**Chapter I**

**Introduction**

* 1. **Project Context**

Technology has made our life easier today from the smallest things to Space, Deep Learning and even Artificial Intelligence (Machine Intelligence), from Automated water sprinkler to Automated Cars that mostly doesn’t need you to steer to drive you to your destination. Technology also help in quality education and effective learning in schools.

Students, teachers and heads must evaluate teachers in their teaching personality, teaching competence, classroom management and classroom presentation skills in school. This is done by Mater Dei Colleges’ Manual Evaluation System which require hard labour and very high to human error. The evaluation system works like this – the evaluator need to go to every room where there is class and get some of that class’ time to make an evaluation of the teacher who is teaching that specific subject. On the other side of evaluation, the Head to teacher evaluation, the head or dean must evaluate teacher under his/her. This is done by where every student and head(dean) fill up the evaluation paper with the scores they think is applicable to the teacher’s personality, teaching competence, classroom management and classroom presentation skills. This system is a hard labour for the evaluator and sometimes the students, teachers and heads in which their time is consume in evaluating the teacher and very prone to human error because of computing the scores that all the MDC’s evaluation form is not easy, more over we have a large student and faculty body now. Also, considering that the scores that everyone gives is a very confidential and can fire a teacher. Designing an Automated Teacher Evaluation System will help improve the accuracy in computing scores of the evaluation of the teachers and making it secure and very confidential yet very easy to use. It will save valuable time for both evaluators, students and teachers. Making an Automated Evaluation System is a standard in todays’ technology level.

By this, we, the researchers proposed the Mater Dei College Automated Teacher Evaluation System. The design and development for this system is standalone and can be view in different system.

Automated Teacher Evaluation System will help students and teachers to evaluate easy and not time consuming, at the same time evaluators have much easier job in evaluation and easier computing in the average grade of the evaluated teachers.

This system provides a much faster way of evaluation than the old way of evaluation we currently used today.

* 1. **Purpose and Description**

The purpose of this project is to develop an automated system for the current manual system. It develops the system to be user-friendly and very accurate and effective system for Mater Dei College. This project is focus on being fully automated where the current manual paper-based system will be improved a big time. This project will help solve the manual-based system problems. It will help the school’s evaluation easy to compute and manage while making it confidential. This system is believed to reduce time consumption and improving the accuracy on computing the average scores for the teacher’s evaluation. Through this, it will record the scores given from the evaluation participant accurately, automatically and confidentially in every evaluation day participants will attend throughout each semester.

* 1. **Objectives**

General Objective:

This project Automated Teacher Evaluation System aims to provide easy user interface that will help students, teachers and evaluator for faster, accurate and efficient way of evaluating teachers.

**Specific Objective:**

* To replace the current existing manual teacher evaluation system to a fully computerized and automated teacher evaluation system.
* To develop an application that will record the participants’ given rates to the teacher of their subjects in more efficient way.
* To develop a system that will use less paper.
* To make the application easy to use where even a non-technical user will use it very well.
* To reduce time, consume and effort in going to every room in MDC.
* To make a teacher to teacher evaluation form.
* To reduce human error occurred in computing each of the evaluation paper one by one from the huge student and faculty body in MDC.
* To display reports to which student if he/she didn’t evaluate a teacher yet.
* To provide accurate average grades in the evaluation.
* To provide backups in case of accident or loss of information.
* To provide confidentiality for the teacher evaluation grades.
  1. **Scope and Limitations**

**Scope**

The scope of this system is to develop an automated teacher evaluation system from the old inefficient manual evaluation system. In this project, students can give ratings to the teacher’s subject he/she is handling. Other teachers and the head/dean can also give grades to each fellow teacher with in his/her department. In this project, you can grade the teacher from 1 to 5 where worst to great. In addition, the grades that each student or teacher gave to the evaluation is confidential and is encrypted only the admin can view it. They can only evaluate the current semester. This system can be used in Desktop and Mobile Platform. This will display the teachers you have evaluated and will be/not yet evaluated. The admin can also view all the student who given/doesn’t yet their ratings, all the head who given/doesn’t yet evaluated and all the teachers that is being evaluated. The heads can also view students and teachers only to their department. Teachers can view their fellow teachers of the same department. Student can view the student to teacher evaluation form.

The Automated Evaluation System was only developed for giving teachers their ratings base on their personality, teaching competence, classroom management and classroom presentation skills. The student can easily rate the teacher base on that. In addition, the admin can view if a certain student, head and teacher who doesn’t yet give their evaluation ratings and can notified them. The questions in the Automated Evaluation System is based on the manual form. The admin can also determine the time frame/range for the evaluation. The system is password protected, only Mater Dei College student that is currently enrolled in that semester can only access their accounts, so is the heads and teachers that is currently working on Mater Dei College. The admin can access and view the ratings the students, teachers and head gave. The system is developed to the purpose of making it efficient, less time consuming while overcoming the manual system problems.

**SYSTEM FEATURES**

* **Time Saver -** This system will help evaluators, students and teachers to have an easy, fast and convenient evaluation.
* **Easy Listing** – This system can list all participants and their basic data that will allow for the current semester. The administrator can see those participants who are done submitting their evaluation and those who are not yet.
* **Ranking** – This system can list all the teachers with their average ratings and rank them by the ratings they’ve got. They will be rank from satisfactory to less satisfactory. The teachers are divided by their respective departments – College, SHS, HS and Elementary.
* **Printing of records** – This system can generate hard copies. The file that will be printed is based on the rankings and departments.
* **User-friendly** – This system is simple, clean, intuitive and reliable. Thus, it is easy to use.
* **Security** – This system has security measure in which only the administrator can view all the sensitive data of the evaluation. Other accounts such as student and teacher accounts are password protected but have a limited access.
* **Flexible Evaluation Time** – The administrator has the privilege to allow evaluation through a certain time frame.
* **Easy Computation** – This system can compute all the ratings given by the participants will be fast, accurate and efficient.

**Limitations**

This project can’t be access outside Mater Dei Colleges’ campus. Students who graduated or will stop cannot access their accounts and evaluate teachers. Student and teachers can’t evaluate the past semester and the coming semester. Teachers and dean/heads can’t evaluate other teachers that are not on the same department. You can’t rate more than 5 or less than 1. The admin can’t change the teacher in the subject or the student of that subject it belongs to the registrar. The heads can’t view and evaluate teachers that are not on his/her department. Students can’t view other than student to teacher evaluation form and his/her teacher whose which subject he/she was enrolled in. Student, heads and teachers can’t evaluate other things than a teacher. This system is not available online for security and confidential purposes. The Instrument update can’t be viewed if the evaluation has started before it.

**CHAPTER II**

**Review of Related Literature/Systems**

This chapter consists of review of literature related with influence of web-based teacher evaluation system on other schools and deals with multiple perspective. Then, it reviews the current issues in manual teacher evaluation and factors affecting adoption of web-based teacher evaluation system. The literature and studies cited in this chapter tackle the different concept, understanding, and ideas, generalization or conclusions and different development related to development of the teacher evaluation system from the past up to the present and which serves as the researchers guide in developing the project. Further it provides a critique review of miscellaneous studies related with other evaluation system. Here are some sample which is of palpability with this system:

**Cebu Institute of Technology University (CIT)** uses a web-based teacher evaluation system to evaluate teacher accurate and faster than the normal manual system. They used their computer lab every time the evaluation will happen. This is easy for both students, teachers and evaluator. By using this it saved a lot of time in rating a teacher and going to each of every room of the campus.

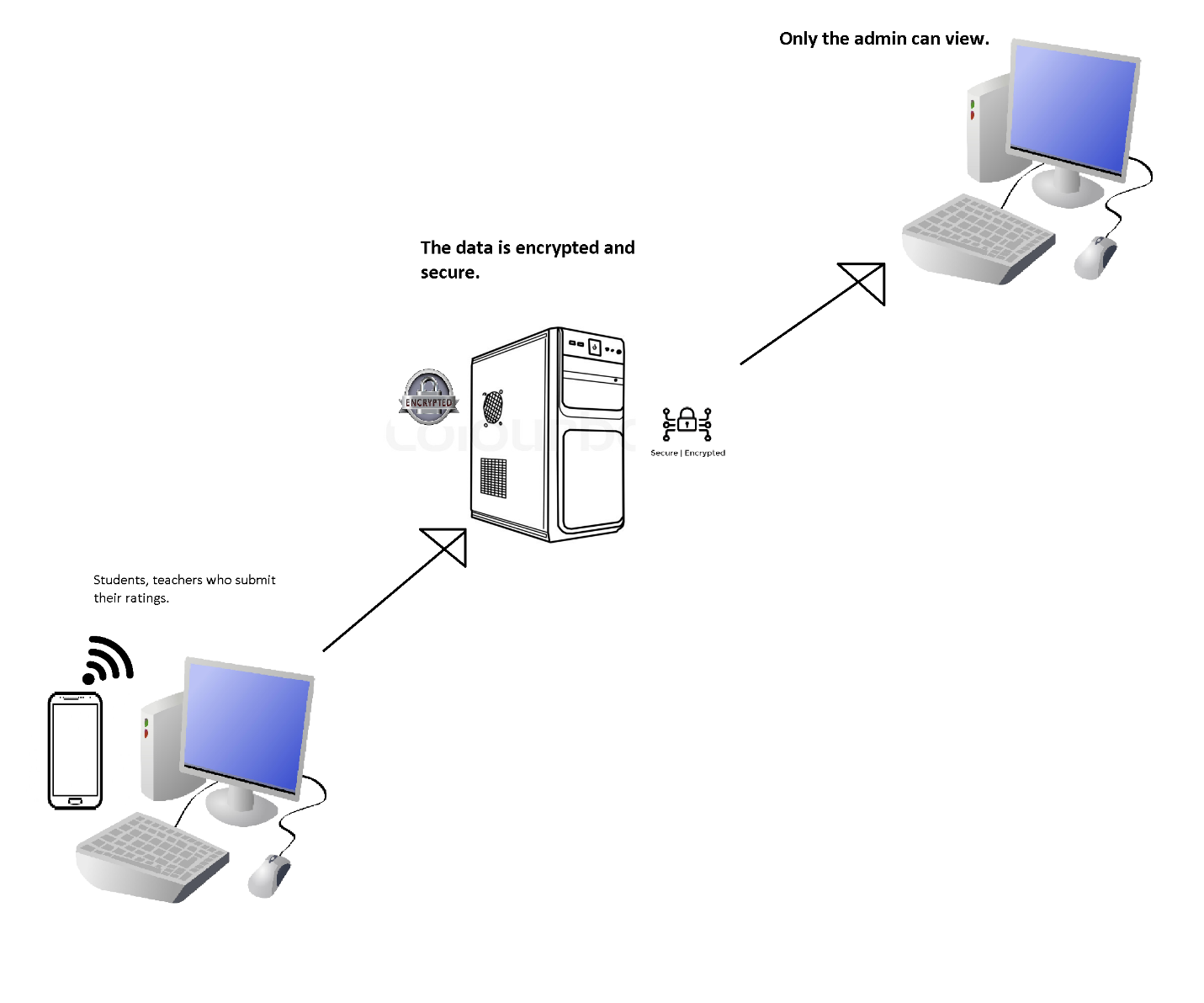
**AMA Computer Learning Center (ACLC)** also uses Web Based Teacher Evaluation System. The Method they used is a like of the SSG Election at MDC where each student will go to the Computer Lab and cast their votes, but in their case Teacher Evaluation rating. By this, it will only consume at least less than a day for a student to evaluate.

**Lazada Philippines** also uses Evaluation System, though it’s not used for evaluating Teachers but for the customers experience of the service of Lazada like the Website User Experience and the Flow of transaction of your Order. They will have sent you an email containing a letter from the COO of Lazada PH, with link to the webpage where you evaluate certain things for their further advancements for their service, website, Customer Service and how easy to use Lazada.

**CHAPTER III**

**TECHNICAL BACKGROUND**

In this chapter we will discuss what are the technical background and the tools that are used in making this project.

**3.1 Architecture Design**

This system will used Web Based Program to display, save and encrypt all the data that the students, teachers and evaluator/admin want. All the ratings that each subject submit to the system will be computed automatically and will be saved and show to the admin/evaluator. The data saved is encrypted and only the admin can view it in decrypted text. The students can only evaluate a teacher of the subject that he/she is enrolled currently in that semester.

After all data/rating from students and teachers the admin can view the average score of each of every teachers in MDC. The data is stored in the database but encrypted and can’t be seen there. Unless you are an admin only you can view those data.

**3.****2 Details of Tools and Technology Used**

The technology that will be used in this system as follow:

**XAMPP 1.7 or higher** - Developers will use this as development tool, to allow website programmer to test their work on their own computers without any access to the Internet.

**MySQL** - pronounced either "My S-Q-L" or "My Sequel," is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

MySQL will use by the researcher because it is one of the database that is compatible with the development of the system.

**PHP: Hypertext Pre-processor (or simply PHP)** is a server-side scripting language designed for web development but also used as a general-purpose programming language. It was originally created by Rasmus Leadoff in 1994, the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Pre-processor.

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

**Hypertext Mark-up Language (HTML)** is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript alongside.

**CSS stands for Cascading Style Sheets** - CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External style sheets are stored in CSS files.

**JavaScript** - jQuery is a Java Script library that allows web developers to add extra functionality to their websites. It is open source and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in web development.

The tools or program we (will) used to code:

**Microsoft Visual Code** - is a source code editor developed by Microsoft for Windows, Linux and macOS. It is a freeware to let the computer programmers to develop software and it is provided by the Microsoft. It allows the programmers to develop applications and websites in platforms such as Visual Basic, Visual C#, Visual C++, JavaScript, PHP etc. The Express Editions is more suitable for novice developers.

**Web Browsers -** A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web. Each individual web page, image, and video is identified by a distinct URL (Uniform Resource Locator or Web address like: https://www.wikipedia.org), enabling browsers to retrieve and display them on the user's device.

Here are some of the browsers:

1. Chrome Browser

2. Mozilla Firefox

3. Microsoft Edge

4. Opera Browser

There are a lot of other browsers but those four are some of them and are very popular.

Here is the hardware that will be used are as follow:

For laptops and/or desktop:

* Minimum 5 GB HDD/SSD space
* At least Intel Pentium and/or AMD Athlon/Ryzen 3 processor
* At least 2 GB RAM

For tablets and mobile

* Android OS 4.2.0 and above
* IOS 9 and/or above

**Chapter IV**

**Methodology**

4.1 Requirements Analysis and Requirements Documentation

In describing how the project is designed, the researcher used a use-case diagram to explain how the project functions.

Use-Case Diagram

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Manage Teacher User | ID:1 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| The Admin wants to manage the teacher’s user into the system. | | |
| Brief Description:  This use case describes how the admin manage or monitor the teacher’s user into the system. | | |
| Trigger: The Admin wants to manage the teacher users of the system. | | |
| Normal Flow of Events:   1. The Admin wants to manage the teacher users of the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the teacher user function. 7. The Admin manage the teacher’s user. 8. The system end. | | |

***Table 4.1.1 Manage Teacher User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Manage Student User | ID:2 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| The Admin wants to manage the student user into the system. | | |
| Brief Description:  This use case describes how the admin manage or monitor the teacher’s user into the system. | | |
| Trigger: The Admin wants to manage the student users of the system. | | |
| Normal Flow of Events:   1. The Admin wants to manage the student users of the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the student user function. 7. The Admin manage the student user. 8. The system end. | | |

***Table 4.1.2 Manage Student User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Add Student-User | ID:3 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| Admin wants to add a student into the system. | | |
| Brief Description:  This use case describes how to register a student-user into the system. | | |
| Trigger: The Admin wants to add a student-user into the system | | |
| Normal Flow of Events:   1. The Admin wants to add a student-user into the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the link to the student table. 7. The Admin will click a student. 8. The Admin will navigate to department list and select one department. 9. The Admin will click generate button. 10. The system ends. | | |

***Table 4.1.3 Add Student-User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Add Teacher-User | ID:4 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| Admin wants to register a teacher into the system. | | |
| Brief Description:  This use case describes how to register a teacher into the system. | | |
| Trigger: The Admin wants to add a teacher the system | | |
| Normal Flow of Events:   1. The Admin wants to add a teacher into the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the link to the teacher table. 7. The Admin will click a teacher. 8. The Admin will navigate to department list and select one department. 9. The Admin will click generate button. 10. The system ends. | | |

***Table 4.1.4 Add Teacher-User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Add Many Teacher-User | ID:5 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| Admin wants to register many teachers into the system. | | |
| Brief Description:  This use case describes how to register a teacher into the system. | | |
| Trigger: The Admin wants to add a teacher the system | | |
| Normal Flow of Events:   1. The Admin wants to add a batch of teacher-user into the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the link to the teacher table. 7. The Admin will click generate many buttons. 8. The system will show list of teachers that has no user yet. 9. The Admin will select teacher. 10. The Admin will choose their departments. 11. The Admin will click generate button. 12. If not successful 13. The system will show notifications what the error is. 14. Else, if successful   The system ends. | | |
|  | | |

***Table 4.1.5 Add Many Teacher-User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Add Many Student-User | ID:6 | Important Level:  High |
| Primary Actor: Admin | Use Case Type: Essential, Detail | |
| Admin wants to register many students into the system. | | |
| Brief Description:  This use case describes how to register a student into the system. | | |
| Trigger: The Admin wants to add a student the system | | |
| Normal Flow of Events:   1. The Admin wants to add a batch of student-user into the system. 2. The Admin starts the application and request to log in. 3. The Admin enters username & password to the system. 4. The system validates the admin username & password from database. 5. The system navigates to home window. 6. The Admin click the link to the student table. 7. The Admin will click generate many buttons. 8. The system will show list of students that has no user yet. 9. The Admin will select student. 10. The Admin will choose their departments. 11. The Admin will click generate button. 12. If not successful 13. The system will show notifications what the error is. 14. Else, if successful.   The system ends. | | |
|  | | |

***Table 4.1.6 Add Many Student-User***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Creating Evaluation Instrument | ID:7 | Important Level:  High |
| Primary Actor:  Admin | Use Case Type: Essential, Detail | |
| The Admin wants to create instrument. | | |
| Brief Description:  This use case describes how the Admin create Evaluation Instrument. | | |
| Trigger: The Admin wants to create Evaluation Instrument. | | |
| Normal Flow of Events:   1. The Admin starts the application. 2. The system navigates to log in window. 3. The Admin provides the username and password. 4. The system will verify the username and password from database and log in to the system. 5. The Admin will go to the Instrument table and clicks create function. 6. The system will show a form for creating instrument 7. The Admin will fill-up the form and save. 8. The system ends. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |

***Table 4.1.7 Creating Evaluation Instrument***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Admin Generates the Evaluation Form for Students | ID:8 | Important Level:  High |
| Primary Actor:  Admin | Use Case Type: Essential, Detail | |
| The Admin wants to generate student-teacher evaluation. | | |
| Brief Description:  This use case describes how the Admin generates the evaluation for the teacher from the students. | | |
| Trigger: The start of evaluating a teacher base on his/her subject. | | |
| Normal Flow of Events:   1. The Admin starts the application. 2. The system navigates to log in window. 3. The Admin click the teacher user function. 4. The Admin choose a teacher. 5. The system will show the teacher’s information. 6. The Admin will go to the subject table below and choose subjects on the table. 7. The Admin will choose the lists of instruments. 8. The Admin will click the evaluate button.   9. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |
|  | | |

***Table 4.1.8 Admin Generates the Evaluation Form for Students***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Admin Generates the Evaluation Form for Deans/Heads | ID:9 | Important Level:  High |
| Primary Actor:  Admin | Use Case Type: Essential, Detail | |
| The Admin wants to generate Dean-teacher evaluation. | | |
| Brief Description:  This use case describes how the Admin generates the evaluation for the teacher from the dean. | | |
| Trigger: The start of evaluating a teacher base on his/her department from their dean. | | |
| Normal Flow of Events:   1. The Admin starts the application. 2. The system navigates to log in window. 3. The Admin click the teacher user function. 4. The Admin choose a department table. 5. The system will show all the teachers in that department. 6. The Admin will click dean evaluation button.   7. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |
|  | | |

***Table 4.1.9 Admin Generates the Evaluation Form for Deans/Heads***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Printing Teachers Scores from the Evaluation | ID:10 | Important Level:  High |
| Primary Actor:  Admin | Use Case Type: Essential, Detail | |
| The Admin wants to generate hard copy of the teacher’s evaluation ranking. | | |
| Brief Description:  This use case describes how the Admin generate hard copy of the teacher’s evaluation ranking. | | |
| Trigger: The start of generating a hard copy of the evaluation result scores. | | |
| Normal Flow of Events:   1. The Admin starts the application. 2. The system navigates to log in window. 3. The Admin click the ranking function. 4. The Admin choose a department table. 5. The system will show all the teachers in that department. 6. The system will show the teachers average evaluation score sort by high to low. 7. The Admin will click PRINT button. 8. The system will show a printable PDF and ready to be printed.   9. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |
|  | | |

***Table 4.1.10 Printing Teachers Scores from the Evaluation***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  The Dean wants Evaluate teacher under his/her department | ID:11 | Important Level:  High |
| Primary Actor:  Dean | Use Case Type: Essential, Detail | |
| The dean wants to evaluate a teacher under his/her department. | | |
| Brief Description:  This use case describes how the Dean evaluate a teacher | | |
| Trigger: The start of generating a hard copy of the evaluation result scores. | | |
| Normal Flow of Events:   1. The Dean starts the application. 2. The system navigates to log in window. 3. The Dean will click the teacher name to evaluate. 4. The Dean will going evaluate a teacher under his/her department. 5. After the dean done putting score evaluate he/she going to submit it.   6. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |

***Table 4.1.11*** ***The Dean wants Evaluate teacher under his/her department***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  The Dean wants to see teacher ranking under his/her department | ID:12 | Important Level:  High |
| Primary Actor:  Dean | Use Case Type: Essential, Detail | |
| The Dean wanted to see a teacher's rank with a teacher under his department. | | |
| Brief Description:  This use case describes how the Dean to see teacher ranking | | |
| Trigger: The start of generating a hard copy of the evaluation result scores. | | |
| Normal Flow of Events:   1. The Dean starts the application. 2. The system navigates to log in window. 3. The Dean will click the teachers function. 4. After the dean click the teachers function it will going show the teacher racking. 5. After the dean done putting score evaluate he/she going to submit it.   6. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |

***Table 4.1.12 The Dean wanted to see a teacher's rank under his department***

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Student wanted to evaluate to teacher | ID:13 | Important Level:  High |
| Primary Actor:  Student | Use Case Type: Essential, Detail | |
| The Student wanted to evaluate a teacher | | |
| Brief Description:  This use case describes how the Student evaluate to teacher | | |
| Trigger: The start of generating a hard copy of the evaluation result scores. | | |
| Normal Flow of Events:   1. The Student starts the application. 2. The system navigates to log in window. 3. The Student will click the name of the teacher and he/she going to evaluate. 4. The Student will evaluate the teacher. 5. After the student evaluate the teacher just click the submit.   6. The system ends. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |

***Table 4.1.13 The Student wanted to evaluate a teacher***

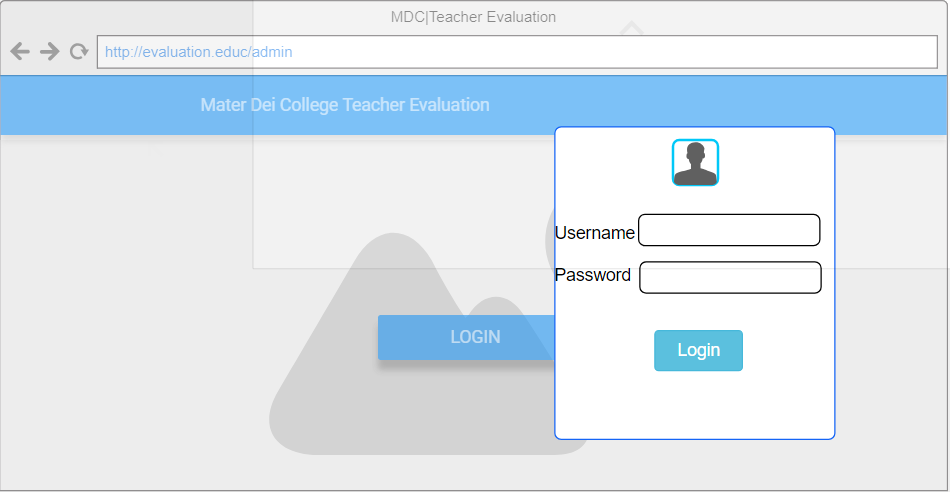
|  |  |  |
| --- | --- | --- |
| Use Case Name:  User wanted to change password | ID:14 | Important Level:  High |
| Primary Actor:  All User | Use Case Type: Essential, Detail | |
| The User wanted to change password | | |
| Brief Description:  This use case describes how the Student to change password | | |
| Trigger: The start of generating a hard copy of the evaluation result scores. | | |
| Normal Flow of Events:   1. The Student starts the application. 2. The system navigates to log in window. 3. The Student will click profile function and it will show the details and the change button. 4. The Student will click the change button and it will going to edit. 5. After the student edit the password just click the save button so that it will going to save.   6. The system end. | | |
| Sub Flows: | | |
| Alternative/Exceptional Flows: | | |

***Table 4.1.14 The Student wanted to change password***

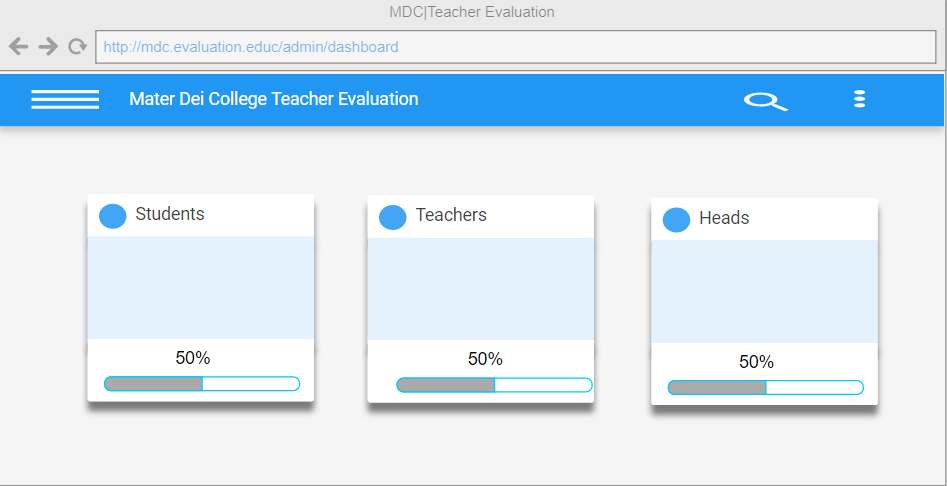
**4.2 Design of Software, Systems, Product, and Processes**

Researcher show a certain view of what to be expect in the system. It is useful for the researcher to guide them on how they develop the system. It is serves as the blue print of the system.

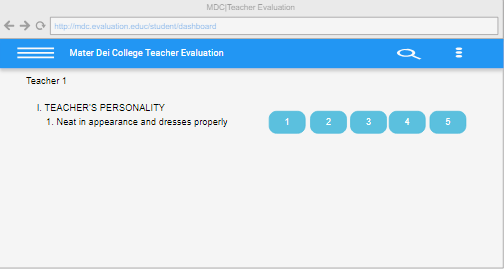
Output and User-Interface Design



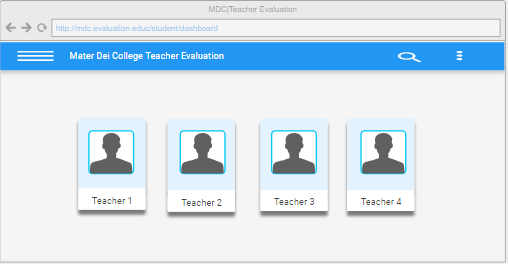
***4.2.1 Log In Interface***



***Figure 4.2.2 Admin Dash Board***

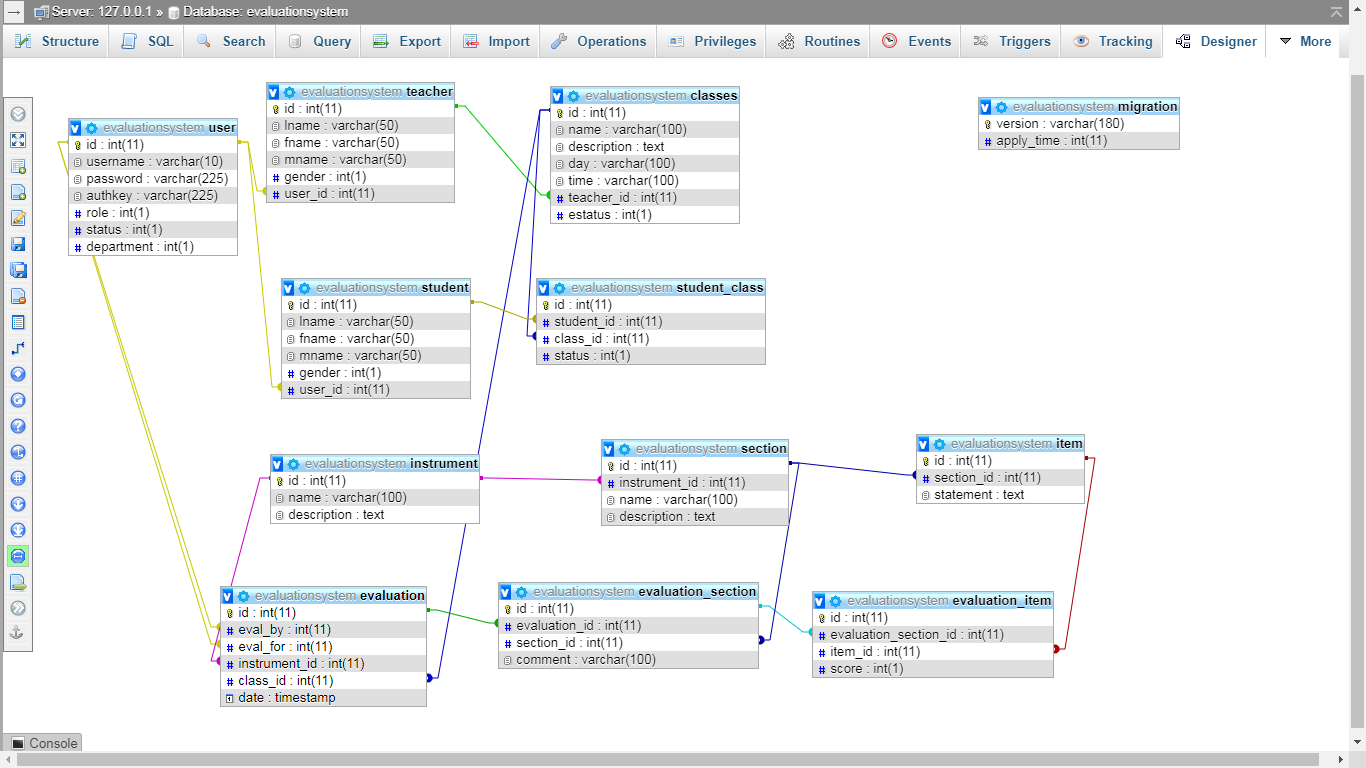
****

***Figure 4.2.3 Evaluation Form Interface***



***Figure 4.2.4 Dean Dashboard Interface***

**4.3 Entity Relationship Diagram**



**4.4 Data Dictionary**

To further understand the data being saved in the database, the data dictionary will help to show the fields of the tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Users Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID number of student  Primary Key |
| username | varchar | 10 | Name of the user |
| password | varchar | 225 | Password of the user |
| authkey | varchar | 225 | Authkey of the user |
| role | int | 1 | Role of the user |
| status | int | 1 | Status of the user |
| department | int | 1 | Department of user |

***Table 4.4.1 Database Table of User***

|  |  |  |  |
| --- | --- | --- | --- |
| **Teachers Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID number of teacher  Primary Key |
| lname | varchar | 50 | Last name of the teacher |
| fname | varchar | 50 | First name of the teacher |
| mname | varchar | 50 | Middle name of the teacher |
| gender | int | 1 | Gender of the teacher |
| User\_id | int | 11 | User ID of the teachers |

***Table 4.4.2 Database Table of Teacher***

|  |  |  |  |
| --- | --- | --- | --- |
| **Students Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the student  Primary Key |
| lname | varchar | 50 | Last name of the student |
| fname | varchar | 50 | First name of the student |
| mname | varchar | 50 | Middle name of the student |
| gender | int | 1 | Gender of the student |
| user\_id | int | 11 | User ID of the student |

***Table 4.4.3 Database Table of Student***

|  |  |  |  |
| --- | --- | --- | --- |
| **Classes Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the class  Primary Key |
| name | varchar | 100 | Name of the class |
| description | text |  | Description of the class |
| day | varchar | 100 | Day of the class |
| time | varchar | 100 | Time of the class |
| teacher\_id | int | 11 | Teacher ID of the class |

***Table 4.4.4 Database Table of Classes***

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_Class Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the student class  Primary Key |
| student\_id | int | 11 | Student ID of the student class |
| class\_id | int | 11 | class ID of the student |
| status | int | 1 | Status of the student |

***Table 4.4.5 Database Table of Student Classes***

|  |  |  |  |
| --- | --- | --- | --- |
| **Instrument Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the instrument  Primary Key |
| name | varchar | 100 | Name of the instrument |
| description | text |  | Description of the instrument |

***Table 4.4.6 Database Table of Instrument***

|  |  |  |  |
| --- | --- | --- | --- |
| **Section Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the section  Primary Key |
| Instrument\_id | int | 11 | Instrument ID of the sectio |
| name | varchar | 100 | class ID of the student |
| description | varchar | 225 | Status of the student |

***Table 4.4.7 Database Table of Section***

|  |  |  |  |
| --- | --- | --- | --- |
| **Item Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the item table |
| section\_id | int | 11 | Section ID of the table item |
| statement | varchar | 11 | Statement use in evaluation |

***Table 4.4.8 Database Table of Item***

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the Evaluation table |
| eval\_by | int | 225 | Evaluated by, field where the person was observed. |
| eval\_for | varchar | 225 | Evaluated for, field where the person was observed by |
| instrument\_id | int | 11 | Instrument Id for the table evaluation |
| class\_id | int | 11 | Class ID for the table evaluation |
| date | date |  | Date being generated the evaluation |
| status | int | 1 | The status of a certain evaluation |

***Table 4.4.9 Database Table of Evaluation***

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation\_Section Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the Evaluation\_Section table |
| evaluation\_id | int | 11 | Evaluated id, of the table evaluation\_section |
| section\_id | id | 11 | Section\_id of the table evaluation\_section |
| comment | varchar | 225 | Comment field in evaluation\_section |

***Table 4.4.9 Database Table of Evaluation\_Section***

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation\_Item Table** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| id | int | 11 | ID of the Evaluation\_Section table |
| evaluation\_section\_id | int | 11 | Evaluated id, of the table evaluation\_section |
| item\_id | int | 11 | Section\_id of the table evaluation\_section |
| score | int | 11 | Comment field in evaluation\_section |

***Table 4.4.10 Database Table of Evaluation\_Item***

**4.5**  **Development and Testing**

To test the system, it must be used by the respondents of the system to evaluate the system and to produce reliable information that makes the project more effective system.

**4.6** **Implementation Plan**

Implementation plan is to help the researcher to build a method on how the project will be implemented. It is very important in order to see the effects and importance of the proposed system from the present system.

**4.7** **Implementation Issues and Challenges**

System Implementation & Results

The system implementation and deployment is the part in which the system will be implemented and deployed into real life to be used. Before starting to develop the MDC Evaluation System, it must be installed and prepare well for the efficient usage of the system and development.

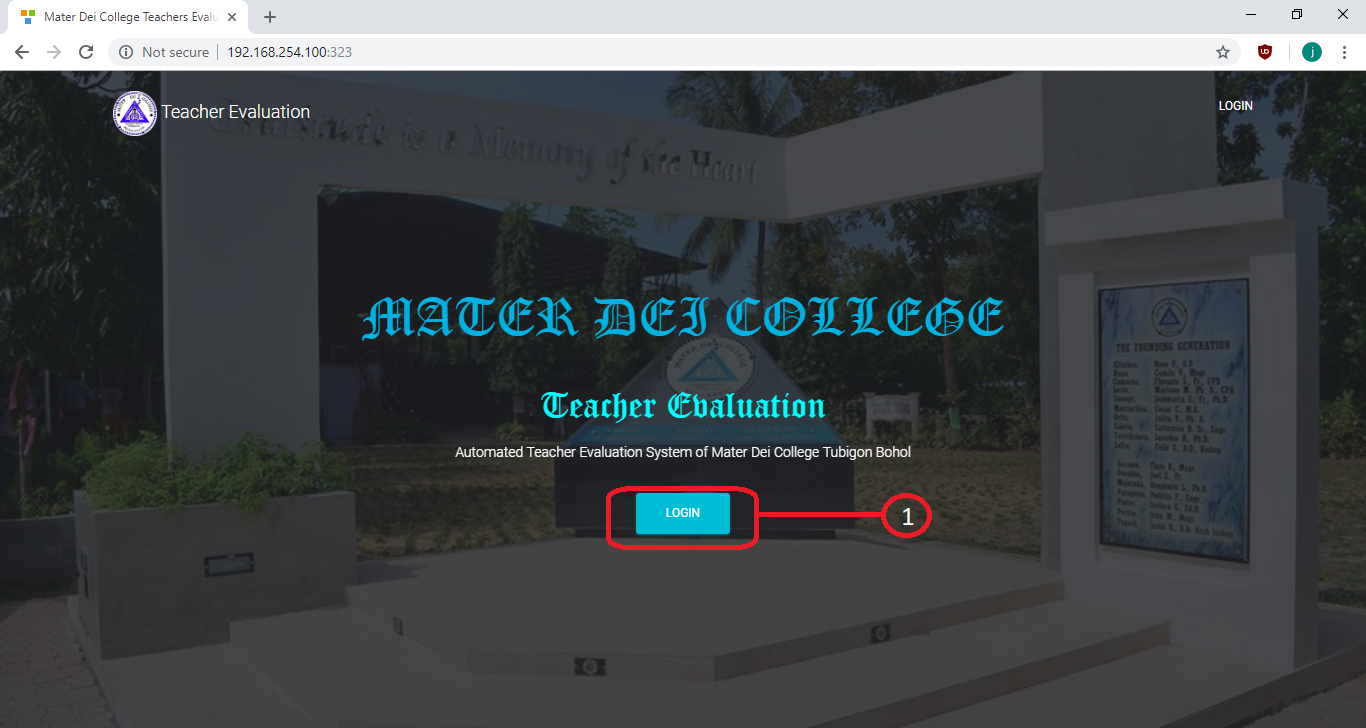
**Chapter V**

**Results and Discussions**

Chapter five shows the screenshots of the system. In this chapter, it will discuss what the screenshot is, and how the screenshot works. Chapter five also answers the objective of the project.

**User-guide**

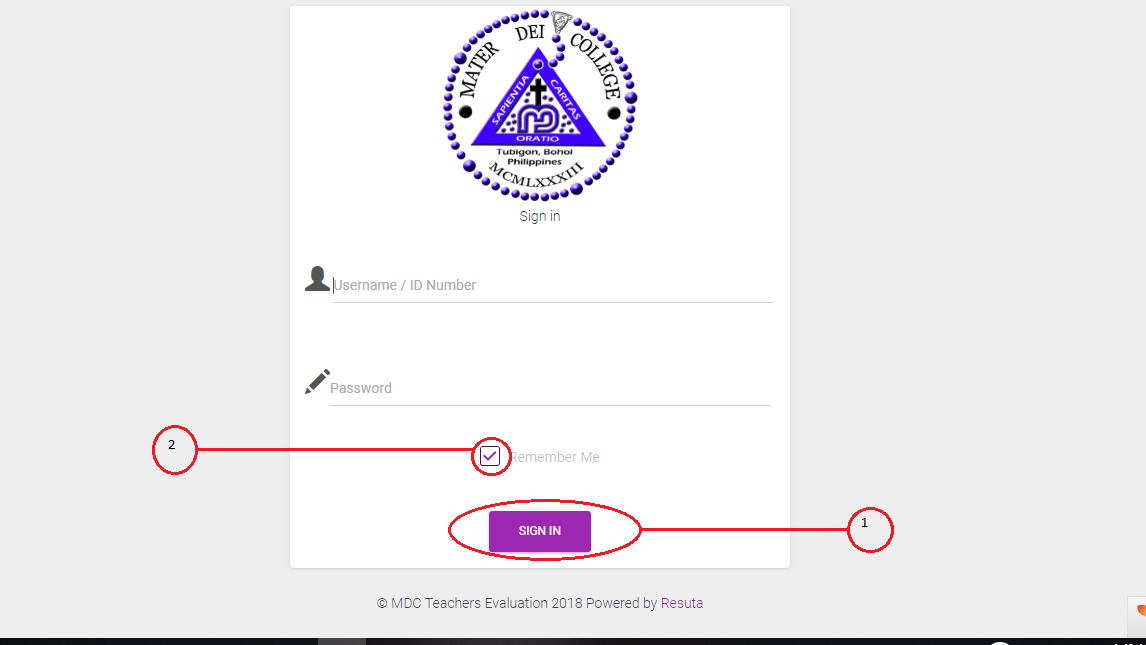
Use-Case Diagram



***Figure 5.1 Landing page Window***

Figure 5.1 is the interface of logging in the system. There are three types of user in the system, the admin, teacher and the student. It follows how the parts work:

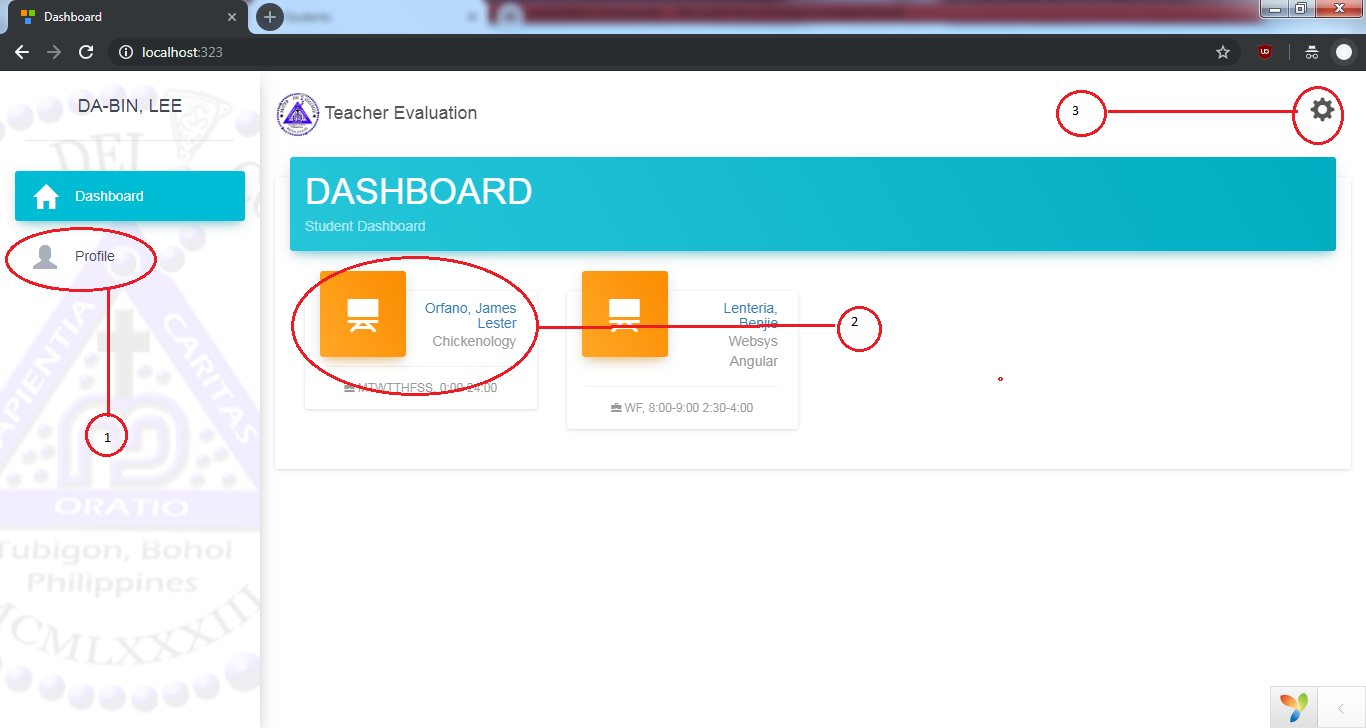
1.This button to show login page.

****

***Figure 5.2 Log in Window***

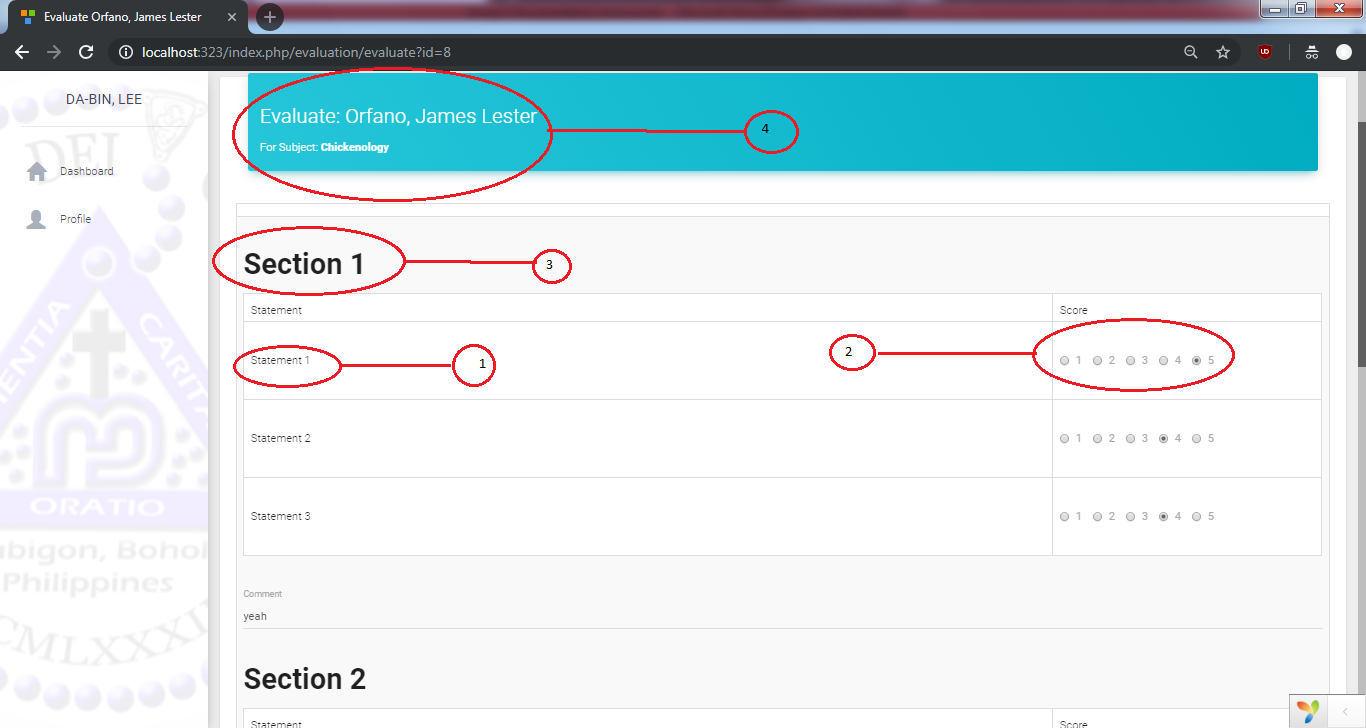
Figure 5.2 is the interface of logging in the system. There are three types of user in the system, the admin, teacher and the student. It follows how the parts work:

1. This button sign in into the system.
2. Once you click this check box your user name will still save

******

***Figure 5.3 Student and Teacher Home Window***

Figure 5.3 is the interface once a Student or Teacher log in the system. It follows how the parts work:

1. This link is show to your profile.
2. This button is where you can sign out the system.
3. This are the teachers, the admin choose to evaluate

***Figure 5.4 Student and Head Evaluation Form***

Figure 5.4 is the interface when you click/select a teacher in your home window. It follows how the parts work:

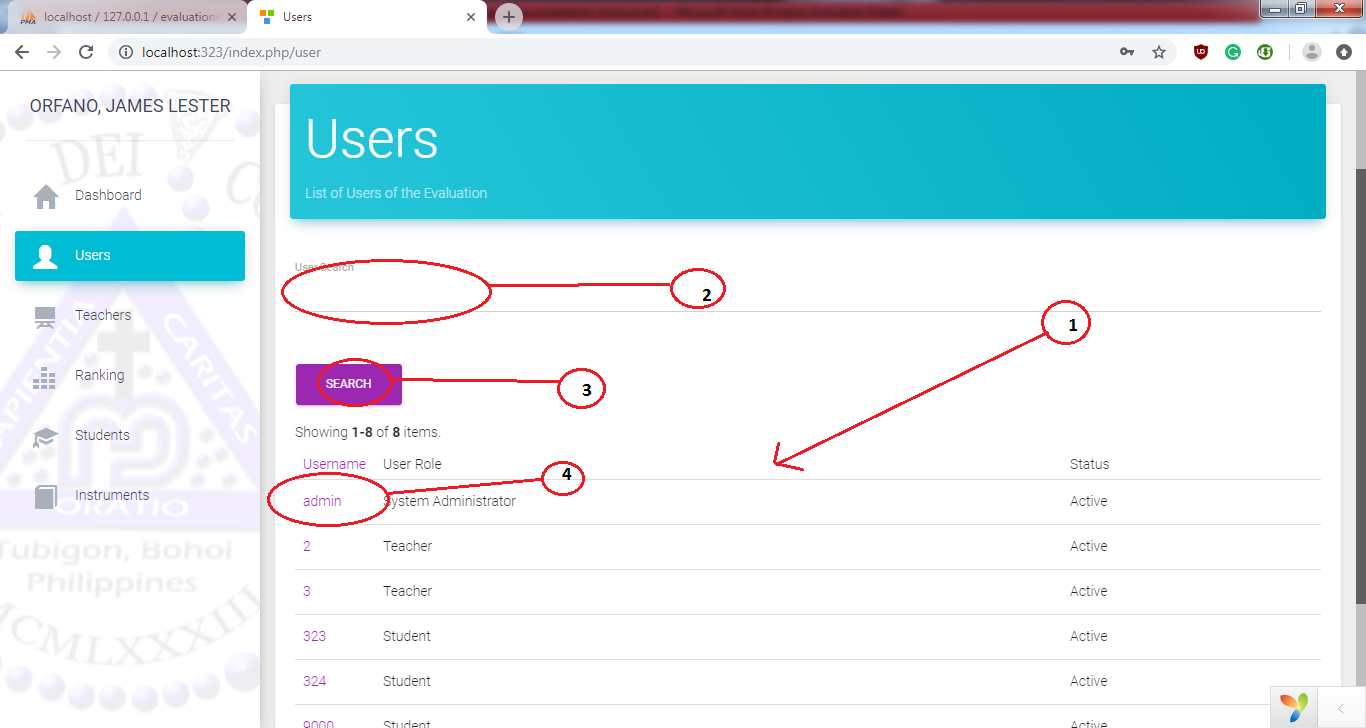
* 1. The instrument section statement.
  2. Selection of scores.
  3. The section.
  4. The teacher details.



***Figure 5.5 Admin Dashboard***

Figure 5.5 is the interface once the Admin log in the system. It follows how the parts work:

1. This will show the admin the list of Users.
2. This will show the admin the list of Teachers.
3. This will show the admin the list of Teacher Ranked.
4. This will show the admin the list of Students.
5. This will show the admin the list of Instruments.
6. This is the calendar and date.

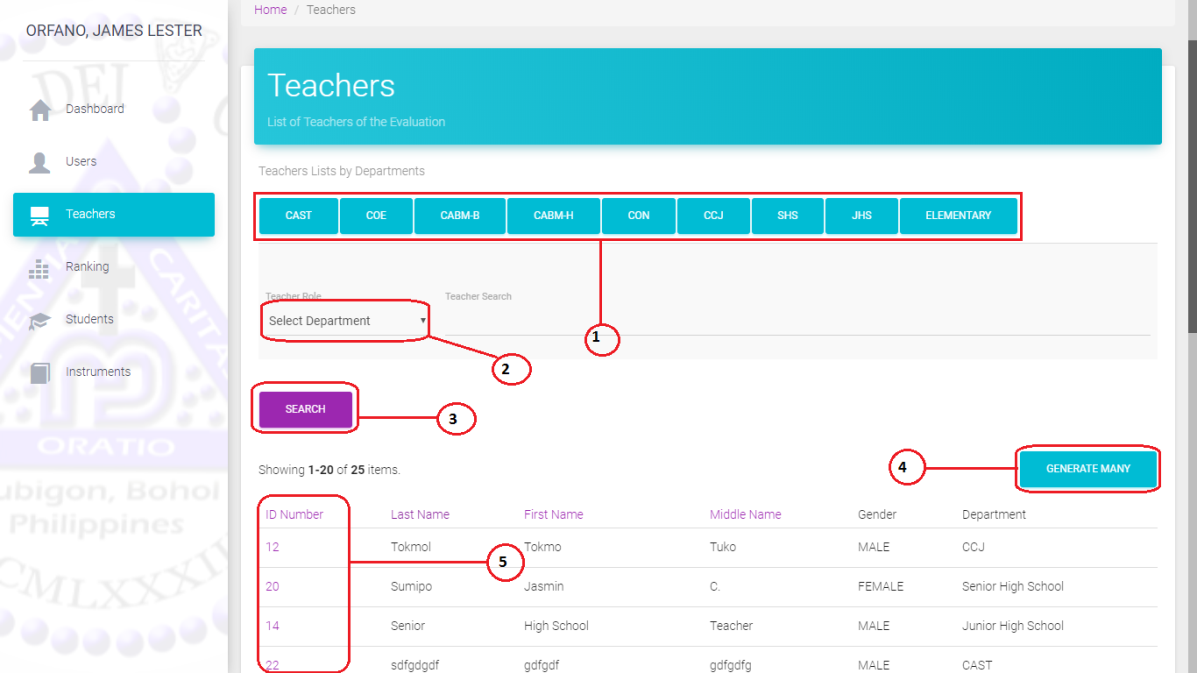


***Figure 5.6 User list***

***Figure 5.6 Users List***

Figure 5.6 is the interface once the Admin click the User link. Only the admin user.

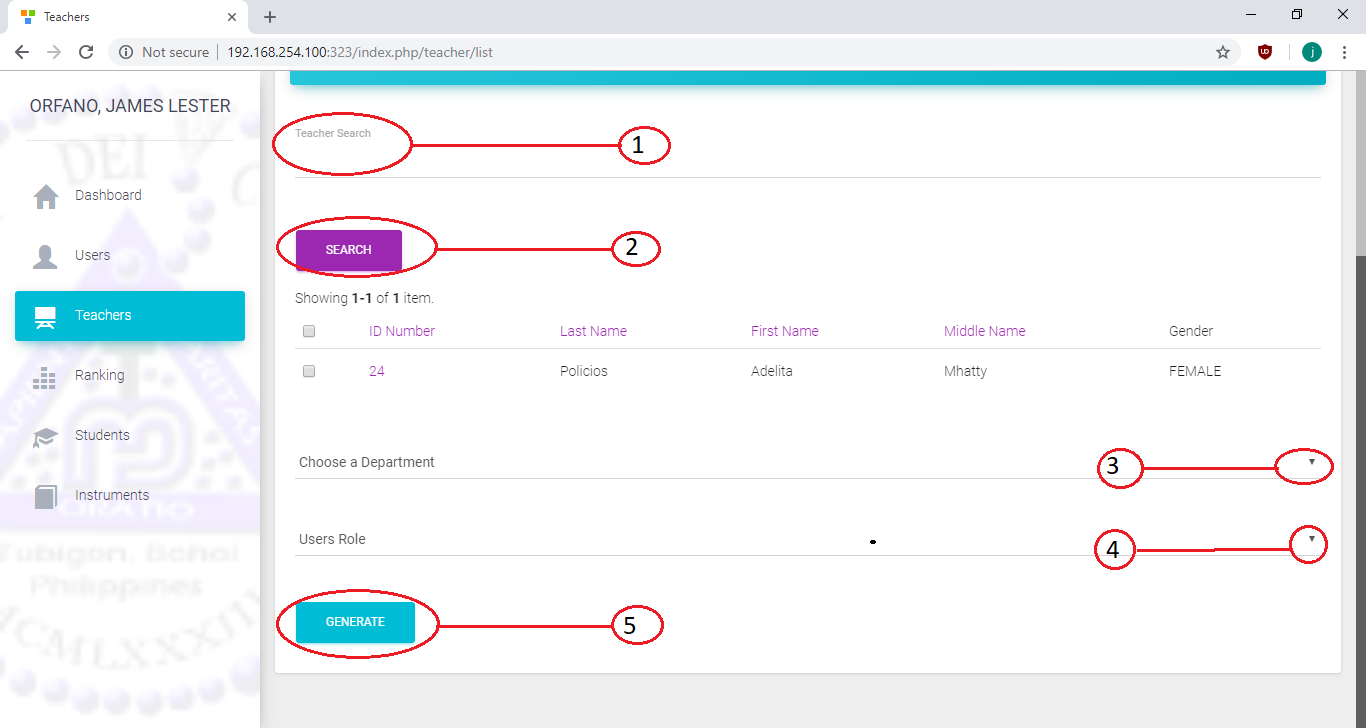
1. The Table of Users
2. This is where you type what you want to search.
3. After you’re done typing you want to search this will be the button you click, the search button.
4. This is the username column when you click one of it it will show you the user details.

******

***Figure 5.7 Teachers List***

Figure 5.7 is the interface once the Admin click the ID link. Only the admin user can see this. It follows how the parts work:

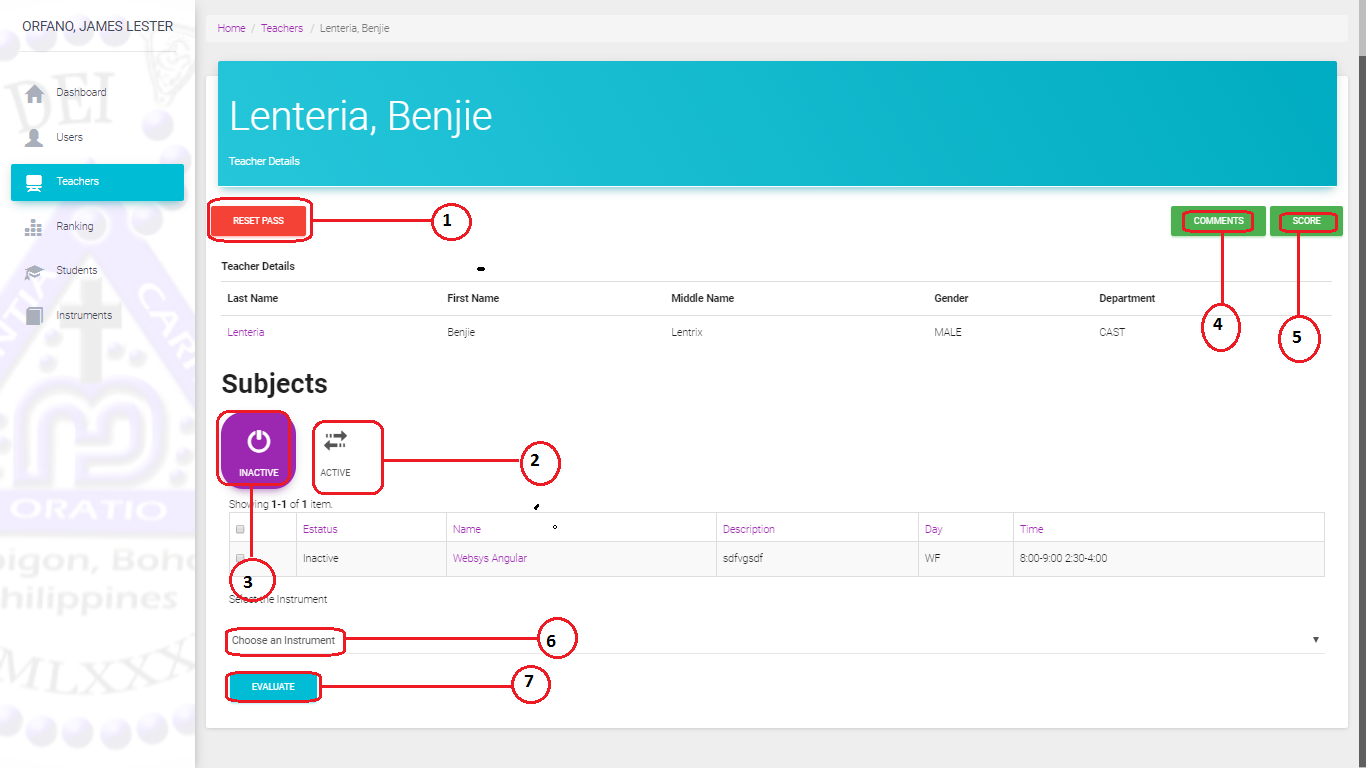
1. This button will show the list of department teachers of the evaluation
2. This dropdown will show the list of the department.
3. After you choose the dropdown just click the search button and it will show the list that you choose.
4. This button will generate new user for teacher.
5. This is the id number of the teacher.



***Figure 5.7.1 Generate New Teacher User***

Figure 5.7.1 is the interface once the Admin click of the Teacher Generate many. Only the admin user can see this. It follows how the parts work:

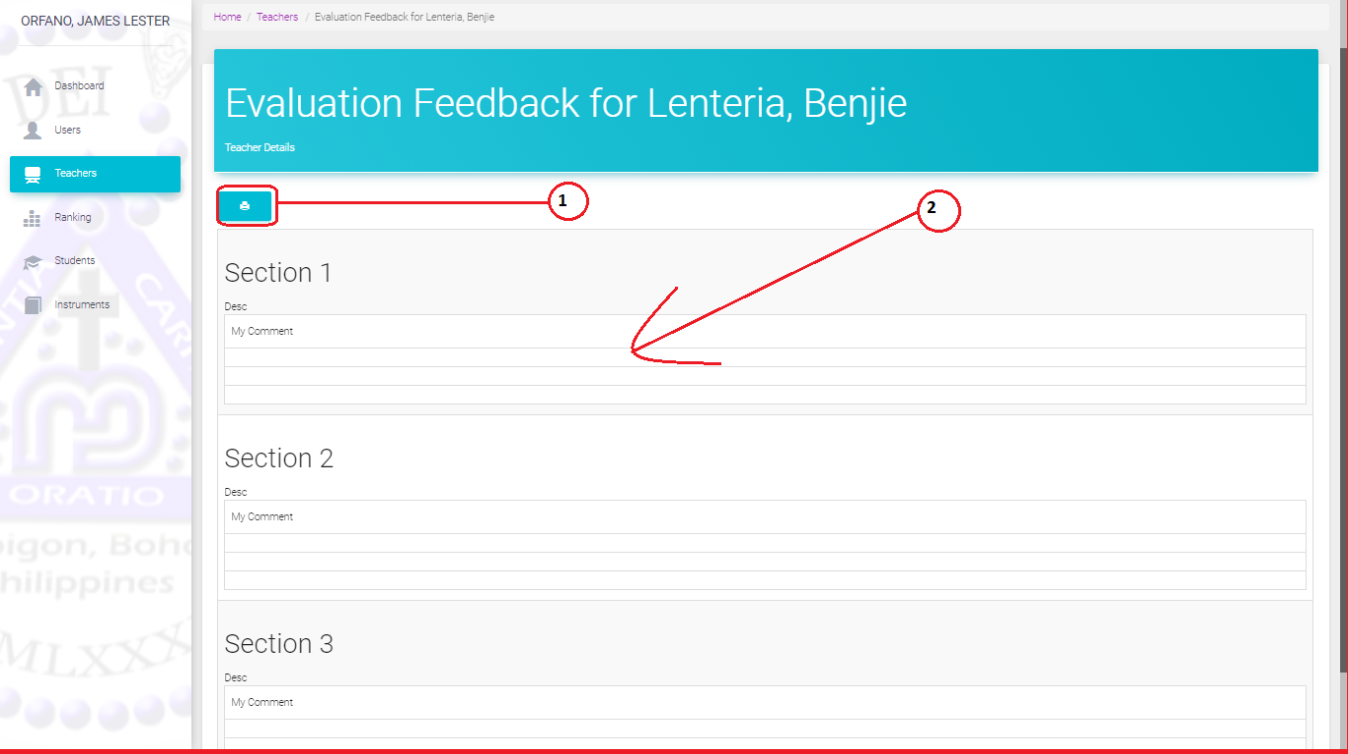
1. This text field where you can text the name of the teacher don’t have account.
2. After you type the name of the teacher just click the search button and it will show the name of the teacher.
3. This dropdown will show the list of departments.
4. This dropdown where you can choose the role of the teacher.
5. This generate button will save the teacher you created.



***Figure 5.7.2 Teacher Details***

Figure 5.7.2 is the interface once the Admin click the User link. Only the admin user can see this. It follows how the parts work:

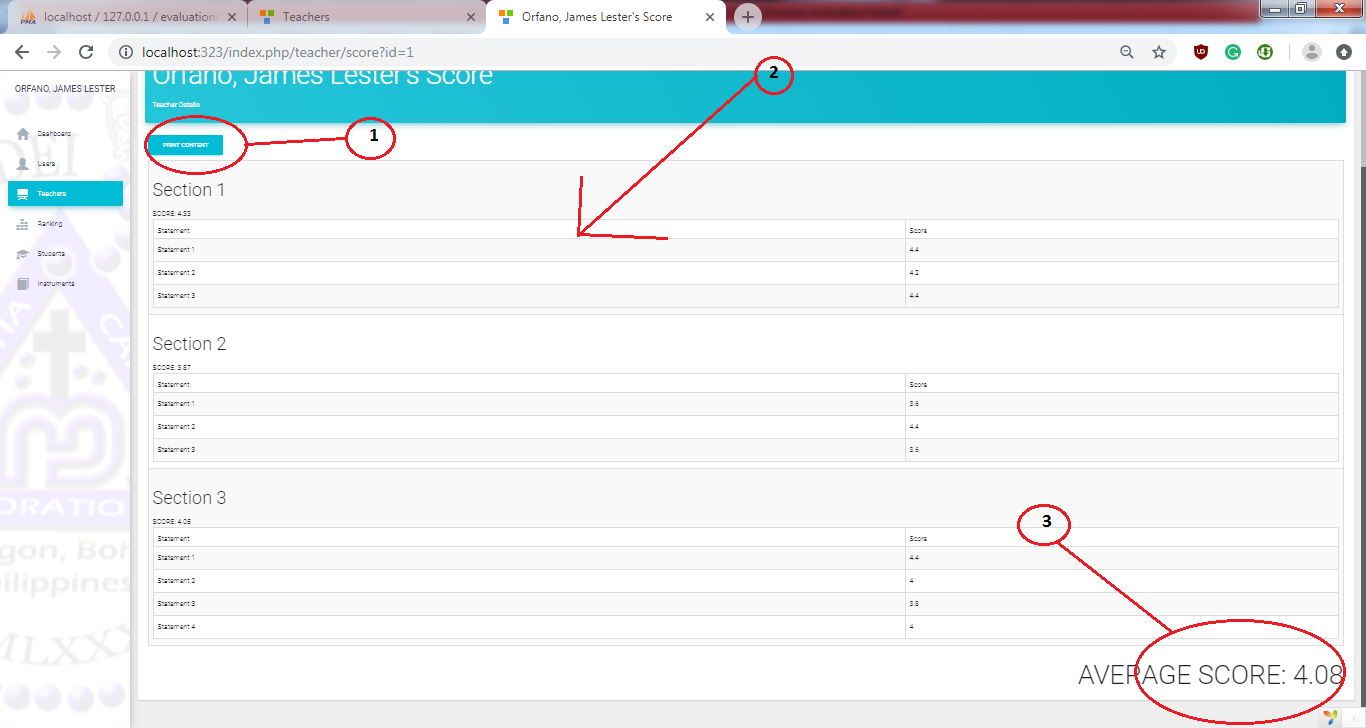
1. This button is used to reset the password of teacher’s user.
2. This tab shows the list of subjects that is active in evaluating process.
3. This tab shows the list of subjects of the teacher that is inactive.
4. This button is used to comment the teacher.
5. This score button shows the score of the teacher by section and average.
6. This dropdown is used to choose instrument of the teacher.
7. This button is used to evaluate to a teacher.



***Figure 5.7.3 Teachers Comment Feedback***

Figure 5.7.3 is the interface once the Admin click the comment button. Only the admin user can see this. It follows how the parts work:

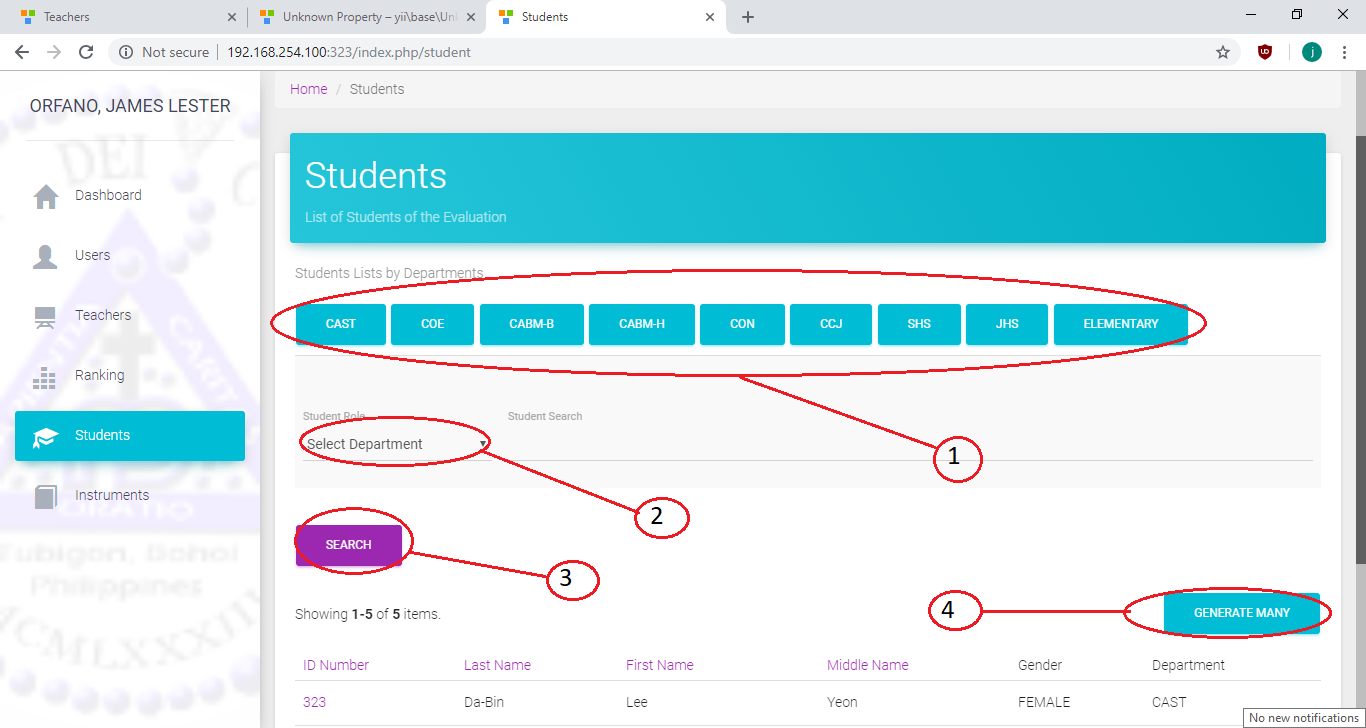
1. This will print the comments given by the teachers students and fellow teacher of same department.
2. This is the comments gave by the teacher student and fellow teacher of same department.



***Figure 5.7.4 Teachers Score***

Figure 5.7.4is the interface once the Admin click one of the teacher’s score button in the Teacher Details. Only the admin user can see this. It follows how the parts work:

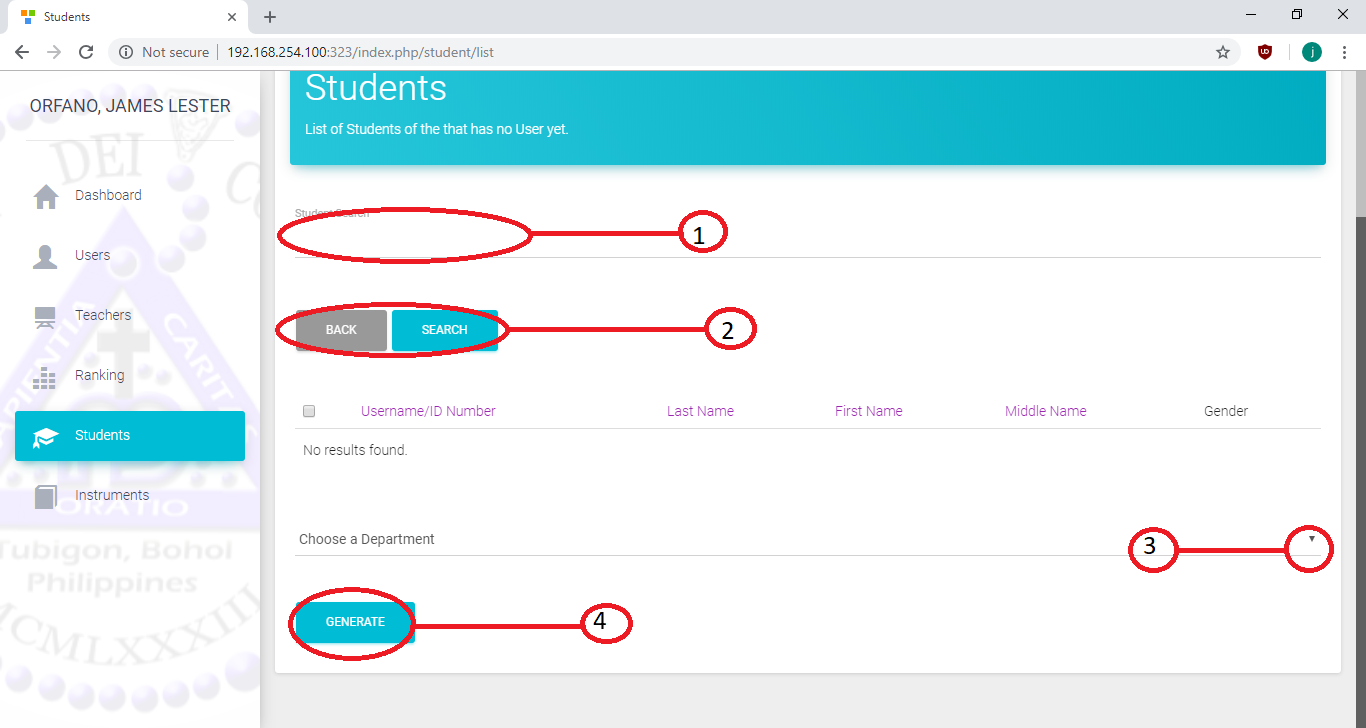
1. This will print the scores.
2. The Table of the evaluation Instrument and scores.
3. The Average score of the teacher.

******

***Figure 5.8 Students List***

Figure 5.8 is the interface once the Admin click the User link. Only the admin user can see this. It follows how the parts work:

1. This button will show the list of department students of the evaluation
2. This dropdown will show the list of the department.
3. After you choose the dropdown just click the search button and it will show the list that you choose.
4. This button will generate new user for students.

******

***Figure 5.8.1 Generate New Student***

Figure 5.8.1 is the interface once the Admin click of the Student Generate many. Only the admin user can see this. It follows how the parts work:

1. This text field where you can text the name of the students don’t have account.
2. This blue button is used to search a student and gray button use to back in the list of students.
3. This dropdown will show the list of departments.
4. This dropdown where you can choose the role of the student.
5. This generate button will save the student you created.

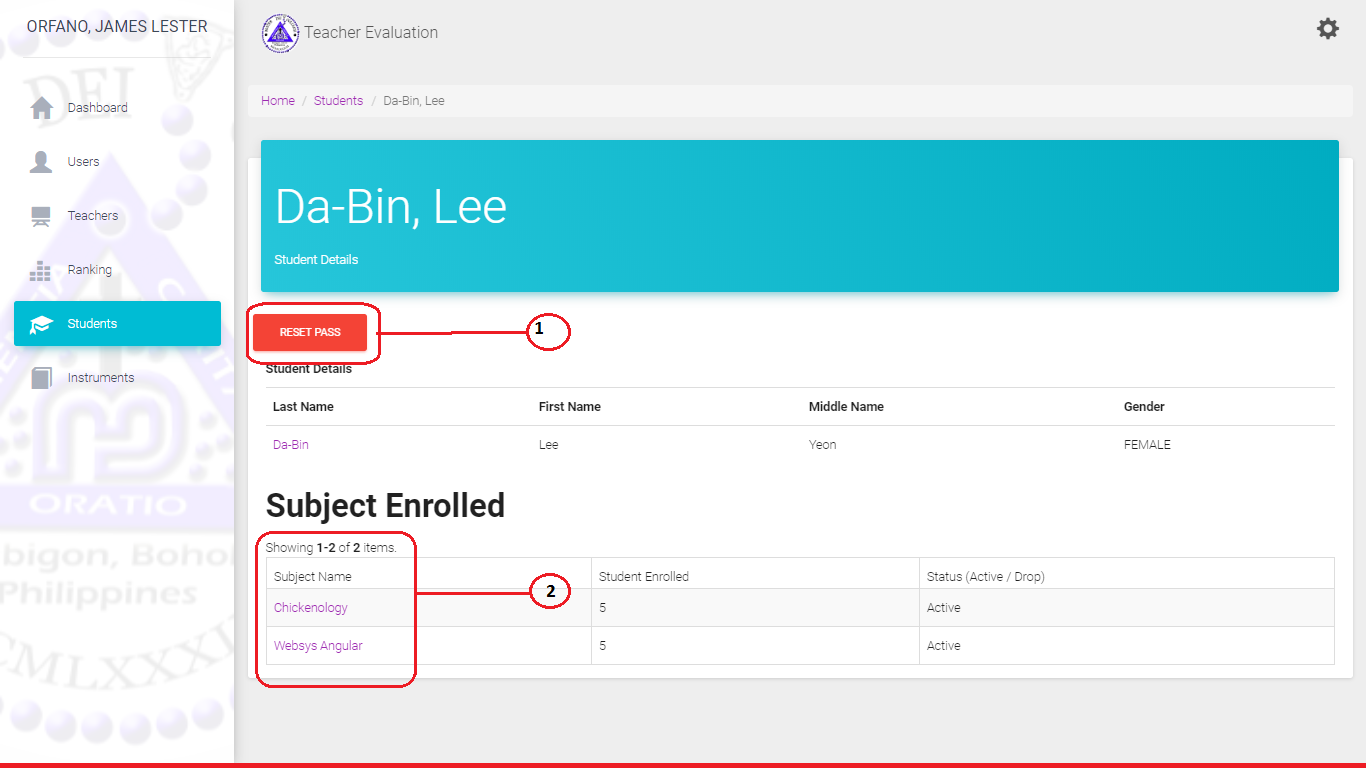
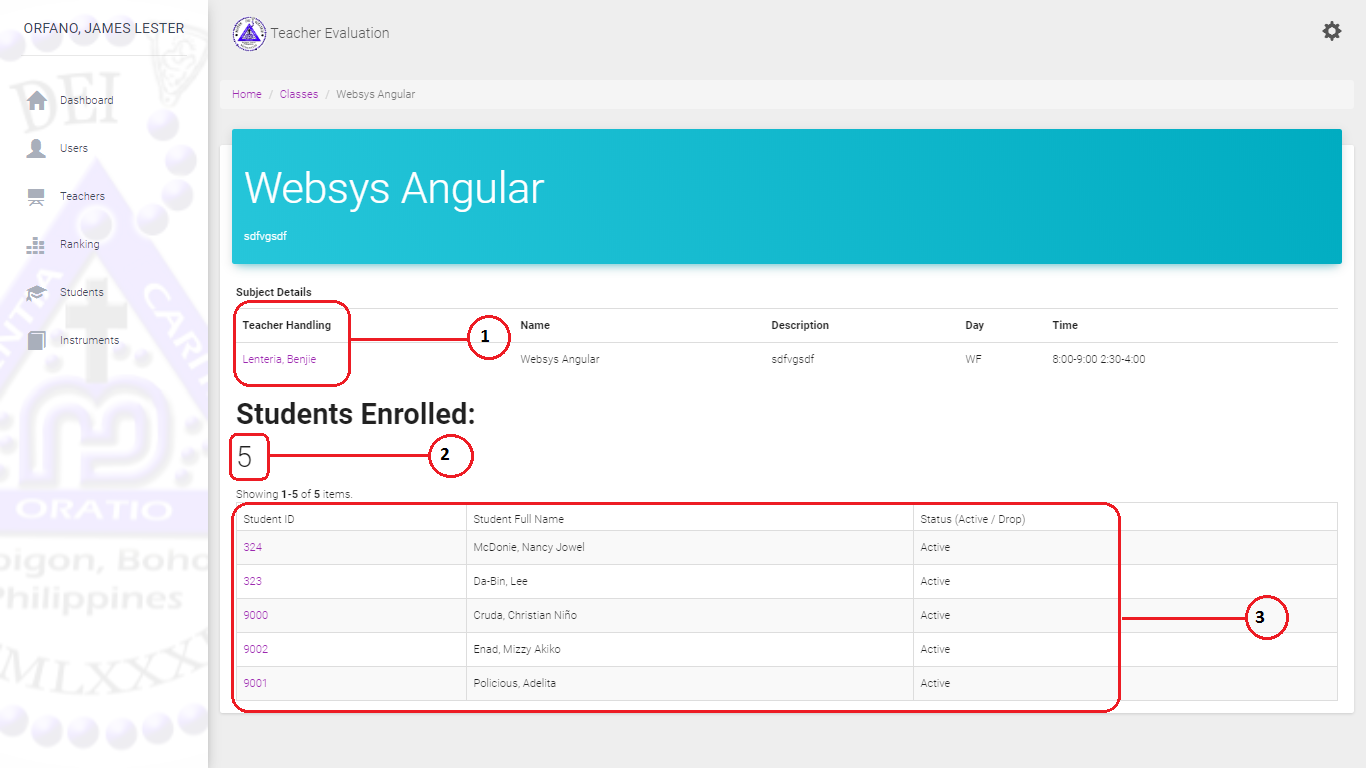
***Figure 5.8.2 Student Details***

Figure 5.8.2 is the interface once the Admin click of the ID Student. Only the admin user can see this. It follows how the parts work:

1. This button use to reset the password of the student user.
2. This is the subject list.

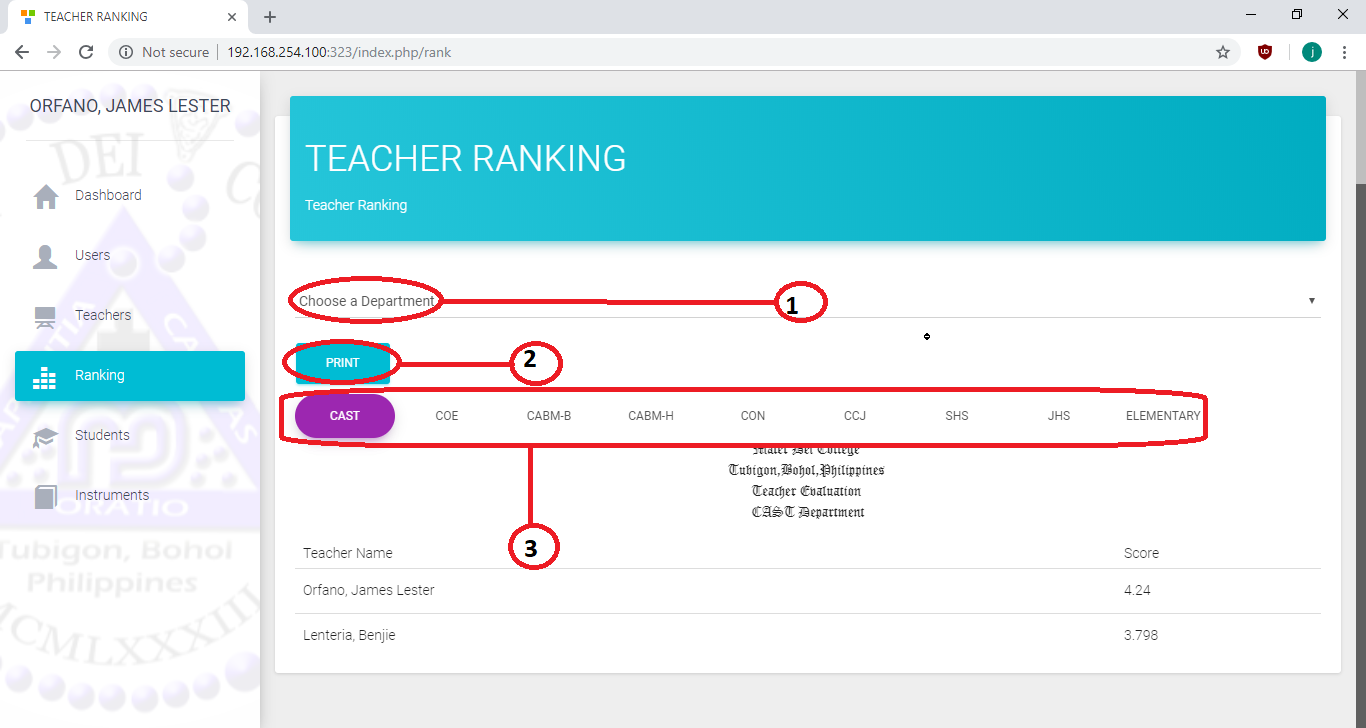


***Figure 5.8.3 Show Students Enrolled Subjects***

Figure 5.8.3 is the interface once the Admin click of the Subject Name.

Only the admin user can see this. It follows how the parts work:

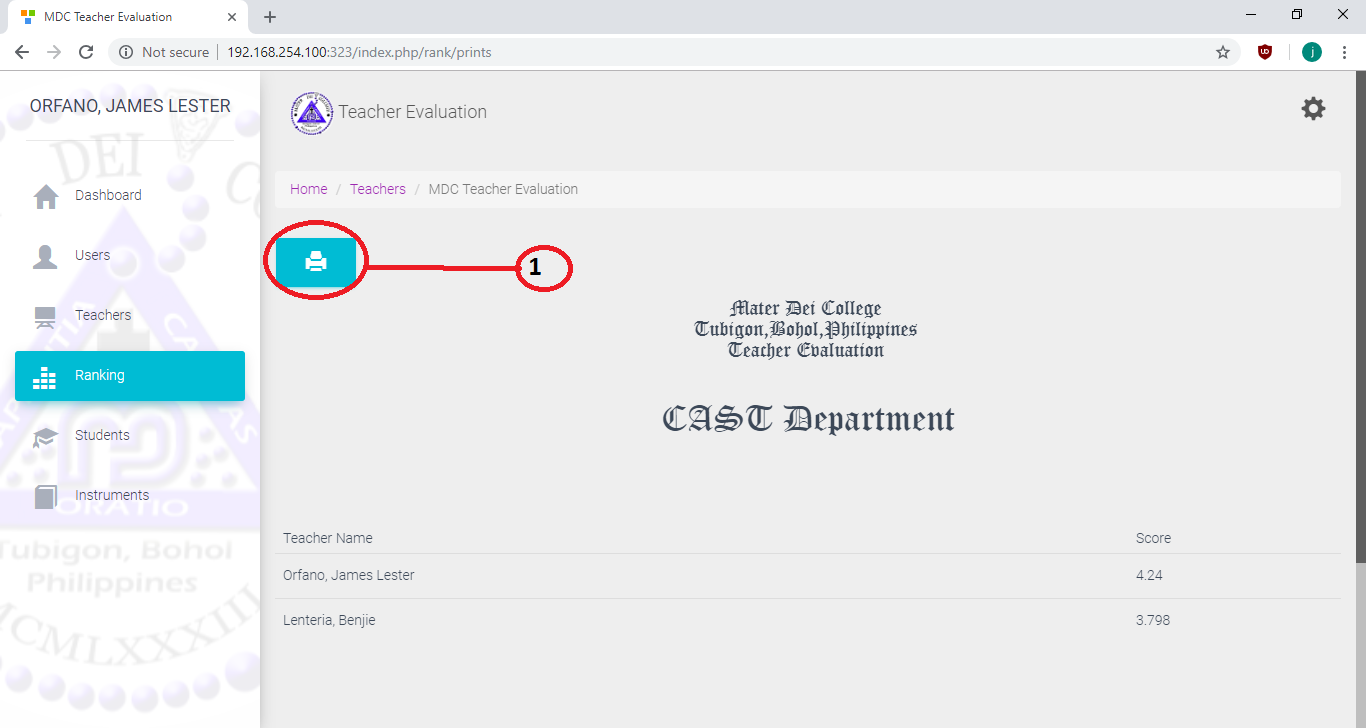
1. The list of the teacher handling.
2. Number of students who enrolled of this subject.
3. List of the students enrolled.



***Figure 5.9 Ranking***

Figure 5.9 is the interface once the Admin click of the ranking. Only the admin user can see this. It follows how the parts work:

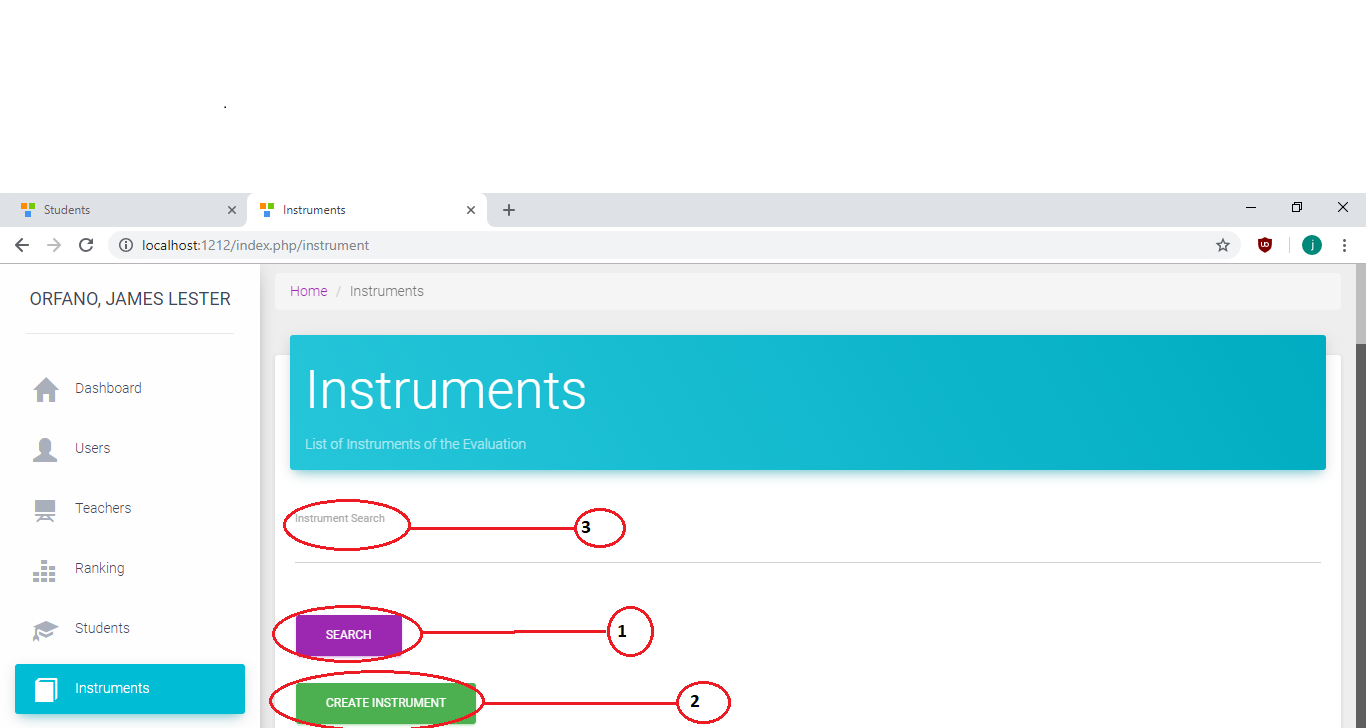
1. This dropdown is used to choose the department.
2. This print button is used to choose to print the result of teacher ranking by department.
3. This button is used to show the teacher sorted by scores.



***Figure 5.9.1 Ranking Printing***

Figure 5.9.1 is the interface once the Admin click of the ranking. Only the admin user can see this. It follows how the parts work:

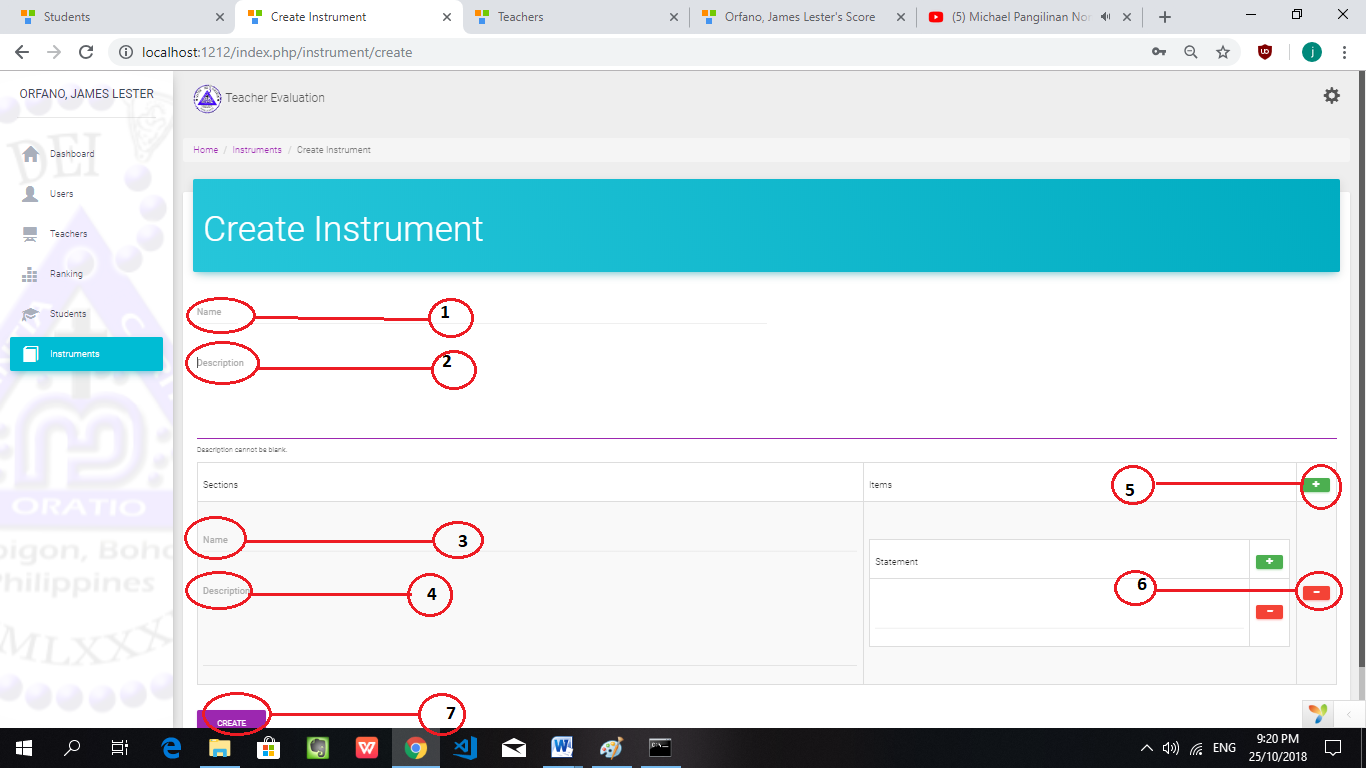
1. This button is used to print.



***Figure 5.10 Instrument***

Figure 5.10 is the interface once the Admin click one Instrument on the Home. Only the admin user can see this. It follows how the parts work:

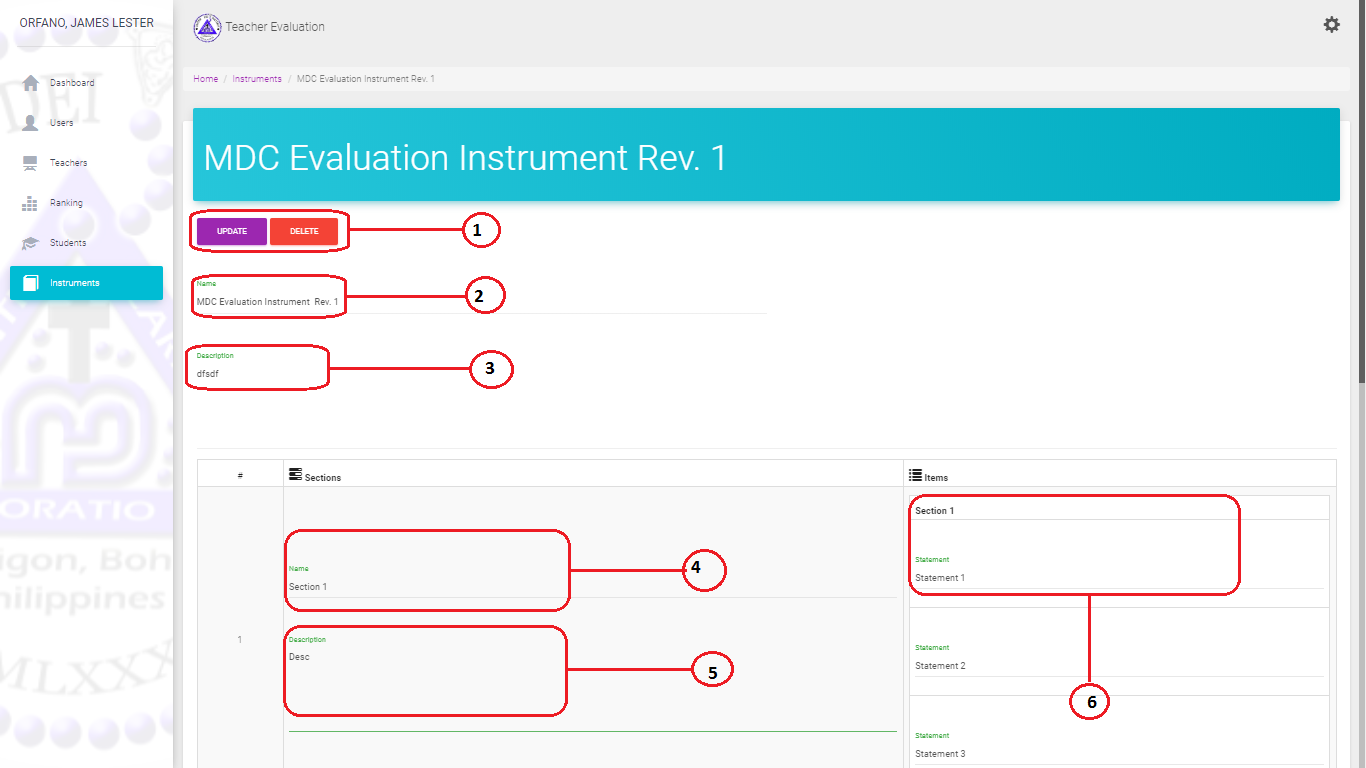
1. This button is used to search easily the instruments that has been added.
2. This button will add new instruments.
3. This text field is used when you type the instruments you want to search.



***Figure 5.10.1 Instrument Create***

Figure 5.10.1 is the interface once the Admin click of the create instrument in the instrument Table. Only the admin user can see this. It follows how the parts work:

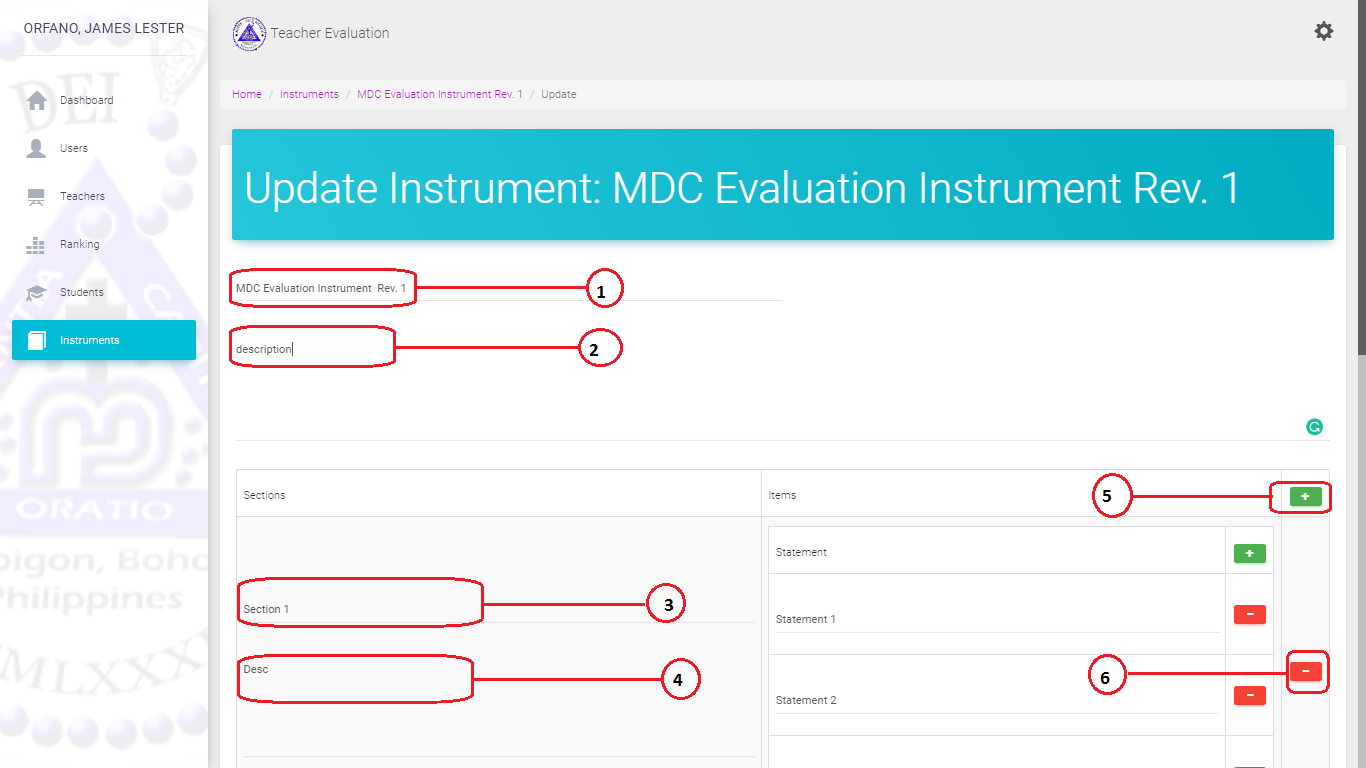
1. Create Instrument: Before the admin evaluate it should create first instrument. Name of the instrument.
2. Description of the instruments.
3. Name of the section.
4. Description of the section.
5. If you click this button it will add item.
6. This button is used to dropdown the item.
7. If you click this button it will save into instrument.



***Figure 5.10.2 Instrument Details***

Figure 5.10.1 is the interface once the Admin click of the create instrument in the instrument Table. Only the admin user can see this. It follows how the parts work:

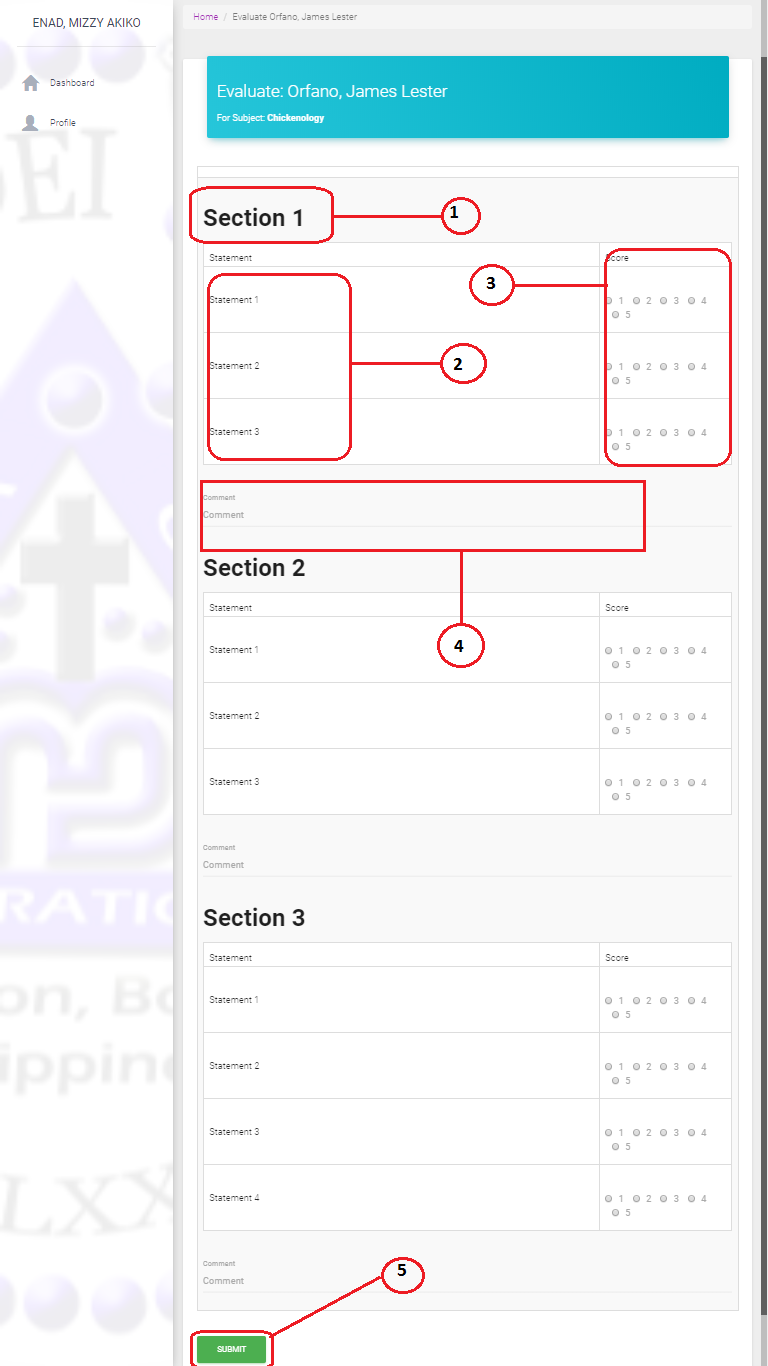
1. This color violet use to update the instrument and other one use to delete the instrument.
2. Name of the Instrument.
3. Description of the instrument.
4. Name of the section.
5. Description of the section.
6. Statement by section.



***Figure 5.10.3 Instrument Details Update***

Figure 5.10.3 is the interface once the Admin click of the Update instrument. Only the admin user can see this. It follows how the parts work.

1. The name of the instrument.
2. The description of the instrument.
3. Name of the section.
4. Description of the section.
5. This button is used to add section or statement.
6. This red button is used to drop the section or statement.



***Figure 5.11 Evaluation Form***

Figure 5.11 is the interface once the Admin click of Evaluate. Only the admin user can see this. It follows how the parts work:

1. The name of the section.
2. The statement of the evaluation.
3. This where you can choose and click the score.
4. This is the where you can text the comment.
5. This button is used to submit the evaluation.

**Chapter VI**

**CONCLUSION AND RECOMMENDATION**

This chapter provides the conclusion of the project based on the findings of the researcher and recommendations that the researchers believed would improve the project if added.

**Conclusion**

In conclusion, automated teacher evaluation system based on Web System, the school will be developed to replace the traditional teacher evaluation system that is currently used. This project will be considered succeed once automated teacher evaluation system based on Web System is developed. This system is designed to make the whole evaluation taking process to become more reliable, convenient, efficient, and accurate. Besides that, with the implementation of Web base system technology will help in reduce errors and evaluation data will be able to compute in easier way.

This project is designed to aim in eliminating spotted problems during initial analysis. The problems spotted are includes time consuming and Tedious work. Prone to human error. It is slow, less reliable and inefficient in computing scores. These problems are the major problems faced by the school.

Therefore, this project is designed in effort to eliminate these problems. Some solution had been applied to eliminate these problems which include the use of Web base system, change the current traditional evaluation system to fully-computerized system, and provide easier way to view report.

**Recommendation**

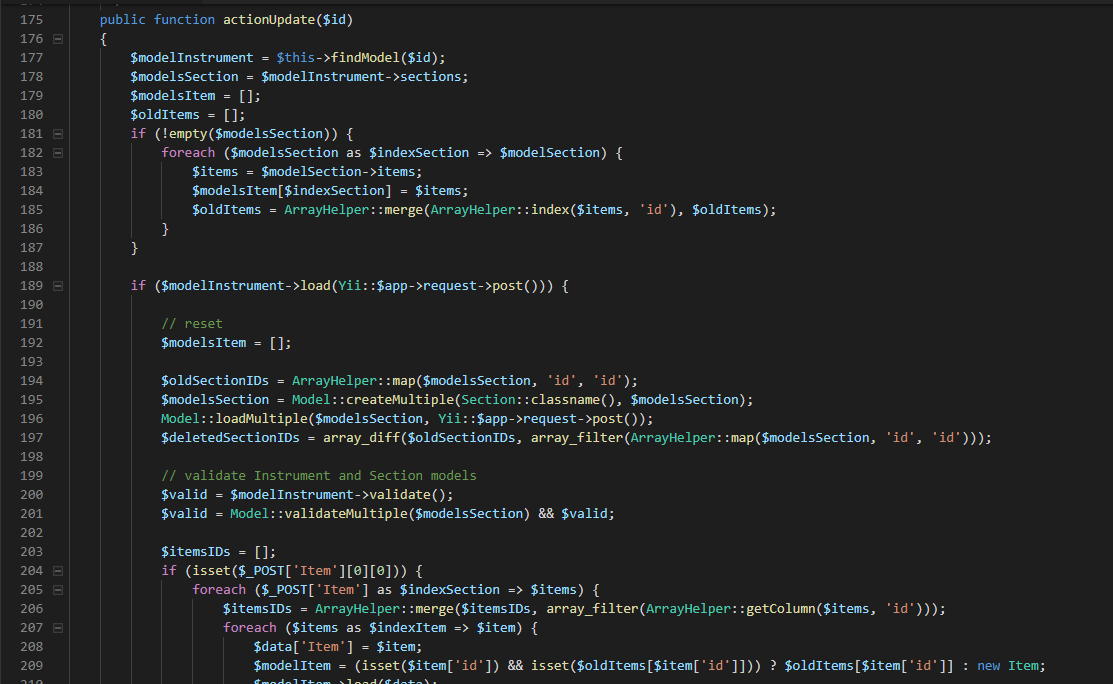
The researchers strongly recommended the implementation of the Automated Teacher Evaluation System based on Web base system. In addition, the following are recommended.

The proposed system is open for further development and enhancement in terms of developing the following features:

1. Generate a printed output for the teacher evaluation record.

**Appendices**

Sample Source Code



**CURRICULUM VITAE**



**ADELITA A. POLICIOS**

**Personal Details**

Birthday: October 17,1993

Address: Pinayagan Norte, Tubigon, Bohol

Phone: 09153689813

Email: [imhaaaatty17@gmail.com](mailto:imhaaaatty17@gmail.com)

Height: 4’13 ft.

Weight: 55 kgs.

**Education**

Elementary

Pinayagan Norte Elementary School

Tubigon, Bohol

2005-2006

Secondary

Holy Cross Academy

Pooc Oriental, Tubigon,Bohol

2009-2010

Tertiary

Mater Dei College

Cabulijan, Tubigon, Bohol

2019

WorkExperience

Production Operator

Cebu Mitsumi Incorporated

Danao City,Cebu

April 2012-May 2015

**Skills**

Software: Soldering Operator

Welder Automation Machine

Language: Tagalog,Visayan, English



**MIZY AKIKO P. ENAD**

**Personal Details**

Birthday: July 14, 1999

Address: Dagohoy, Inabanga, Bohol

Phone: 09363537812

Email: [mizyakiko123@gmail.com](mailto:mizyakiko123@gmail.com)

Height: 5’2ft

Weight: 50 kgs.

**Education**

Elementary

Dagohoy Elementary School

Dagohoy, Inabanga, Bohol

2010-2011

Secondary

Nahawan National High School

Nahawan, Clarin, Bohol

2014-2015

Tertiary

Mater Dei College

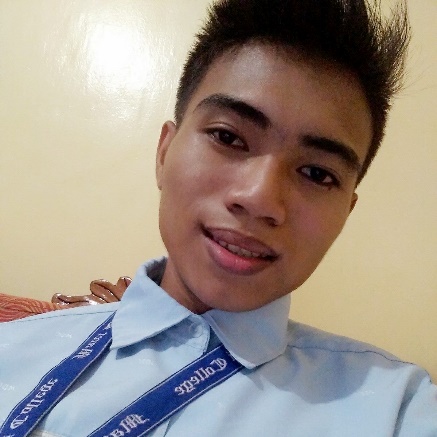
Cabulijan, Tubigon, Bohol

2019

**Skills**

Language: Tagalog, Visayan, English

Software: Microsoft Applications



**CHRISTIAN NIÑO P. CRUDA**

**Personal Details**

Birthday: December 06, 1998

Address: Calunasan, Calape, Bohol

Phone: 09664585472

Email: [juggthug357@gmail.com](mailto:juggthug357@gmail.com)

Height: 5’2 ft

Weight: 45 kgs

**Education**

Elementary:

Mandaug Elementary School

Abucayan Sur, Calape, Bohol

2010-2011

Secondary:

Mayor Anunciacion R. Tuazon National School of Fisheries

Calunasan, Calape, Bohol

2014-2015

Tertiary:

Mater Dei College

Cabulijan, Tubigon, Bohol

2019

**Skills**

Language: Tagalog, Visayan, English

Software: Microsoft Applications



**JAMES LESTER ORFANO**

Personal Details

Birthday: January 03, 1999

Address: Pinayagan Sur, Tubigon, Bohol

Phone: 09060044485

Email: [resuta566@gmail.com](mailto:resuta566@gmail.com)

Height: 5’4 ft.

Weight: 90 kgs.

**Education**

Elementary:

Pinayagan Sur Elementary School

Pinayagan Sur, Tubigon, Bohol

2010-2011

Secondary:

Holy Cross Academy

Pooc Oriental, Tubigon, Bohol

2014-2015

Tertiary:

Mater Dei College

Cabulijan, Tubigon, Bohol

2019

**Skills**

Language: Tagalog, Visayan, English

Software: Microsoft Applications, knowledgeable in some programming languages.