CLOUD-BASED EVENT TABULATION SYSTEM

A proposal
Presented to the faculty of the
College for Research and Technology of
Cabanatuan

In partial fulfilment of the requirements
For the degree of Bachelor of Science in Information Technology

By:

Acutillar, Angeline S.

Aguilar, Aljune S.

Picaña, Ronell M.

Taguiam, Merson L.

Mr. William Aldana
Adviser

July 2021

Chapter I

INTRODUCTION

1.1 Background

In today's new environment, we have to learn how to handle not just us in-person activities, but also our interactive programs. Virtual event management involves the same steps as planning you’re in-person event, but with the extra difficulty of making sure, the content is twice as captivating. Although in-person activities have added the benefit of transport, networking, and free food, a virtual experience depends heavily on the material to keep participants involved. When managing a virtual event, making sure that the speakers are equipped to deliver digitally their content and that content is interesting and concise.

As the COVID-19 health crisis erupted, activities were the first field to ban by the government. Ensuring the protection of each individual and planning a face-to-face interaction would also be the last thing to do.

Due to the fast-growing technology, establishments and organizations in the country are using new technologies to allow them to serve clients easily and effectively and to develop, and to provide correct tabulation as well as fall count and anticipation as soon as possible and to solve many problems with computer technology.

Computers are more capable of handling large volumes of data more efficiently than ever before. Anything can have achieved with the aid of computers in terms of computational and logical manipulation. Nowadays, creativity in things has arisen. Automation of machines and appliances and other computer-controlled technologies are trends in which the primary aim of technology is to make life more beautiful, reduce people's workload and make work easier.

Tabulation means putting a data in a table. Table means a diagram, which has a few vertical and horizontal lines. It is a presentation of a data in a lucid form. It is the act or process of tabulating and table displaying data in a compact form. A tabulation system for delivery to

a medium of data information suitably arranged for tabulation of character series and ruled lines, and a control for controlling the data information arrangement applied to the medium.

Some automated scoring program built in Excel and uses embedded formulas to calculate the final scores and rank the contestant However, when considering the disadvantages of spreadsheets, it appears to be a much riskier option.

Cloud server is a virtual server (rather than a physical server) running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely. They are also known as virtual servers. Cloud servers have all the software they require to run and can function as independent units. A cloud server gives the business user stability and security because any software problems are isolated from your environment. Other cloud servers will not effect on your cloud server and vice versa. If another user overloads their cloud server, this will have no impact on your cloud server, unlike with physical servers.

But in most cases, the cloud is more secure than on premise data centers because cloud providers have made — and continue to make — significant investments to ensure data protection. Many cloud services for business have security features built in, including application role-based authentication.

In this study, the researchers aiming to develop an application that is more secure, fast, user-friendly, and can produce a reliable and accurate results to prevent human error. The study is a combined two systems, an electronic voting system and dynamic tabulation system. The proposed study focuses on the flexibility of the tabulations which can use in any kind of event such as Dance Competition, Beauty Contest, Talent Competitions that include or involve in any particular tabulating events, and also in conducting voting election. The Cloud Based Event Tabulation System is designed and developed specifically to cope up with the pandemic crisis people facing right now. The researchers will make it possible to conduct a contactless event tabulation which can cast votes and scores online.

Conceptual Framework

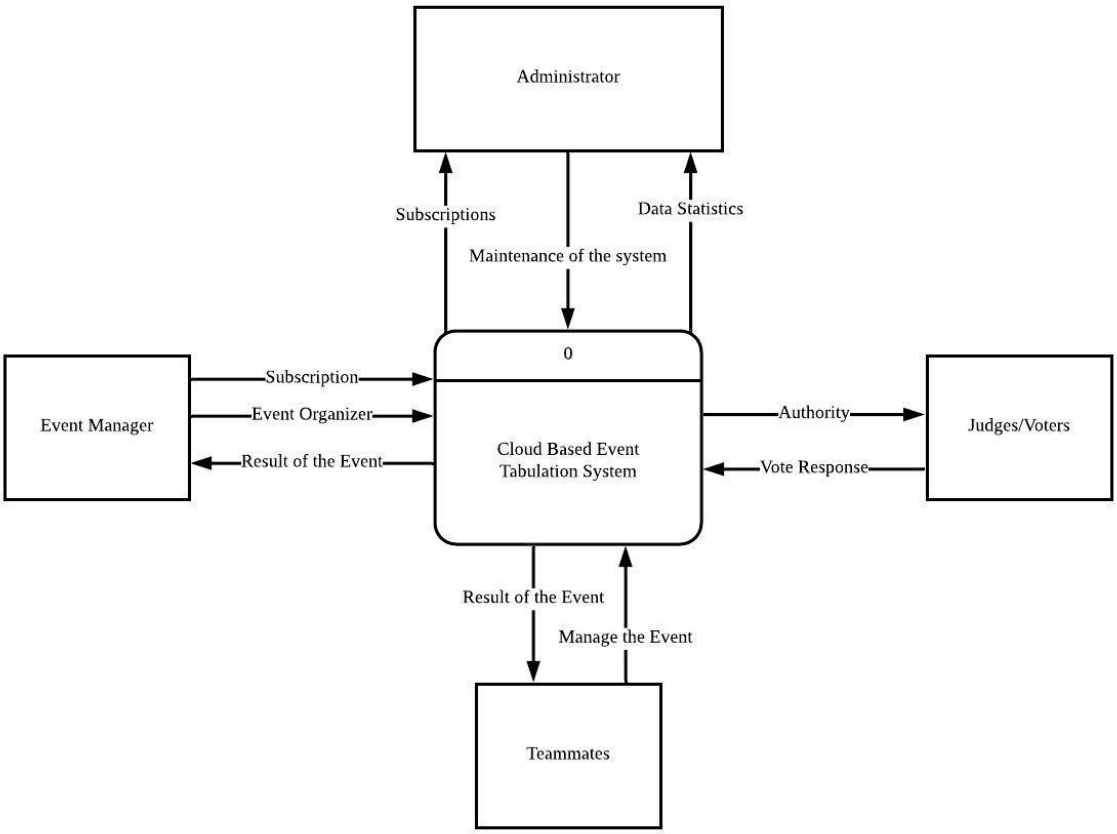


Figure 1: <u>Context Diagram of Cloud Based Event Tabulation System</u>

The context diagram illustrates the structure of Cloud Based Event Tabulation System. It shows the relationship among the Administrator, Event Manager, Teammates, and Judges/Voters. The proposed system is designed to be a flexible that can be use in any kind of tabulation. The administrator, which is the researchers has all the privilege to monitor the subscribers and also the users in every event. To access the system, event managers need a subscription depending on how long it will be used. Subscription feature is one of the researchers' way to maintain the continuous use of the system. Event manager is the one will assign the specific task for the teammates; the event mangers can modify what task they will be assigning to teammates. The judges/voters have a restriction that they can only sign up their account, view candidates, and cast votes.

1.2 Statement of the Problem

This study aimed to develop and implement a Cloud Based Event Tabulation System. Specifically aiming to seek an answer to the following questions in terms of;

1. Security,
2. Accuracy,
3. Usability,
4. Reliability and
5. Flexibility

1.3 Objectives of the Study

General Objectives

The proposed system will reduce the paperwork in terms of tabulating scores in every event. The system also implements the contactless in holding an event.

Specific Objectives

To develop a system called Cloud Based Event Tabulation System that:

- To ensure a fast and secure of casting votes
- To allow a contactless tabulation
- Provide a user-friendly interface for the users to access
- To enumerate the challenges encountered with the existing system.
- To identify the accuracy of the system.
- To identify the advantage of having a well-developed tabulating system.

1.4 Significance of the Study

Through this study, it would be beneficial to the following:

- **Event Organizer:** The main beneficiary of this study is the event organizers. Making it possible to conduct a contactless event would be such a big help for them.
- **Administrator**: It will help them to make their workings stress-free.
- **Judges/Voters:** The system has its flexibility feature that the judges and voters can use it in any device at any time. The system designed for web application that will definitely fitted in any device.
- **Future Researchers:** It will help them on their future research and give them an idea how to develop their project.

1.5 Scope and Limitation

Scope

In this study, the researchers aiming to develop an application that is more secure, fast, user-friendly, and can produce a reliable and accurate results to prevent human error. The study is a combined two systems, an electronic voting system and dynamic tabulation system. The proposed study focuses on the flexibility of the tabulations, which can use in any kind of event such as Dance Competition, Beauty Contest, Talent Competitions that include or involve in any particular tabulating events, and in conducting voting election. The Cloud Based Event Tabulation System is designed and developed specifically to cope up with the pandemic crisis people facing right now. The researchers will make it possible to conduct a contactless event tabulation which can cast votes and scores online. Apart from that, the system will also have the following features:

- This project can always add, edit, delete, view, upload picture, and save information.
- Capable of customization.

- Multi flat form capability.
- Can assign specific percentage to the given category and criteria.
- Administrators/Users/Judges have their own usernames and passwords to access the interface personally and for the security purposes.
- Can print the reports or the results instantly.
- Capable of watching live on actual event using the installed camera.
- Provide tally of the results for verification.

Limitation

The study entitled "Cloud Based Event Tabulation System", will be having a limitation regarding with the following:

- It will not be working on a lower version of a web browser application which is contrary to the system requirements of the research study
- The proposed system will be having a limitation regarding with the internet speed. It will require a 1mbps+ for its download and 500kbps+ for its upload to access it and used all the feature of the system**.**

1.6 Definition of Terms

Cloud server: A virtual server running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely.

Cloud webhosting: A web hosting that uses the resources of several clustered servers. Which is that your website uses the virtual resources of several servers to accommodate all the aspects of hosting your site.

Tabulation: Means putting a data in a table. It is the act or process of tabulating and table displaying data in a compact form.

MBPS: The acronym Mbps stands for "megabits per second." It is a measure of internet bandwidth. In simple terms, bandwidth is the download rate of your internet connection. It is the maximum speed at which you can download data from the internet onto to your computer or mobile device.

Multiplatform: The ability for a software system to run on different computer hardware and operating systems with little or no changes.

Chapter II

REVIEW OF RELATED LITERATURE

According to Afable, Shoven M. [AFAB2020], on Multi-user Automated Pageant Tabulation System, “The developers highly recommend implementation of the pageant automated tabulation system in the community during pageant event for accurate, fast, easy to use and efficient tabulation system. This system was intended to help them calculate, tabulate and generate results of the pageant in a timely and accurate manner. The main objective of the study is to develop a LAN-based automated tabulating system for beauty pageant that can (1) display the criteria to be graded; (2) allow judges to enter scores in the system based on criteria; (3) compute, display and print result for specific event automatically; and (4) consolidate and display judges’ scores of specific event for easy monitoring by the admin.”

With the help of the system, the tabulation and computation of the scores for each category of the event will be accurate, faster, and with reliable result. The system was intended to help them calculate, tabulate and generate results of the pageant in a timely and accurate manner. The system was developed and ensured its efficiency and user-friendliness. One of the greatest advantages of the software is that its structure is simple, especially the executable code, user friendly and interactive. The system was developed using a Visual Basic, it has also an integrated development environment (IDE) with easy-to-use tools and utilities that allows for rapid development of software programs.

Based on Campbell, Bryan A. et al [CAMP2011] developed a mobile voting system for the iPhone to increase voter participation by allowing voters to use familiar technology that was published on September 1, 2011 and compared its usability with traditional voting platforms. The results showed that the mobile voting system was not as efficient as the other voting methods in total interaction time.

According to Ahmed, Waqas on Android based Event tabulation [AHME2014] “Task automation has become quite popular among mobile users thanks to apps like Tasked. Tasked allows you to configure different scenarios based on current location, time, event and so on, and automatically trigger different actions based on those scenarios. For instance, if you want your phone to automatically switch to silent mode when connected for charging, or need to turn off Wi-Fi when you’re on the road, you can easily achieve that using Tasker. This Mobile-based events tabulation System is if you have a technology or have an application it is easy for the users to give score and calculate the score of every contestant in every category.”

From the study of Fedena [FEDE2015], automated pageant scoring and Tabulation System Developed at the University of Northern Philippines for use in the Miss CCIT (College of Communication and Information Technology) Pageant and the University wide Miss UNP Pageant. It allows any number of judges to enter their scores for varying rounds throughout the pageant. Upon completing each round, and ultimately the pageant, PDF reports can be generated and printed the reports have lines for the judges to sign indicating that their scores are accurate. This ensures a complete paper trail of the final pageant results.

According to Swierenga, et al. [SWIE02012] Usable voting systems are key to a successful voting experience for everyone, but are especially important for persons with disabilities. Voting systems need to be designed so that these voters can effectively interact with a voting system in a reasonable amount of time and without discomfort. The overall goal of their study was to develop a suitable, rigorous test protocol for a Voting System Test Laboratory (VSTL) to conduct usability conformance testing of accessible voting systems with persons who are blind, have low vision, or have dexterity impairments in order to ensure that they can vote independently. First, they conducted a gap analysis and formal study of existing conformance tests and methodologies for conducting usability tests for accessibility. Then, they developed and refined tests and protocols appropriate to the selected demographic groups,

before conducting and analyzing dry runs of the test protocol using multiple voting systems to get expert timings. The materials that were developed are intended to be used in pilot testing.

Based on the study of Tripathi, Ankita [TRIP2016] Online event management system is an online event management system software project that serves the functionality of an event manager. Providing most of the functionality required for an event type e.g. (marriage, dance show, birthday party, etc.), the system then allows the user to select date and time of event, place and the event equipment. All the data is logged in the database and the user is given a receipt number for his booking. The data is then sent to administrator (website owner) and they may interact with the client as per his requirement.

Event Management defined by Jain, Anjali [JAIN2015], that it can be broadly explained as an application of project management to create and develop large scale events such as festivals, conferences, athletic competitions, ceremonies, formal parties, concerts, conventions and more. Online event management tools and event management software are winning over the traditional event management methods due to their efficiency. An event management software ensures you focus on those things that need your attention the most. A capable software would efficiently take care of, sending invites, registrations, sales, payment collection execute offers if any (early bird or VIP) and equip you with all the details with just a few clicks on your smart device saving you the unnecessary hassle.

Most online event management systems allow you to quickly set up an events page, manage ticketing, and instantly start sending invites to your prospective guests. The good ones will also allow you to post your event details on social media with easy share buttons on the events page. Posting details on social media about your event on a relevant platform may help fetch you more appropriate registrations. These easy social share buttons will also motivate your attendees to share the event details on their social networks.

According to Saleem, Amir [SALE2017] Event management is the application to manage and development of festivals, events and conferences. The proposed work involves study of identifying the target of budget, cost, and analysis. Post event analysis and ensuring a return on investment have become significant drivers for the event industry. This is an online event management system, software project that serves the functionality of an event manager. The project provides most of the basic functionality required for an event. It allows the user to select from list of event types. Events Management System is very helpful for events. Event organizer is an application under project management for managing festivals or social events like gathering, colleges, events, conferences etc. To understand use of the application, consider the flow of actions happening, by this application user can register the students, after registering, user can login, after login, event details including name, contact, address, venue of the event, date, event conducting time, cost of events etc. After receiving SMS student can register through application.

Prajapati, Vaishali [PRAJ2017] stated that, "The need for event management system is client can easily get information about event at his home. Anyone can see gallery in which user can see photo-shoots, decoration, birthday celebration images, college event's images, anniversary celebration images etc. Every user has their own user id & password by using they can log into user area and pay for decoration charges, hall booking etc. By using this online event management system project, it reduces many worries of the user. User can book what they want according to their requirements and also pay online. Everything is done at only one place."

Based on the article in National Democratic Institute [2013], Tabulation process at all levels should be fully transparent for party representatives and observers. It should be able to witness by the observer all the data being uploaded or entered into the tabulation computers. If observers have collected results protocols from polling stations, they should be able to verify that these figures have been properly recorded at each higher level of the tabulation process.

The full tabulation from the central level down to the polling station should be publicly available on the Internet in an easily verifiable format. Electronic voting and counting technologies allow for quicker tabulation and transmission of results when compared to paper-based systems, but election authorities must ensure that these processes are undertaken with as much transparency as possible and with a strict focus on the security of results data. Results can be transmitted either through secure communication channels or by encrypted data. And as these security measures are taken to safeguard the data, election authorities must ensure that observers and oversight groups are able to observe the data being uploaded. Results data from the polling station to central level should be made publicly available online.

Chapter III

METHODOLOGY

3.1 Gantt Chart

Figure 3.1: Gantt Chart

The researcher started to plan and think regarding with the title, design, and features of the system. (Green Shade) On the last week of November until December, the researcher started to analyze the information gathered for their documentation. (Red Shade) In this phase, the researcher started building up the design of the system. (Yellow Shade). From February until March, the researcher starts coding to build the proposed system. (Dark Blue Shade) On the 3rd week of March up to the last week of April, the researcher started to test the system. (Sky Blue Shade) And lastly, on the month of May, the researcher implements the proposed system. (Purple Shade)

3.2 V-Model

V-Model a software development life cycle (SDLC) that emphasizes the concept of "**Verification and Validation**".

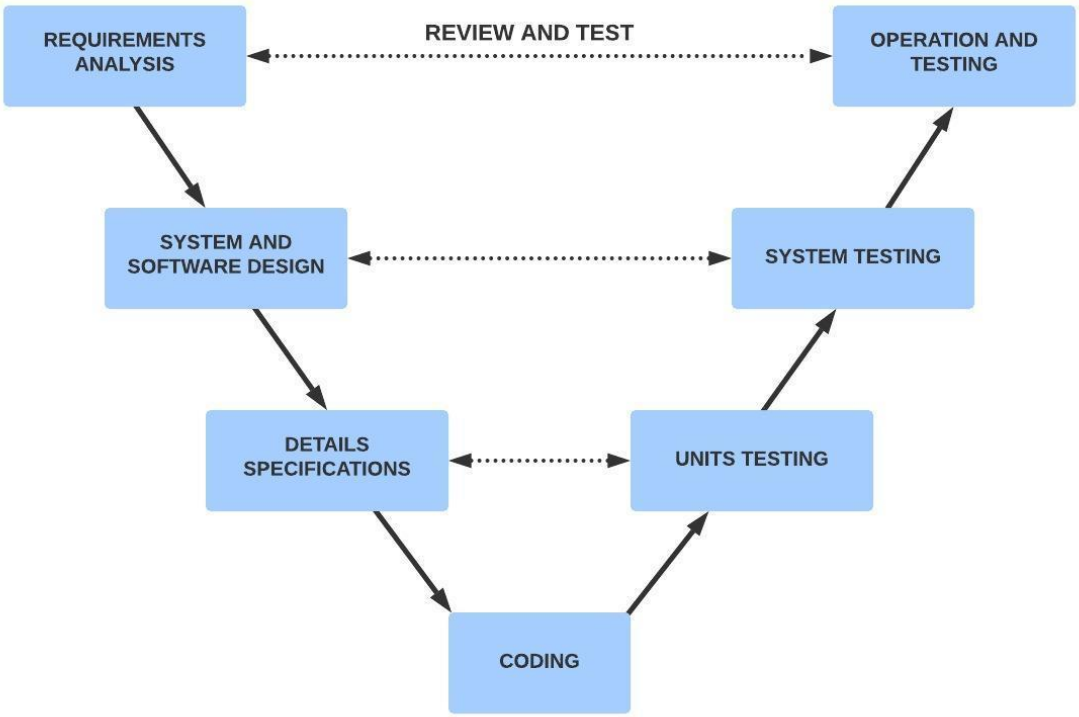


Figure 3.2: V-Model

- ✓ **Requirements Analysis**
    1. Information gathering for tabulation and election.
    2. Basic information of the event such as Title of the event, descriptions, location etc. same as information of the tabulation and election.
    3. Familiarize with category, criteria and sub criteria and its percentage.
    4. List of Judge or judge profile for the tabulation event.
    5. List of contestants, participants or members of tabulation or election event.
    6. Identify how does the computations and the securities worked for preventing vulnerability.

7. Categorize the satisfactions come to judges and voters when it comes technology.
8. Determine how the generating results can work fast by the accuracy of 100%.
9. Identify and applying the Cloud computing.

- ✓ **System and Software Design**

1. Front end: HTML5, CSS3, JavaScript
2. Back end: PHP
3. Database: MySQL
4. Online access: Cloud web hosting
5. Security: Laravel framework

- ✓ **Details Specifications**

1. The Platform to be used for PHP MySQL is XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P).
2. For security, PHP Laravel framework is used, it is an open-source PHP framework designed to make developing web apps easier and faster through built-in features.
3. For online access Cloud web hosting, it is hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely.
4. Bootstrap, a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
5. Vue.js, an open-source model–view/view model front end JavaScript framework for building user interfaces and single-page applications.
6. Formulas, to compute Category, criteria and sub criteria for the generated result.
7. FPDF, PHP class which allows to generate PDF files to print reports.

- ✓ **Coding**

  The implementation of the code for functionality:

  1. Landing page
  2. Login, registration subscription panel
  3. Administrator panel
  4. Event manager panel
  5. Voter panel
  6. Judge panel
  7. Chat box panel
  8. Result panel
  9. Reports

- ✓ **Units Testing**

  1. Testing the Landing page
  2. Testing the functionality of Login, registration subscription panel
  3. Testing the functionality of Administrator panel
  4. Testing the functionality of Event manager panel
  5. Testing the functionality of Voter panel
  6. Testing the functionality of Judge panel
  7. Testing the functionality of Chat box panel
  8. Testing and accessing the Result panel
  9. Testing the Reports and printing

- ✓ **System Testing**

  Testing, crawling, accessing, checking the whole system from back end to front end from administrator to user if the system can use.

- ✓ **Operation and Testing**

  Beta testing, and User testing, this is the part the requirements for the system is to meet the client's expectation, a testing phase involved in the finished product in a real user environment will be planned.

3.3 Flowchart

A Flowchart represents a workflow or process, it is a diagram of the sequence of movements or actions of people or things involved in a complex system or activity.

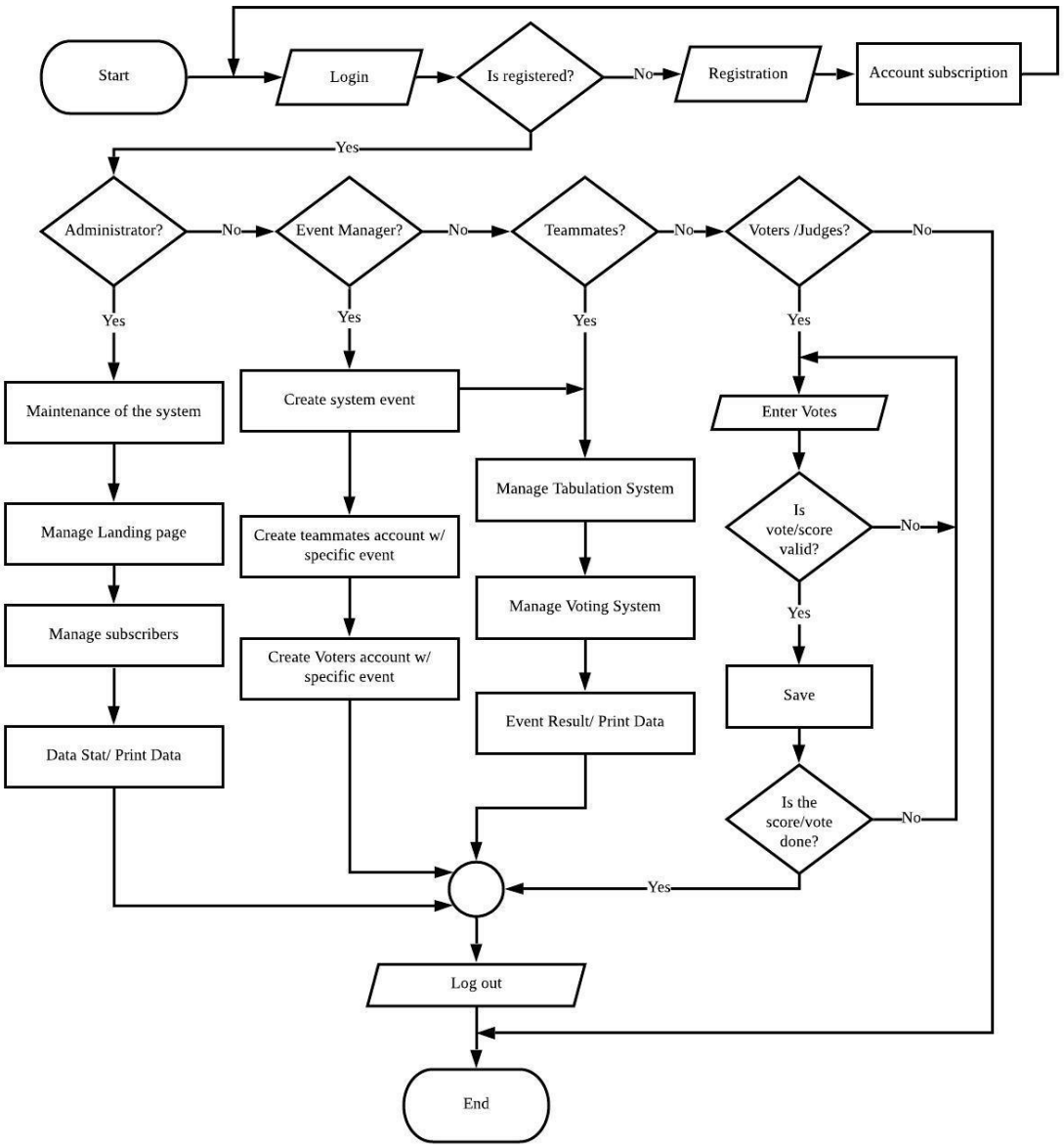


Figure 3.3: Flowchart

Figure 3.3 showed the flow of the whole system. First step is the login, if the user is not yet registered, it will be directed to the registration form. Next, if the user has already an account or successfully created one then it will be forwarded to the dashboard, wherein there will be an option of what account do the users have – Administrator, Event Manager, Teammates, or Voters/Judges. The Administrator has the access in maintenance of the system,

manage the landing page for future announcement, manage the subscribers, and can process the data ant print. The Event Manager can create and organize an event such as tabulation or election, and also can create an account the Teammates, Judges/Voters. The accessibility for Teammates has a limitation on what particular area the event managers given to them, including the managing the tabulation or voting system. The Voters/Judges can cast votes and ask for the validation and save the data to be process.

3.4 Data Flow Diagram

The Data Flow Diagram is the graphical representation of the flow of data from one component to another component in any information system.

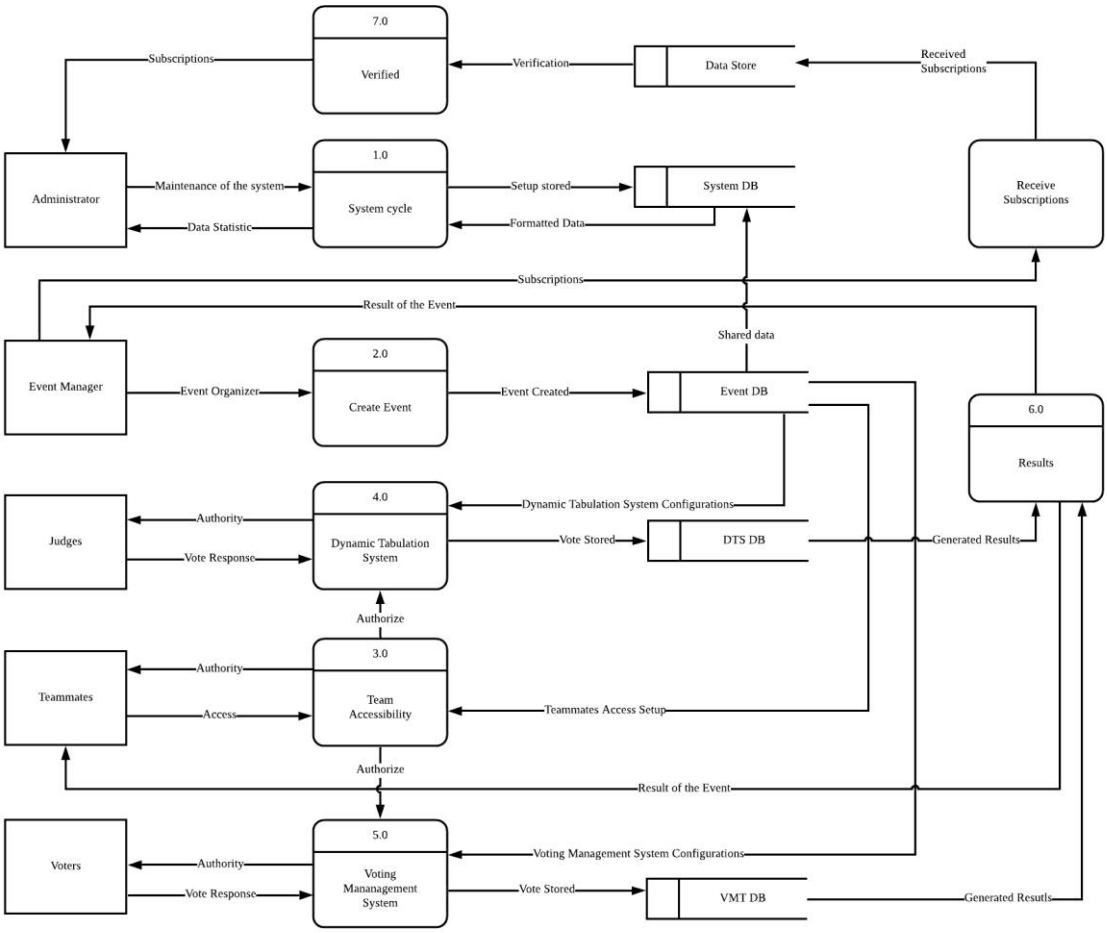


Figure 3.4: Data Flow Diagram

There are five entities in this diagram: First, the entity administrator who can manage the whole cycle of the system, also responsible for maintaining the system. Next, the entity event manager that is responsible in creating and organizing events such as tabulation or election, etc. Third, the entity teammates wherein the accessibility of this entity depends on what restrictions the event manager set in the account of a particular teammate. And lastly, the entity judges and voters that will be the one who will cast votes in response to the system.

3.5 Entity Relationship Diagram

The entity relationship diagram (ERD) shows the relationship between tables in the database using the design of “Crow's foot notation” and it shows the relationship of the tables in the database of the entire system.

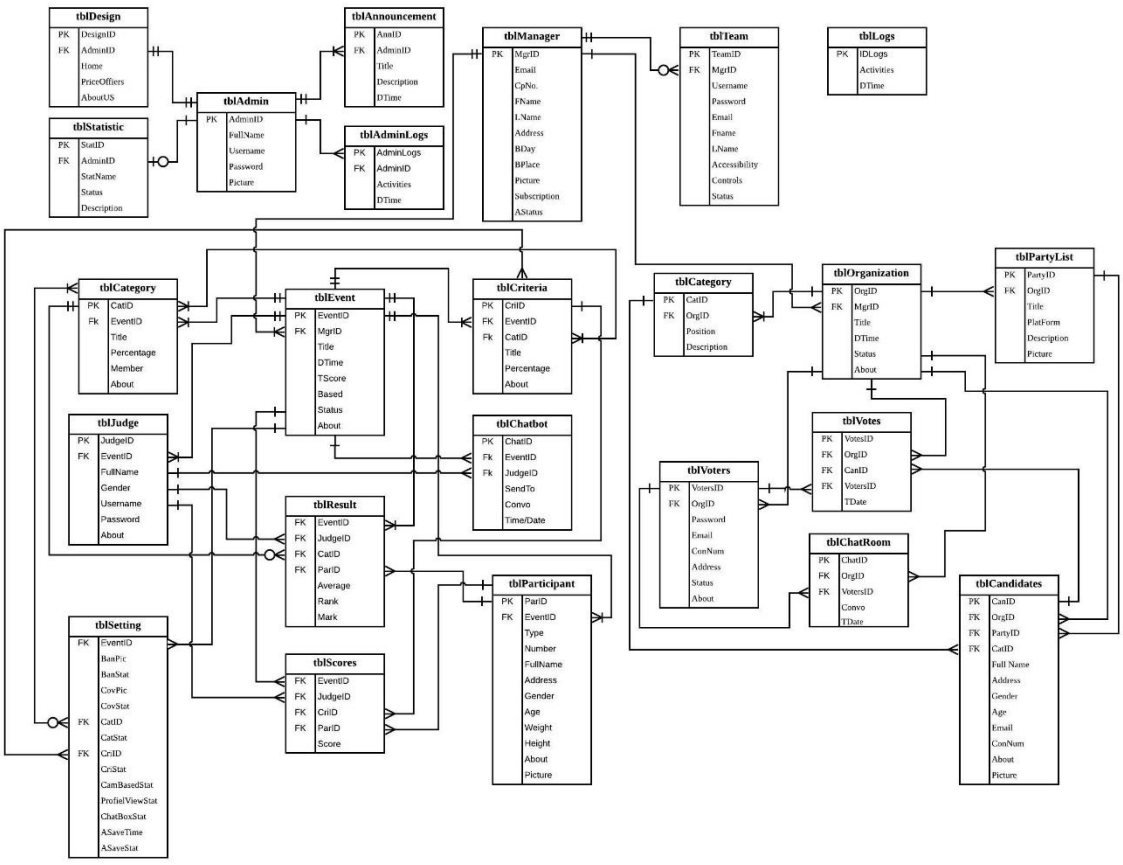


Figure 3.5: Entity Relationship Diagram

The figure 3.5 have the Fourth normal form (4NF), Basically a normalization that process of efficiently organizing the data's in a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and deletion anomalies.

The diagram will describe where located the "PK" as Primary key connected to (FK) Foreign key which telling the relationship between each other. The primary key "AdminID" in the table "tblAdmin" has a common field in the "tblAdmin, tblStatistic, tblAdminLogs, tblAnnouncement" to serves as the Foreign key. Determine the involvement of manipulating of the Landing page or design of the system that can track the last who manipulate of it, same with the relationship were the primary connected to the other table.

The "tblAdmin" exactly the super admin who manipulate and maintaining of the system. The "tblManager" with Primary Key of "MgrID" serve as Event manager that connected with the "tblEvent" which is telling to the Event manager knowing what exactly event of the Event manager have done or created, and it was also told to "tblTeam" the relationship between them, through Event manager you can dictate "tblTeam" where it can be accessed to the Event by Event Manager. Simply that the Administrator, Event manager, teammates, voters and judges are all involve to the whole function of ERD.

Proposed System Requirements

Table 3.1

Software Specification

SOFTWARE	
SOFTWARE	**DESCRIPTION**
WINDOWS 7	Operating System
XAMPP	Database Management Tool
Visual Studio Code	Code Editor
Adobe Photoshop CS6	Photo Editing Tool
Bootstrap	CSS Framework
LARAVEL	PHP Framework
VUE JS	JavaScript Framework

The table above shows the operating system and the software used to create the proposed system. The operating system that used to create this project was Windows 7. The programming language used was PHP Laravel. For the design, the system interface used Bootstrap 5, Vue JS and Adobe Photoshop. The code editor used to develop the system was Visual Studio Code, and XAMPP to create a local web server for testing and deployment.

Table 3.2

Hardware Specification

HARDWARE	
ITEM	**DESCRIPTION**
Processor	Dual-Core 2.0 GHz
RAM	2GB DDR3
Video Card	512GB
Android Phone	2GB
Camera	13MP

The table above shows the hardware specification needed used to create the proposed system. It requires a Dual-Core processor for the operating system, The Memory or RAM must be at least 2GB to run the system and Video Card at least 512Mb for good quality of the images, video and soft running of the software. The android phones with at least 2GB of memory, and a camera for the live video to the actual event.

Table 3.3

System Requirements

RECOMMENDED	
ITEM	**DESCRIPTION**
Web browser	HTML5, CSS3 and JavaScript supported.
Processor	with at least 1 gigahertz (GHz) 32-bit (x86) or 64-bit (x64) processor.
RAM	2 GB
Network	Internet connection.

The table above shows the recommended system requirements of the user that can be able to access the proposed system. The web browser application used to navigate and operate the functionalities, and it must be HTML 5, CSS3, and JavaScript supported. The processor the brain of the device with at least 1 gigahertz (GHz) 32-bit (x86) or 64-bit (x64) processor. The Memory or RAM must be at least 2GB to run the system, and Network for the Internet connections or accessibly of the proposed system.

Chapter IV

Result and Discussion

This chapter presents the system description, the result of the study. The following figures illustrate the design and its functions.

The Cloud-Based Event Tabulation System is a system that can automatically calculate the scores rendered by its users. The proposed study focuses on the flexibility of the tabulations which can use in any kind of event and can tabulate scores for each category, and generate results fast and accurate.

The Cloud-Based Event Tabulation System developed using Laravel. It is an open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller architectural pattern and based on Symphony. Laravel scores better than other web frameworks because of its advanced features and development tools that facilitate rapid web application development. It also helps website developers simplify their development process with clean and reusable code. It's one of the few frameworks that has the flexibility and rich features to make it a perfect platform for creating websites and online applications.

To give the system a responsive design, the proponents used Bootstrap. Responsive design is a way to put together a website so that it automatically scales its content and elements to match the screen size on which it viewed. Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project.

The proponents used XAMPP, the most popular software package which is used to set up a PHP development environment for web services by providing all the required software components. XAMPP is a AMP stack which stands for Cross platform, Apache, MySQL, PHP, Perl with some additional administrative software tools such as PHPMyAdmin (for database access), FileZilla FTP server, Mercury mail server and JSP Tomcat server.

Other commonly known software packages like XAMPP are WAMP, LAMP, and others. The XAMPP server is used to test PHP pages. It works as local server. It contains a MySQL database to manage or save data on a local server. XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl.

The proponents used Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

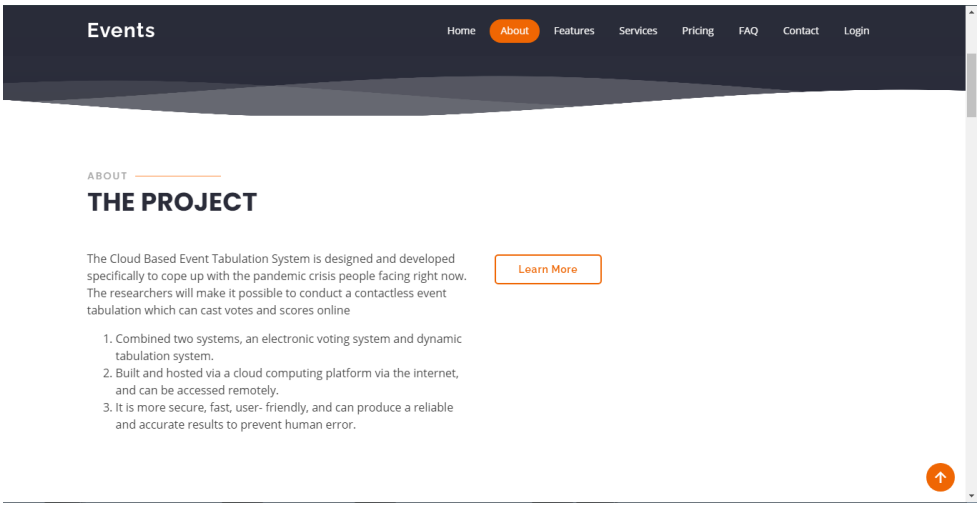


Figure 4.1 Landing Page

The Cloud-Based Event Tabulation System is a combination of two systems – Tabulation and Electronic Voting System. **Figure 4.1** shows the Landing Page providing a brief introduction about the system, including the information of the developers, announcements, features, services offered, pricing, contact, and the login button.

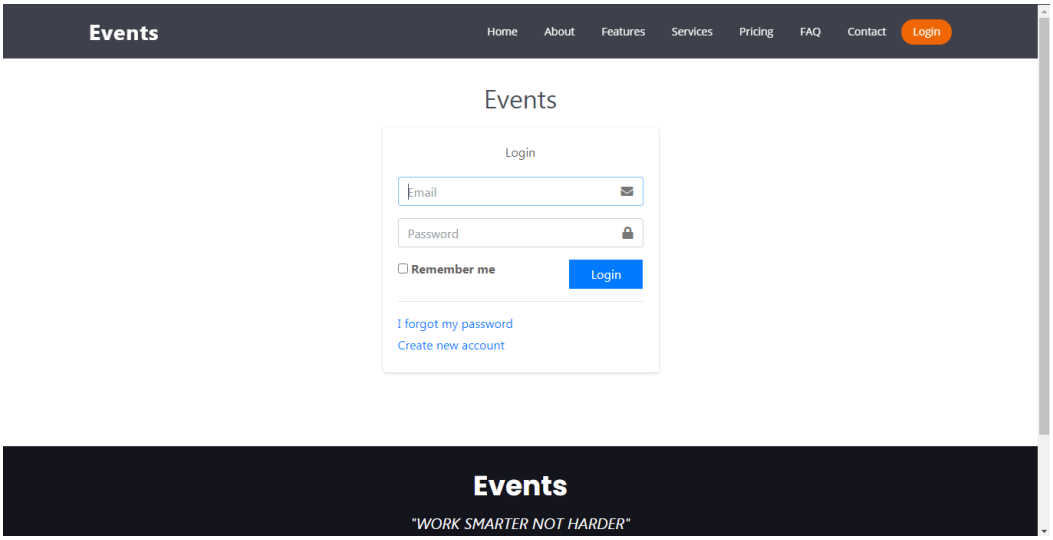


Figure 4.1.1 Event Manager Login Form

Figure 4.2 showed the login form for the Event Manager. The appointed Event Manager is required to register with the system and provide the basic and personal information needed that will be stored in database. After the procedure the event manager need to verify their account using an email. Once the account was verified the appointed event manager may proceed to login and will be able to manage and monitor the entire system.

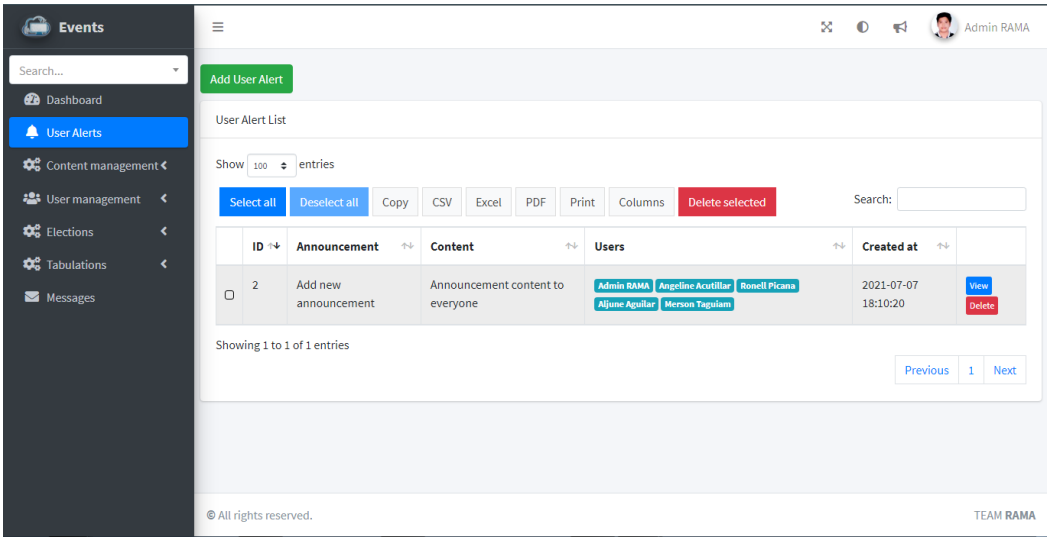


Figure 4.2.1 Event Manager DashBoard

One of the feature of the Cloud-Based Event Tabulation System is the Event Manager Dashboard. In this section, the appointed event manager has the access depending on the restrictions given by the administrator. The Event Manager may access on the User Management, Update and Deleting Users on the list, future announcements – including the description in every event, adding roles on Election and Tabulation.

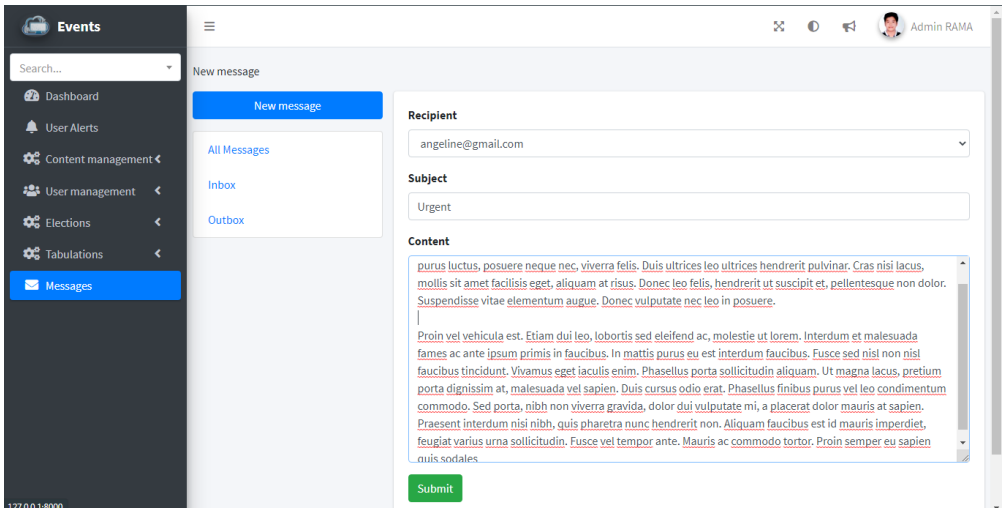


Figure 4.3 Messages

Another feature of the system is the Messages. This serves as the chat section for the judges, event managers, and the administrator. This features allow the event managers, administrator, and most especially the judges to send a message to their fellow judges or to event managers and administrators if they need some assistance.

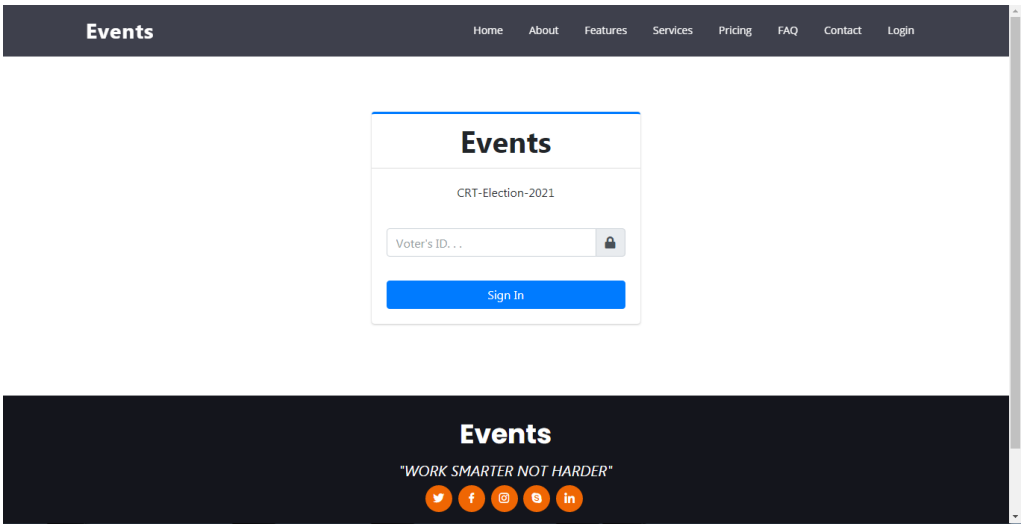


Figure 4.4 E-Voting Login

Figure 3 shows the specified login form for the voters in Electronic Voting System. Only the registered voters may log in this area, and once the voters account is finished in voting it will not be able to login again.

Figure 4.4.1 E-Voting User Dashboard

This figure shows all the candidates according to what position they are in. Every candidate has their own pictures attached to identify and familiarize the voters on the candidates. It has also the name, address, gender, contact numbers, brief description attached in every candidate photo.

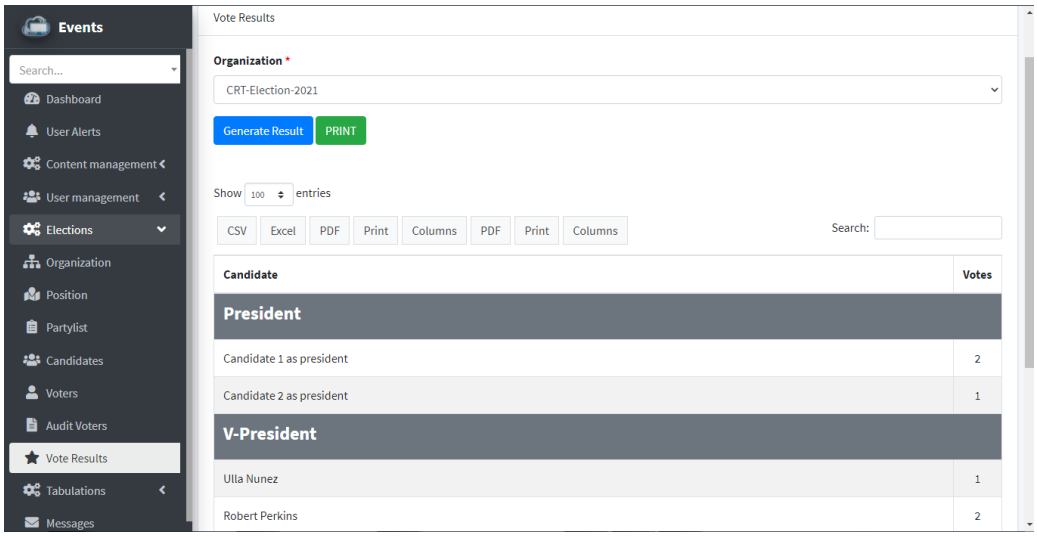


Figure 4.4.1 E-Voting Generate Results

This figure shows the generated results of the electronic voting with accuracy. It displays the winners in every position and the votes earned of every candidate. This one is the most important data needed to generate result in voting system.

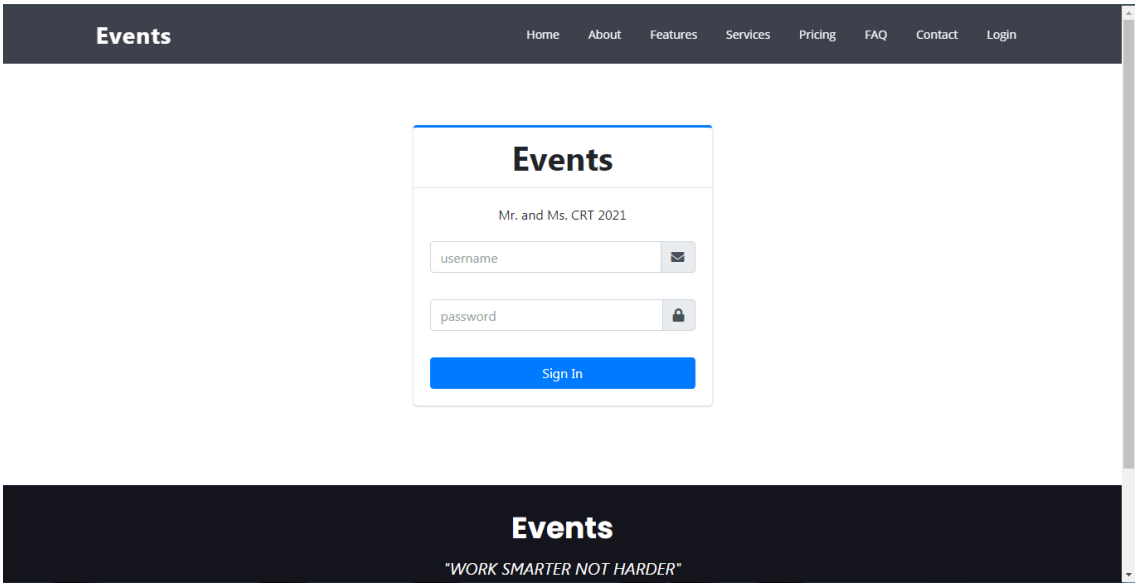


Figure 4. 5 Dynamic Tabulation Login

Figure 4.5 shows the specified login form for the judges in Dynamic Tabulation System. Each judges have their unique username and password based on the information entered in the registration.

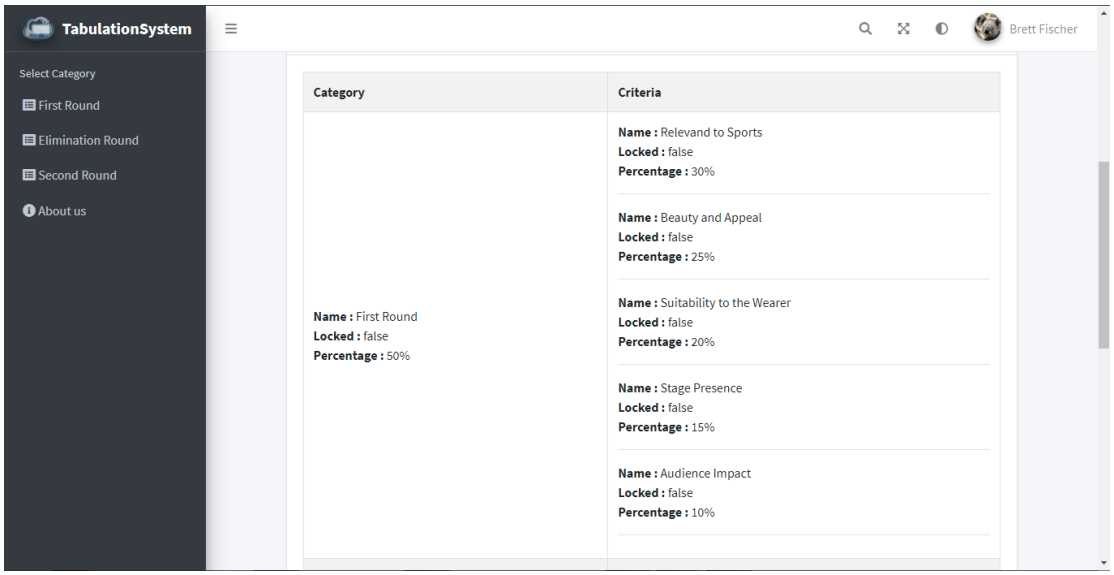


Figure 4.6 Overview of Criteria of Event

This figure shows the criteria for the current event. In every round and criteria, it has a "Locked" label that indicates if it is still accessible or not. In this figure, the percentage in every criterion also indicates to inform the judges.

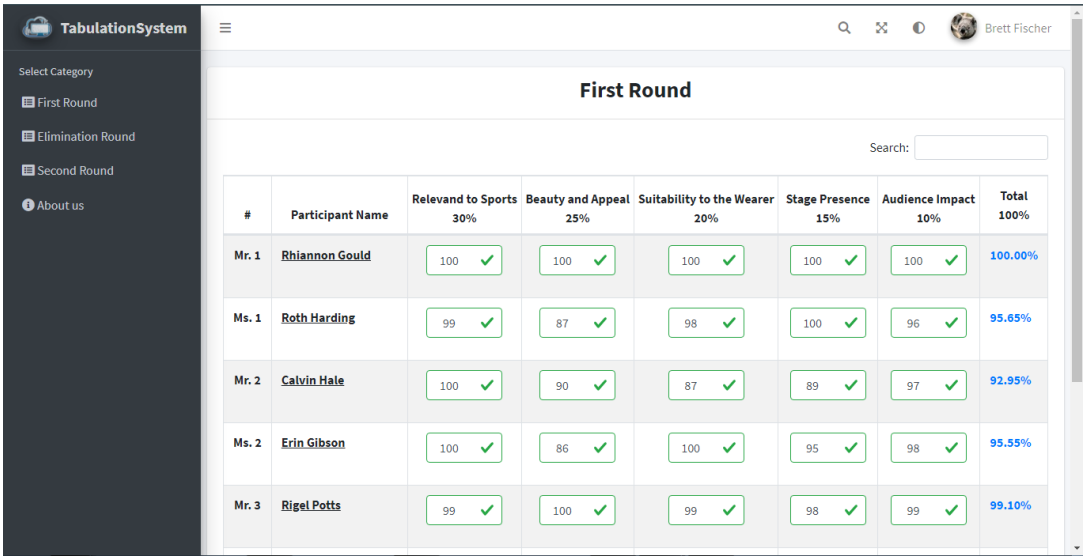


Figure 4.6 Score Board of Judges

Figure 4.6 shows the total scores gained by the candidate with accuracy. It shows the candidate scores gained in every criterion. This one of the important data needed in generating the results in every event to choose the winner.

Figure 4.7 Dynamic Tabulation User Dashboard

Figure 4.7 shows a locked sign. One of the feature of Dynamic Event Tabulation is that once any of the First, Elimination, and Second Round is done – the judges can't be access the previous round to prevent the changing of scores in every candidate.

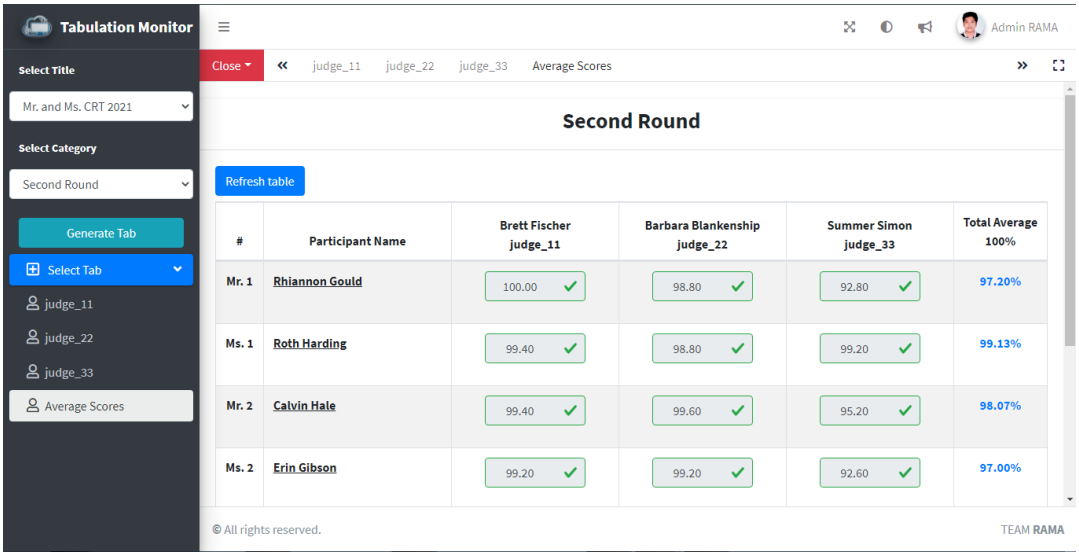


Figure 4.8 Dynamic Tabulation Monitor Mode

This figure is the Real-time judges' checker if all judges already voted. In that case, the admin will be able to monitor they are going to lock the said round for the candidates. This will help to decrease the waste votes for the candidate due to locked round.

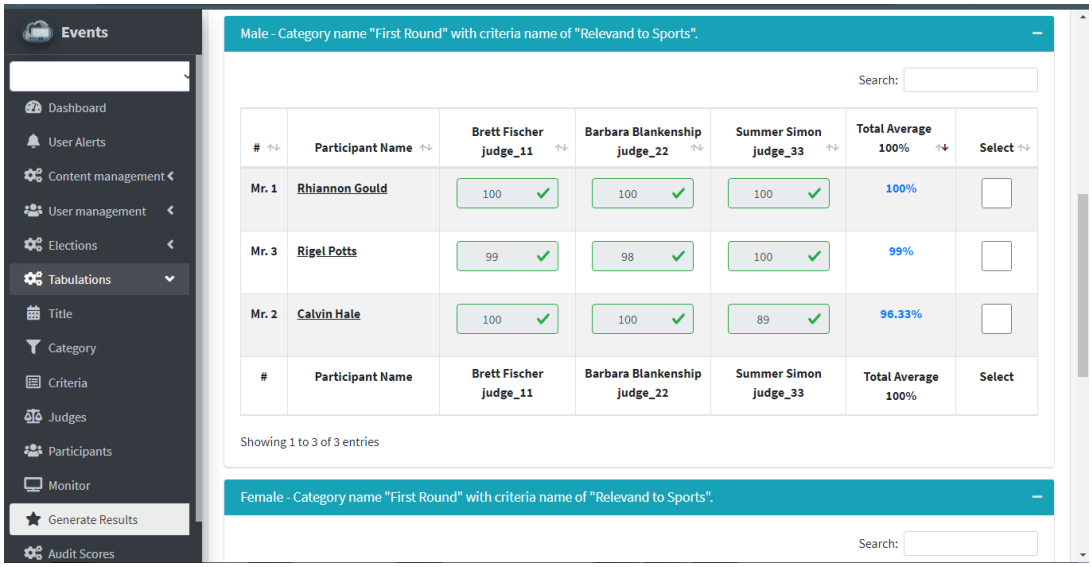


Figure 4.9 Dynamic Tabulation Generated Scores from Criteria

Figure 4.9 illustrates the generated scores from the criteria with accuracy. At the bottom part of the table, the name of the candidate who gained the highest scores from the judges will display as the winners in this round of the competition.

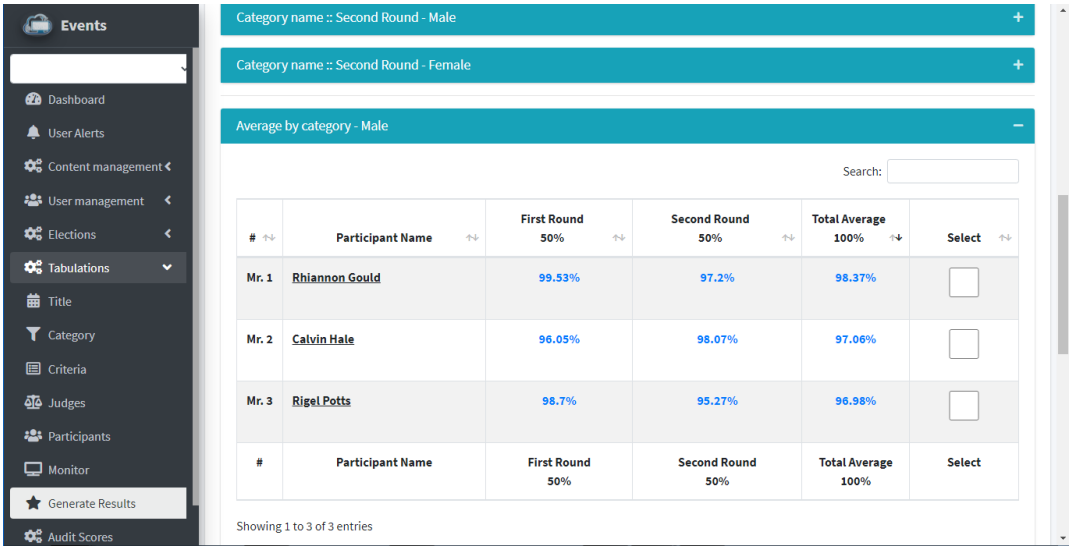


Figure 4.9 Dynamic Tabulation Generated Scores from Category

This figure shows the scores of every candidate gained in every round. It generates and validate by all the judges voting scores to verify the winner. The Event Manager has only the ability to generate result by ranking. It has also an options if the Event Manager want to export it as a pdf file or an excel file to announce the winner of the event.

Chapter V
Conclusion and Recommendation

Conclusion

Based on the result of the study the proponents were able to create a tabulation system with the following capabilities:

- The system has a secured login form for the judge and admin.
- The system can manage judges and criteria.
- The system can monitor the given scores by the judges.
- The cloud-web hosting is better than the traditional web hosting.

Recommendation

Below are the researchers' recommendation to improve the pageant tabulation system by including the following features:

- The Cloud-Based Event Tabulation System should have price list for the fund of maintenance of the system.
- The system should allow the viewer to vote for their preferred candidate for the audience impact.
- For the future researcher, it will serve as a guide for their project.

Bibliography

1. Afable, Shoven M., et al (2020) Multi-user Automated Pageant Tabulation. International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-9 Issue-3, February, 2020.

2. Campbell, Bryan A. (2011) Mobile Voting System, Online: Retrieved on December 9, 2020, from https://journals.sagepub.com/doi/abs/10.1177/1071181311551230

3. Fedena MIT (2015) Agjudgetayon – Automated pageant scoring andTabulation System, Online: Retrieved on November 28, 2020 from http://www.findbestopensource.com/product/agjudgetayon.

4. Jain, Anjali (2015) Event Managers Blog, Online: Retrieved on January 2, 2021 from https://blog.townscript.com/online-event-management-software

5. Prajapati, Vaishali (2017) Event Management System, Online: Retrieved on January 3, 2021 from https://www.academia.edu/35039153/Event_Management_System

6. Saleem, Amir (2017) Review Paper on an Event Management System, Amir Saleem et al, International Journal of Computer Science and MobileComputing, Vol.6 Issue.7, July- 2017, pg. 40-43

7. Swierenga, et al (2012) Testing Usability Performance of Accessible Voting Systems, Online: Retrieved fon December 26, 2020 from http://usability.msu.edu/research/projects/accessible-voting-systems

8. Mendoza, José Roberto (2017). *PHP Succinctly*, Syncfusion Inc, 2501 Aerial Center Parkway #200 Morrisville, NC 27560, USA.

9. Dyer, Russell (2015). *Learning MySQL and MariaDB*, O'Reilly Media, Inc. 1005 Gravenstein Highway North Sebastopol, CA95472.

10. Lambert, Matt (2016). *Learning Bootstrap 4 Second Edition*, Packt Publishing, Birmingham, United Kingdom.

Appendices

Definition of Terms

Azure: Microsoft Azure, often referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

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

Cloud server: A virtual server running in a cloud computing environment. It is built, hosted and delivered via a cloud computing platform via the internet, and can be accessed remotely.

Tabulation: Means putting a data in a table. It is the act or process of tabulating and table displaying data in a compact form.

MBPS: The acronym Mbps stands for "megabits per second." It is a measure of internet bandwidth. In simple terms, bandwidth is the download rate of your internet connection. It is the maximum speed at which you can download data from the internet onto to your computer or mobile device.

Multiplatform: The ability for a software system to run on different computer hardware and operating systems with little or no changes.

Database: a structured set of data held in a computer, especially one that is accessible in various ways.

System: composed of different modules, conceptual framework output.

Admin: the administration of a business, organization, etc.

Data Flow Diagram: a top-down analysis and design tool, which consist primarily of rectangle with rounded corners representing how data and people interact.

XAMMP (MYSQL) : is a free and open-source cross-platform web server solution stack package developed by Apache Friends,[3] consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Laravel: Open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller architectural pattern and based on Symfony.

Html: (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

JS Library: JavaScript libraries contain various functions, methods, or objects to perform practical tasks on a webpage or JS-based application.

WAMP: WampServer refers to a solution stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language.

User's Manual

Step 1. XAMPP Installation

1. The first step of the process is to download application XAMPP at official website *https://www.apachefriends.org/download.html* to run as localhost the files and database without purchasing cloud webhosting.

2. Installing XAMPP is relatively simple, once you download it go through the installation wizard until it is complete.

Step 2: Set up the Files and Database into XAMPP

Now that we have everything needed to recreate the site locally, we can start working with XAMPP. The first step is to copy the name "dev-events-orig" folder and the content is inside. Copy into your XAMPP inside of the htdocs folder, and when you open, the path folder will be look like this (C:\xampp\htdocs\ dev-events-orig\).

1. Open XAMPP Control Panel, start Apache and MySQL, then click the Admin tab on MySQL.

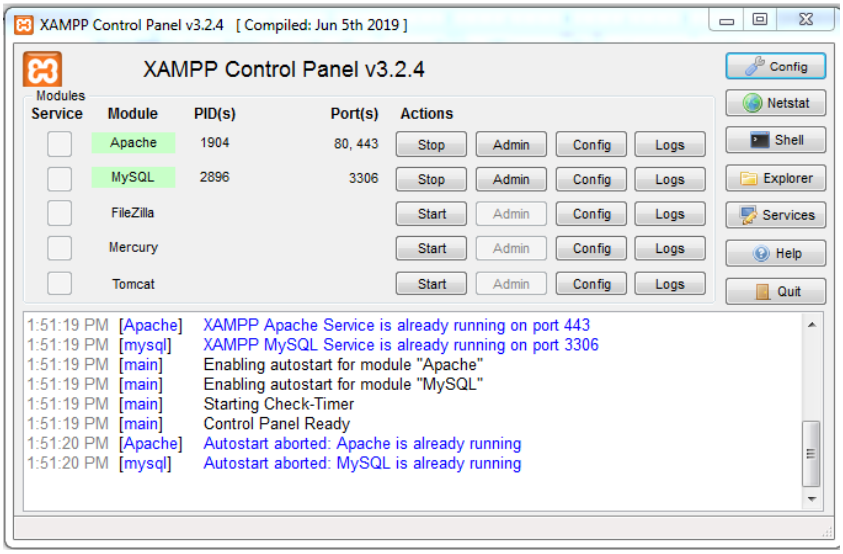


Figure 1. XAMPP Control Panel

2. Once phpMyAdmin is open, create a new database with the name events_orig.

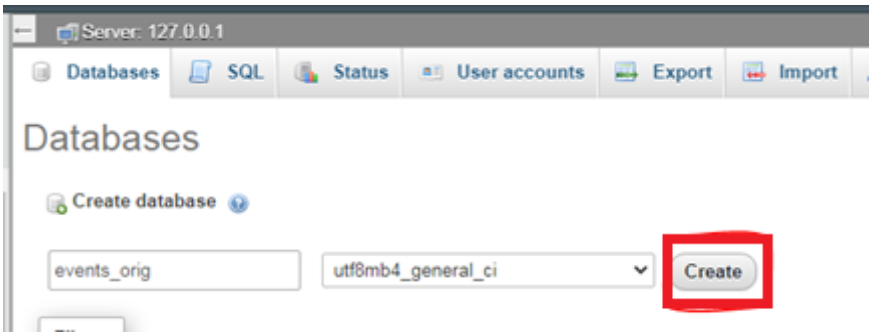


Figure 2. Create Database

3. On the sidebar, click the newly created database and click the import tab. With the import tab open, select the SQL file and press Go at the bottom. This will complete copying the database locally.

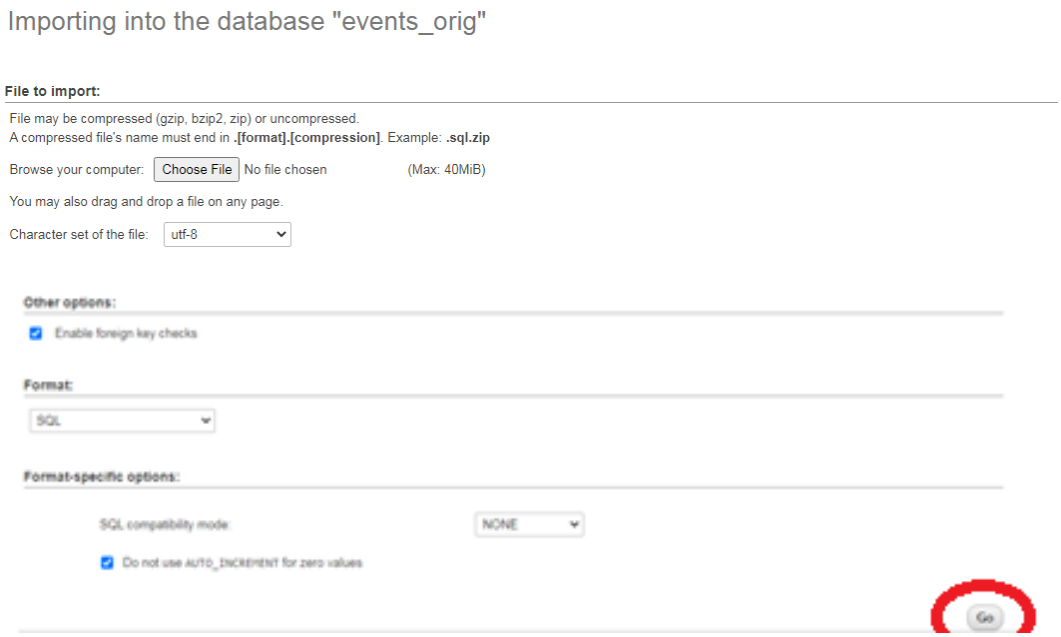


Figure 3. Import Database

Step 3: Set up Virtual Environment

Now that we all of our files and database setup, we need to create a virtual domain for the site. Open up C:\xampp\apache\conf\extra\httpd-vhosts.conf. At the bottom of the file add the following code:

```
<VirtualHost *:80>
    ServerName localhost
    DocumentRoot "C:\xampp\htdocs"
    <Directory "C:\xampp\htdocs">
    </Directory>
</VirtualHost>
 <VirtualHost *:80>
    ServerName events-tabulation.cloud.com
    DocumentRoot "C:\xampp\htdocs\dev-events-orig\public"
    <Directory "C:\xampp\htdocs\dev-events-orig\public">
       DirectoryIndex index.php
    </Directory>
</VirtualHost>
```

Next, open up C:\Windows\System32\drivers\etc\ hosts at the bottom of the file add the following code:

```
127.0.0.1 localhost
127.0.0.1 events-tabulation.cloud.com
```

Next, you may now verify the virtual environment by open this link in your browser http://events-tabulation.cloud.com

Take note: You may not be able to open this link http://events-tabulation.cloud.com/ until the setup environment is completed.

Step 4: Create New Account

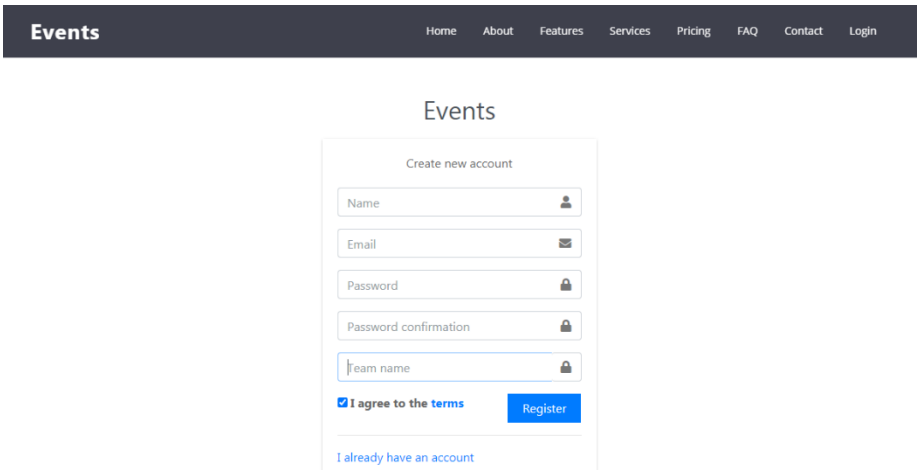


Figure 4. Event Manager Registration Form

Register as Event manager by going to http://events-tabulation.cloud.com/register. Once the registration became valid, go to your email to verify the account and it will redirect to the Admin dashboard.

Take note: It requires internet connection for registration an account.

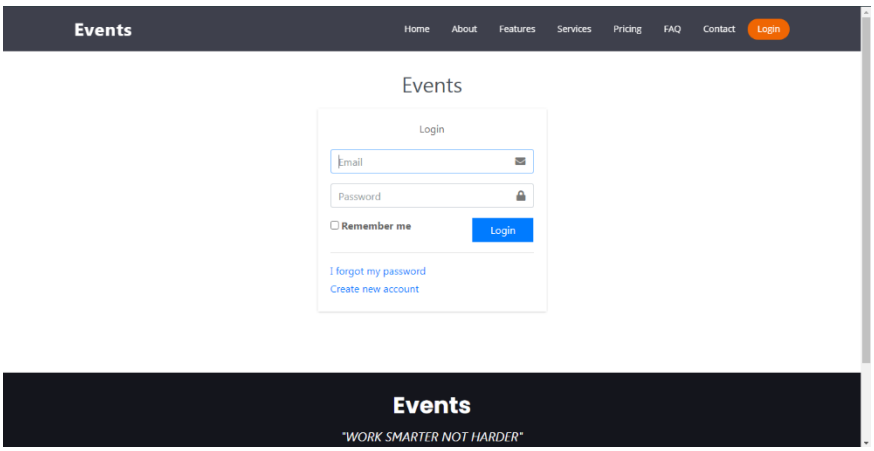


Figure 4.1. Event Manager Registration Form

Once your account verified through email verification, you may now be able to login your account by going to http://events-tabulation.cloud.com/ login.

Step 5: User management setup Roles

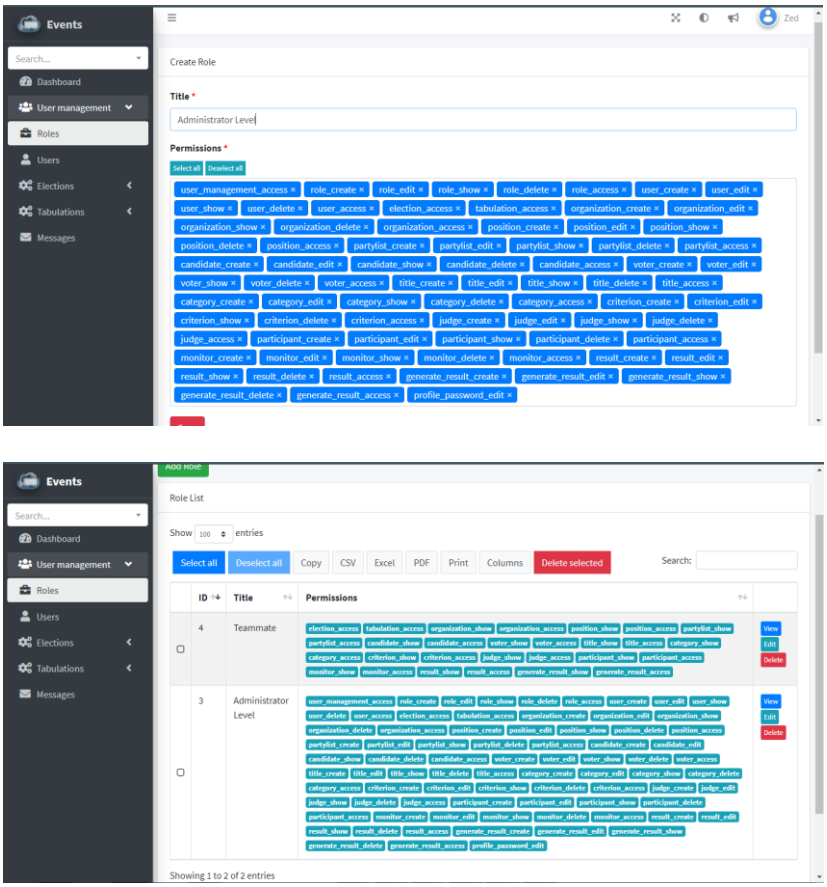
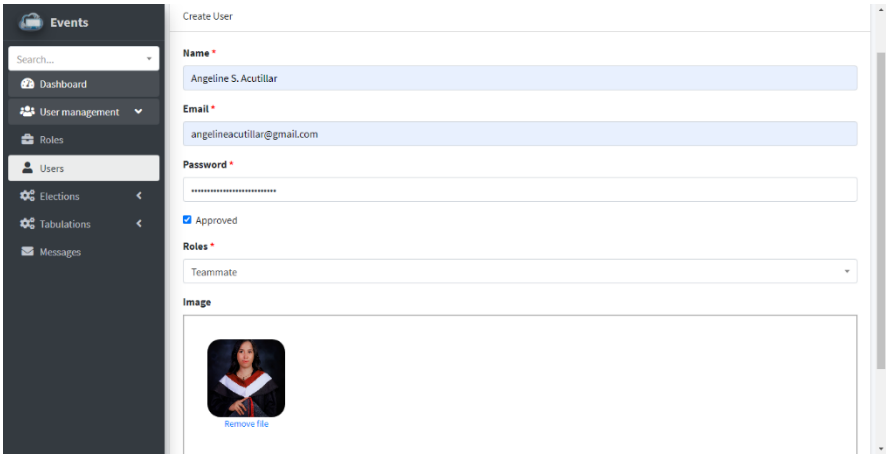


Figure 5 & 6. Roles

After login you may see the Event Manager dashboard, User management contain 2 options the Roles and Users, to add roles click add Roles, sign for a title and choose your permission, the purpose of permission is to create restriction or filter for every teammate you wanted to be filtered for accessibility of the functionalities.

Step 6: User management setup Users



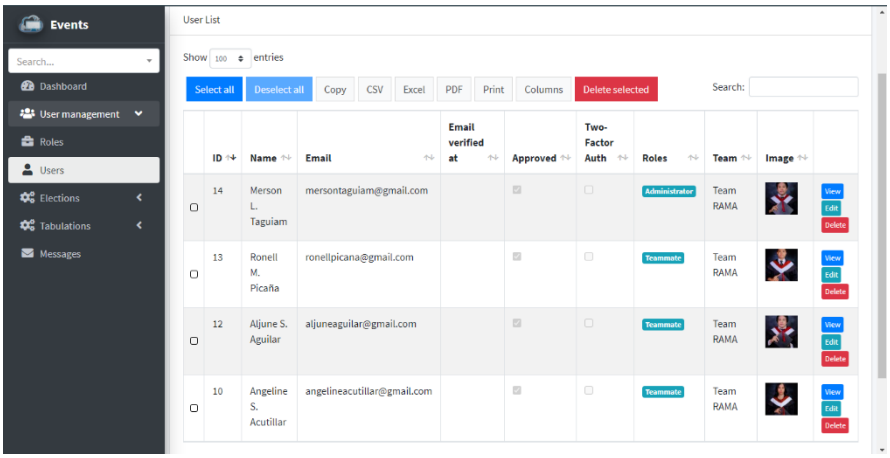


Figure 7 & 8. Users

This figure showed the users or teammates. Event Manager was able to create an unlimited account for his teammates by filling up the name, email, password, and roles, after registration the account are still need to verify by using email verification sent to the email and still required the internet connection upon registration; every account was able to see or manipulate each tabulation that has been created by Event manager.

Step 7: Elections setup Organization

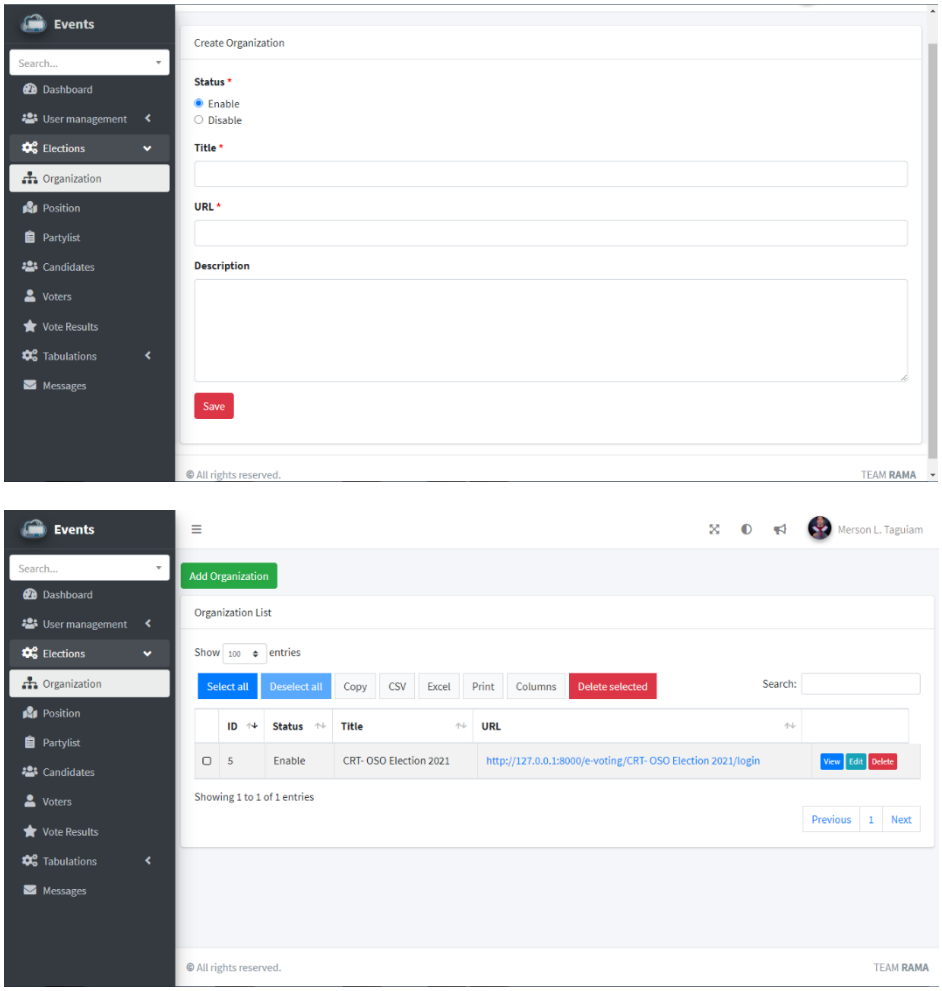


Figure 9 & 10. Organizations

This figure showed the Organization tab of Elections, by clicking Add Organization button to create for a new election, There's an option of the status by choosing enable which means active, disable which means inactive, add title for new election and customizable URL for user link.

Step 8: Elections setup Positions

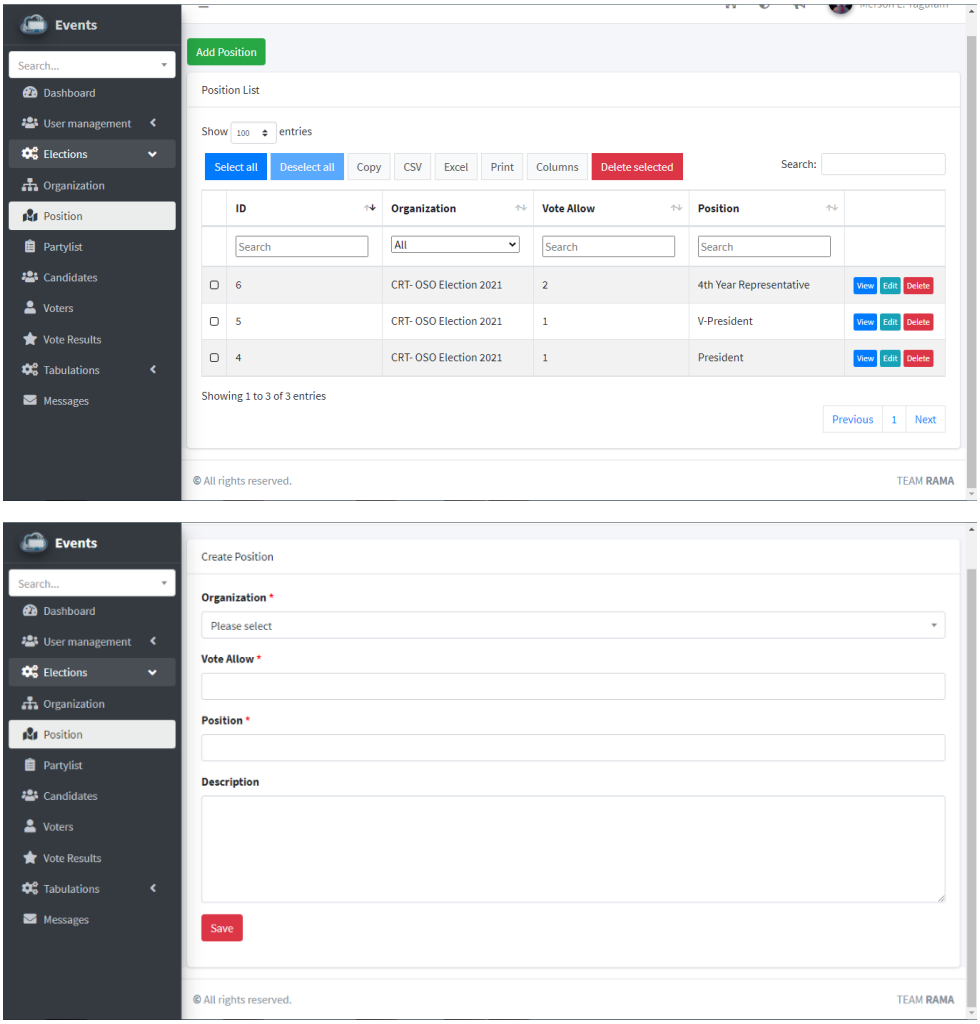
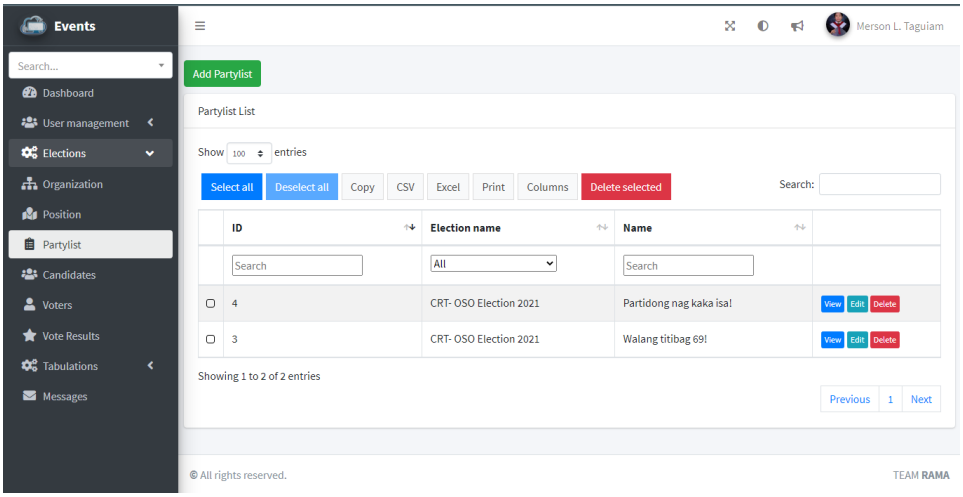


Figure 11 & 12. Positions

This figure showed the Positions tab of Elections, by clicking Add Position button to create a position to your Organization created before, by choosing which organization belong this new position you want to create, vote allow which means how many candidates allowed for this position, and description for the position.

Step 10: Elections setup Party lists



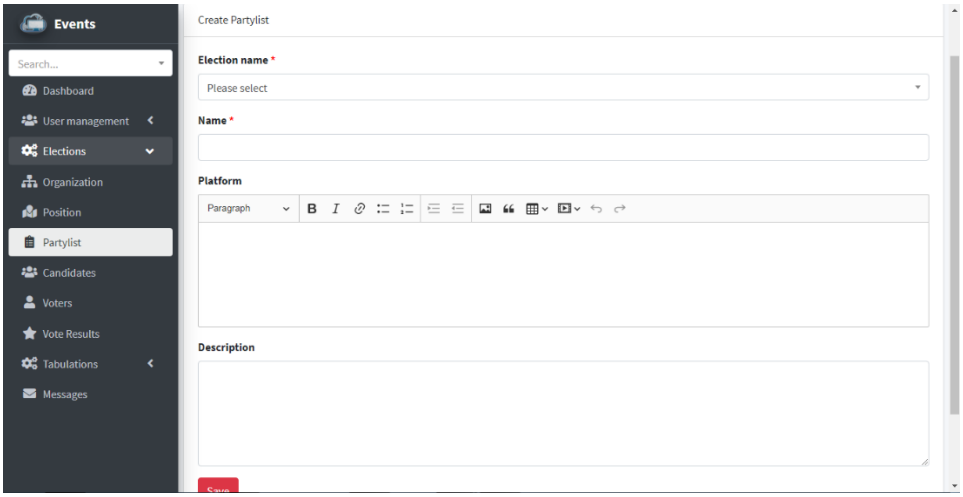


Figure 13 & 14. Party Lists

This figure showed the Party lists tab of Elections, by clicking Add Party list button, There's an option of which election belong this party list, party list name, Platform and description.

Step 11: Elections setup Candidates

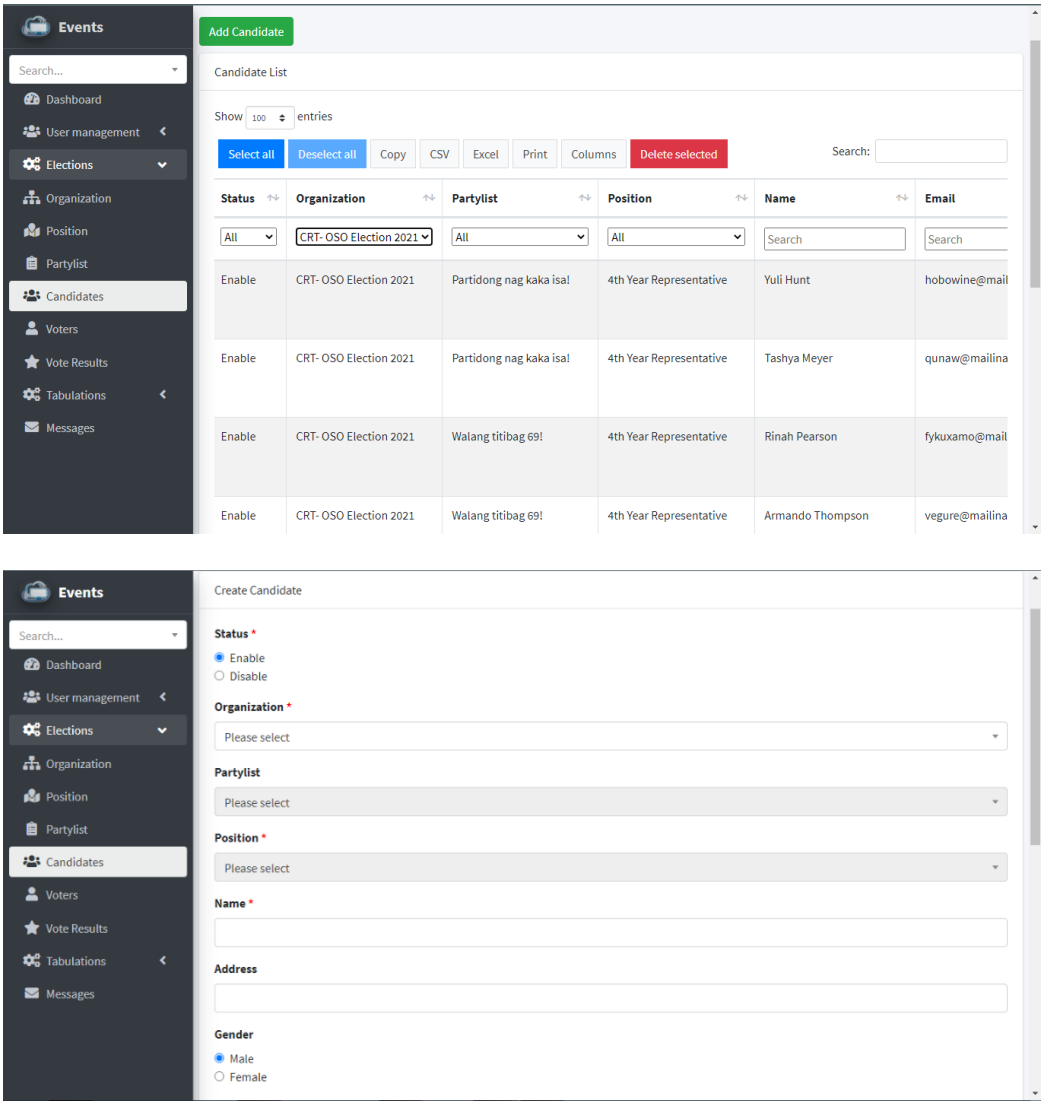


Figure 15 & 16. Candidates

This figure showed the Candidates tab of Elections, by clicking Add Candidate button, There's an option of status to determine that the candidate if enable it is active or if disable it is inactive, and also options of which organization belong this Candidate, to what position belong this candidate, the name, address, gender, email, contact #, description and images of the candidate.

Step 12: Elections setup Voters

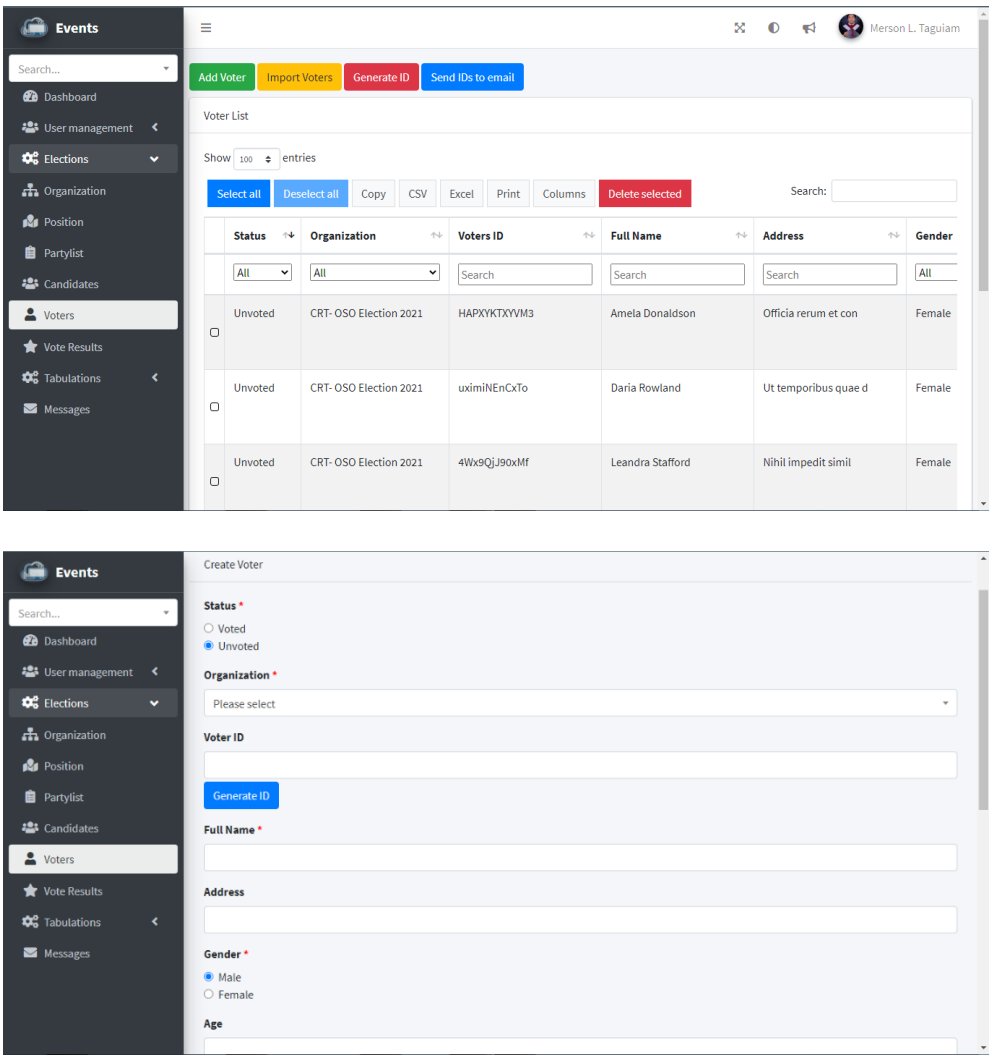
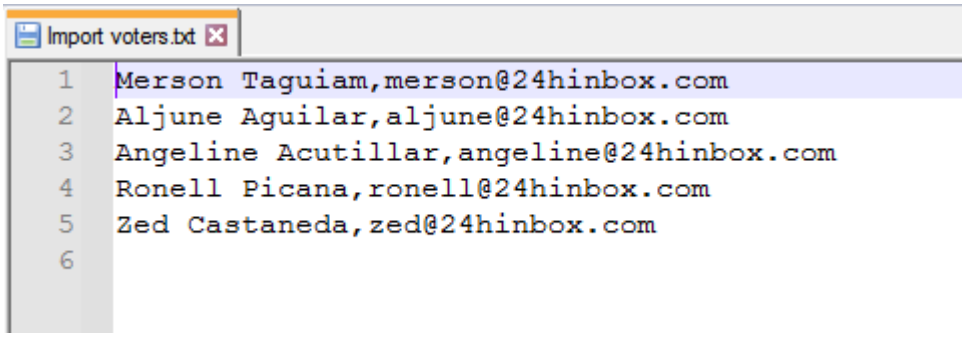


Figure 17 & 18. Voters

This figure showed the Voters tab of Elections, by clicking Add Voter, There's an option of status to determine the voter if enable it is active or if disable it is inactive, and also options of which organization belong this voter, voter's ID this is the credential of the voter to vote and to login, full name, address, gender, age, email, contact #, description, image.



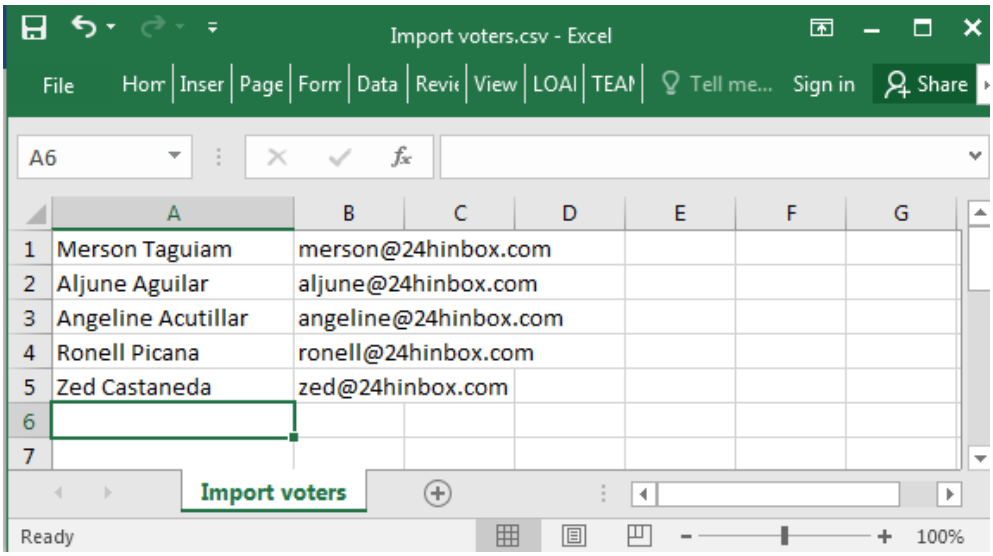


Figure 19 & 20. Voters

This is the example of the file format for .txt and .csv extension.

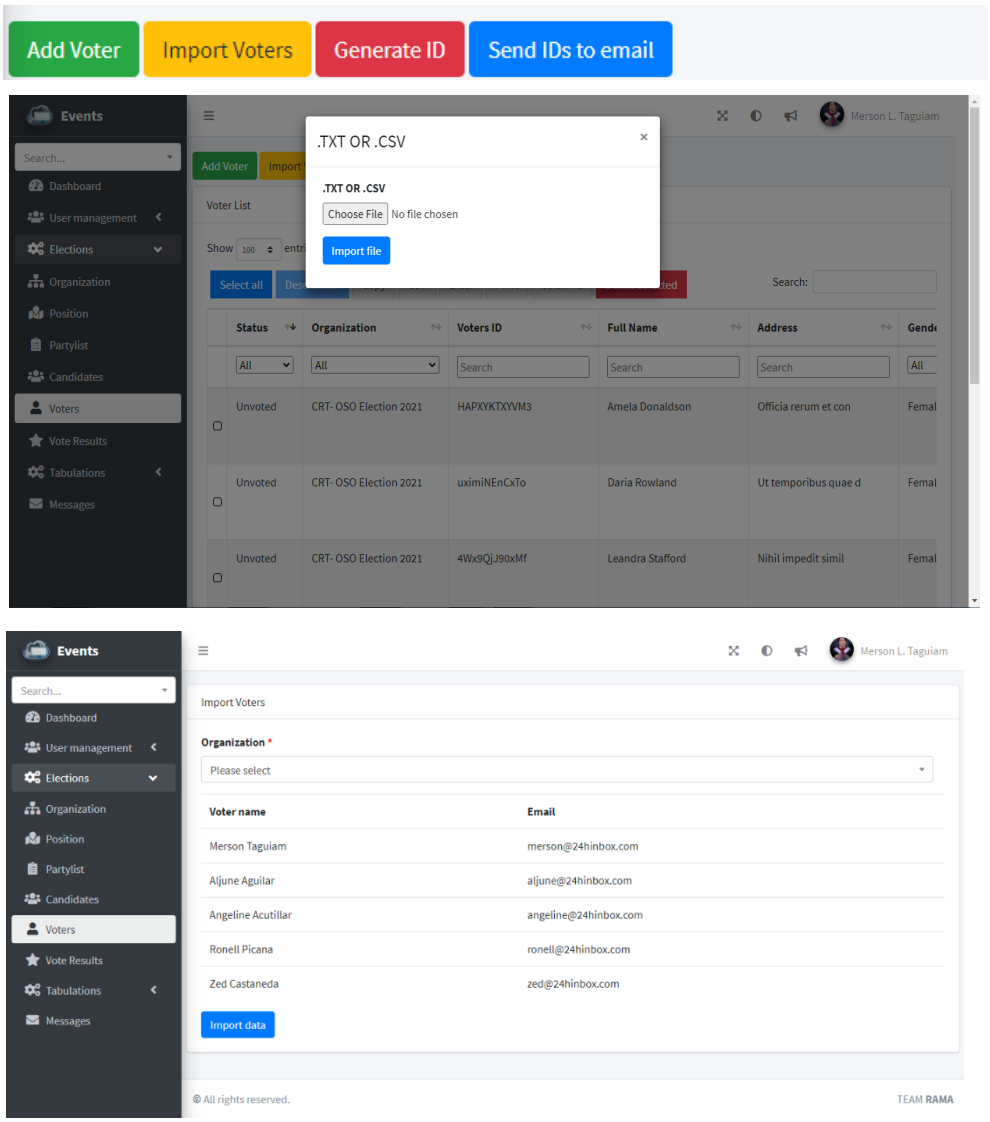
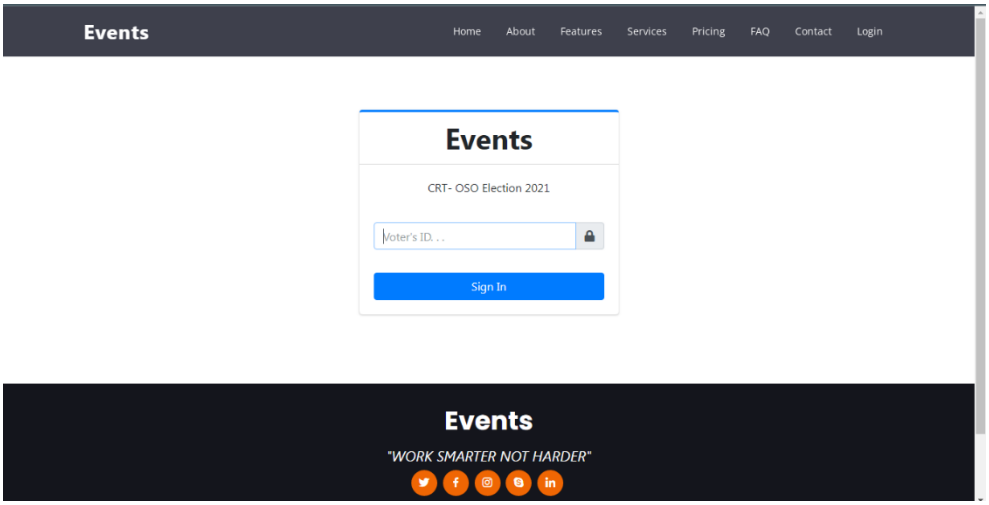


Figure 21 & 22. Voters

This figure showed the Imported Voters, by clicking Import Voters and click import file. Choose which organization belong the imported voters and click import data. After the import data you may now able to generate the id by clicking Generate ID button for the imported Voters and after generating ID's you have an option of Sending ID's button to email and it required internet connection.

Step 13: Elections - Voting System



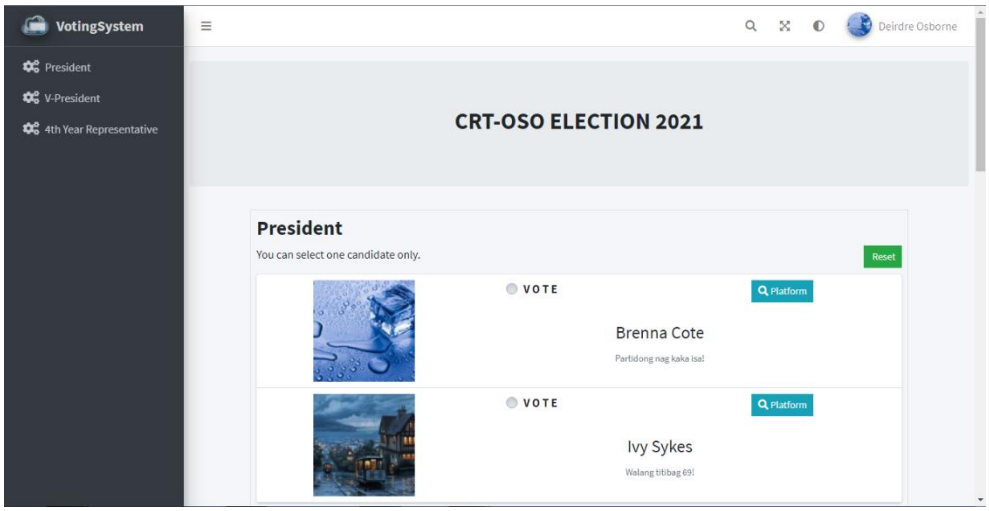


Figure 23 & 24. Login & Dashboard

This figure showed the current example of election for 2021. Just paste the valid voter's ID and click sign in, after sign in you can choose candidates and click submit at the bottom of the page.

Step 14: Elections – Vote Results

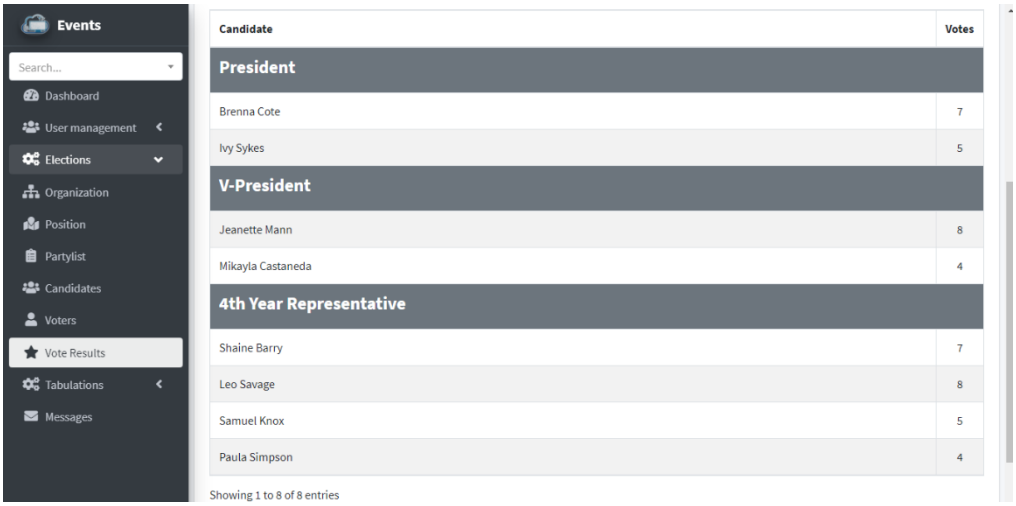
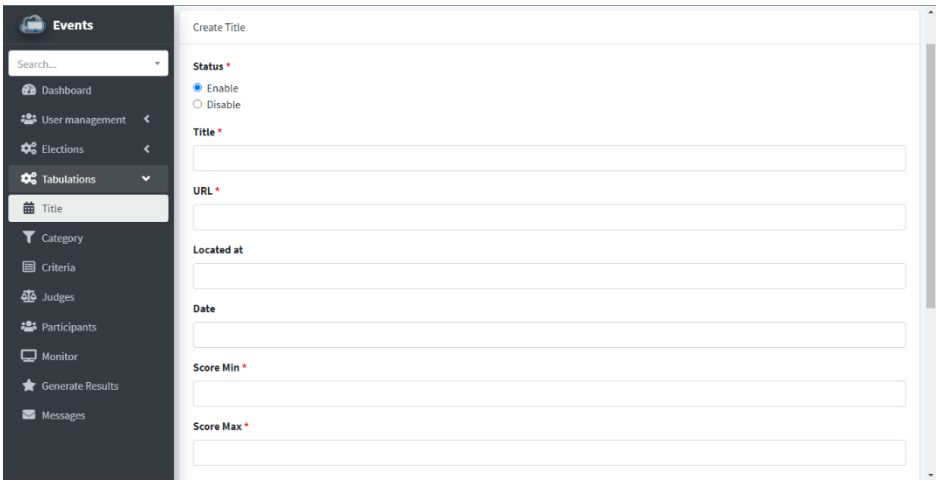


Figure 25. Vote results

This figure showed the results of the election by generating from the Event manager dashboard by choosing which organization or election and click Generate Result button.

Step 15: Tabulation setup Title



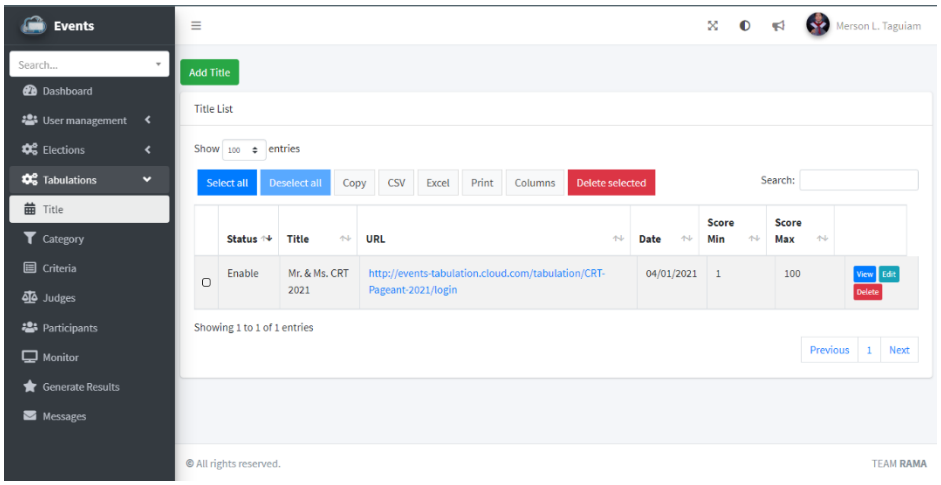


Figure 26 & 27. Title

This figure showed the tabulation tab of Elections, by clicking Add title button to create for a new tabulation event, There's an option of the status by choosing enable which means active, disable which means inactive, add title for new tabulation, customizable URL for user link, located at, date, minimum score and maximum score and the base type for the event.

Step 16: Tabulation setup Category

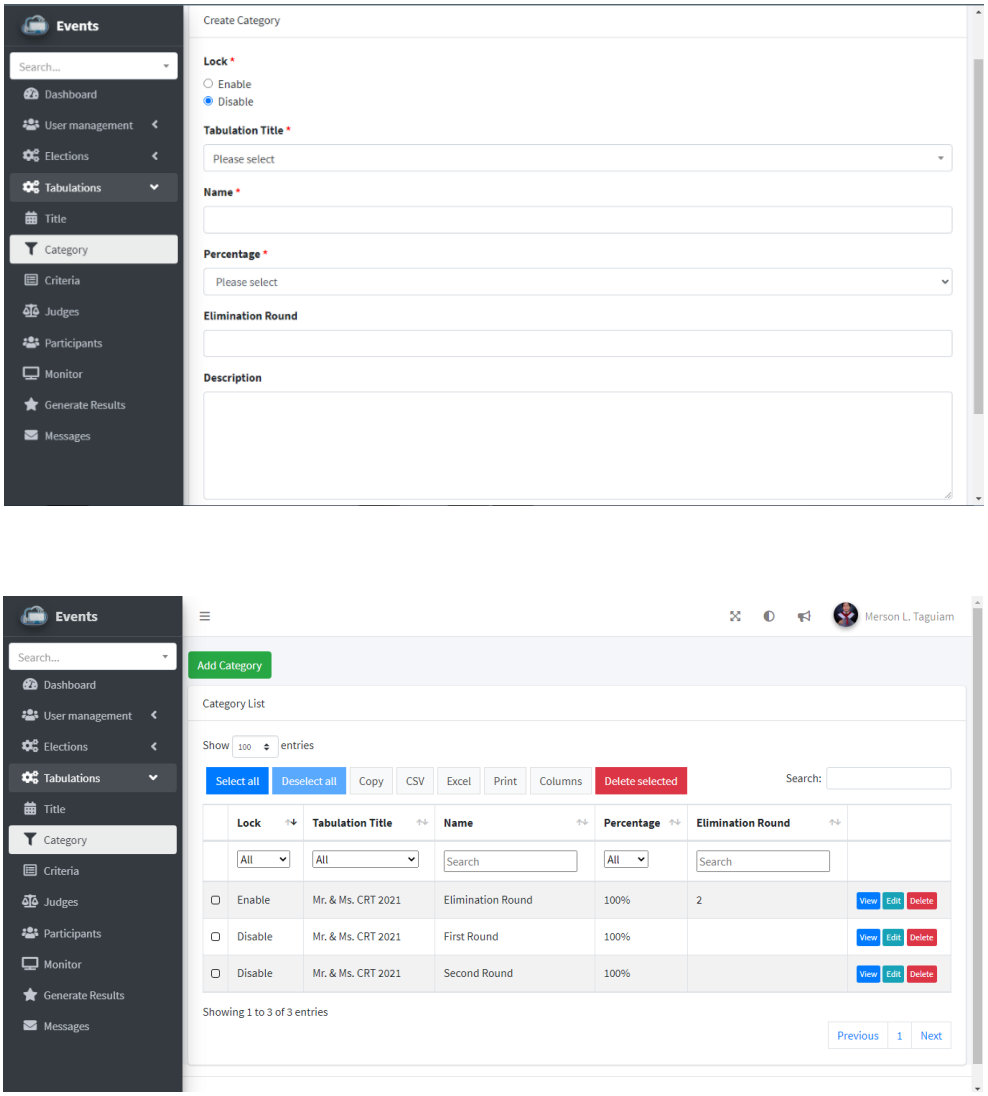
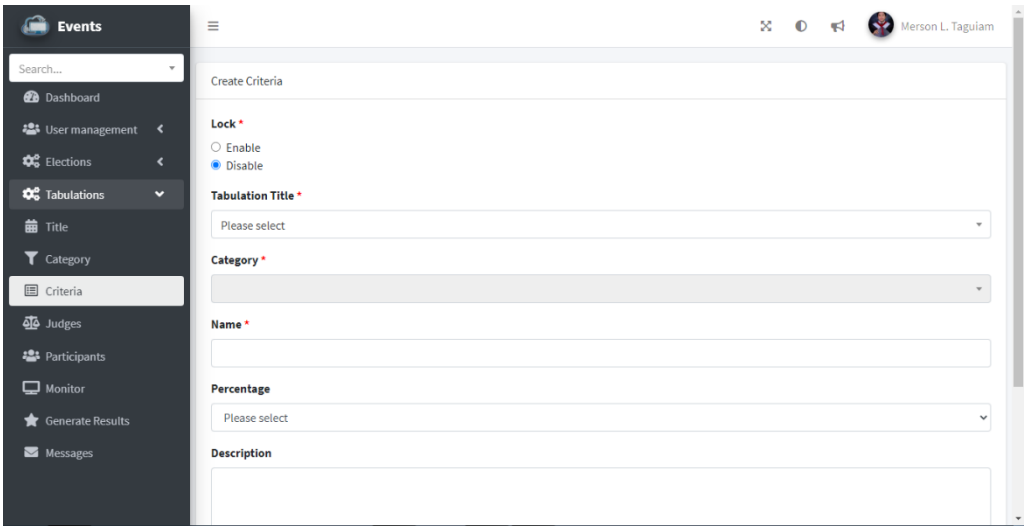


Figure 28 & 29. Categories

This figure showed the Category tab of tabulation, by clicking Add category button, There's an option of which tabulation belong this category, name of the category, percentage of the category, elimination round and description.

Step 17: Tabulation setup Criteria



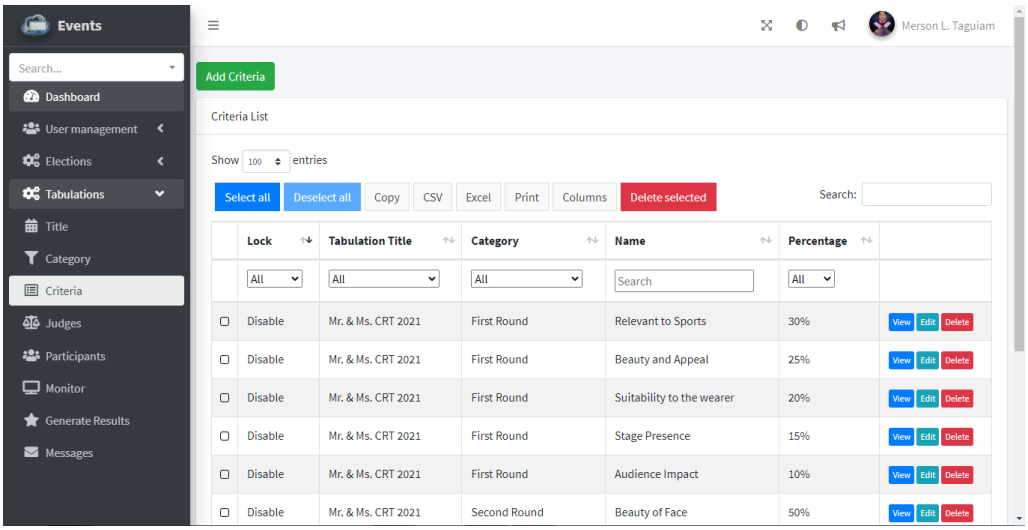
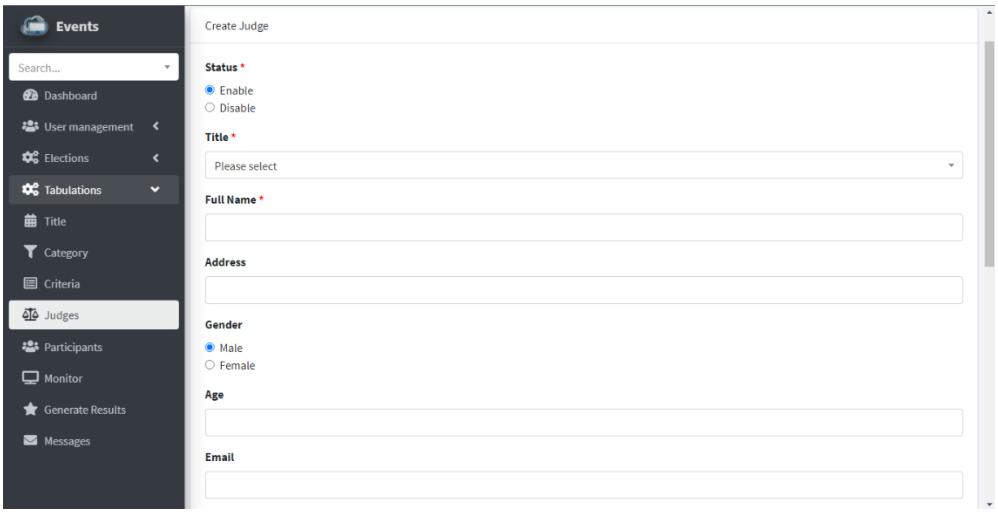


Figure 30 & 31. Criteria

This figure showed the Criteria tab of tabulation, by clicking Add criteria button, There's an option of which tabulation belong this criteria, which category belong this criteria, name of the criteria, percentage and description.

Step 18: Tabulation setup Judges



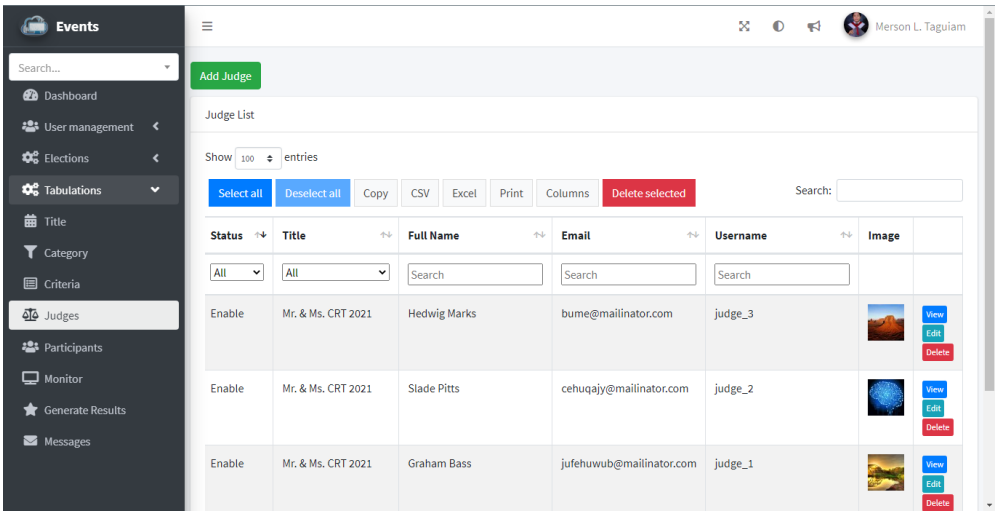
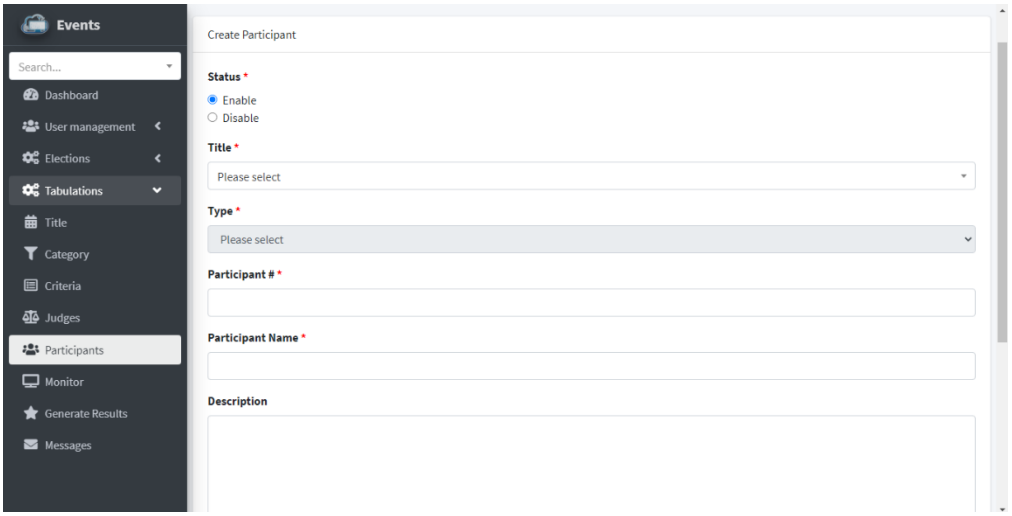


Figure 32 & 33. Judges

This figure showed the Judges tab of tabulation, by clicking Add Judge button, There's an option of which tabulation belong this judge, full name, address, age, email, gender, username and password for the login credential of the judges into the system.

Step 19: Tabulation setup Participants



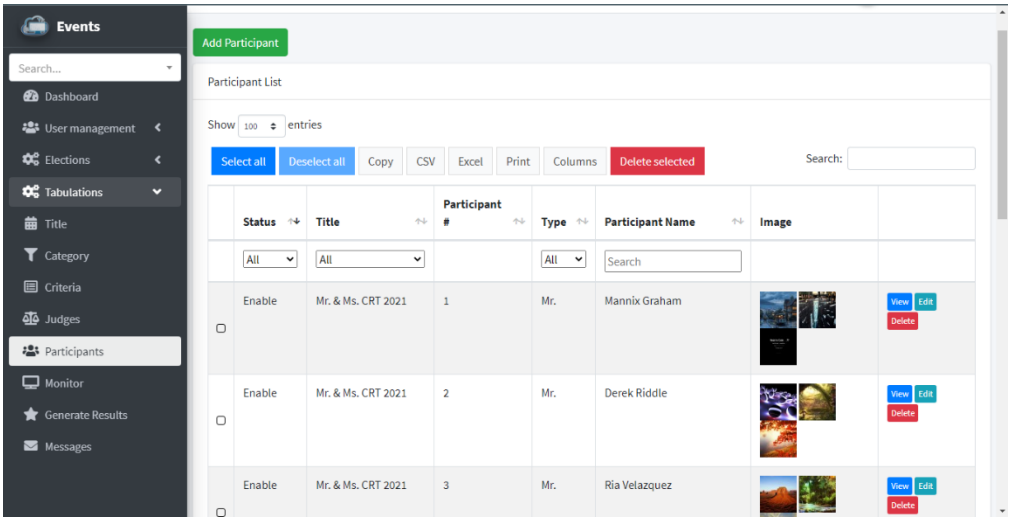


Figure 34 & 35. Participants

This figure showed the Participants tab of tabulation, by clicking Add Participant button, There's an option of which tabulation belong this judge, full name, address, age, email, gender, username and password for the login credential of the judges into the system.

Step 20: Tabulation setup Monitor

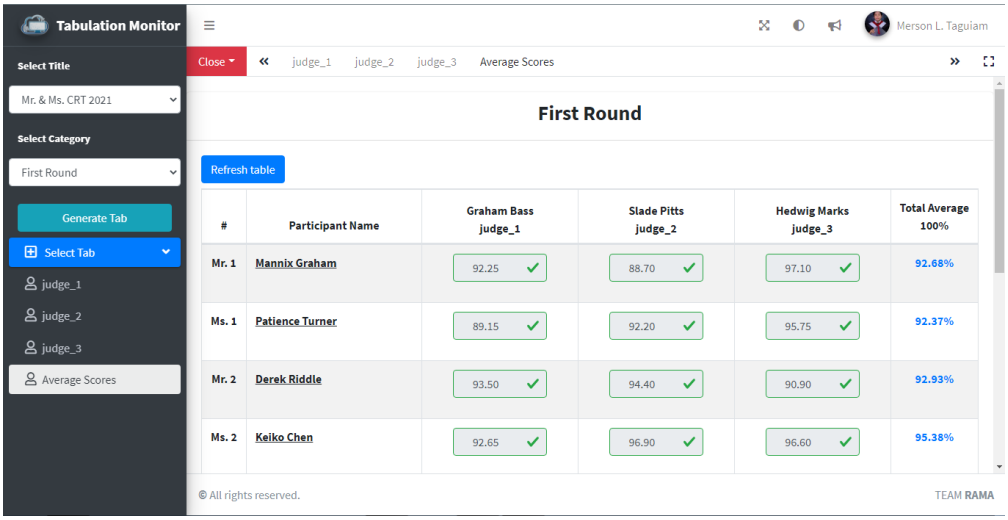


Figure 36. Monitor

This figure showed the monitor tab of tabulation, by clicking monitor it will open another tab, select title and choose which category you want to see or monitor and click generate tab to see judges, by clicking the judges tab it shows the score board of the judge.

Step 21: Tabulation setup Generate Results

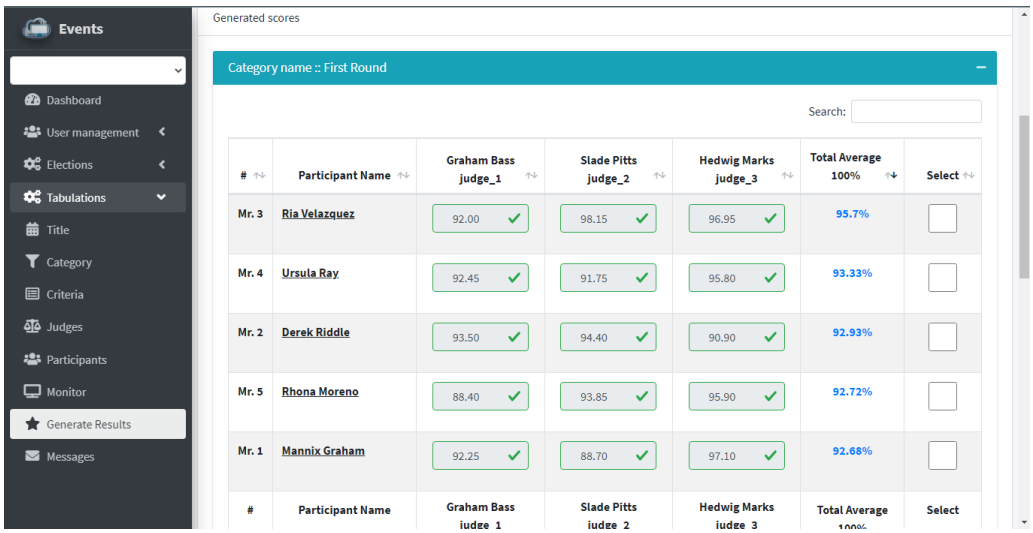


Figure 37. Generate Results

This figure showed the Overall Result by clicking the tab of Generate Results of tabulation. Choose tabulation title and click create new, choose category or criteria and click next and it will show the over all result by choosing option and clicking the generate button.

Source Code

Landingpage.blade.php

```
@extends('layouts.homenav')
@section('content')
  <!-- ======= Header Section ======= -->
  <section id="hero" class="d-flex flex-column justify-content-end align-items-center">
    <div id="heroCarousel" class="container carousel carousel-fade" data-ride="carousel">
      @foreach ($homepages as $key=>$homepage)
        @if ($key == 0)
          <!-- Slide 1 -->
          <div class="carousel-item active">
            <div class="carousel-container">
            <h2 class="animate__animated animate__fadeInDown">{{$homepage-
>title}}</span></h2>
              <p class="animate__animated fanimate__adeInUp">
                @if(!(empty($homepage->content))) {!! $homepage->content !!} @endif
              </p>
              </div></div>
        @else
          <div class="carousel-item">
            <div class="carousel-container">
            <h2 class="animate__animated animate__fadeInDown">{{$homepage-
>title}}</h2>
              <p class="animate__animated animate__fadeInUp">
                @if(!(empty($homepage->content))) {{$homepage->content}} @endif
              </p>
              </div> </div>
        @endif
      @endforeach
     <a class="carousel-control-prev" href="#heroCarousel" role="button" data-slide="prev">
      <span class="carousel-control-prev-icon bx bx-chevron-left" aria-
hidden="true"></span>
      <span class="sr-only">Previous</span>
     </a>
     <a class="carousel-control-next" href="#heroCarousel" role="button" data-slide="next">
      <span class="carousel-control-next-icon bx bx-chevron-right" aria-
hidden="true"></span>
      <span class="sr-only">Next</span>
     </a>
    </div>
    <svg class="hero-waves" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 24 150 28 "
preserveAspectRatio="none">
     <defs>
      <path id="wave-path" d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18 88-18 58 18
88 18 v44h-352z">
     </defs>
     <g class="wave1">
      <use xlink:href="#wave-path" x="50" y="3" fill="rgba(255,255,255, .1)">
     </g>
     <g class="wave2">
      <use xlink:href="#wave-path" x="50" y="0" fill="rgba(255,255,255, .2)">
     </g>
```

```
    <g class="wave3">
      <use xlink:href="#wave-path" x="50" y="9" fill="#fff">
    </g>
  </svg>
 </section><!-- End Hero -->
 <main id="main">
  <!-- ======= About Section ======= -->
  <section id="about" class="about">
   <div class="container">
    <div class="section-title" data-aos="zoom-out">
     <h2>About</h2>
     <p>The Project</p>
    </div>
    <div class="row content" data-aos="fade-up">
     <div class="col-lg-6">
        @if(!(empty($aboutpage->content))) {!! $aboutpage->content !!} @endif
     </div>
     <div class="col-lg-6 pt-4 pt-lg-0">
      <a href="#" class="btn-learn-more">Learn More</a>
     </div> </div></div>
  </section><!-- End About Section -->
  <!-- ======= Goals Features Section ======= -->
  <section id="features" class="services">
   <div class="container">
    <div class="section-title" data-aos="zoom-out">
     <h2>FEATURES</h2>
     <p>WHAT WE ARE WORKING ON</p>
    </div>
    <div class="row">
     <div class="col-lg-4 col-md-6">
      <div class="icon-box" data-aos="zoom-in-left">
       <div class="icon"><i class="fas fa-user-lock"" style="color: #1422a1;"></i></div>
       <h4 class="title"><a href="">SECURITY</a></h4>
       <p class="description ">The quality or state of being secure, free from danger,
freedom from fear or anxiety.</p>
      </div>         </div>
     <div class="col-lg-4 col-md-6 mt-5 mt-md-0">
      <div class="icon-box" data-aos="zoom-in-left" data-aos-delay="100">
       <div class="icon"><i class="fas fa-copy" style="color: #be9d0a;"></i></div>
       <h4 class="title"><a href="">ACCURACY</a></h4>
       <p class="description">The quality or state of being correct or precise.</p>
      </div> </div>
     <div class="col-lg-4 col-md-6 mt-5 mt-lg-0 ">
      <div class="icon-box" data-aos="zoom-in-left" data-aos-delay="200">
       <div class="icon"><i class="las la-file-alt" style="color: #3aafa9;"></i></div>
       <h4 class="title"><a href="">RELIABILITY</a></h4>
       <p class="description">The consistently good in quality or performance and able to
be trusted.</p>
      </div> </div>
     <div class="col-lg-4 col-md-6 mt-5">
      <div class="icon-box" data-aos="zoom-in-left" data-aos-delay="300">
       <div class="icon"><i class="fas fa-users-cog" style="color:#33a724;"></i></div>
       <h4 class="title"><a href="">USABILITY</a></h4>
       <p class="description">To provide a condition for its users to perform the tasks
safely, effectively, and efficiently.</p>
```

```
          </div> </div>
        <div class="col-lg-4 col-md-6 mt-5">
          <div class="icon-box" data-aos="zoom-in-left" data-aos-delay="400">
            <div class="icon"><i class="fas fa-desktop" style="color: #94b115;"></i></div>
            <h4 class="title"><a href="">FLEXIBILITY</a></h4>
            <p class="description">The ability to be easily modified and willingness to change
or compromise. </p>
          </div></div>
        <div class="col-lg-4 col-md-6 mt-5">
          <div class="icon-box" data-aos="zoom-in-left" data-aos-delay="500">
            <div class="icon"><i class="fas fa-chalkboard-teacher" style="color:
#4680ff;"></i></div>
            <h4 class="title"><a href="">USER-FRIENDLY</a></h4>
            <p class="description">Easy to learn, use, understand, or deal with user.</p>
          </div></div></div> </div>
    </section><!-- End Services Section -->
    </section><!-- End F.A.Q Section -->
    <!-- ======= Team Section ======= -->
    <section id="team" class="team">
      <div class="container">
        <div class="section-title" data-aos="zoom-out">
          <h2>Team</h2>
          <p>Our Hardworking Team</p>
        </div>
        <div class="row">
          <div class="col-lg-3 col-md-6 d-flex align-items-stretch">
            <div class="member" data-aos="fade-up">
              <div class="member-img">
                <img src="{{asset('assets')}}/img/team/team-1.jpg" class="img-fluid" alt="">
                <div class="social">
                  <a href=""><i class="icofont-twitter"></i></a>
                  <a href=""><i class="icofont-facebook"></i></a>
                  <a href=""><i class="icofont-instagram"></i></a>
                  <a href=""><i class="icofont-linkedin"></i></a>  </div>
              </div>
              <div class="member-info">
                <h4>Merson L. Taguiam</h4>
                <span>Project Manager</span>
              </div>  </div> </div>
          <div class="col-lg-3 col-md-6 d-flex align-items-stretch">
            <div class="member" data-aos="fade-up" data-aos-delay="100">
              <div class="member-img">
                <img src="{{asset('assets')}}/img/team/team-2.jpg" class="img-fluid" alt="">
                <div class="social">
                  <a href=""><i class="icofont-twitter"></i></a>
                  <a href=""><i class="icofont-facebook"></i></a>
                  <a href=""><i class="icofont-instagram"></i></a>
                  <a href=""><i class="icofont-linkedin"></i></a>
                </div> </div>
              <div class="member-info">
                <h4>Angeline S. Acutillar</h4>
                <span>Technical Writer</span>
              </div> </div> </div>
          <div class="col-lg-3 col-md-6 d-flex align-items-stretch">
            <div class="member" data-aos="fade-up" data-aos-delay="200">
```

```
            <div class="member-img">
              <img src="{{asset('assets')}}/img/team/team-3.jpg" class="img-fluid" alt="">
              <div class="social">
                <a href=""><i class="icofont-twitter"></i></a>
                <a href=""><i class="icofont-facebook"></i></a>
                <a href=""><i class="icofont-instagram"></i></a>
                <a href=""><i class="icofont-linkedin"></i></a>
              </div>  </div>
            <div class="member-info">
              <h4>Aljune S. Aguilar</h4>
              <span>System Analyst</span>
            </div>
          </div>
        </div>         <div class="col-lg-3 col-md-6 d-flex align-items-stretch">
          <div class="member" data-aos="fade-up" data-aos-delay="300">
            <div class="member-img">
              <img src="{{asset('assets')}}/img/team/team-4.jpg" class="img-fluid" alt="">
              <div class="social">
                <a href=""><i class="icofont-twitter"></i></a>
                <a href=""><i class="icofont-facebook"></i></a>
                <a href=""><i class="icofont-instagram"></i></a>
                <a href=""><i class="icofont-linkedin"></i></a>
              </div>  </div>
            <div class="member-info">
              <h4>Ronell M. Picaña</h4>
              <span>Q.A. Tester</span>
            </div> </div> </div> </div></div></main>
<!-- End #main -->
@endsection
```

Login.blade.php

```
@extends('layouts.homenav')
@section('styles')
<!-- Font Awesome -->
<link href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}" rel="stylesheet" />
<link href="{{ asset('css/adminlte.min.css') }}" rel="stylesheet" />
@endsection
@section('content')
<div class="container py-5 my-5">
   <div class="row justify-content-center">
      <div class="login-box">
         <div class="login-logo">
            <a class="font-weight-bolder" href="{{url('/')}}">{{ trans('panel.site_title') }}</a>
</div>
         <!-- /.login-logo -->
         <div class="card">
            <div class="card-body login-card-body">
               <p class="login-box-msg">
                  {{ trans('global.login') }}
               </p>
               @if(session()->has('message'))
               <p class="alert alert-info">
                  {{ session()->get('message') }}
               </p>
               @endif
               <form action="{{ route('login') }}" method="POST">
                  @csrf
                  <div class="input-group mb-3">
                     <input id="email" type="email"
                        class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" required
                        autocomplete="email" autofocus placeholder="{{
trans('global.login_email') }}"
                        name="email" value="{{ old('email', null) }}">
                     <div class="input-group-append">
                        <div class="input-group-text">
                           <span class="fas fa-envelope"></span>
                        </div>
                     </div>
                     @if($errors->has('email'))
                     <div class="invalid-feedback">
                        {{ $errors->first('email') }}
                     </div>
                     @endif
                  </div>
                  <div class="input-group mb-3">
                     <input id="password" type="password"
                        class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}"
name="password"
                        required placeholder="{{ trans('global.login_password') }}">
                     <div class="input-group-append">
                        <div class="input-group-text">
                           <span class="fas fa-lock"></span>
                        </div>
                     </div>
```

```
                    @if($errors->has('password'))
                    <div class="invalid-feedback">
                        {{ $errors->first('password') }}
                    </div>
                    @endif
                </div>
                <div class="row">
                    <div class="col-8">
                        <div class="icheck-primary">
                            <input type="checkbox" id="remember">
                            <label for="remember">
                                {{ trans('global.remember_me') }}
                            </label>
                        </div>
                    </div>
                    <!-- /.col -->
                    <div class="col-4">
                        <button type="submit" class="btn btn-primary btn-block btn-flat">
                            {{ trans('global.login') }}
                        </button>
                    </div>
                    <!-- /.col -->
                </div>
            </form>
            <div class="social-auth-links text-center mb-3">
                <hr>
                {{-- <p>- OR -</p>
                <a href="#" class="btn btn-block btn-primary">
                    <i class="fab fa-facebook mr-2"></i> Sign in using Facebook
                </a>
                <a href="#" class="btn btn-block btn-danger">
                    <i class="fab fa-google-plus mr-2"></i> Sign in using Google+
                </a> --}}
            </div>
            <!-- /.social-auth-links -->
            @if(Route::has('password.request'))
            <p class="mb-1">
                <a href="{{ route('password.request') }}">
                    {{ trans('global.forgot_password') }}
                </a>
            </p>
            @endif
            <p class="mb-1">
                <a class="text-center" href="{{ route('register') }}">
                    {{ trans('global.register') }}
                </a>
            </p>
        </div>
        <!-- /.login-card-body -->
    </div>
</div>
<!-- /.login-box -->
</div>
</div>
@endsection
```

Announcement.blade.php

```
@extends('layouts.admin')
@section('content')
@can('user_alert_create')
    <div style="margin-bottom: 10px;" class="row">
        <div class="col-lg-12">
            <a class="btn btn-success" href="{{ route('admin.user-alerts.create') }}">
                {{ trans('global.add') }} {{ trans('cruds.userAlert.title_singular') }}
            </a>
        </div>
    </div>
@endcan
<div class="card">
    <div class="card-header">
        {{ trans('cruds.userAlert.title_singular') }} {{ trans('global.list') }}
    </div>
    <div class="card-body">
        <table class=" table table-bordered table-striped table-hover ajaxTable datatable
datatable-UserAlert">
            <thead>
                <tr>
                    <th width="10">
                    </th>
                    <th>
                        {{ trans('cruds.userAlert.fields.id') }}
                    </th>
                    <th>
                        {{ trans('cruds.userAlert.fields.alert_text') }}
                    </th>
                    <th>
                        {{ trans('cruds.userAlert.fields.alert_link') }}
                    </th>
                    <th>
                        {{ trans('cruds.userAlert.fields.user') }}
                    </th>
                    <th>
                        {{ trans('cruds.userAlert.fields.created_at') }}
                    </th>
                    <th>
                         
                    </th>
                </tr>
            </thead>
        </table>
    </div>
</div>@endsection
@section('scripts')
@parent
<script>
    $(function () {
  let dtButtons = $.extend(true, [], $.fn.dataTable.defaults.buttons)
@can('user_alert_delete')
```

```
  let deleteButtonTrans = '{{ trans('global.datatables.delete') }}';
  let deleteButton = {
    text: deleteButtonTrans,
    url: "{{ route('admin.user-alerts.massDestroy') }}",
    className: 'btn-danger',
    action: function (e, dt, node, config) {
      var ids = $.map(dt.rows({ selected: true }).data(), function (entry) {
          return entry.id
      });
      if (ids.length === 0) {
        alert('{{ trans('global.datatables.zero_selected') }}')
        return
      }
      if (confirm('{{ trans('global.areYouSure') }}')) {
        $.ajax({
          headers: {'x-csrf-token': _token},
          method: 'POST',
          url: config.url,
          data: { ids: ids, _method: 'DELETE' }})
          .done(function () { location.reload() })
      }
    }
  }
  dtButtons.push(deleteButton)
@endcan
  let dtOverrideGlobals = {
    buttons: dtButtons,
    processing: true,
    serverSide: true,
    retrieve: true,
    aaSorting: [],
    ajax: "{{ route('admin.user-alerts.index') }}",
    columns: [
      { data: 'placeholder', name: 'placeholder' },
{ data: 'id', name: 'id' },
{ data: 'alert_text', name: 'alert_text' },
{ data: 'alert_link', name: 'alert_link' },
{ data: 'user', name: 'users.name' },
{ data: 'created_at', name: 'created_at' },
{ data: 'actions', name: '{{ trans('global.actions') }}' }
    ],
    orderCellsTop: true,
    order: [[ 1, 'desc' ]],
    pageLength: 100,
  };
  let table = $('.datatable-UserAlert').DataTable(dtOverrideGlobals);
  $('a[data-toggle="tab"]').on('shown.bs.tab click', function(e){
      $($.fn.dataTable.tables(true)).DataTable()
          .columns.adjust();
  });
});
</script>
@endsection
```

ContentManagement.blade.php

```
@extends('layouts.admin')
@section('content')
@can('homepage_create')
    <div style="margin-bottom: 10px;" class="row">
        <div class="col-lg-12">
            <a class="btn btn-success" href="{{ route('admin.homepages.create') }}">
                {{ trans('global.add') }} {{ trans('cruds.homepage.title_singular') }}
            </a>
        </div>
    </div>
@endcan
<div class="card">
    <div class="card-header">
        {{ trans('cruds.homepage.title_singular') }} {{ trans('global.list') }}
    </div>
    <div class="card-body">
        <table class=" table table-bordered table-striped table-hover ajaxTable datatable
datatable-Homepage">
            <thead>
                <tr>
                    <th width="10">

                    </th>
                    <th>
                        {{ trans('cruds.homepage.fields.id') }}
                    </th>
                    <th>
                        {{ trans('cruds.homepage.fields.title') }}
                    </th>
                    <th>
                         
                    </th>
                </tr>
            </thead>
        </table>
    </div>
</div>@endsection
@section('scripts')
@parent
<script>
    $(function () {
  let dtButtons = $.extend(true, [], $.fn.dataTable.defaults.buttons)
@can('homepage_delete')
  let deleteButtonTrans = '{{ trans('global.datatables.delete') }}';
  let deleteButton = {
    text: deleteButtonTrans,
    url: "{{ route('admin.homepages.massDestroy') }}",
    className: 'btn-danger',
    action: function (e, dt, node, config) {
      var ids = $.map(dt.rows({ selected: true }).data(), function (entry) {
          return entry.id
      });
```

```
      if (ids.length === 0) {
        alert('{{ trans('global.datatables.zero_selected') }}')
        return
      }

      if (confirm('{{ trans('global.areYouSure') }}')) {
        $.ajax({
          headers: {'x-csrf-token': _token},
          method: 'POST',
          url: config.url,
          data: { ids: ids, _method: 'DELETE' }})
          .done(function () { location.reload() })
      }
    }
  }
  dtButtons.push(deleteButton)
@endcan
  let dtOverrideGlobals = {
    buttons: dtButtons,
    processing: true,
    serverSide: true,
    retrieve: true,
    aaSorting: [],
    ajax: "{{ route('admin.homepages.index') }}",
    columns: [
      { data: 'placeholder', name: 'placeholder' },
{ data: 'id', name: 'id' },
{ data: 'title', name: 'title' },
{ data: 'actions', name: '{{ trans('global.actions') }}' }
    ],
    orderCellsTop: true,
    order: [[ 1, 'desc' ]],
    pageLength: 100,
  };
  let table = $('.datatable-Homepage').DataTable(dtOverrideGlobals);
  $('a[data-toggle="tab"]').on('shown.bs.tab click', function(e){
      $($.fn.dataTable.tables(true)).DataTable()
          .columns.adjust();
  });
});</script>
@endsection
```

UserManagement.blade.php

```
@extends('layouts.admin')
@section('content')
@can('user_create')
    <div style="margin-bottom: 10px;" class="row">
        <div class="col-lg-12">
            <a class="btn btn-success" href="{{ route('admin.users.create') }}">
                {{ trans('global.add') }} {{ trans('cruds.user.title_singular') }}
            </a>
        </div>
    </div>
@endcan
```

```
<div class="card">
   <div class="card-header">
      {{ trans('cruds.user.title_singular') }} {{ trans('global.list') }}
   </div>
   <div class="card-body">
      <div class="table-responsive">
         <table class=" table table-bordered table-striped table-hover datatable datatable-
User">
            <thead>  <tr>
                  <th width="10">

                  </th>
                  <th>
                     {{ trans('cruds.user.fields.id') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.name') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.email') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.email_verified_at') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.approved') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.two_factor') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.roles') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.team') }}
                  </th>
                  <th>
                     {{ trans('cruds.user.fields.image') }}
                  </th>
                  <th>
                      
                  </th>
               </tr>
            </thead>
            <tbody>
               @foreach($users as $key => $user)
                  @if($user->id != auth()->user()->id)
                     <tr data-entry-id="{{ $user->id }}">
                        <td>
                        </td>
                        <td>
                           {{ $user->id ?? '' }}
                        </td>
                        <td>
                           {{ $user->name ?? '' }}
```

```
                    </td>
                    <td>
                        {{ $user->email ?? '' }}
                    </td>
                    <td>
                        {{ $user->email_verified_at ?? '' }}
                    </td>
                    <td>
                        <span style="display:none">{{ $user->approved ?? '' }}</span>
                        <input type="checkbox" disabled="disabled" {{ $user->approved ?
'checked' : '' }}>
                    </td>
                    <td>
                        <span style="display:none">{{ $user->two_factor ?? '' }}</span>
                        <input type="checkbox" disabled="disabled" {{ $user->two_factor ?
'checked' : '' }}>
                    </td>
                    <td>
                        @foreach($user->roles as $key => $item)
                            <span class="badge badge-info">{{ $item->title }}</span>
                        @endforeach
                    </td>
                    <td>
                        {{ $user->team->name ?? '' }}
                    </td>
                    <td>
                        @if($user->image)
                            <a href="{{ $user->image->getUrl() }}" target="_blank"
style="display: inline-block">
                                <img src="{{ $user->image->getUrl('thumb') }}">
                            </a>
                        @endif
                    </td>
                    <td>
                        @can('user_show')
                            <a class="btn btn-xs btn-primary" href="{{
route('admin.users.show', $user->id) }}">
                                {{ trans('global.view') }}
                            </a>
                        @endcan

                        @can('user_edit')
                            <a class="btn btn-xs btn-info" href="{{ route('admin.users.edit',
$user->id) }}">
                                {{ trans('global.edit') }}
                            </a>
                        @endcan

                        @can('user_delete')
                            <form action="{{ route('admin.users.destroy', $user->id) }}"
method="POST" onsubmit="return confirm('{{ trans('global.areYouSure') }}');"
style="display: inline-block;">
                                <input type="hidden" name="_method" value="DELETE">
                                <input type="hidden" name="_token" value="{{ csrf_token()
}}">
```

```
                                <input type="submit" class="btn btn-xs btn-danger" value="{{
trans('global.delete') }}">
                            </form>
                        @endcan
                    </td>
                </tr>
            @endif
        @endforeach
    </tbody>
</table>
</div>
</div>
</div>
@endsection
@section('scripts')
@parent
<script>
    $(function () {
  let dtButtons = $.extend(true, [], $.fn.dataTable.defaults.buttons)
  @can('user_delete')
  let deleteButtonTrans = '{{ trans('global.datatables.delete') }}'
  let deleteButton = {
    text: deleteButtonTrans,
    url: "{{ route('admin.users.massDestroy') }}",
    className: 'btn-danger',
    action: function (e, dt, node, config) {
    var ids = $.map(dt.rows({ selected: true }).nodes(), function (entry) {
      return $(entry).data('entry-id')
    });
    if (ids.length === 0) {
      alert('{{ trans('global.datatables.zero_selected') }}')
      return
    }
    if (confirm('{{ trans('global.areYouSure') }}')) {
      $.ajax({
      headers: {'x-csrf-token': _token},
      method: 'POST',
      url: config.url,
      data: { ids: ids, _method: 'DELETE' }})
      .done(function () { location.reload() })
    }
    }
  }
  dtButtons.push(deleteButton)
  @endcan
  $.extend(true, $.fn.dataTable.defaults, {
    orderCellsTop: true,
    order: [[ 1, 'desc' ]],
    pageLength: 100,
  });
  let table = $('.datatable-User:not(.ajaxTable)').DataTable({ buttons: dtButtons })
  $('a[data-toggle="tab"]').on('shown.bs.tab click', function(e){
    $($.fn.dataTable.tables(true)).DataTable()
      .columns.adjust();
  });
```

```
    })
</script>
@endsection
```

E-votingAdmin.blade.php

```
@extends('layouts.admin')
@section('content')
@can('organization_create')
    <div style="margin-bottom: 10px;" class="row">
        <div class="col-lg-12">
            <a class="btn btn-success" href="{{ route('admin.organizations.create') }}">
                {{ trans('global.add') }} {{ trans('cruds.organization.title_singular') }}
            </a>
        </div>
    </div>
@endcan
<div class="card">
    <div class="card-header">
        {{ trans('cruds.organization.title_singular') }} {{ trans('global.list') }}
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class=" table table-bordered table-striped table-hover datatable datatable-
Organization">
                <thead>
                    <tr>
                        <th width="10">
                        </th>
                        <th>
                            {{ trans('cruds.organization.fields.id') }}
                        </th>
                        <th>
                            {{ trans('cruds.organization.fields.status') }}
                        </th>
                        <th>
                            {{ trans('cruds.organization.fields.title') }}
                        </th>
                        <th>
                            {{ trans('cruds.organization.fields.slug') }}
                        </th>
                        <th>
                             
                        </th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($organizations as $key => $organization)
                        <tr data-entry-id="{{ $organization->id }}">
                            <td>
                            </td>
                            <td>
                                {{ $organization->id ?? '' }}
```

```
                    </td>
                    <td>
                        {{ App\Models\Organization::STATUS_RADIO[$organization->status]
?? '' }}
                    </td>
                    <td>
                        {{ $organization->title ?? '' }}
                    </td>
                    <td>
                        @if($organization->slug)
                            <a href="{{ URL::to('/') . '/e-voting/' . $organization->slug . '/login'}}"
target="_blank" >{{ URL::to('/') . '/e-voting/' . $organization->slug . '/login' }}</a>
                        @else
                            {{ '' }}
                        @endif
                    </td>
                    <td>
                        @can('organization_show')
                            <a class="btn btn-xs btn-primary" href="{{
route('admin.organizations.show', $organization->id) }}">
                                {{ trans('global.view') }}
                            </a>
                        @endcan
                        @can('organization_edit')
                            <a class="btn btn-xs btn-info" href="{{
route('admin.organizations.edit', $organization->id) }}">
                                {{ trans('global.edit') }}
                            </a>
                        @endcan
                        @can('organization_delete')
                            <form action="{{ route('admin.organizations.destroy', $organization-
>id) }}" method="POST" onsubmit="return confirm('{{ trans('global.areYouSure') }}');"
style="display: inline-block;">
                                <input type="hidden" name="_method" value="DELETE">
                                <input type="hidden" name="_token" value="{{ csrf_token() }}">
                                <input type="submit" class="btn btn-xs btn-danger" value="{{
trans('global.delete') }}">
                            </form>
                        @endcan
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
  </div>
 </div>
</div>
@endsection
@section('scripts')
@parent
<script>
  $(function () {
  let dtButtons = $.extend(true, [], $.fn.dataTable.defaults.buttons)
@can('organization_delete')
  let deleteButtonTrans = '{{ trans('global.datatables.delete') }}'
```

```
  let deleteButton = {
    text: deleteButtonTrans,
    url: "{{ route('admin.organizations.massDestroy') }}",
    className: 'btn-danger',
    action: function (e, dt, node, config) {
      var ids = $.map(dt.rows({ selected: true }).nodes(), function (entry) {
          return $(entry).data('entry-id')
      });
      if (ids.length === 0) {
        alert('{{ trans('global.datatables.zero_selected') }}')
        return
      }
      if (confirm('{{ trans('global.areYouSure') }}')) {
        $.ajax({
          headers: {'x-csrf-token': _token},
          method: 'POST',
          url: config.url,
          data: { ids: ids, _method: 'DELETE' }})
          .done(function () { location.reload() })
      }
    }
  }
  dtButtons.push(deleteButton)
@endcan
  $.extend(true, $.fn.dataTable.defaults, {
    orderCellsTop: true,
    order: [[ 1, 'desc' ]],
    pageLength: 100,
  });
  let table = $('.datatable-Organization:not(.ajaxTable)').DataTable({ buttons: dtButtons })
  $('a[data-toggle="tab"]').on('shown.bs.tab click', function(e){
      $($.fn.dataTable.tables(true)).DataTable()
          .columns.adjust();
  });
})
</script>
@endsection
```

TabulationAdmin.blade.php

```
@extends('layouts.admin')
@section('content')
@can('title_create')
    <div style="margin-bottom: 10px;" class="row">
        <div class="col-lg-12">
            <a class="btn btn-success" href="{{ route('admin.titles.create') }}">
                {{ trans('global.add') }} {{ trans('cruds.title.title_singular') }}
            </a>
        </div>
    </div>
@endcan
<div class="card">
    <div class="card-header">
        {{ trans('cruds.title.title_singular') }} {{ trans('global.list') }}
    </div>
```

```
    <div class="card-body">
        <div class="table-responsive">
            <table class=" table table-bordered table-striped table-hover datatable datatable-
Title">
                <thead>
                    <tr>
                        <th width="10">
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.status_2') }}
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.title') }}
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.slug') }}
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.date') }}
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.score_min') }}
                        </th>
                        <th>
                            {{ trans('cruds.title.fields.score_max') }}
                        </th>
                        <th>
                             
                        </th>
                    </tr>
                </thead>
                <tbody>
                    @foreach($titles as $key => $title)
                        <tr data-entry-id="{{ $title->id }}">
                            <td>
                            </td>
                            <td>
                                {{ App\Models\Title::STATUS_2_RADIO[$title->status_2] ?? '' }}
                            </td>
                            <td>
                                {{ $title->title ?? '' }}
                            </td>
                            <td>
                                <a href="{{ URL::to('/') . '/tabulation/' . $title->slug . '/login'}}"
target="_blank" >{{ URL::to('/') . '/tabulation/' . $title->slug . '/login' }}</a>
                            </td>
                            <td>
                                {{ $title->date ?? '' }}
                            </td>
                            <td>
                                {{ $title->score_min ?? '' }}
                            </td>
                            <td>
                                {{ $title->score_max ?? '' }}
                            </td>
```

```
                    <td>
                        @can('title_show')
                            <a class="btn btn-xs btn-primary" href="{{ route('admin.titles.show', $title->id) }}">
                                {{ trans('global.view') }}
                            </a>
                        @endcan
                        @can('title_edit')
                            <a class="btn btn-xs btn-info" href="{{ route('admin.titles.edit', $title->id) }}">
                                {{ trans('global.edit') }}
                            </a>
                        @endcan
                        @can('title_delete')
                            <form action="{{ route('admin.titles.destroy', $title->id) }}" method="POST" onsubmit="return confirm('{{ trans('global.areYouSure') }}');" style="display: inline-block;">
                                <input type="hidden" name="_method" value="DELETE">
                                <input type="hidden" name="_token" value="{{ csrf_token() }}">
                                <input type="submit" class="btn btn-xs btn-danger" value="{{ trans('global.delete') }}">
                            </form>
                        @endcan
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
  </div>
 </div>
</div>
@endsection
@section('scripts')
@parent
<script>
  $(function () {
  let dtButtons = $.extend(true, [], $.fn.dataTable.defaults.buttons)
@can('title_delete')
  let deleteButtonTrans = '{{ trans('global.datatables.delete') }}'
  let deleteButton = {
    text: deleteButtonTrans,
    url: "{{ route('admin.titles.massDestroy') }}",
    className: 'btn-danger',
    action: function (e, dt, node, config) {
      var ids = $.map(dt.rows({ selected: true }).nodes(), function (entry) {
          return $(entry).data('entry-id')
      });
      if (ids.length === 0) {
        alert('{{ trans('global.datatables.zero_selected') }}')
        return
      }
      if (confirm('{{ trans('global.areYouSure') }}')) {
        $.ajax({
          headers: {'x-csrf-token': _token},
          method: 'POST',
```

```
        url: config.url,
        data: { ids: ids, _method: 'DELETE' }})
        .done(function () { location.reload() })
    }
   }
  }
  dtButtons.push(deleteButton)
@endcan
  $.extend(true, $.fn.dataTable.defaults, {
    orderCellsTop: true,
    order: [[ 1, 'desc' ]],
    pageLength: 100,
  });
  let table = $('.datatable-Title:not(.ajaxTable)').DataTable({ buttons: dtButtons })
  $('a[data-toggle="tab"]').on('shown.bs.tab click', function(e){
     $($.fn.dataTable.tables(true)).DataTable()
        .columns.adjust();
  });
})
</script>
@endsection
```

Messages.blade.php

```
@extends('admin.messenger.template')
@section('title', $title)
@section('messenger-content')
<div class="row">
   <div class="col-lg-12">
      <div class="list-group">
         @forelse($topics as $topic)
            <div class="row list-group-item d-flex">
               <div class="col-lg-4">
                  <a href="{{ route('admin.messenger.showMessages', [$topic->id]) }}">
                     @php($receiverOrCreator = $topic->receiverOrCreator())
                        @if($topic->hasUnreads())
                           <strong>
                              {{ $receiverOrCreator !== null ? $receiverOrCreator->email : '' }}
                           </strong>
                        @else
                           {{ $receiverOrCreator !== null ? $receiverOrCreator->email : '' }}
                        @endif
                  </a>
               </div>
               <div class="col-lg-5">
                  <a href="{{ route('admin.messenger.showMessages', [$topic->id]) }}">
                     @if($topic->hasUnreads())
                        <strong>
                           {{ $topic->subject }}
                        </strong>
                     @else
                        {{ $topic->subject }}
                     @endif
                  </a>
               </div>
```

```
                <div class="col-lg-2 text-right">{{ $topic->created_at->diffForHumans()
}}</div>
                <div class="col-lg-1 text-center">
                   <form action="{{ route('admin.messenger.destroyTopic', [$topic->id]) }}"
method="POST" onsubmit="return confirm('{{ trans('global.areYouSure') }}');">
                      <input type="hidden" name="_method" value="DELETE">
                      @csrf
                      <input type="submit" class="btn btn-xs btn-danger" value="{{
trans('global.delete') }}">
                   </form>
                </div>
             </div>
             @empty
             <div class="row list-group-item">
                {{ trans('global.you_have_no_messages') }}
             </div>
          @endforelse
       </div>
    </div>
</div>
@endsection
```

E-votingUserDashboard.blade.php

```
@extends('layouts.election')
@section('content')
<div class="jumbotron">
   <h2 class="page-header text-center title text-black"><b>{{$elections->title}}</b></h2>
</div>
@if ($voterData->status == "true")
   <div class="jumbotron">
      <h4 class="page-header text-center title text-black"><b>You have already voted for this
election</b></h4>
      {{-- SUBMIT BUTTON --}}
      <div class="text-center my-5">
         <button type="button" class="btn btn-primary btn-flat p-5" id="preview" data-
toggle="modal" data-target="#viewBallow"><i
               class="fa fa-file-text"></i>
            View ballot</button>
      </div>
   </div>
   <div class="modal fade" id="viewBallow" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLongTitle" aria-hidden="true">
      <div class="modal-dialog" role="document">
      <div class="modal-content">
         <div class="modal-header">
         <h5 class="modal-title" id="exampleModalLongTitle">Your Votes</h5>
         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
         </button>
         </div>
         <div class="modal-body">
            <table class="table table-bordered">
               <thead>
                 <tr>
```

```
                    <th scope="col">Position</th>
                    <th scope="col">Candidate</th>
                  </tr>
                </thead>
                <tbody>
                    @foreach ($AuditVoters as $AuditVoter)
                        <tr>
                            <td scope="row">{{$AuditVoter->position->position}}</td>
                            <td>{{$AuditVoter->candidate->name}}</td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
        <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        </div>
      </div>
      </div>
   </div>
@else
<!-- Main content -->
<section class="content">
   <div class="row d-flex justify-content-center">
      <div class="col-sm-10 col-sm-offset-1">
         <div class="alert alert-danger alert-dismissible" id="alert" style="display:none;">
            <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
            <span class="message"></span>
         </div>
         <!-- Voting Ballot -->
         <form method="POST" action="{{ url('e-voting/'.$slug.'/') }}"
enctype="multipart/form-data">
            @csrf
            @foreach ($possitionData as $key => $possition)
               {{-- LOAD POSITION --}}
               <div class="row" id="position{{$key+1}}">
                  <div class="col-sm-12 border p-2">
                     <div class="box box-solid" id="8">
                        <div class="box-header with-border">
                           <h3 class="box-title"><b>{{$possition->position}}</b></h3>
                        </div>
                        <div class="box-body">
                           <p>
                              @if ($possition->vote_allow == 1)
                                 You can select one candidate only.
                              @elseif ($possition->vote_allow > 1)
                                 You may vote up to {{$possition->vote_allow}} candidates.
                              @else
                                 Selecting candidate is not availble.
                              @endif

                              <span class="float-right">
```

```
                        <button type="button" class="btn btn-success btn-sm btn-flat
reset"
                            data-desc="president"><i class="fa fa-refresh"></i>
                            Reset</button>
                        </span>
                    </p>
                    <div id="candidate_list">
                        <div class="card mb-3 w-100 ">
                            @php ($count = 0)
                            @foreach ($CandidateData as $key => $Candidate)
                                @if ($possition->id == $Candidate->position_id)
                                        {{-- Candidates --}}
                                    <div class="row no-gutters border-bottom">
                                        <div class="col-md-4 d-flex justify-content-center p-2">
                                            <img src="{{$Candidate->image-
>getUrl('preview')}}" width="50%">
                                        </div>
                                        <div class="col-md-7 w-100">
                                            <div class="d-flex justify-content-between px-5 py-
2">

                                                @if ($possition->vote_allow == 1)
                                                    <div class="custom-control custom-radio">
                                                        <input class="custom-control-input"
type="radio" id="vote{{$key}}" name="position{{$possition->id}}" value="{{$Candidate-
>id}}" required>
                                                        <label for="vote{{$key}}" class="custom-
control-label">V O T E</label>
                                                    </div>
                                                @else
                                                    <div class="custom-control custom-checkbox">
                                                        <input class="custom-control-input" data-
vote-allow="{{$possition->vote_allow}}" type="checkbox" id="vote{{$key}}"
name="position{{$possition->id}}[]" value="{{$Candidate->id}}">
                                                        <label for="vote{{$key}}" class="custom-
control-label">V O T E</label>
                                                    </div>
                                                @endif

                                                <span class="btn btn-info btn-sm btn-flat reset"
data-toggle="modal" @if(!(empty($Candidate->partylist->name))) data-
target="#partylist{{$Candidate->partylist->id}}" @else data-target="#partylistDefault"
@endif>
                                                    <i class="fa fa-search"></i> Platform</span>
                                                </div>
                                                <div class="card-body text-center">
                                                    <h4>{{$Candidate->name}}</h4>
                                                    <p class="card-text"><small class="text-muted">

                                                        @if (!(empty($Candidate->partylist->name)))
                                                            {{$Candidate->partylist->name}}
                                                        @else
                                                        @endif
                                                    </small></p>
                                                </div>
```

```
                                    </div>
                                    <hr>
                                </div>
                                @php ($count =  $count + 1)
                            @endif
                        @endforeach
                        @if ($count == 0)
                            <h3 class="text-center p-5 text-secondary">No candidate
available!</h3>
                        @endif

                        {{-- Candidates --}}
                        {{-- <div class="row no-gutters border-bottom">
                            <div class="col-md-4 d-flex justify-content-center p-2">
                                <img src="{{asset('assets')}}/img/team/team-1.jpg"
width="50%">
                            </div>
                            <div class="col-md-7 w-100">
                                <div class="d-flex justify-content-between px-5 py-2">
                                    <div class="custom-control custom-radio">
                                        <input class="custom-control-input" type="radio"
id="president2"
                                            name="president" value="false" required>
                                        <label for="president2" class="custom-control-label">V
O T
                                            E</label>
                                    </div>
                                    <span class="btn btn-info btn-sm btn-flat reset"
                                        data-desc="president"><i class="fa fa-search"></i>
                                        Platform</span>
                                </div>
                                <div class="card-body text-center">
                                    <h3>Merson Labro. Taguiam Jr.</h3>
                                    <p class="card-text"><small class="text-muted">PARTY-
LIST</small></p>
                                </div>
                            </div>
                            <hr>
                        </div> --}}
                    </div>
                </div>
            </div>
        </div>
        <div class="card mb-3 w-100 ">
        </div>

    </div>
@endforeach
{{-- SUBMIT BUTTON --}}
<div class="text-center my-5">
    {{-- <button type="button" class="btn btn-success btn-flat" id="preview"><i
            class="fa fa-file-text"></i>
        Preview</button> --}}
```

```
                <button type="submit" class="btn btn-primary btn-flat w-100 p-3"
name="vote"><i
                    class="fa fa-check-square-o"></i>
                  Submit</button>
              </div>
            </form>
            <!-- End Voting Ballot -->
        </div>
    </div>
</section>
@endif
  <div class="modal fade" id="partylistDefault" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLongTitle" aria-hidden="true">
    <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
      <h5 class="modal-title" id="exampleModalLongTitle">PARTYLIST
PLATFORM</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
      </div>
      <div class="modal-body">
        <h3 class="text-center p-5 text-secondary">No Platform available!</h3>
      </div>
      <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
    </div>
</div>
  @foreach ($partylistData as $key => $partylist)
      <!-- Preview Modal Platform-->
      <div class="modal fade" id="partylist{{$partylist->id}}" tabindex="-1" role="dialog"
aria-labelledby="exampleModalLongTitle" aria-hidden="true">
        <div class="modal-dialog" role="document">
        <div class="modal-content">
          <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLongTitle">PARTYLIST
PLATFORM</h5>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
          </div>
          <div class="modal-body">
          {!! $partylist->platform !!}
          </div>
          <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
          </div>
        </div>
        </div>
      </div>
  @endforeach
```

```
<!-- Preview -->
<div class="modal fade" id="preview_modal">
   <div class="modal-dialog">
      <div class="modal-content">
         <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
               <span aria-hidden="true">&times;</span></button>
            <h4 class="modal-title">Vote Preview</h4>
         </div>
         <div class="modal-body">
            <div id="preview_body"></div>
         </div>
         <div class="modal-footer">
            <button type="button" class="btn btn-default btn-flat pull-left" data-
dismiss="modal"><i
                  class="fa fa-close"></i> Close</button>
         </div>
      </div>
   </div>
</div>
@endsection
@section('scripts')
<script>
   $(document).ready(function () {
      $('input[type=radio]').change(function () {
         $(this).parents().eq(3).siblings().removeClass('text-white bg-primary');
         $(this).parents().eq(3).addClass('text-white bg-primary');
         //alert(test);
      });       $('.reset').click(function (e) {

$(this).parent().parent().siblings().children().children().find('input[type=radio]').prop('checked
', false);

(this).parent().parent().siblings().children().children().find('input[type=checkbox]').prop('chec
ked', false);
         $(this).parent().parent().siblings().children().children().removeClass('text-white bg-
primary');

      });
      //$(this).parents().eq(3).addClass('text-white bg-primary');
      $('input[type=checkbox]').change(function () {

         //to get value of data-* or data-vote-allow
         $(this).data('vote-allow');           //var limit = $(this).attr("data-vote-allow")
         var limit = $(this).data('vote-allow');           if ($(this).prop('checked')) {
            if ($('input[type=checkbox]:checked').siblings().length <= limit) {
                  $(this).parents().eq(3).addClass('text-white bg-primary');
               }else{
                  $(this).prop('checked', false);
                  alert("You have reached the maximum vote of " + limit);
               }
         }else{
            $(this).parents().eq(3).removeClass('text-white bg-primary');
         }
         //alert(test);
```

```
        });
        $('input.single-checkbox').on('change', function(evt) {
           if($(this).siblings(':checked').length >= limit) {
              this.checked = false;
           }
        });
    });

</script>
@endsection
```

TabulationUserDashboard.blade.php

```
@extends('layouts.tabulation')
@section('styles')
  <!-- DataTables -->
  <link rel="stylesheet" href="{{ asset('plugins/datatables-
bs4/css/dataTables.bootstrap4.min.css') }}">
  <link rel="stylesheet" href="{{ asset('plugins/datatables-
responsive/css/responsive.bootstrap4.min.css') }}">
  <link rel="stylesheet" href="{{ asset('plugins/datatables-
buttons/css/buttons.bootstrap4.min.css') }}">
@endsection
@section('content')
{{-- <div class="jumbotron">
    <h2 class="page-header text-center title text-black"><b>{{$categories-
>where('id',$categorySelected)->first()->name}}</b></h2>
</div> --}}
    <!-- Content Wrapper. Contains page content -->
<!-- Main content -->
<section class="content">
    @if ($categories->where('id',$categorySelected)->first()->status == 'true')
        <div class="card">
            <div class="card-body">
                <h3 class="text-center title text-black"><b>{{$categories-
>where('id',$categorySelected)->first()->name}}</b></h3>
            </div>
        </div>
        <img src="{{ asset('assets/img/lockCover.png') }}" class="rounded mx-auto d-block
img-fluid" alt="Responsive image">
    @else
        <div class="row d-flex justify-content-center">
            <div class="col-sm-10 col-sm-offset-1">
                <div class="alert alert-danger alert-dismissible" id="alert" style="display:none;">
                    <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
                    <span class="message"></span>
                </div>
            </div>
            <div class="col-12">
                <div class="card">
                    <div class="card-header">
                    <h3 class="text-center title text-black"><b>{{$categories-
>where('id',$categorySelected)->first()->name}}</b></h3>
                    </div>
                    <!-- /.card-header -->
                    <div class="card-body">
                    <table id="example1" class="table table-bordered table-striped {{
Session::get('main-body') ? "table-dark" : "" }} disabled">
                        <thead>
                            <tr>
                            <th width="10" class="text-center">#</th>
                            <th class="text-center text-nowrap px-4">Participant Name</th>
                                @foreach ($criterias as $criteria)
```

```
                        <th class="text-center text-nowrap px-0 mx-0 py-2">{{$criteria->name}}<br>{{$criteria->percentage}}%</th>
                        @endforeach
                        <th width="10" class="text-center">Total<br>100%</th>
                        </tr>
                    </thead>

                    <tbody>
                    @foreach ($participants as $rowcount => $participant)
                        <tr >
                            <td width="10" class="text-center text-nowrap" data-participant-id="{{$participant->id}}"  data-row-count="{{$rowcount+1}}">
                                <b>
                                    @if ($participant->type == 1)
                                        Mr. {{$participant->number}}
                                    @elseif ($participant->type == 2)
                                        Ms. {{$participant->number}}
                                    @elseif ($participant->type == 3)
                                        Grp. {{$participant->number}}
                                    @elseif ($participant->type == 4)
                                        Team {{$participant->number}}
                                    @endif
                                </b>
                            </td>
                            <td class="text-nowrap">
                                <a href="" style="color: inherit;" data-toggle="modal" data-target="#participantPreview{{$participant->id}}">
                                    <u>
                                        <b>{{$participant->name}}</b>
                                    </u>
                                </a>
                            </td>
                            @php
                                $a = 'A';
                            @endphp
                            @foreach ($criterias as $key => $criteria)
                            @php                                $scoreValue = '';
                                if (!empty($judgeScores->where('criteria_id', $criteria->id )->where('participant_id', $participant->id )->first()->scores)) {
                                    $scoreValue = $judgeScores->where('criteria_id', $criteria->id )->where('participant_id', $participant->id )->first()->scores;
                                }
                            @endphp
                            <td class="text-center" >
                                <div class="form-group has-validation d-flex justify-content-center">
                                    <div class="text-center">
                                        <div class="spinner-grow" role="status">
                                            <span class="sr-only">Loading...</span>
                                        </div>
                                    </div>
                                    <input class="d-none text-center form-control @if($scoreValue) is-valid @endif}}" style="width: 6em"
                                        min="{{$tabulations->score_min}}"
                                        max="{{$tabulations->score_max}}"
                                        type="number" name="{{$a++ . ($rowcount+1)}}"
```

```
                                data-percentage="{{$criteria->percentage}}"
                                data-criteria-id="{{$criteria->id}}"
                                data-row-count="{{$rowcount+1}}"
                                data-participant-id="{{$participant->id}}"
                                value="{{$scoreValue}}"
                                />
                            </div>
                        </td>
                        @endforeach
                        <td>
                            <label class="text-primary TotalAverage{{$rowcount+1}}"
                            data-participant-id="{{$participant->id}}">
                                0.00%
                            </label>

                        </td>
                    </tr>
                @endforeach
                </tbody>
                <tfoot>
                    <tr>
                        <th width="10" class="text-center">#</th>
                        <th>Participant Name</th>
                        @foreach ($criterias as $criteria)
                        <th class="text-center">{{$criteria->name}}<br>{{$criteria-
>percentage}}%</th>
                        @endforeach
                        <th width="10" class="text-center">Total<br>100%</th>
                    </tr>
                </tfoot>
            </table>
            </div>
            <!-- /.card-body -->
        </div>
        <!-- /.card -->
    </div>
  </div>
@endif
</section>
{{-- PREVIEW PARTICIPANTS PROFILE --}}
@foreach ($participants as $rowcount => $participant)
<div class="modal fade" id="participantPreview{{$participant->id}}"  tabindex="-1"
role="dialog" aria-labelledby="myLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
      <h5 class="modal-title" id="exampleModalLabel">Profile</h5>
      <button type="button" class="close" data-dismiss="modal" aria-label="Close">
        <span aria-hidden="true">&times;</span>
      </button>
      </div>
      <div class="modal-body">
        <table class="table table-bordered table-striped">
          <tbody>
            <tr class="text-wrap text-center">
```

```
                    <td colspan="2" style="text-align: center; vertical-align: middle;">

                        @if ($participant->image->count() > 0)
                            @foreach($participant->image as $key => $media)
                                <picture>
                                    <img src="{{ $media->getUrl() }}" class="img-fluid img-
thumbnail" alt="{{ $participant->name }}">
                                </picture>
                            @endforeach
                        @else
                            <picture>
                                <img src="{{ asset('assets/img/defaultuser.png') }}" class="img-
fluid img-thumbnail" alt="{{ $participant->name }}">
                            </picture>
                        @endif
                    </td>
                </tr>
                <tr>
                    <th>
                        Participant Number
                    </th>
                    <td>
                        @if ($participant->type == 1)
                            Mr. {{$participant->number}}
                        @elseif ($participant->type == 2)
                            Ms. {{$participant->number}}
                        @elseif ($participant->type == 3)
                            Grp. {{$participant->number}}
                        @elseif ($participant->type == 4)
                            Team {{$participant->number}}
                        @endif
                    </td>
                </tr>
                <tr>
                    <th>
                        Participan Name
                    </th>
                    <td>
                        {{ $participant->name }}
                    </td>
                </tr>
                <tr>
                    <th>
                        Others
                    </th>
                    <td>
                        {{ $participant->description }}
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
    <div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
```

```
            </div>
          </div>
        </div>
      </div>
      @endforeach
      @endsection
      @section('scripts')
      <!-- jQuery -->
      {{-- <script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script> --}}
      <!-- Bootstrap 4 -->
      <script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
      <!-- DataTables  & Plugins -->
      <script src="{{ asset('plugins/datatables/jquery.dataTables.min.js') }}"></script>
      <script src="{{ asset('plugins/datatables-bs4/js/dataTables.bootstrap4.min.js') }}"></script>
      <script src="{{ asset('plugins/datatables-responsive/js/dataTables.responsive.min.js')
      }}"></script>
      <script src="{{ asset('plugins/datatables-responsive/js/responsive.bootstrap4.min.js')
      }}"></script>
      <script src="{{ asset('plugins/datatables-buttons/js/dataTables.buttons.min.js') }}"></script>
      <script src="{{ asset('plugins/datatables-buttons/js/buttons.bootstrap4.min.js') }}"></script>
      <script src="{{ asset('plugins/jszip/jszip.min.js') }}"></script>
      <script src="{{ asset('plugins/pdfmake/pdfmake.min.js') }}"></script>
      <script src="{{ asset('plugins/pdfmake/vfs_fonts.js') }}"></script>
      <script src="{{ asset('plugins/datatables-buttons/js/buttons.html5.min.js') }}"></script>
      <script src="{{ asset('plugins/datatables-buttons/js/buttons.print.min.js') }}"></script>
      <script src="{{ asset('plugins/datatables-buttons/js/buttons.colVis.min.js') }}"></script>

      <!-- Page specific script -->
      <script>
        $(function () {
          $("#example1").DataTable({
            "responsive": true, "lengthChange": false, "autoWidth": false, "paging": false,"ordering":
      false,
          //   "buttons": ["csv", "excel", "pdf"]
          }).buttons().container().appendTo('#example1_wrapper .col-md-6:eq(0)');
          $('#example2').DataTable({
            "paging": false,
            "lengthChange": false,
            "searching": false,
            "ordering": true,
            "info": true,
            "autoWidth": false,
            "responsive": true,
          });
        });
      </script><script>
          $(document).ready(function () {
              var CategoryStatus = null;
              function refreshCompute(){
                  var rowCount = '{{$participants->count()}}';
                  for (let index = 1; index <= rowCount; index++) {

                      var JudgeScore = 'input[name="A' + index + '"]';
                      ComputeTotalScore(JudgeScore);
                  }  }
```

```
$(window).on('load', function() {
    refreshCompute();
    categoryLocked();
    setInterval(moveItem, 5000);
    setInteral(
        function()
        {
            categoryLocked();
        }, 5000);
});
function categoryLocked(){
    $.ajax({
        url: "{{ url('tabulation/'.$slug.'/' . $categories->where('id',$categorySelected)-
>first()->id . '/categorylock')}}",
        data: {
            cat_id: "{{$categories->where('id',$categorySelected)->first()->id}}",
            title_id: "{{$tabulations->id}}"
        },
        success: function (data) {

            $.each(data, function (id, value) {
                console.log(value);
                CategoryStatus = value;
            });} });}
function criteriaLocked($this){
    //var returnValue;
        $.ajax({
            url: "{{ url('tabulation/'.$slug.'/' . $categories->where('id',$categorySelected)-
>first()->id . '/criterialock')}}",
            data: {
                cri_id: $($this).data('criteria-id')
                },
            success: function (data) {
                $.each(data, function (id, value) {
                    //$(JudgeScore).last().val(value);
                    // console.log(value);
                    // returnValue = value;
                    JudgeSaveScores(value, $this);
                     //return returnValue;
                });
            },
            errors: function (data) {
                    //console.log(data);
                }
        });

        //return returnValue;
}
function JudgeSaveScores($myStatus, $this){
    if ($myStatus == 'false') {
        //Save Judge Scores
        $.ajax({

            url: "{{ url('tabulation/'.$slug.'/' . $categories->where('id',$categorySelected)-
>first()->id . '/savescore')}}",
```

```
                data: {
                      //"_token": "{{ csrf_token() }}",
                      title_id: "{{$tabulations->id}}",
                      cat_id: "{{$categories->where('id',$categorySelected)->first()->id}}",
                      cri_id: $($this).data('criteria-id'), //data-criteria-id
                      judge_id: "{{$judgeData->id}}",
                      participant_id: $($this).data('participant-id'),
                      score_val: $($this).val()
                   },
                success: function (data) {
                      setTimeout(
                      function()
                      {
                         //ComputeTotalScore(this);
                         ComputeTotalScore($this);
                      }, 1000);
                      //ComputeTotalScore(this);
                      $($this).blur();
                      $($this).removeClass('is-invalid');
                      $($this).addClass('is-valid');
                },
                errors: function (data) {
                } });
            }else{
               alert('Unable to save score. The criteria is already locked!');
               $($this).removeClass('is-valid');
               $($this).addClass('is-invalid');
               $($this).last().removeClass('d-none')
               $($this).last().siblings().addClass('d-none')
               $.ajax({
                  url: "{{ url('tabulation/'.$slug.'/' . $categories->where('id',$categorySelected)-
>first()->id . '/getscores')}}",
                data: {
                      //"_token": "{{ csrf_token() }}",
                      title_id: "{{$tabulations->id}}",
                      cat_id: "{{$categories->where('id',$categorySelected)->first()->id}}",
                      cri_id: $($this).data('criteria-id'), //data-criteria-id
                      judge_id: "{{$judgeData->id}}",
                      participant_id: $($this).data('participant-id')
                   },
                success: function (data) {
                      $.each(data, function (id, value) {
                            if (value) {
                                $($this).last().val(value);
                                ComputeTotalScore($this);
                            }else{
                               alert('poge');
                            }  });
 },
                errors: function (error) {
                      console.log(error);
                } });           }     }
          function JudgeSaveAverage($this){
               //Save Judge Scores
               var averageScore = $($this).html()
```

```
                averageScore = averageScore.substring(0, averageScore.length-1)
            $.ajax({
                // type: "POST",
                //headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')},

                url: "{{ url('tabulation/'.$slug.'/' . $categories->where('id',$categorySelected)->first()->id . '/saveaverage')}}",
                data: {
                      //"_token": "{{ csrf_token() }}",
                      title_id: "{{$tabulations->id}}",
                      cat_id: "{{$categories->where('id',$categorySelected)->first()->id}}",
                      judge_id: "{{$judgeData->id}}",
                      participant_id: $($this).data('participant-id'),
                      score_val: averageScore

                  },
                success: function (data) {
                  console.log(data)
                },
                errors: function (data) {
                  //$position.html('<option value="" selected>Please select</option>');
                    //console.log(data)
                  console.log(data)
                }
            });       }
        $(document).on('change', 'input[type="number"]', function(e) {
          // $(this).siblings(".fa-times").removeClass('d-none');
          // $(this).siblings(".fa-check").addClass('d-none');
          //$('.fa-times').val();
          $(this).removeClass('is-valid');
          $(this).addClass('is-invalid');

        });

        //live events keypress
        $(document).on('keypress', 'input[type="number"]', function(e) {
          var keycode = (event.keyCode ? event.keyCode : event.which);
          if(keycode == '13'){
            if (CategoryStatus == 'true') {
              alert('Unable to save score. The category is already locked!');
              return;
            }else{
              //alert('You pressed a "enter" key in textbox');
              var minScore = parseFloat($(this).attr("min"));
              var maxScore = parseFloat('{{$tabulations->score_max}}');
              var valScore =  parseFloat($(this).val());
              var $this = $(this);
              if (valScore >= minScore && valScore <= maxScore) {
                $(this).addClass('d-none')
                $(this).siblings().removeClass('d-none')
                //check if criterias is locked!!
                criteriaLocked(this);
                //COMPUTATION BEGIN!!!
              }else{
                $(this).removeClass('is-valid');
```

```
                $(this).addClass('is-invalid');
                alert('Invalid score!');
            }   }   }
    });
     //alphabet increment
     function nextChar(cchar) {
            return String.fromCharCode('A'.charCodeAt() + cchar);
     }
     function ComputeTotalScore($this){
            //$($this).addClass('border border-danger');
            //COMPUTATION BEGIN!!!
            var maxScore = '{{$tabulations->score_max}}'
            var rowCount = $($this).data('row-count');
            var criteriaCount = '{{$criterias->count()}}'
            var ScoreData = parseFloat(0);
            var TotalScore = parseFloat(0);
            var CriteriaPercentage = parseFloat(0);
            var JudgeScore;
            for (let i = 1; i <= criteriaCount; i++) {
               //reset
               ScoreData = 0
               CriteriaPercentage = 0
               JudgeScore = 'input[name="' + nextChar(i-1) + rowCount + '"]';

               //get score
               ScoreData = parseFloat($(JudgeScore).last().val());
               CriteriaPercentage = parseFloat($(JudgeScore).last().data('percentage'))/100;
               if (isNaN(ScoreData)) {
                  //skip not a number
               }else{
                  $(JudgeScore).last().removeClass('is-invalid');
                  $(JudgeScore).last().addClass('is-valid');
                  //compute percentage
                  ScoreData = parseFloat(ScoreData) * parseFloat(CriteriaPercentage);
                  TotalScore = parseFloat(TotalScore) + parseFloat(ScoreData);
               }
               $(JudgeScore).last().removeClass('d-none');
               $(JudgeScore).last().siblings().addClass('d-none');
            }
            //compute final output
            TotalScore = (parseFloat(TotalScore) / parseFloat(maxScore)) * 100;
            $averageTotal = '.TotalAverage' + rowCount;
            $($averageTotal).html(TotalScore.toFixed(2) + '%');
JudgeSaveAverage($averageTotal);
     }
     var oTable = $('#example1').DataTable();
     $("#myInput").on("keyup", function(e) {
        e.preventDefault();          oTable.search( $(this).val()).draw();
     });
     $("#closeSearch").click(function (e) {
        e.preventDefault();
        $("#myInput").val('');
        oTable.search( $(this).val()).draw();
     });
     //responsive mode
```

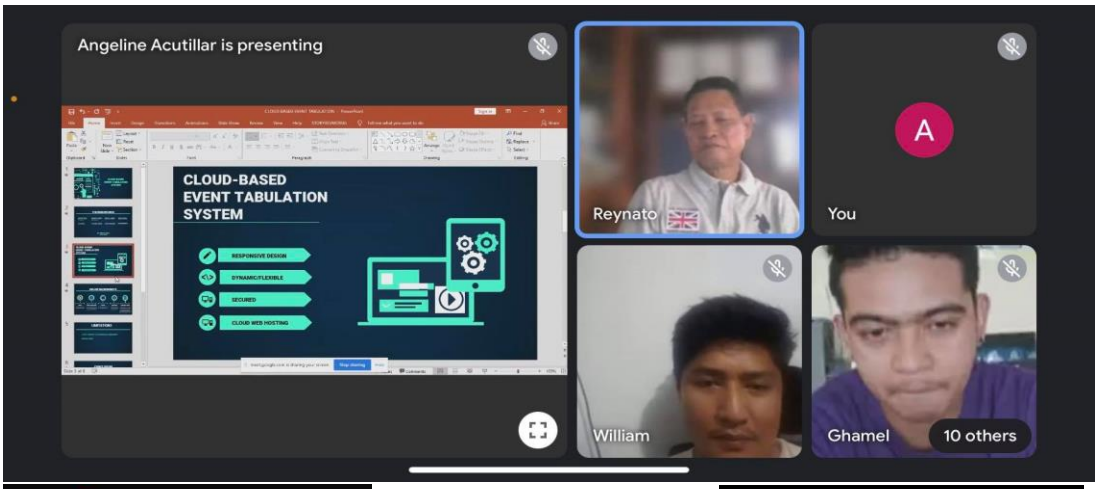
```
$('[tabindex="0"]').click(function (e) {
    //alert($(this).data('participant-id'));
    var ParticipantID = $(this).data('participant-id');
    //var Scores = ["90", "80"];
    var criteriaCount = '{{$criterias->count()}}';
    var rwCount = $(this).data('row-count');
    var $this = $(this);
    var JudgeScore;
    setTimeout(
        function()
        {
            for (let i = 1; i <= criteriaCount; i++) {
                JudgeScore = 'input[name="' + nextChar(i-1) + rowCount + '"]';
                $.ajax({
                    url: "{{ url('tabulation/'.$slug.'/' . $categories-
>where('id',$categorySelected)->first()->id . '/getscores')}}",
                    data: {
                        //"_token": "{{ csrf_token() }}",
                        title_id: "{{$tabulations->id}}",
                        cat_id: "{{$categories->where('id',$categorySelected)->first()-
>id}}",
                        cri_id: $(JudgeScore).data('criteria-id'), //data-criteria-id
                        judge_id: "{{$judgeData->id}}",
                        participant_id: ParticipantID
                    },
                    success: function (data) {
                        JudgeScore = 'input[name="' + nextChar(i-1) + rowCount + '"]';
                        $.each(data, function (id, value) {
                            $(JudgeScore).last().val(value);
                        });
                        ComputeTotalScore(JudgeScore);

                    },
                    errors: function (data) {
                        //console.log(data)

                    }
                });
                //COMPUTATION BEGIN!!!
            }
        }, 100);
    setTimeout(
        function()
        {
            // for (let i = 1; i <= criteriaCount; i++) {

            //     JudgeScore = 'input[name="' + nextChar(i-1) + rowCount + '"]';

            //     $(JudgeScore).last().val(Scores[i-1]);

            //     }
            //     //COMPUTATION BEGIN!!!
            // ComputeTotalScore($this);

        }, 500);
```

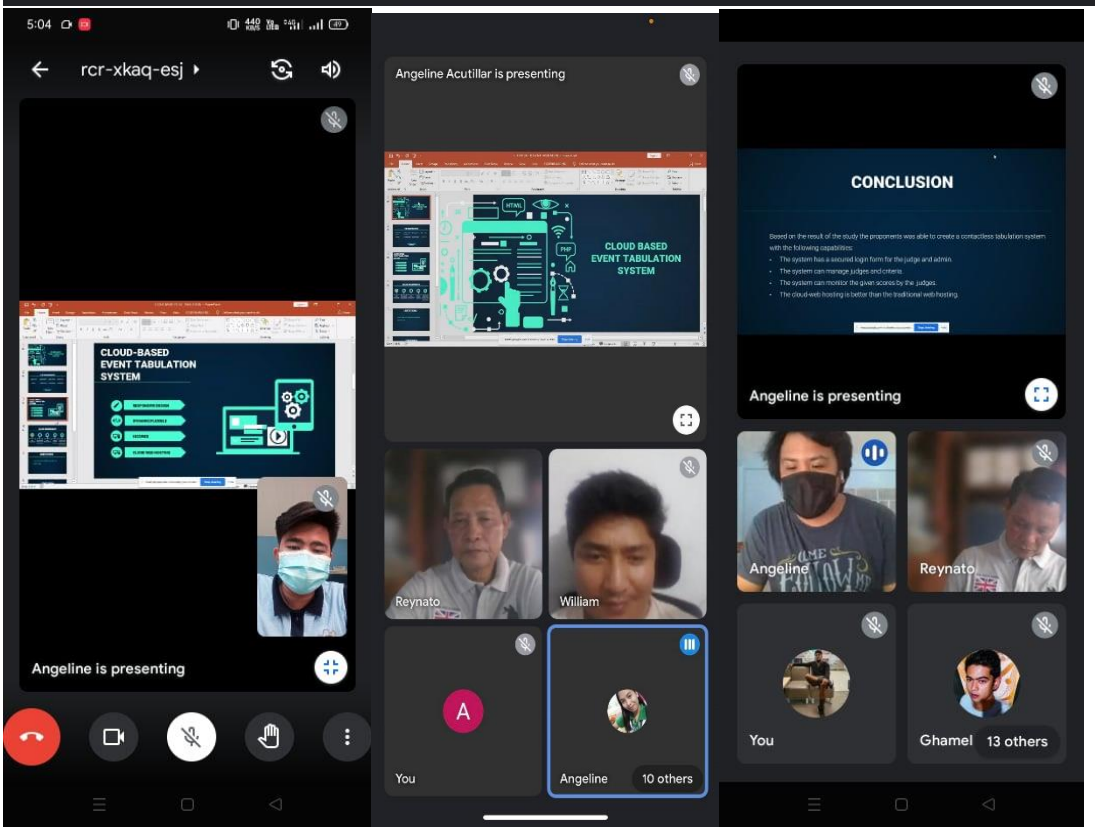
```
        });
    });

</script>
@endsection
```

Documentation

The Team RAMA's capstone final defense was held on June 29, 2021 at 5:05 pm. The following images below are the screenshots during the virtual final defense

.





Curriculum Vitae

Merson Labro Taguiam

PERSONAL INFORMATION

EDUCATIONAL BACKGROUND

Tertiary: **College for Research and Technology**
Bachelor of Science in Information Technology
Burgos Avenue, Cabanatuan City, Nueva Ecija
2017 – 2021

Curriculum Vitae

Ronell Miguel Picaña

PERSONAL INFORMATION

EDUCATIONAL BACKGROUND

Tertiary College for Research and Technology
Bachelor of Science in Information Technology
San Jose City, Nueva Ecija
CY: 2019-2021

Curriculum Vitae

Aljune Solis Aguilar

PERSONAL INFORMATION

EDUCATIONAL BACKGROUND

Tertiary College for Research and Technology
Bachelor of Science in Information Technology
San Jose City, Nueva Ecija
CY: 2017-2021

Curriculum Vitae

Angeline Sinobago Acutillar

PERSONAL INFORMATION

EDUCATIONAL BACKGROUND

Tertiary College for Research and Technology
Bachelor of Science in Information Technology
San Jose City, Nueva Ecija
SY: 2020-2021