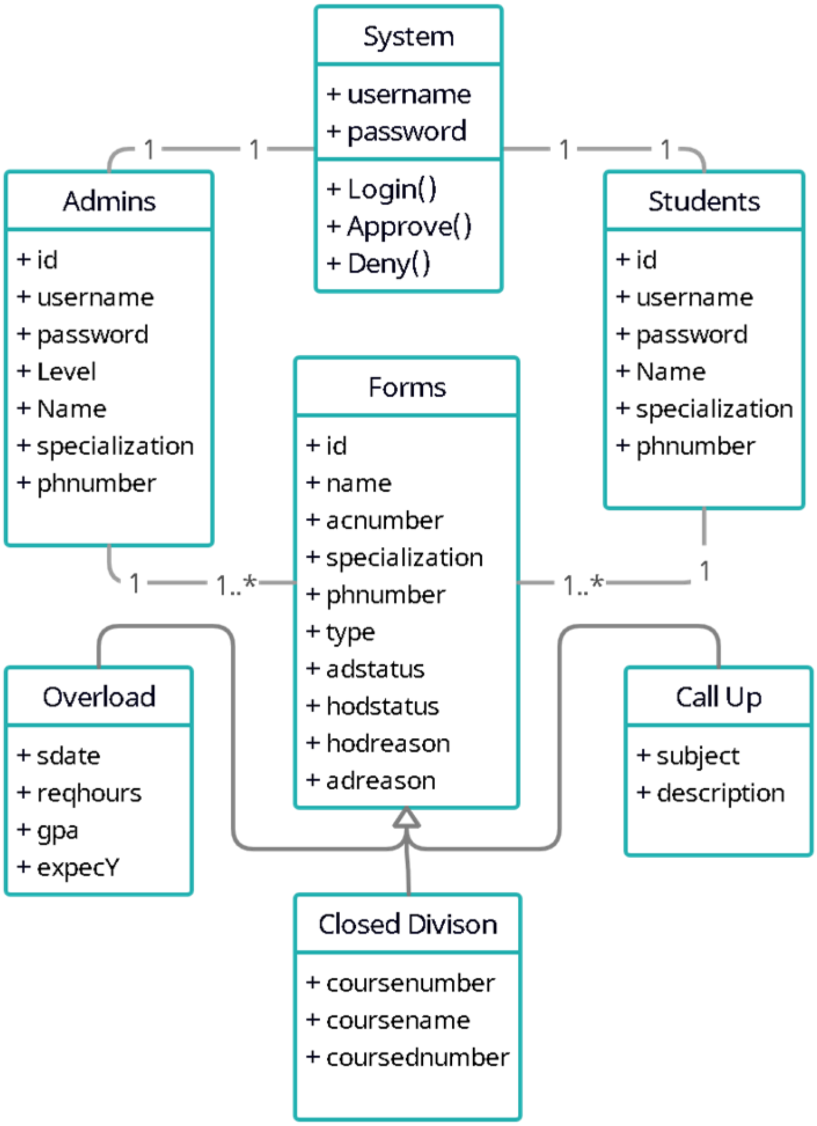
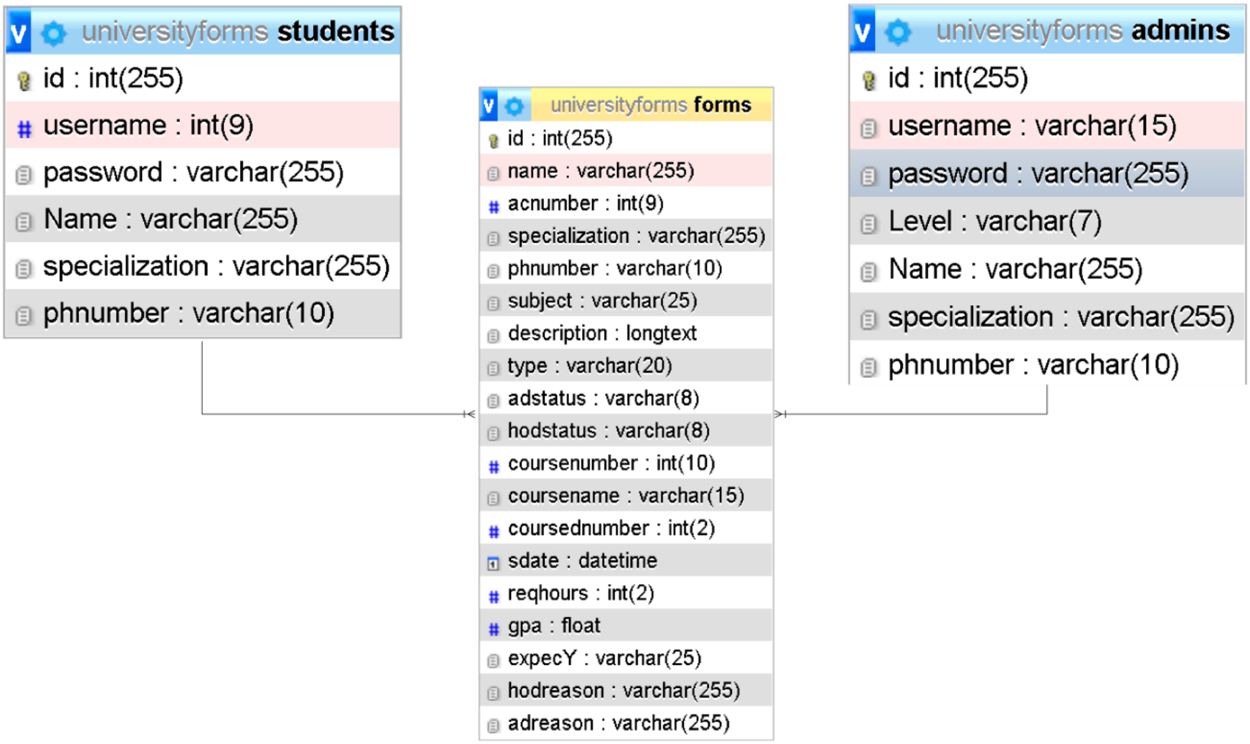


Figure (4): Class Diagram

5.3 - Relational Database Table

A relational database is a digital database based on the relational model of data. A software system used to maintain relational databases is a relational database management system. That said we will see the relational database for this project in Figure (5).



universityforms students

id : int(255)
username : int(9)
password : varchar(255)
Name : varchar(255)
specialization : varchar(255)
phnumber : varchar(10)

universityforms forms

id : int(255)
name : varchar(255)
acnumber : int(9)
specialization : varchar(255)
phnumber : varchar(10)
subject : varchar(25)
description : longtext
type : varchar(20)
adstatus : varchar(8)
hodstatus : varchar(8)
coursenumber : int(10)
coursename : varchar(15)
coursednumber : int(2)
sdate : datetime
reqhours : int(2)
gpa : float
expecY : varchar(25)
hodreason : varchar(255)
adreason : varchar(255)

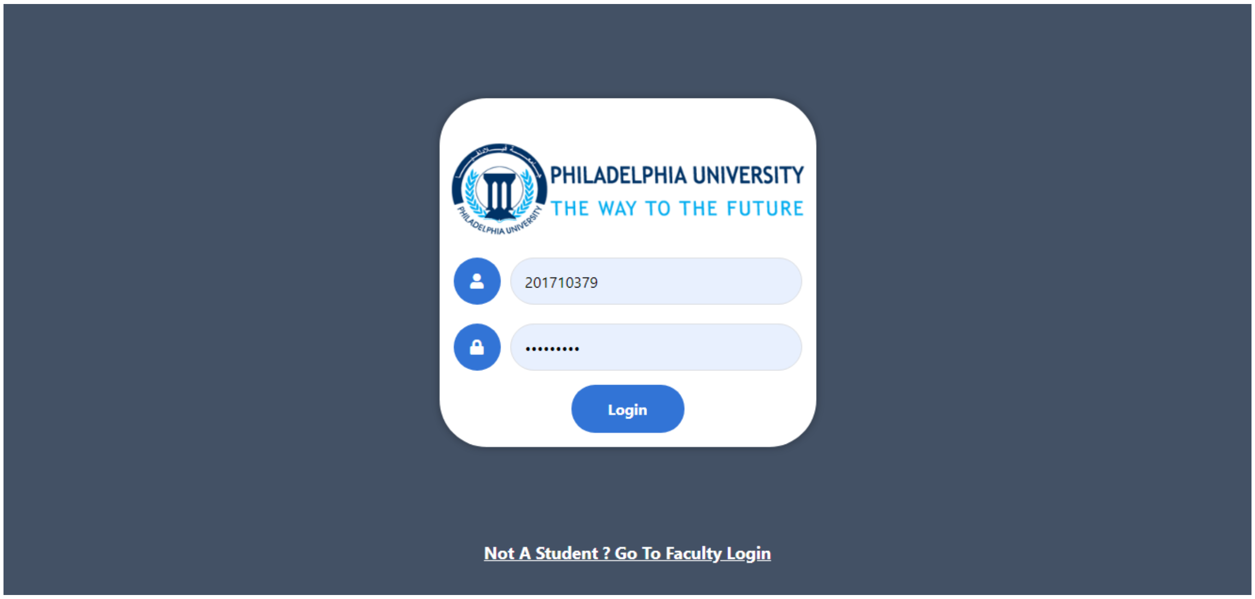
universityforms admins

id : int(255)
username : varchar(15)
password : varchar(255)
Level : varchar(7)
Name : varchar(255)
specialization : varchar(255)
phnumber : varchar(10)

Figure (5): Relational Database Table

5.4 - User Interface Design

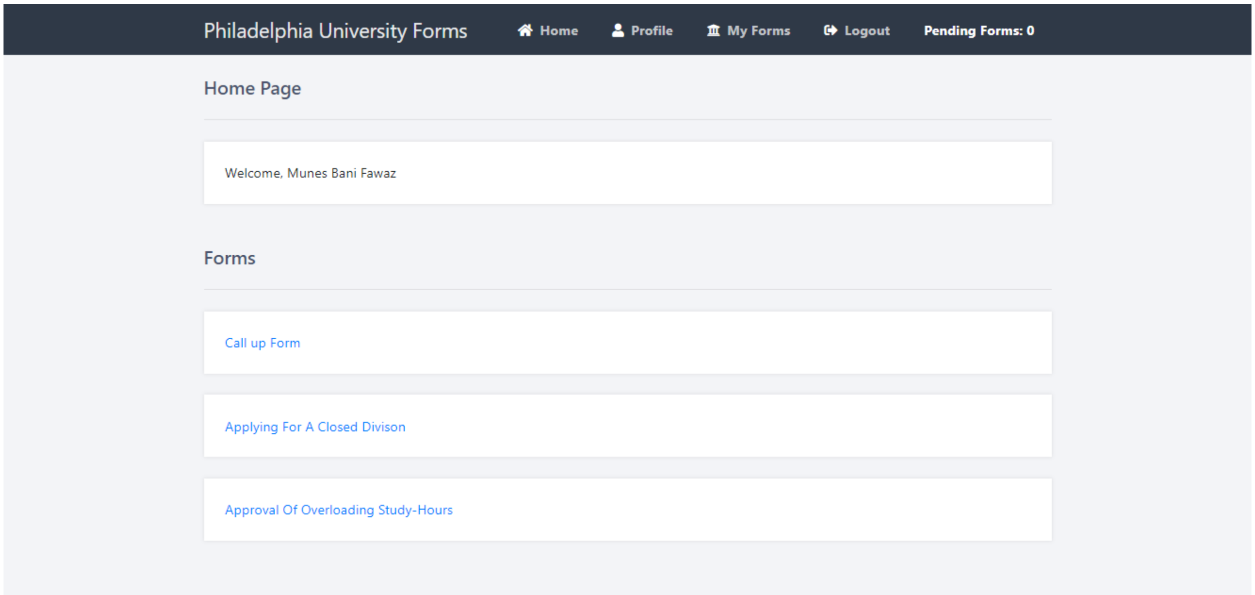
User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand and to use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture as demonstrated in Figures (6-11): User Interface Design.



201710379

•••••••••

Figure (6): User Interface Design (Login Page)



Home Page

Welcome, Munes Bani Fawaz

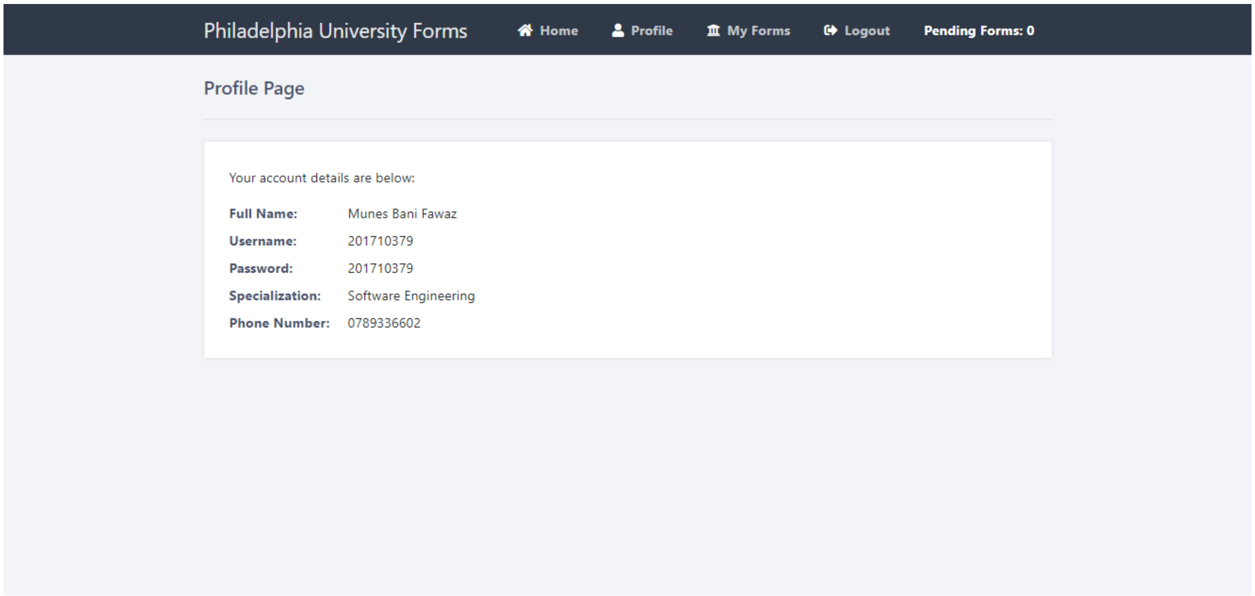
Forms

Call up Form

Applying For A Closed Divison

Approval Of Overloading Study-Hours

Figure (7): User Interface Design (Home Page)

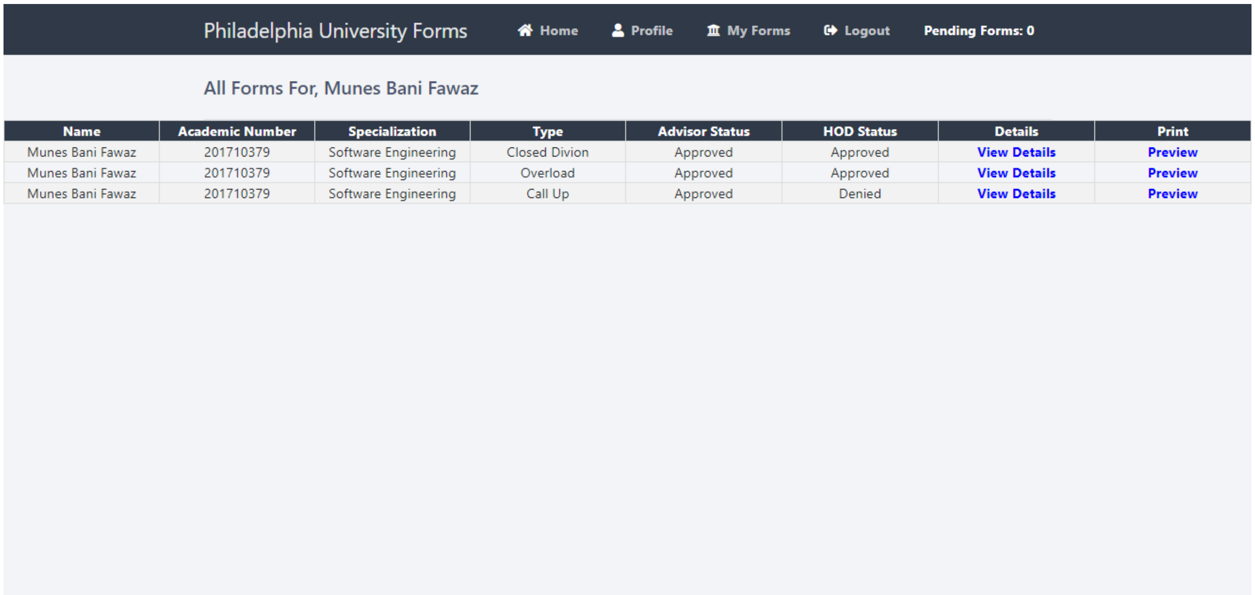


Profile Page

Your account details are below:

Full Name:	Munes Bani Fawaz
Username:	201710379
Password:	201710379
Specialization:	Software Engineering
Phone Number:	0789336602

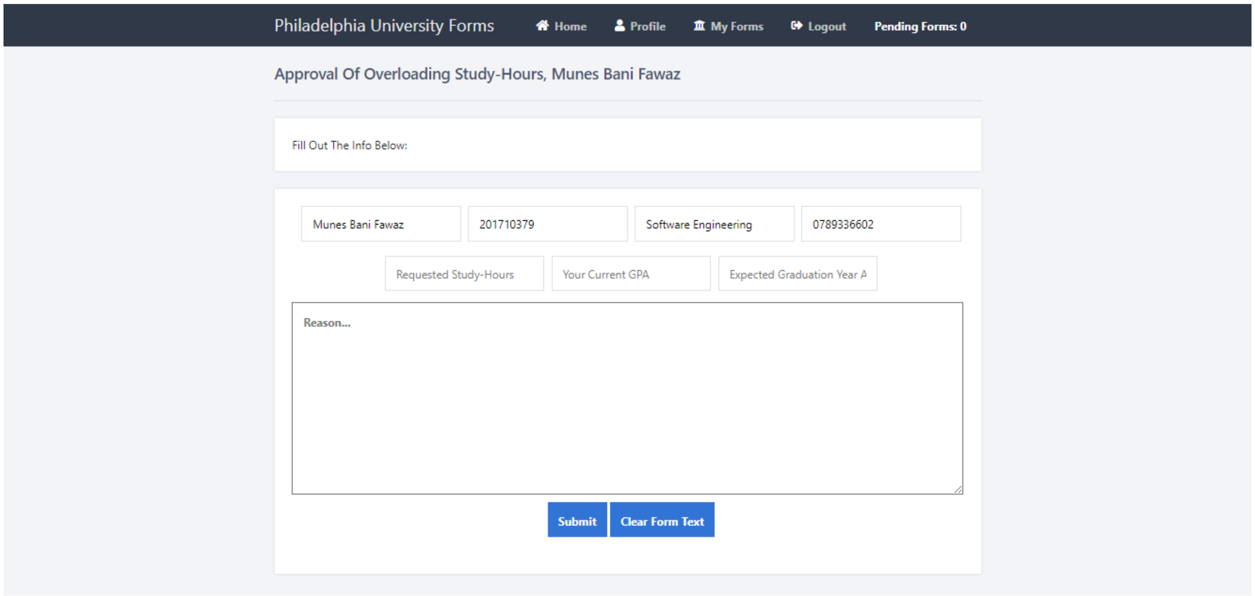
Figure (8): User Interface Design (Profile Page)



All Forms For, Munes Bani Fawaz

Name	Academic Number	Specialization	Type	Advisor Status	HOD Status	Details	Print
Munes Bani Fawaz	201710379	Software Engineering	Closed Divion	Approved	Approved	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Overload	Approved	Approved	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Call Up	Approved	Denied	View Details	Preview

Figure (9): User Interface Design (My Forms Page)



Approval Of Overloading Study-Hours, Munes Bani Fawaz

Fill Out The Info Below:

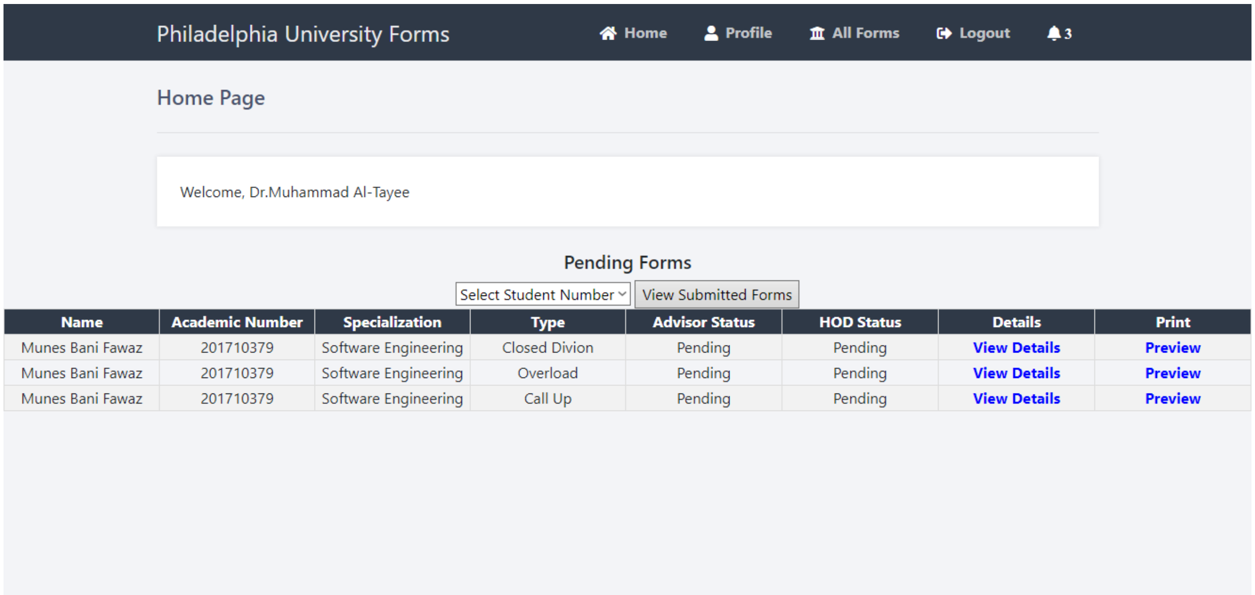
Munes Bani Fawaz	201710379	Software Engineering	0789336602
Requested Study-Hours	Your Current GPA	Expected Graduation Year A	

Reason...

Figure (10): User Interface Design (Student Overload Form Page)

The next Figure is the Admin Home Page

(Rest of the admin pages share the same concept as in the Figures Before).



Home Page

Welcome, Dr.Muhammad Al-Tayee

Pending Forms

Name	Academic Number	Specialization	Type	Advisor Status	HOD Status	Details	Print
Munes Bani Fawaz	201710379	Software Engineering	Closed Divion	Pending	Pending	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Overload	Pending	Pending	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Call Up	Pending	Pending	View Details	Preview

Figure (11): User Interface Design (Admin Home Page)

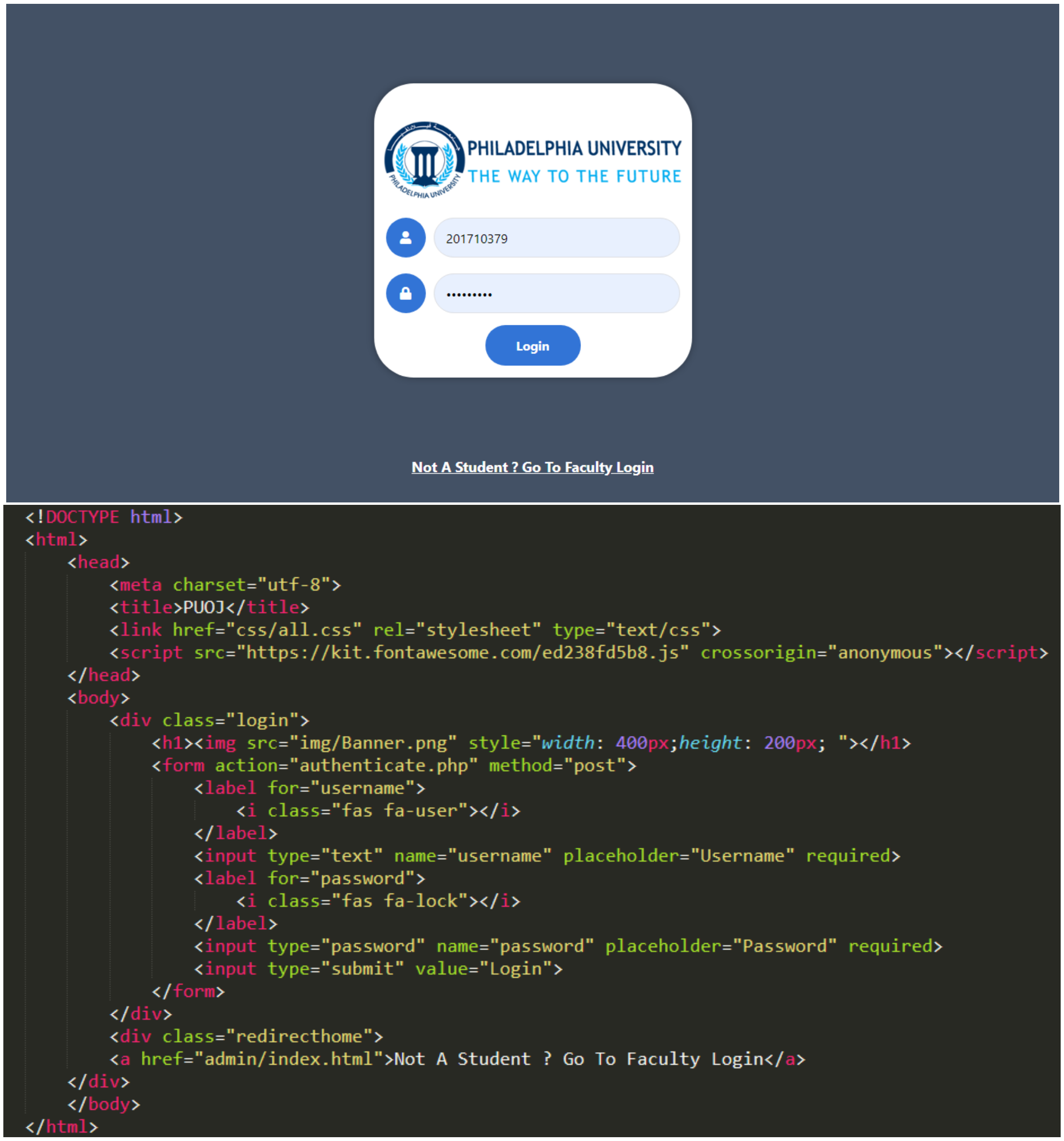
Chapter 6 - Implementation

6.1 - Introduction about Implementation

Software implementation refers to the process of adopting and integrating a software application into a business workflow. Implementation of new tools and software can be complex, depending on the size of the software so in this chapter we are going to demonstrate some of the implementation used in our project.

6.2 - Implementing the Login Method

We need a login form for our websites users to interact with and enter their details. So first we have our index.html file that will be the access portal to the user.



201710379

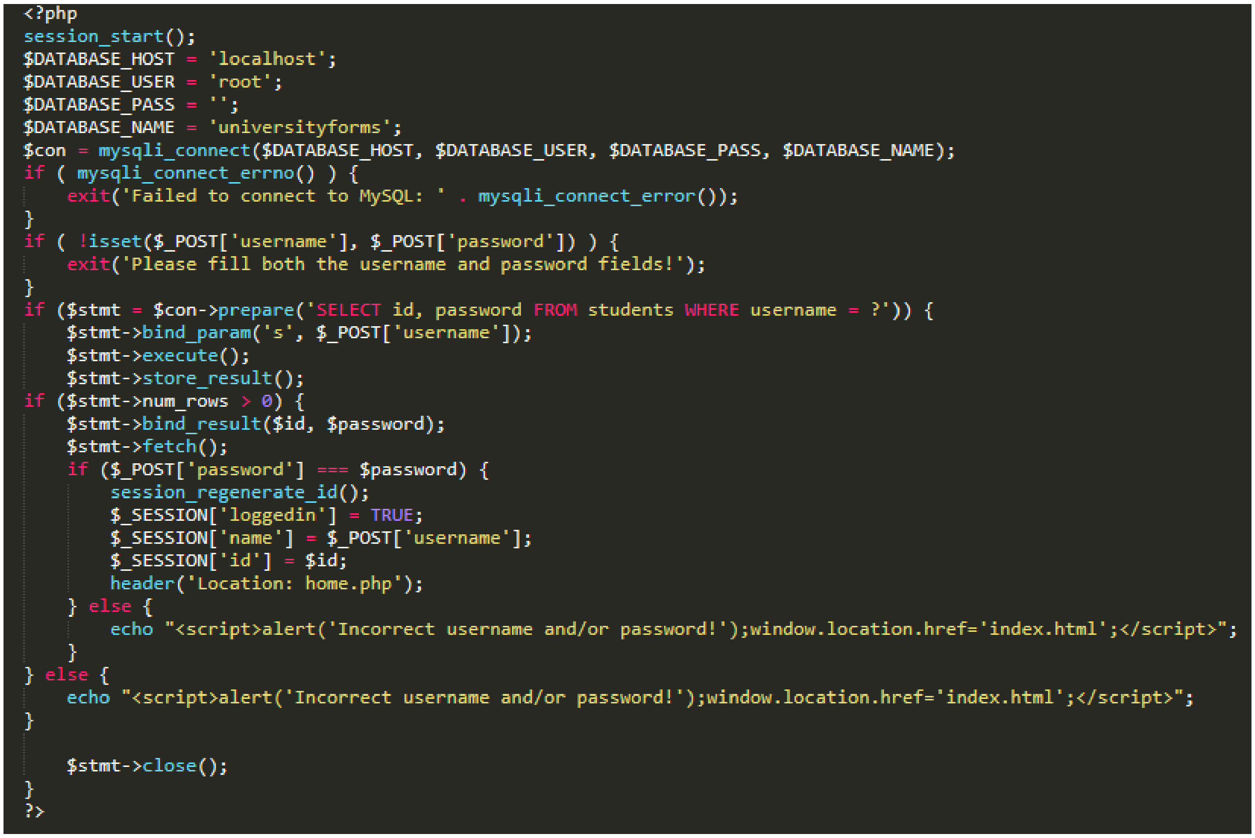
•••••••••

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>PUOJ</title>
        <link href="css/all.css" rel="stylesheet" type="text/css">
        <script src="https://kit.fontawesome.com/ed238fd5b8.js" crossorigin="anonymous"></script>
    </head>
    <body>
        <div class="login">
            <h1><img src="img/Banner.png" style="width: 400px;height: 200px; "></h1>
            <form action="authenticate.php" method="post">
                <label for="username">
                    <i class="fas fa-user"></i>
                </label>
                <input type="text" name="username" placeholder="Username" required>
                <label for="password">
                    <i class="fas fa-lock"></i>
                </label>
                <input type="password" name="password" placeholder="Password" required>
                <input type="submit" value="Login">
            </form>
        </div>
        <div class="redirecthome">
        <a href="admin/index.html">Not A Student ? Go To Faculty Login</a>
    </div>
    </body>
</html>
```

Figure (12): Implementation (Login Page Design & Code)

6.2.1 - Implementing the Login Authentication Method

Now after the user enters his login data a authenticate.php file will handle the request in order to see if the user's access is granted.

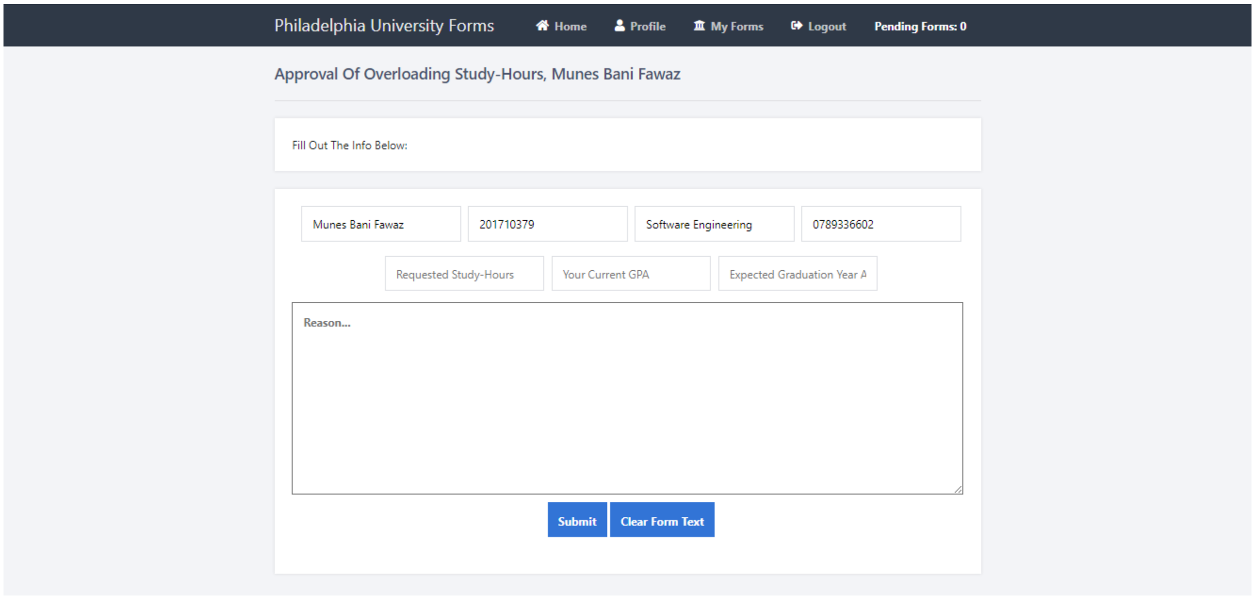


```php
<?php
session_start();
$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'universityforms';
$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
if ( mysqli_connect_errno() ) {
    exit('Failed to connect to MySQL: ' . mysqli_connect_error());
}
if ( !isset($_POST['username'], $_POST['password']) ) {
    exit('Please fill both the username and password fields!');
}
if ($stmt = $con->prepare('SELECT id, password FROM students WHERE username = ?')) {
    $stmt->bind_param('s', $_POST['username']);
    $stmt->execute();
    $stmt->store_result();
if ($stmt->num_rows > 0) {
    $stmt->bind_result($id, $password);
    $stmt->fetch();
    if ($_POST['password'] === $password) {
        session_regenerate_id();
        $_SESSION['loggedin'] = TRUE;
        $_SESSION['name'] = $_POST['username'];
        $_SESSION['id'] = $id;
        header('Location: home.php');
    } else {
        echo "<script>alert('Incorrect username and/or password!');window.location.href='index.html';</script>";
    }
} else {
    echo "<script>alert('Incorrect username and/or password!');window.location.href='index.html';</script>";
}

    $stmt->close();
}
?>
```

Figure (13): Implementation (Login Authentication Code)

6.3 - Implementing the Form Method

Now the main purpose of the project is that the student applies for a form in the university to be approved by the advisor and head of department so this next fragment of code is going to demonstrate a form for the university that the student fills in order to get approval of overloading his study hours.



Approval Of Overloading Study-Hours, Munes Bani Fawaz

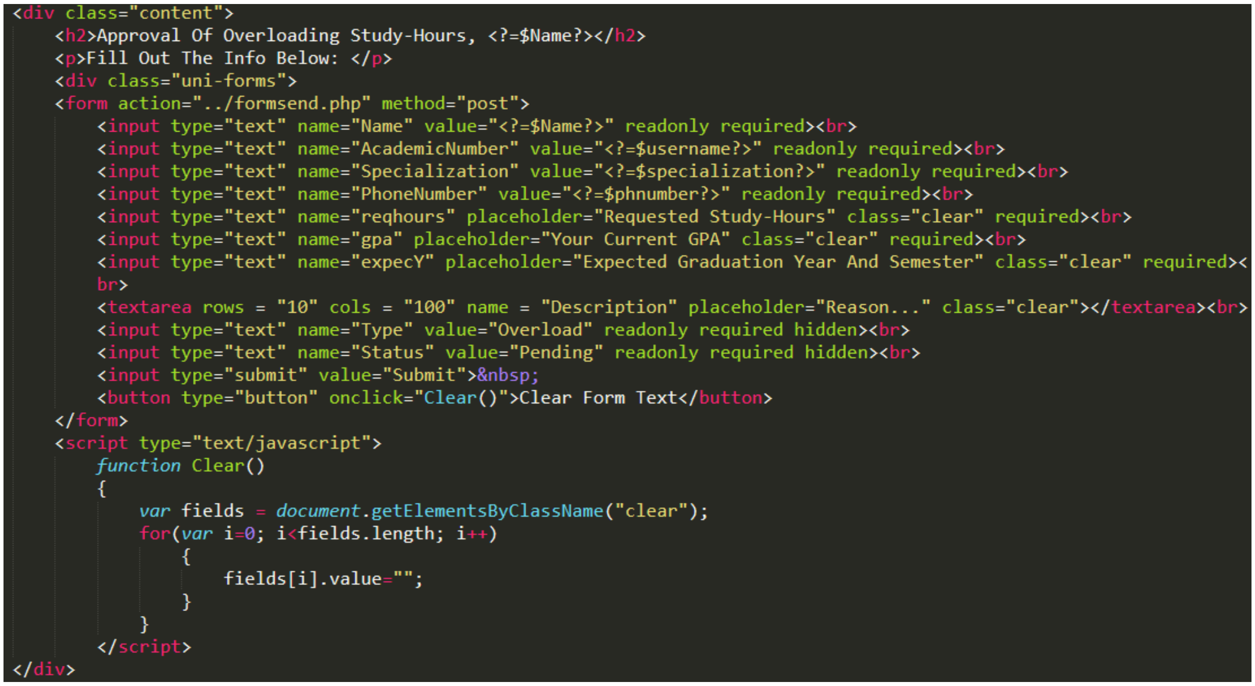
Fill Out The Info Below:

Munes Bani Fawaz

201710379

Software Engineering

0789336602

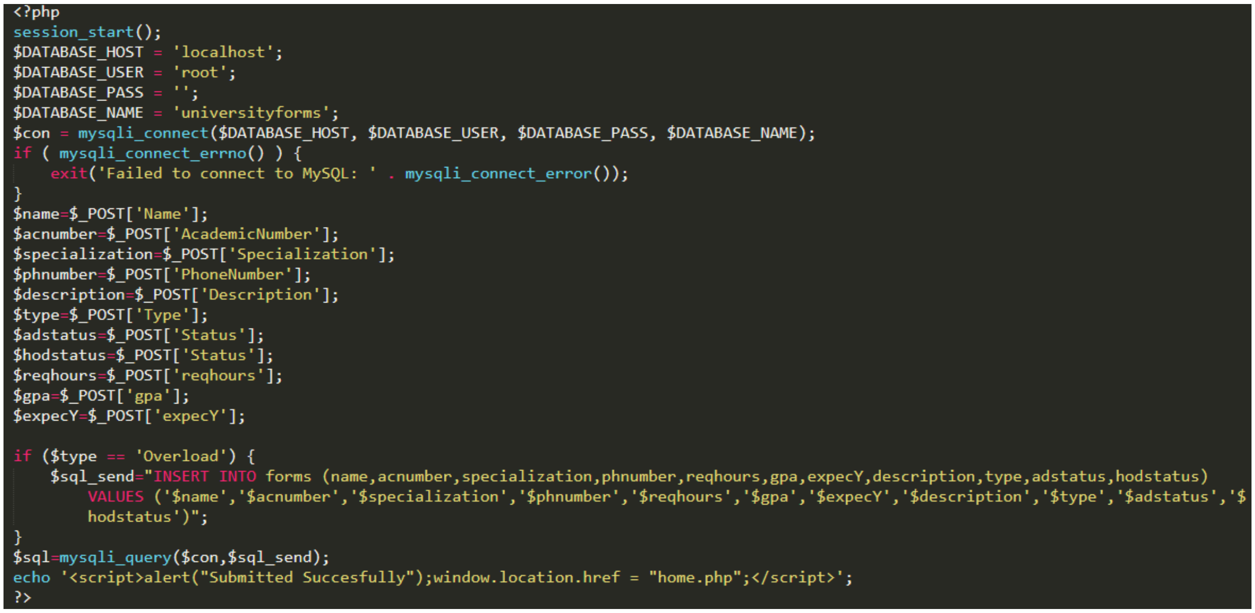


```
<div class="content">
    <h2>Approval Of Overloading Study-Hours, <?=$Name?></h2>
    <p>Fill Out The Info Below: </p>
    <div class="uni-forms">
    <form action="../formsend.php" method="post">
        <input type="text" name="Name" value="<?=$Name?>" readonly required><br>
        <input type="text" name="AcademicNumber" value="<?=$username?>" readonly required><br>
        <input type="text" name="Specialization" value="<?=$specialization?>" readonly required><br>
        <input type="text" name="PhoneNumber" value="<?=$phnumber?>" readonly required><br>
        <input type="text" name="reqhours" placeholder="Requested Study-Hours" class="clear" required><br>
        <input type="text" name="gpa" placeholder="Your Current GPA" class="clear" required><br>
        <input type="text" name="expecY" placeholder="Expected Graduation Year And Semester" class="clear" required><
        br>
        <textarea rows = "10" cols = "100" name = "Description" placeholder="Reason..." class="clear"></textarea><br>
        <input type="text" name="Type" value="Overload" readonly required hidden><br>
        <input type="text" name="Status" value="Pending" readonly required hidden><br>
        <input type="submit" value="Submit">&nbsp;
        <button type="button" onclick="Clear()">Clear Form Text</button>
    </form>
    <script type="text/javascript">
        function Clear()
        {
            var fields = document.getElementsByClassName("clear");
            for(var i=0; i<fields.length; i++)
            {
                fields[i].value="";
            }
        }
        </script>
</div>
```

Figure (14): Implementation (Form Applying Design & Code)

6.3.1 - Implementing the Form Sending Method

Now after filling out the previous form and clicking submit there is a file called formsend.php that will handle the sending of the form as demonstrated next:



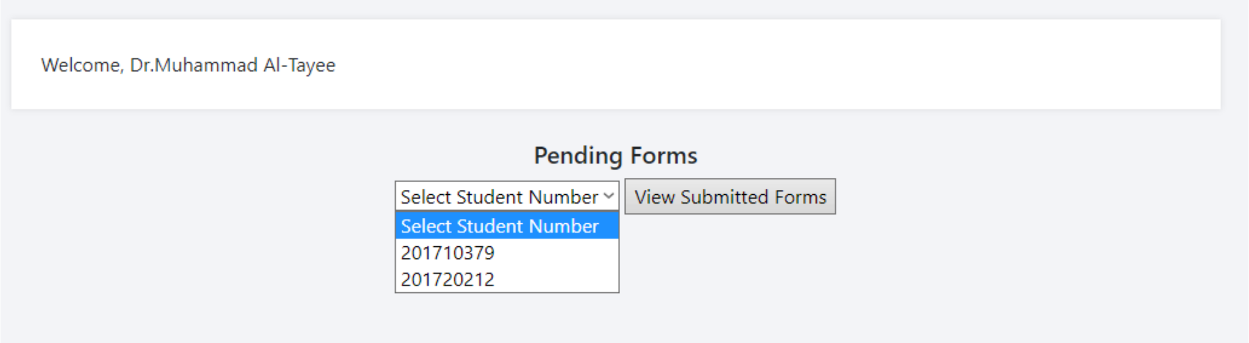
```php
<?php
session_start();
$DATABASE_HOST = 'localhost';
$DATABASE_USER = 'root';
$DATABASE_PASS = '';
$DATABASE_NAME = 'universityforms';
$con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
if ( mysqli_connect_errno() ) {
    exit('Failed to connect to MySQL: ' . mysqli_connect_error());
}
$name=$_POST['Name'];
$acnumber=$_POST['AcademicNumber'];
$specialization=$_POST['Specialization'];
$phnumber=$_POST['PhoneNumber'];
$description=$_POST['Description'];
$type=$_POST['Type'];
$adstatus=$_POST['Status'];
$hodstatus=$_POST['Status'];
$reqhours=$_POST['reqhours'];
$gpa=$_POST['gpa'];
$expecY=$_POST['expecY'];

if ($type == 'Overload') {
    $sql_send="INSERT INTO forms (name,acnumber,specialization,phnumber,reqhours,gpa,expecY,description,type,adstatus,hodstatus)
        VALUES ('$name','$acnumber','$specialization','$phnumber','$reqhours','$gpa','$expecY','$description','$type','$adstatus','$
        hodstatus')";
}
$sql=mysqli_query($con,$sql_send);
echo '<script>alert("Submitted Succesfully");window.location.href = "home.php";</script>';
?>
```

Figure (15): Implementation (Form Send Code)

6.4 - Implementing the Admin Portal Home Page

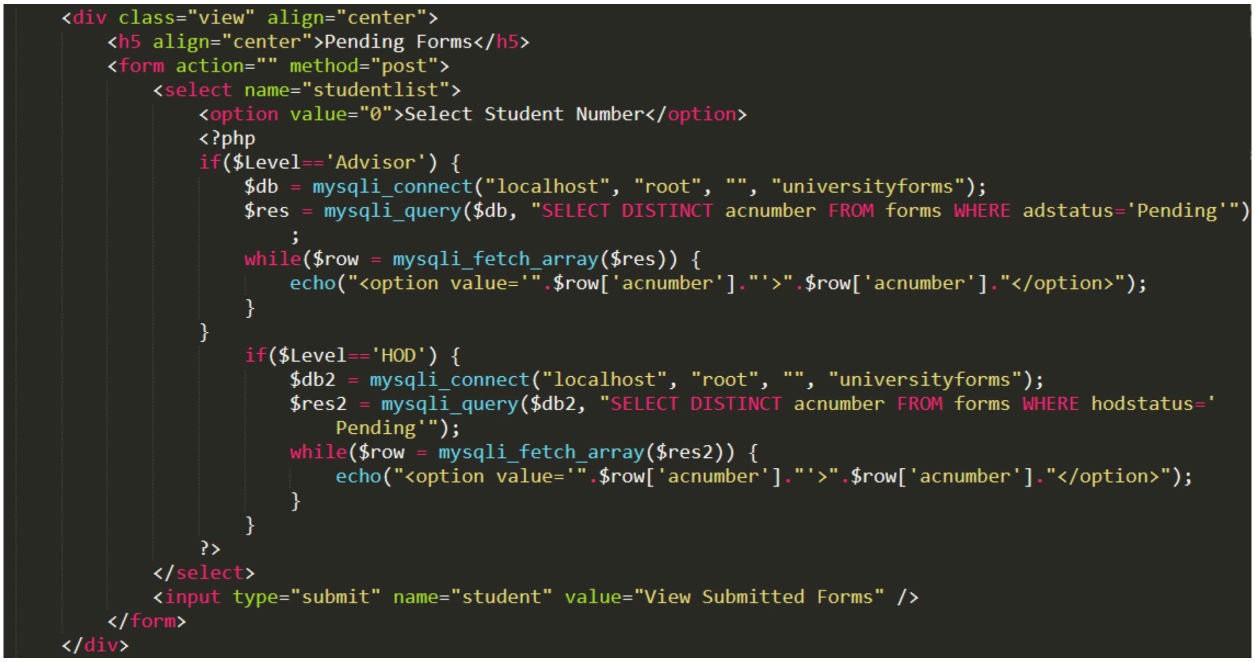
Next the Admin Portal is where these forms are going to show up so we need to list them in a table after the administrative chooses the academic number of the student from a dropdown-list to see the applied forms of the student.

The List:



Welcome, Dr.Muhammad Al-Tayee

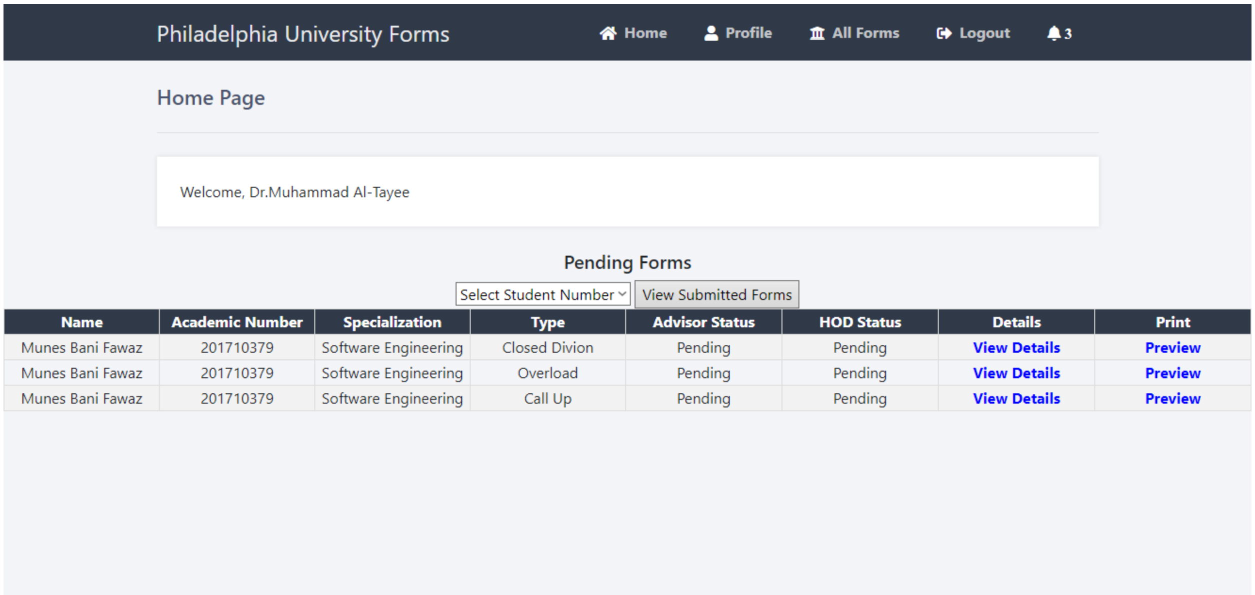
Pending Forms



```
<div class="view" align="center">
    <h5 align="center">Pending Forms</h5>
    <form action="" method="post">
        <select name="studentlist">
            <option value="0">Select Student Number</option>
            <?php
            if($Level=='Advisor') {
                $db = mysqli_connect("localhost", "root", "", "universityforms");
                $res = mysqli_query($db, "SELECT DISTINCT acnumber FROM forms WHERE adstatus='Pending'")
                    ;
                while($row = mysqli_fetch_array($res)) {
                    echo("<option value='".$row['acnumber']."'>".$row['acnumber']."</option>");
                }
            }
                if($Level=='HOD') {
                    $db2 = mysqli_connect("localhost", "root", "", "universityforms");
                    $res2 = mysqli_query($db2, "SELECT DISTINCT acnumber FROM forms WHERE hodstatus='
                        Pending'");
                    while($row = mysqli_fetch_array($res2)) {
                        echo("<option value='".$row['acnumber']."'>".$row['acnumber']."</option>");
                    }
                }
            ?>
        </select>
        <input type="submit" name="student" value="View Submitted Forms" />
    </form>
</div>
```

Figure (16): Implementation (Admin Portal Student List Design & Code)

The table after clicking on a student number:

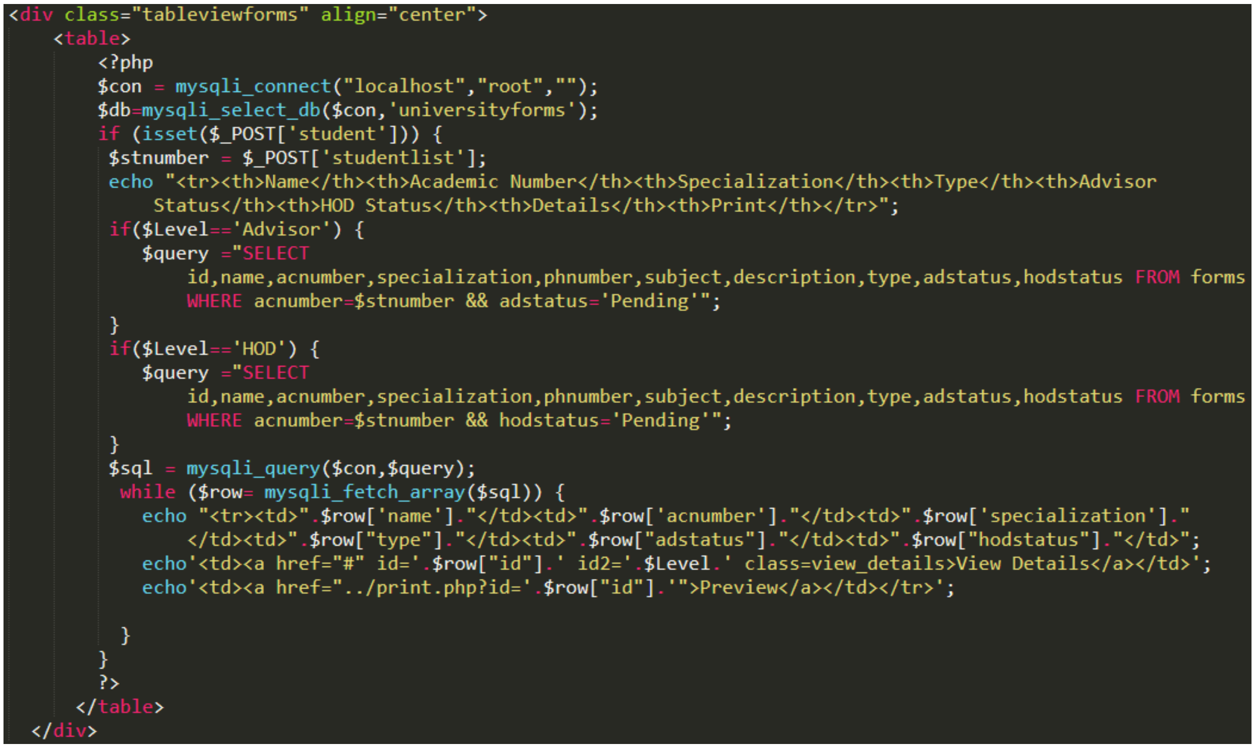


Home Page

Welcome, Dr.Muhammad Al-Tayee

Pending Forms

Name	Academic Number	Specialization	Type	Advisor Status	HOD Status	Details	Print
Munes Bani Fawaz	201710379	Software Engineering	Closed Divion	Pending	Pending	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Overload	Pending	Pending	View Details	Preview
Munes Bani Fawaz	201710379	Software Engineering	Call Up	Pending	Pending	View Details	Preview



```
<div class="tableviewforms" align="center">
    <table>
        <?php
        $con = mysqli_connect("localhost","root","");
        $db=mysqli_select_db($con,'universityforms');
        if (isset($_POST['student'])) {
         $stnumber = $_POST['studentlist'];
         echo "<tr><th>Name</th><th>Academic Number</th><th>Specialization</th><th>Type</th><th>Advisor
             Status</th><th>HOD Status</th><th>Details</th><th>Print</th></tr>";
         if($Level=='Advisor') {
            $query ="SELECT
                id,name,acnumber,specialization,phnumber,subject,description,type,adstatus,hodstatus FROM forms
                WHERE acnumber=$stnumber && adstatus='Pending'";
         }
         if($Level=='HOD') {
            $query ="SELECT
                id,name,acnumber,specialization,phnumber,subject,description,type,adstatus,hodstatus FROM forms
                WHERE acnumber=$stnumber && hodstatus='Pending'";
         }
         $sql = mysqli_query($con,$query);
          while ($row= mysqli_fetch_array($sql)) {
            echo "<tr><td>".$row['name']."</td><td>".$row['acnumber']."</td><td>".$row['specialization']."
                </td><td>".$row["type"]."</td><td>".$row["adstatus"]."</td><td>".$row["hodstatus"]."</td>";
            echo'<td><a href="#" id='.$row["id"].' id2='.$Level.' class=view_details>View Details</a></td>';
            echo'<td><a href="../print.php?id='.$row["id"].'">Preview</a></td></tr>';

          }
        }
        ?>
      </table>
  </div>
```

Figure (17): Implementation (Admin Portal Table Design & Code)

6.5 - Implementing the Admin Approval / Denial Options

Now after reviewing a form the administrate needs either to accept or deny the form and this small fragment of code will be responsible for that (The next are the advisor tools as shown in the deny phase if the advisor denies the form will be automatically denied by the head of department) as shown in the next steps:

Step 1: Choosing Approve or Deny

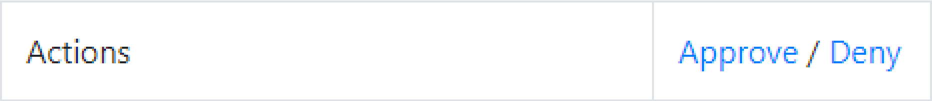
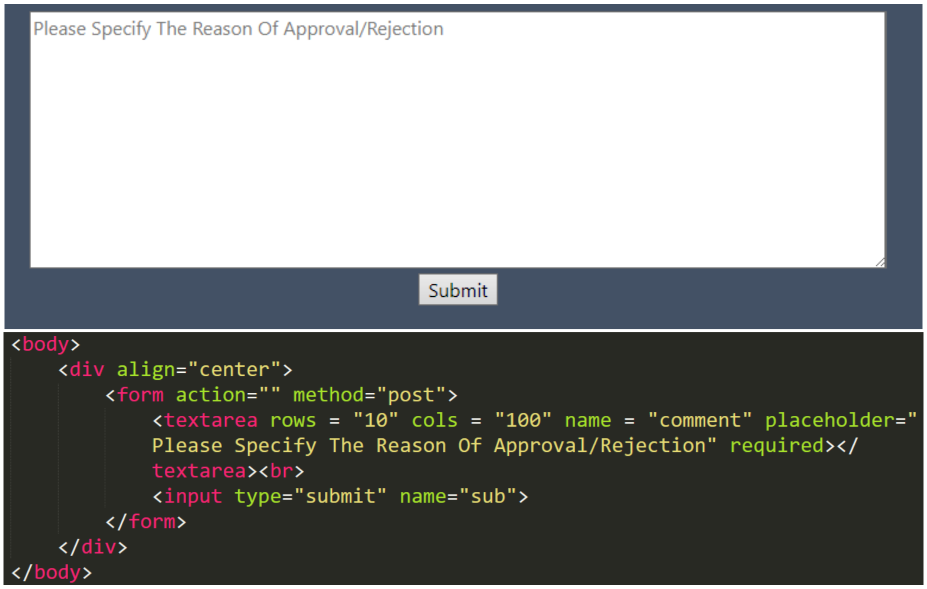


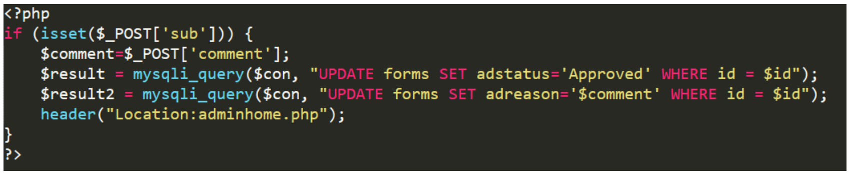
Figure (18): Implementation (Admin Approval / Denial Tools Design)

Step 2: Specify Reason of Approval or Rejection and clicking submit



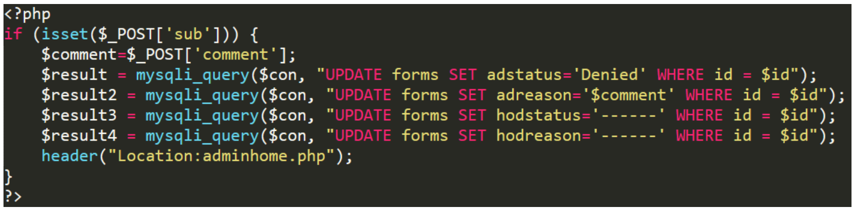
```
<body>
    <div align="center">
        <form action="" method="post">
            <textarea rows = "10" cols = "100" name = "comment" placeholder="
            Please Specify The Reason Of Approval/Rejection" required></
            textarea><br>
            <input type="submit" name="sub">
        </form>
    </div>
</body>
```

Figure (19): Implementation (Admin Approval / Denial Design & Design)



```
<?php
if (isset($_POST['sub'])) {
    $comment=$_POST['comment'];
    $result = mysqli_query($con, "UPDATE forms SET adstatus='Approved' WHERE id = $id");
    $result2 = mysqli_query($con, "UPDATE forms SET adreason='$comment' WHERE id = $id");
    header("Location:adminhome.php");
}
?>
```

Figure (20): Implementation (Admin Approve Code)

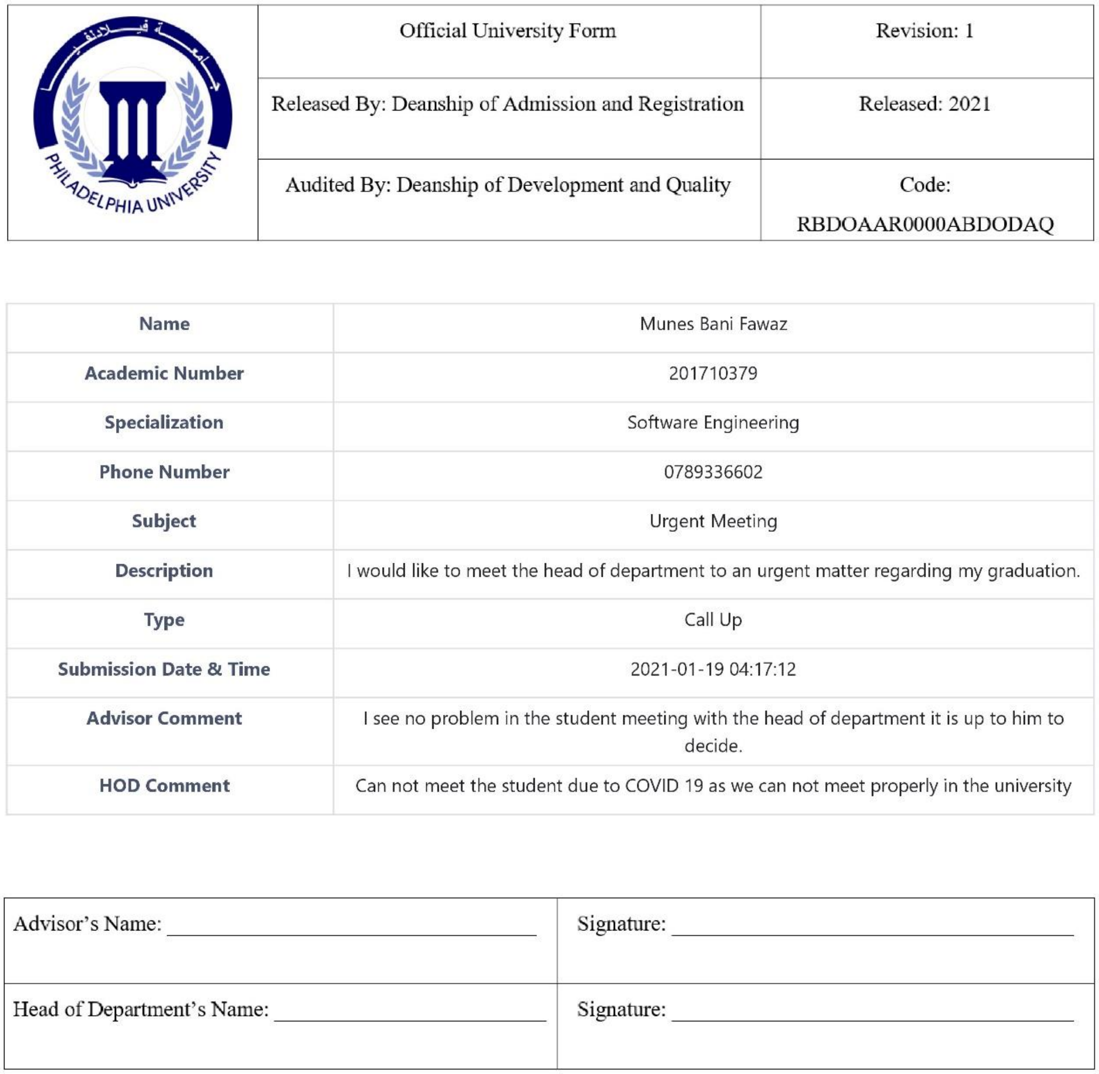


```
<?php
if (isset($_POST['sub'])) {
    $comment=$_POST['comment'];
    $result = mysqli_query($con, "UPDATE forms SET adstatus='Denied' WHERE id = $id");
    $result2 = mysqli_query($con, "UPDATE forms SET adreason='$comment' WHERE id = $id");
    $result3 = mysqli_query($con, "UPDATE forms SET hodstatus='------' WHERE id = $id");
    $result4 = mysqli_query($con, "UPDATE forms SET hodreason='------' WHERE id = $id");
    header("Location:adminhome.php");
}
?>
```

Figure (21): Implementation (Admin Deny Code)

6.6 - Implementing the Print Function

Another important function is to have the ability to print the form in a university standard as demonstrated below followed with the code on the next page:



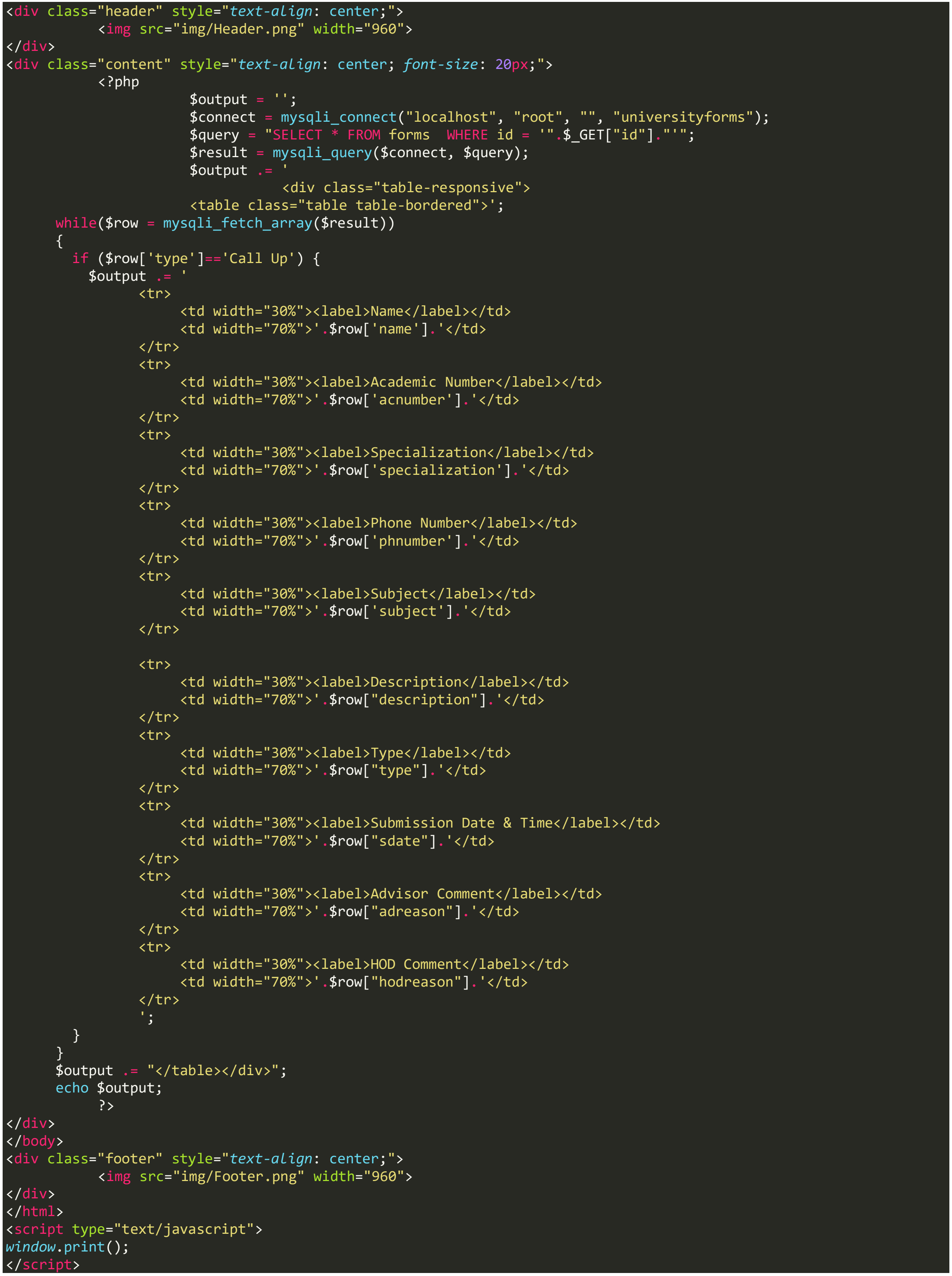
	Official University Form	Revision: 1
	Released By: Deanship of Admission and Registration	Released: 2021
	Audited By: Deanship of Development and Quality	Code: RBDOAAR0000ABDODAQ

Name	Munes Bani Fawaz
Academic Number	201710379
Specialization	Software Engineering
Phone Number	0789336602
Subject	Urgent Meeting
Description	I would like to meet the head of department to an urgent matter regarding my graduation.
Type	Call Up
Submission Date & Time	2021-01-19 04:17:12
Advisor Comment	I see no problem in the student meeting with the head of department it is up to him to decide.
HOD Comment	Can not meet the student due to COVID 19 as we can not meet properly in the university

Advisor's Name: ______________________________	Signature: ________________________________
Head of Department's Name: ______________________	Signature: ________________________________

Figure (22): Implementation (Print Design)

6.6.1 - Implementing the Print Function



```
<div class="header" style="text-align: center;">
           <img src="img/Header.png" width="960">
</div>
<div class="content" style="text-align: center; font-size: 20px;">
           <?php
                      $output = '';
                      $connect = mysqli_connect("localhost", "root", "", "universityforms");
                      $query = "SELECT * FROM forms  WHERE id = '".$_GET["id"]."'";
                      $result = mysqli_query($connect, $query);
                      $output .= '
                                 <div class="table-responsive">
                      <table class="table table-bordered">';
      while($row = mysqli_fetch_array($result))
      {
        if ($row['type']=='Call Up') {
          $output .= '
                <tr>
                     <td width="30%"><label>Name</label></td>
                     <td width="70%">'.$row['name'].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Academic Number</label></td>
                     <td width="70%">'.$row['acnumber'].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Specialization</label></td>
                     <td width="70%">'.$row['specialization'].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Phone Number</label></td>
                     <td width="70%">'.$row['phnumber'].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Subject</label></td>
                     <td width="70%">'.$row['subject'].'</td>
                </tr>

                <tr>
                     <td width="30%"><label>Description</label></td>
                     <td width="70%">'.$row["description"].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Type</label></td>
                     <td width="70%">'.$row["type"].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Submission Date & Time</label></td>
                     <td width="70%">'.$row["sdate"].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>Advisor Comment</label></td>
                     <td width="70%">'.$row["adreason"].'</td>
                </tr>
                <tr>
                     <td width="30%"><label>HOD Comment</label></td>
                     <td width="70%">'.$row["hodreason"].'</td>
                </tr>
                ';
        }
      }
      $output .= "</table></div>";
      echo $output;
           ?>
</div>
</body>
<div class="footer" style="text-align: center;">
           <img src="img/Footer.png" width="960">
</div>
</html>
<script type="text/javascript">
window.print();
</script>
```

Chapter 7 - System Testing

7.1 - Introduction about System Testing

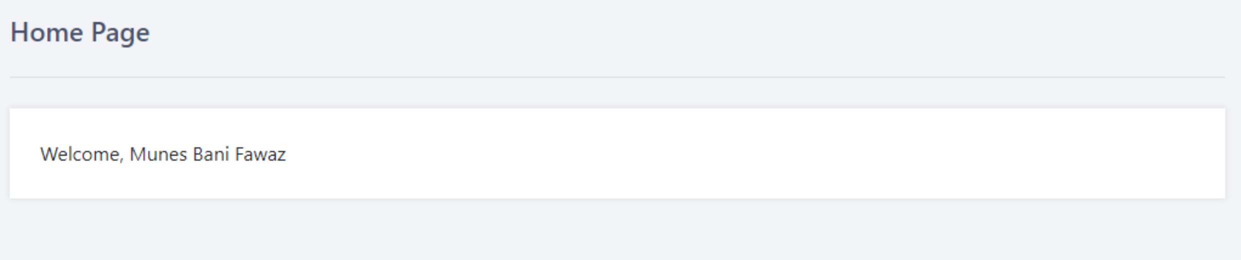
Software testing is nothing but an art of investigating software to ensure that its quality under test is in line with the requirement of the client. Software testing is carried out in a systematic manner with the intent of finding defects in a system.

7.2 - Black Box Testing

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. Black box testing is a powerful testing technique because it exercises a system end-to-end. In the next table we will see some tests performed on the website followed with the images of the expected outputs as written in the table.

Test #	Description	Input	Expected Output	Pass / Fail
1	Testing Login	Username: 201710379 Password: 201710379	Condition (Login Successful) Welcome, Munes Bani Fawaz Figure (23)	Passed
2	Testing Login	Username: 201710379 Password: 200010000	Incorrect username and / or password! Figure (24)	Passed
3	Call Up Form Filling / Sending	Subject: Random Text Description: Random Text	Submitted Successfully Figure (25)	Passed
4	Call Up Form Filling / Sending	Subject: NO TEXT Description: Random Text	Please Fill Out This Field Figure (26)	Passed
5	Administrative Form Approval	Reason: Random Text	Condition:(Redirect to home page and submit successfully) Figure (27)	Passed
6	Administrative Form Approval	Reason: NO TEXT	Please Fill Out This Field Figure (28)	Passed
7	Administrative Form Denial	Reason: Random Text	Condition:(Redirect to home page and submit successfully) Figure (29)	Passed
8	Administrative Form Denial	Reason: NO TEXT	Please Fill Out This Field Figure (30)	Passed

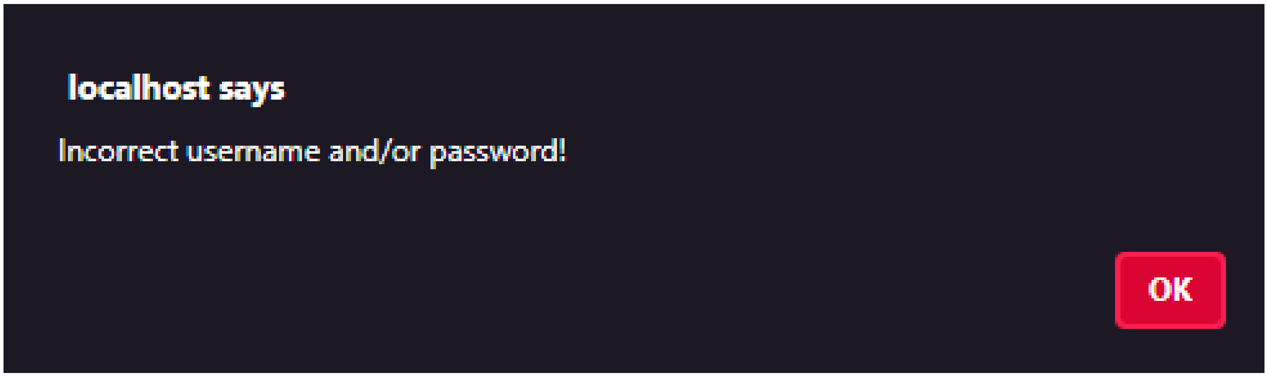
Table (2): Black Box (Testing)



Home Page

Welcome, Munes Bani Fawaz

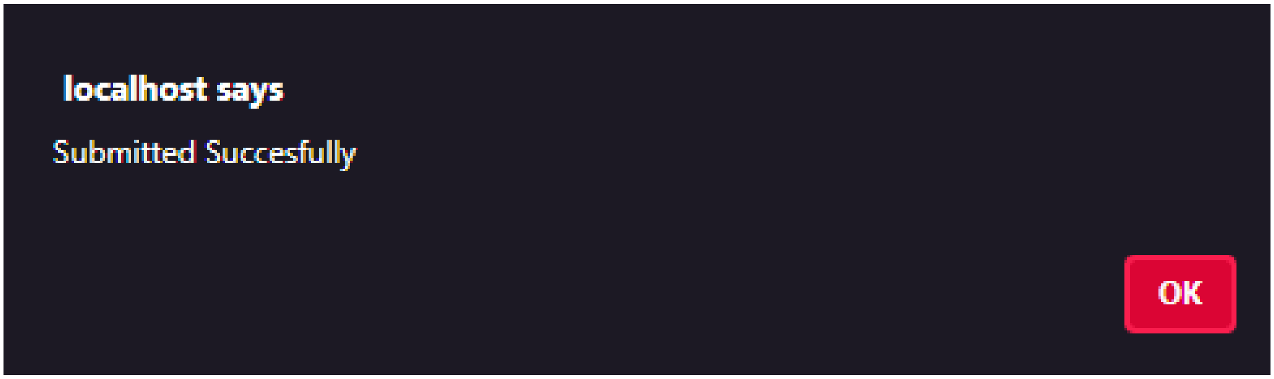
Figure (23): Black Box (Testing Case #1)



localhost says

Incorrect username and/or password!

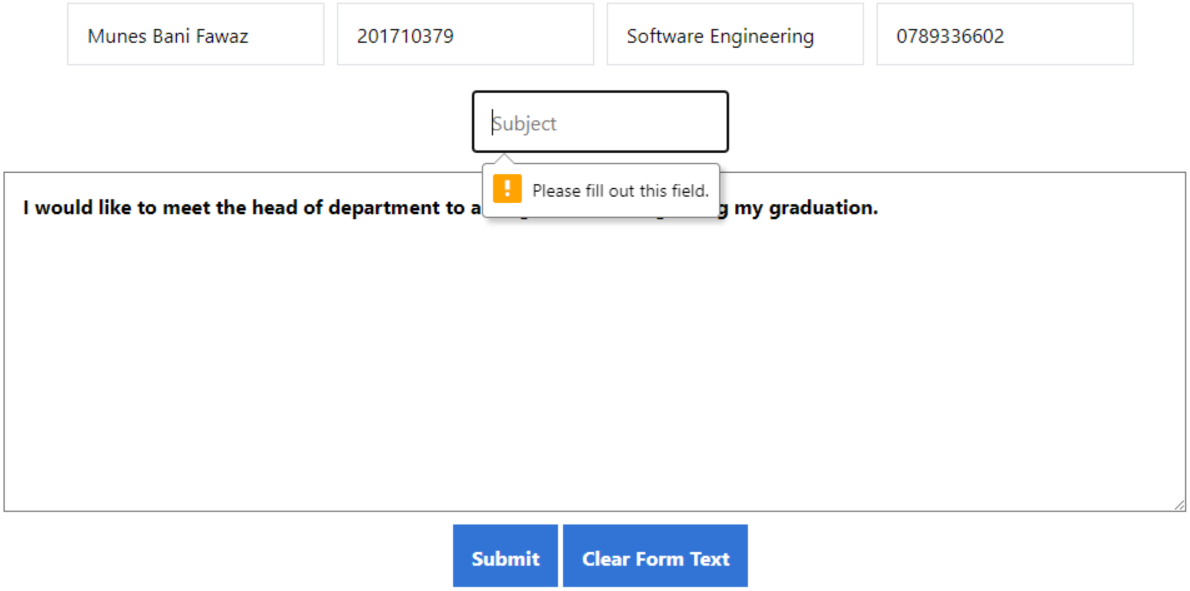
Figure (24): Black Box (Testing Case #2)



localhost says

Submitted Succesfully

Figure (25): Black Box (Testing Case #3)



Munes Bani Fawaz

201710379

Software Engineering

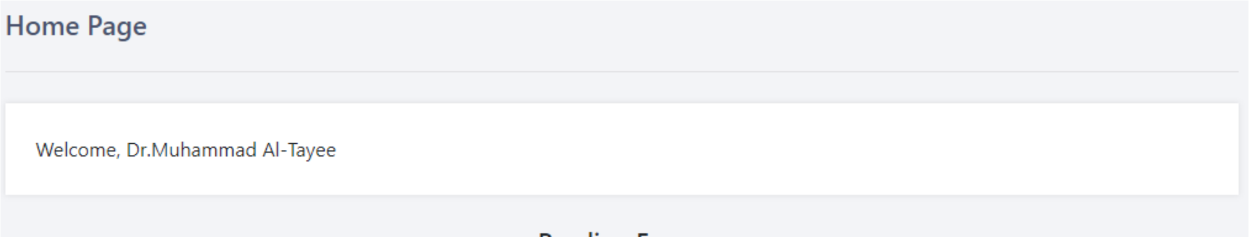
0789336602

Subject

Please fill out this field.

I would like to meet the head of department to a g my graduation.

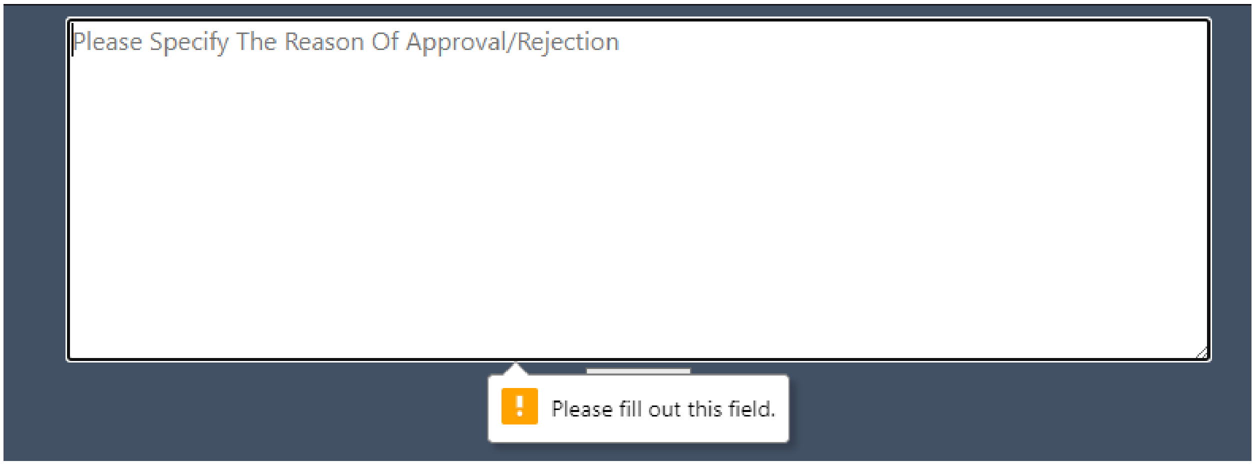
Figure (26): Black Box (Testing Case #4)



Home Page

Welcome, Dr.Muhammad Al-Tayee

Figure (27 & 29): Black Box (Testing Case #5 & #7)



Please Specify The Reason Of Approval/Rejection

Please fill out this field.

Figure (28 & 30): Black Box (Testing Case #6 & #8)

7.3 - Testing by Using

To perform this test, multiple usage scenarios were given and described. A set of testing data were written for each scenario and the data was matching the real-world examples. The operations were performed separately by different users on different machines. Following are samples of the testing data.

Operation	Test data
Apply for Callup Form	Subject: Urgent Meeting! Description: I would like to meet with the head of department due a problem in my study plan regarding my graduation.
Apply for Closed Division	Course Number: 732314124 Course Name: C# Course Division Number: 1
Apply for Overloading Study Hours	Requested Study Hours: 21 Current GPA: 91.5 Expected Graduation Year/Semester: 2021/First Semester Reason: Because I am a graduate.
Approving Forms	Approving Forms on Advisor and Head of Department Accounts.
Denying Forms	Denying Forms on Advisor and Head of Department Accounts.
Printing Forms	Printing any form in student portal or admin portal.

Table (3): Testing by Using (Samples)

7.3.1 - Testing by Using Results

Operation	Result
Apply for Callup Form	Applied Successfully
Apply for Closed Division	Applied Successfully
Apply for Overloading Study Hours	Applied Successfully
Approving Forms	Approved Successfully
Denying Forms	Denied Successfully
Printing Forms	Printed Successfully

Table (4): Testing by Using (Results)

7.4 - Check List of Software Testing

#	Question	Answer
1	Does it really work as expected?	Yes
2	Does it meet the user's requirements?	Yes
3	Is it what the users expect?	Yes
4	Do the users like it?	Yes
5	Is it compatible with our other systems?	Yes
6	How does it scale when more users are added?	N/A
7	Which areas need more work?	Adding more forms to the student portal
8	Is it ready for release?	As a beta yes, as a full release no
9	How well does it work?	Excellent
10	What does it mean to you that "it works"?	It met all requirement specifications and functions without errors
11	How do you know it works? What evidence do you have?	Testing done with excellent results
12	In what ways could it seem to work but still have something wrong?	Nothing
13	What might cause it to not to work well?	N/A

Table (5): Check List of Software Testing

Chapter 8 - Conclusion & Future work

8.1 - Conclusion

Philadelphia University Forms is an effective and useful website that could be applied to the current university website as an independent portal. It’s designed to help the university regarding students, advisors and head of departments to automate the form applying system and making it online instead of the need to come to the university to apply for a form. The website passed all the functional tests with a high level of reliability and robustness.

8.2 - Future work

More work could be done to the student and admin portals. The portals could be translated to Arabic or any other language that suits the university and serves its students. It can be implemented with the university system directly in order to give students more information when applying to a form like perquisites to topics etc. The admin portal can be upgraded with a tab that shows all the history of the student that applied a form in regards to his topic grades, warnings and more information. The Portals can also be upgraded to include a private chat system between the student and his advisor or head of department. Also, more forms can be added easily to the system as the coding was done separated for the forms so copying and pasting the same design with some alternations should suffice.

References

- Refsnes Data. “W3Schools.” Internet: https://www.w3schools.com/, 1998.
- Mark Otto, Jacob Thornton. “Bootstrap.” Internet: https://getbootstrap.com, August.19,2011.
- Brendan Eich. “JavaScript” Internet: https://www.javascript.com/, December.4,1995.
- John Resig, Brandon Aaron, Jörn Zaefferer. “jQuery.” Internet: https://jquery.com, August.26,2006.
- Jeff Atwood, Joel Spolsky. “Stack Overflow.” Internet: https://stackoverflow.com, September.15,2008.

Appendix

Project Submission

1. **Supervisor:** Dr. Muhammad Al-Tayee.
2. **Project Title:** University Forms.
3. **Goals and Objectives:** 1. Saving time 2. Lowering cost 3. Easy access 4. Reducing paperwork 5. Preventing loss of Data.
4. **Brief description of the project:** Our project is a Website called "University Forms" that gives the Students the advantage to apply for different university forms applications online so we save time and effort and to make the application online instead of needing physical interaction.
5. **References:**
    a. Refsnes Data. “W3Schools.” Internet: https://www.w3schools.com/, 1998 [Oct.28,2020].
    b. Mark Otto, Jacob Thornton. “Bootstrap.” Internet: https://getbootstrap.com, August.19,2011 [Oct.28,2020].
    c. Brendan Eich. “JavaScript” Internet: https://www.javascript.com/, December.4,1995 [Oct.26,2020].
6. **Project Requirements (Hardware & Software):** Any Windows-based device or Android/IOS based phone with an Internet Browser that supports PHP+HTML5 (Google Chrome recommended).
7. **Company or organization (If applicable):** No Company or Organization.
8. **Prerequisite:** 1. Understanding the project 2. Setting up the development requirement 3. Brainstorming main ideas. 4. Passing (90 Academic hours) + (Department Approval)
9. **Project Specialization:** Software Engineering.
10. **Time schedule:** Mainly working on Mondays and Wednesdays.

Munes Bani Fawaz - 201710379 Ibrahim Hammad - 201720212

Supervisor Signature: Date:

Note: This is completed by the supervisor, and submitted to the Graduation Project Committee