

**A Web Management System for Gensan Youth Orchestra
and Choir Corporation**

**A Capstone Project
Presented to the Faculty of the
Information and Communications Technology Program
STI College General Santos**

**In Partial Fulfilment
of the Requirements for the Degree
Bachelor of Science in Information Technology**

**Andy Jay Borre
John Michael S. Panizales
Owen Lourd L. Ravina
Calvin Drakke I. Rulete**

October 2024

ENDORSEMENT FORM FOR ORAL DEFENSE

TITLE OF RESEARCH: A Web Management System for
Gensan Youth Orchestra and Choir
Corporation

NAME OF PROPONENTS: Andy Jay Borre
John Michael S. Panizales
Owen Lourd L. Ravina
Calvin Drakke I. Rulete

In Partial Fulfilment of the Requirements
for the degree Bachelor of Science in Information Technology
has been examined and is recommended for Oral Defense.

ENDORSED BY:

Allyssa Kaye A. Parcon
Capstone Project Adviser

APPROVED FOR ORAL DEFENSE:

Ivy Grace C. Laurente
Capstone Project Coordinator

NOTED BY:

Anthony Mark F. Silong, MIT
Program Head

OCTOBER 2024

APPROVAL SHEET

This capstone project titled "**A Web Management System for Gensan Youth Orchestra and Choir Corporation**", prepared and submitted by **Andy Jay Borre, John Michael S. Panizales, Owen Lourd L. Ravina, and Calvin Drakke I. Rulete**, in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology, has been examined and is recommended for acceptance and approval.

Allyssa Kaye A. Parcon
Capstone Project Adviser

Accepted and approved by the Capstone Project Review Panel
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Information Technology

Dan Henly P. Sales
Panel Member

Dr. Ann Gilyn B. Premarion
Panel Member

Anthony Mark F. Silong, MIT
Lead Panelist

Jolina M. Migriño, LPT
Capstone Project Coordinator

Anthony Mark F. Silong, MIT
Program Head

Noted:

December 2024

ACKNOWLEDGEMENTS

The researchers would like to extend their gratitude to the following individuals and organizations:

To their capstone project coordinator, Ma'am Ivy Grace C. Laurente, for providing invaluable guidance, coordination, and support throughout the research process to ensure that the project stayed on track and met all necessary requirements; their capstone project adviser, Ma'am Allyssa Kaye A. Parcon, who offered expert advice, constructive feedback, and mentorship that greatly contributed to the refinement and success of this research; their capstone project review panel Dan Henley Sales, Dr. Ann Gilyn B. Premarion, and Anthony Mark F. Silong MIT, who provided thorough evaluation, insightful suggestions, and critical assessments that enhanced the quality and rigor of this study; and the researchers' parents and/or guardians, who provided unwavering support, encouragement, and understanding.

To the faculty members within STI College of General Santos who had provided insights upon the creation of this capstone project, Sir Darwin Berja, during the early developments of this capstone project for helping tremendously in crafting the early drafts of Chapter 1 as well as conceptualizing of which topic and host company should the proponents pursue in the creation of this capstone project.

To the researcher's friends and inspirations for their unwavering support and motivation throughout the research process. Moreover, to anyone else who helped in any way that made the research much easier to complete, whether with technical help, moral support, or other assistance. Finally, the researchers would like to express their gratitude to General Santos Youth Orchestra and Choir Corporation, especially the CEO, Sir Fernan Globen Talonding, their host company, for giving them a chance to work together and for their assistance, resources and cooperation, all of which were essential for the project's successful completion.

ABSTRACT

Title of research: **A Web Management System for Gensan Youth Orchestra and Choir Corporation**

Researchers: **Andy Jay Borre**
John Michael S. Panizales
Owen Lourd L. Ravina
Calvin Drakke I. Rulete

Degree: **Bachelor of Science in Information Technology**

Date of Completion: **December 2024**

Keywords: **web management system, ticketing system, appointment reservation, GYOCC, music lessons, cloud computing**

The development of a comprehensive web management system for the Gensan Youth Orchestra and Choir Corporation (GYOCC); the primary goal of this system was to enhance the organization's operational efficiency and user experience. The system incorporated various modules, including ticketing, appointment booking, and comprehensive reports and analytics which utilized cloud computing, data analytics, and secured payment gateways; the system significantly boosted the efficiency and effectiveness of GYOCC's services, the main features of the system included online ticket sales, real-time booking updates, and detailed reports on ticket sales, event attendance, and lesson participation, where the user-friendly interface was designed to facilitate student's access to learning materials, enabled them to join lesson sessions, book appointments for events, and purchase tickets with ease, meanwhile, for administrators, the system offered robust tools to manage user roles, oversee events, and ensured smooth operations. Extensive alpha testing was conducted with GYOCC members, instructors, and administrators, yielding high satisfaction rates because users praised the system for its ease of use, streamlined booking processes, and overall functionality. As such, the feedback highlighted the effectiveness of automated processes and the intuitive design of the modules for booking appointments, ticketing, and event management. This web management system not only enhanced GYOCC's service delivery but also reduced manual workload, significantly improving the user experience by integrating modern technological solutions.

TABLE OF CONTENTS

	Page
Title Page	i
Endorsement form for Oral Defense	ii
Approval Sheet	iii
Acknowledgments	iv
Abstract	v
Table of Contents	vi
Introduction	1
Project Context	1
Purpose and Description	4
Objectives	6
Scope and Limitations	7
Review of Related Literature/Studies/Systems	12
Methodology	18
Technical Background	20
Requirements Analysis	29
Requirements Documentation	32
Design of Software, System, Product, and/or Processes	38
Development	40
Results and Discussion	42
Testing	42
Description of Prototype	44
Implementation Plan	47
Conclusion	49
References	50
Appendices	53
Resource Persons	54
Sample Input/Output/Reports	56
User's Guide	60
Photo Documentation	90

Accomplishment and Consultation Form	94
Transcription	103
Revision's List	112
Figures and Tables	116
Certificates	157
Personal Technical Vitae	162
Relevant Source Code	167

INTRODUCTION

Project Context

The Gensan Youth Orchestra Choir Corporation is an organization that specializes in providing music lessons, choir, and orchestral performances to nurture the talent of young musicians of various instruments in the city of General Santos; they also host performances in events such as weddings, corporate events, city festivals, etc. Founded in May 2014, GYOCC serves as a training ground for aspiring musicians and singers, and GYOCC polished these hidden gems and instilled the values of discipline, teamwork, and cultural appreciation. The organization quickly rose to prominence by securing two gold medals and certificates in November 2018 during the Cantata Macau International Choral Competition and Festival. Since its founding, GYOCC competed internationally and provided quality lesson services to aspiring musicians and talents (Gensan Youth Orchestra and Choir Corporation, 2024).

However, as GYOCC gained popularity and its services became used more often by several clients, it encountered several challenges due to the overwhelming demand for its services, including how to manage booking appointment services, ticketing services, lesson sessions, and event services for performances and concerts.

The proposed capstone project aimed to develop a web management system for GYOCC in response to these challenges. The web app was composed of several modules to complement the services offered by GYOCC, such as a reports & analysis module, which tracks ticket sales, event attendance, and monthly bookings to aid organizational decision-making. One of the features of the system was an integrated calendar module, which manages event scheduling for performances, auditions, and lessons. This module provided real-time updates, allowing staff and instructors to organize bookings and lessons easily. At the same time, the system checked schedules to avoid conflicts and warned the staff appropriately if a schedule conflict was found. This web management system not only aided GYOCC's operations efficiently and smoothly but also elevated the experience for the staff, instructors, members, and event performers inside the organization.

In relation to the proposed web management system, it was essential to emphasize modern technology, specifically cloud computing, data analytics, and web-based platforms, in optimizing GYOCC's operations. According to a study by Sutabri, Wijaya, Seprina, and Amalia (2023), internet services for accessing information became increasingly popular due to their extensive reach and accessibility. These technologies effectively promote, manage, and share data across organizations. Because of the integration of modern technology, it was great for promoting and publishing information.

Moreover, the function and importance of implementing a ticket management module was further solidified by what this journal article states: ticketing sales and management was intended to solve member difficulties, such as those encountered when ordering tickets or searching for information on tickets or other items. For example, assisting all consumers whose access to ticketing counters was limited by distance and time, aiding customers and ticket sellers in the purchasing and selling process, advertising the company, and giving services to customers about online and on-time ticket booking (Sutabri, Wijaya, Seprina, & Amalia, 2023). Henceforth, implementing a ticket management module improved GYOCC's operational capabilities for concerts, ensuring that every ticket sold and every audience member was accounted for, with all transactions logged to address any discrepancies effectively.

Furthermore, the booking appointment module of the system maintained records and logs of each aspiring learner's progress, schedules, and payment history, providing the organization with business insights into improving their services and keeping track of and managing event concerts or performances provided by GYOCC. According to this article, the benefits of an appointment system in correspondence to an academic or educational purpose offered additional advantages, including more targeted instruction, more information about learner needs, more flexibility in rerouting appointment requests, and more effective use of staff and student time—all while maintaining the possibility for human intervention in the system when necessary (Morris & McDermott, 2021). What the article stated provides a direct solution to one of the main problems that GYOCC was experiencing, such as the system's booking appointment modules, which were important and must be implemented to solve these problems. Implementing the system not only

solved one of their current problems but also digitalized this currently manual process within the organization. The system enhanced the efficiency of the operations within GYOCC and elevated their audience, members, and instructors' experience, which displayed a seamless integration of the organization's concerts and instructional undertakings, which further solidified their reputation as a pillar in the entertainment industry, specifically orchestra in the local community of General Santos and the future of the entire region.

Purpose and Description

The primary purpose of the web management system for GYOCC was to improve their ticketing processes in concerts further, as well as booking appointment scheduling for private and public meetings by tackling the current problems in which the manual booking system leads to delays and inefficiencies for members and staff. Additionally, the manual handling of event scheduling and booking appointments often resulted in scheduling conflicts. A Web Management System for the Gensan Youth Orchestra and Choir Corporation system addressed these challenges by providing a streamlined, automated platform that ensures faster, more efficient processes.

The proponents wanted to clarify that the members specified in this capstone project are members of the website, not to be confused with members of the actual GYOCC's orchestra and choir.

The system was an online platform accessible through a browser that served as a central hub for GYOCC, allowing services such as concert ticketing and booking appointments for meeting schedules to be easily accessed and managed by staff and interested individuals. A unified platform simplifies interaction with the organization, ensuring smoother operations and an enhanced user experience for members engaging with GYOCC's services. The system focused on booking appointments that allowed interested individuals in hiring GYOCC to do an organized event with their specified details of place, type of theme, and date, warn staff, instructors, and superusers on conflicting dates, upcoming schedules, and provide a calendar of performance or concert events that are lined up within their schedules.

One main component related to booking appointments was the ticketing system, which allowed customers to purchase tickets online or at physical sales outlets. The purchased tickets contained details about the user, including a unique ticket ID linked to their user ID; the ticket information was encrypted, which only the system could decode to prevent data leak, and to further add security, the proponents added a two-factor authentication implementation.

The one-time lessons offered by GY OCC were managed through the instructor component of the system. This component focused on scheduling individual or group lessons for instructors, separate from the general event booking appointment service used for concerts and other organizational events. Each instructor had a specific schedule for lessons, and students could book based on their preferred instrument, instructor, and available scheduled times set by the instructor.

In addition, the instructor component worked with two other components: the resource sub-component, which managed physical resources like instruments and uniforms, and the repertoire sub-component, which handles e-learning materials such as musical pieces, video tutorials, and other learning aids. These materials were accessible to students via the website.

The main goal of this system was to provide members with the ability to book lessons tailored to their preferences, allowed them to book an appointment to hire the orchestra and choir of the organization for their public or private events, book an appointment for auditions to join the organization as a choir or orchestral member, and purchase tickets for events that were offered by the organization, whether ongoing or upcoming. Additionally, this component used the payment management system, as these services were part of GY OCC's paid offerings.

Furthermore, the system also provided an analytical report, such as ticket sales for the organization after each event or as specified by a group of events in a specific date range. The system recorded attendance through the QR codes scanned during the event. The system also recorded the payment transactions processed through the payment methods in the system.

The system had an optional email subscription to notify users about future events. A FAQ (Frequently Asked Questions) was included to help members navigate the website and answer frequently asked questions.

Objectives

The main objective of this study was to develop and implement a web management system for GYOCC that addressed the problems and challenges faced by the Gensan Youth Orchestra and Choir Corporation (GYOCC). The proposed system reduces manual load, enhances customer experience, and improves resource management through the booking appointment module, ticket management module, member management module, and reports and analytics module. The proposed system provided a specific solution to GYOCC by achieving the following objectives.

The following were the specific objectives of the study:

1. To develop a module that allowed administrators and superusers to manage each user level within the system.

This module enabled detailed role-based access control, ensuring that each user level, from members to organizers, instructors, and administrators, had appropriate permissions and access to the system's features.

2. To develop a system that could book appointment meetings for events that are public or private through an integrated calendar.

The integrated calendar streamlines the scheduling process, allowing members to seamlessly book appointments for various public or private events based on their preferences. This reduces schedule conflicts and enhances organizational efficiency.

3. To develop a system that could generate unique tickets for concerts.

This feature ensured that tickets were securely generated and delivered to customers upon purchase. It included security measures such as unique identifiers, hashes, and two-factor authentication to prevent fraud.

4. To develop a system to generate analytics and reports, including ticket sales, event attendance, and monthly booking appointments.

The analytics and reports provided valuable insights into organizational performance by tracking metrics such as ticket sales, event attendance, and booking appointment trends. These reports aided in strategic decision-making and resource allocation.

Scope and Limitations

Scope

The web management system for GYOCC encompassed a comprehensive solution tailored to the specific needs of the Gensan Youth Orchestra and Choir Corporation (GYOCC). The system addressed the challenges faced by GYOCC in managing tickets for concerts and performances, bookings and reservations, lesson management, reports and analytics, and member management.

There are a total of 5 user levels, and each user level has a different scope of what modules they could access:

1. Superuser – The superuser oversaw all member, instructor, admin, and staff roles. The superuser created these roles through the groups and permissions of the member management module. The superuser could also generate summaries of past activities, such as payments, events, lessons, tickets, etc.

In the member management module, the superuser managed group roles and permissions. In the reports and analytics module, the superuser could generate summaries of past activities. Lastly, the superuser could see the current event calendar and any associated submodules through the appointment reservation/booking management module.

2. Administrator / Staff – administrators and staff manage the day-to-day management of members below their role, including instructors and members; they also could oversee user profiles, ticketing operations, event operations, lesson operations, borrowing of organization assets, overseeing internal messaging, keeping social media integrations working, keeping the repertoire resources up to date, and generate summaries of past activities. The administrator/staff had access to almost all modules as they oversaw the usage of these modules by member roles below them.

2.1. Appointment Reservation / Booking Management Module – the administrator or staff could manage the booking management module for when edits, cancellations, or approval of booking was needed as requested by the booker or head organizer; they could also manage the ticket management submodule for when it was used in an ongoing event, they could check the validity of the tickets in combination with the two-factor authentication submodule. Lastly, the administrators and staff also oversee the payment process within the organization; if a certain error or mishaps occur, they are the ones to address these issues to the members that are affected by it; the administrator and staff also manage the two-factor authentication module of the proposed system for validation checking of tickets to be used in a certain event, this module goes hand in hand with the ticket management submodule.

2.2. Reports and Analysis Module – the reports and analysis module were also accessible to the administrators and staff; this was for when they needed to generate reports to find conclusive evidence of certain aspects since the operational running of the system as well as use these reports for informed decisional making, the proposed system provides the following analytics with csv reports: specific event, events within a specific date range, monthly bookings, and lessons within a specific date range.

2.3. Member Management Module – the administrators and staff also had access to the member management module to oversee member, organizer, or instructor profiles to be kept in line with the set rules and guidelines of the organization; the administrators and staff could also manage member roles of instructors, members, and organizers a like for cases where a member promoted and hired as an organizer or instructor within the organization, administrator, and staff also had access to the messaging submodule for internal communication where a member may suggest, tell a concern, or provide feedback within a certain aspect of the organization or the system, the administrators and staff also oversee the messages sent within this internal messaging module as they should follow proper messaging communication guidelines set by the organization. Lastly, administrators and staff helped maintain the social media interaction submodule to keep the website's news or activity page up to date with other social media pages that the organization had.

3. Organizers — organizers were the entities responsible for managing an event that was assigned to them, including ticket and member management. The organizer's main job was to keep the event running smoothly and according to the wishes of the member client who requested it.

3.1. Member Management Module — particularly the messaging submodule and the two-factor authentication submodule. The former was for internal messaging of inquiries, feedback, suggestions, or concerns, while the latter focused on securing the instructor's account with two-factor authentication.

3.2. Appointment Reservation / Booking Management Module — The organizers could manage booking appointment reservations as well as related events and tickets. This allowed the organizer to do his / her job mainly in managing this module. It also allowed the organizer to scan tickets for events assigned to him/her.

4. Instructors — instructors were entities responsible for lessons that were assigned to them. Instructors could set the instruments that they want to teach, including the number of students that they could manage.

4.1. Lesson Management Module – the instructors needed this module to keep track of a member student's performance throughout the lessons provided by the instructors; in the case of instructors wanting to supplement the lesson at home, they may recommend member students to browse through the repertoire management submodule's resources of the proposed system where online learning materials could be browsed based upon their chosen instrument, as for an instructor's physical lessons the instructors could access the resource management submodule mainly on the borrowing of resources within the organization when the instructor needs musical pieces or instruments to be borrowed for the use of lessons for the student's use or for his/her use, he or she may request it in this submodule.

4.2. Member Management Module – particularly the messaging submodule and the two-factor authentication submodule, the first one for internal messaging of inquiries, feedback, suggestions, or concerns, while the latter focuses on securing the instructor's account with

two-factor authentication.

5. Members – this membership had three main types but was interconnected to one another, meaning what they saw was the same as one another; one member could be a member client that was interested in hiring the organization to do a concert or perform in accordance with their events, another member can be a member that was interested in attending upcoming or ongoing events within the organization, they purchase tickets to the selected events. Lastly, the member student was for members interested in learning musical pieces and instruments through the website by requesting a lesson schedule with their selected instructor and instrument.

5.1. Lesson Management Module — where member students could join a lesson with their chosen instrument, instructor, and schedule. Suppose a student wanted to learn an instrument but did not have it. In that case, they could access the resource management submodule, where member students could request to borrow a certain resource within the organization to use for their lessons.

5.2. Appointment Reservation / Booking Management Module – where member clients could hire the organization to perform for their events, including specific details regarding that event; they could also access the events where members could find the latest upcoming events or workshops and information regarding them and if they saw an event they wanted to attend to they had access to ticket management submodule where member customers can purchase tickets for events that they want to attend.

5.3. Repertoire Management Submodule – where member students had access to online learning materials for instruments that they were actively learning.

5.4. Member Management Module – particularly the messaging submodule and the two-factor authentication submodule, the first one for internal messaging of inquiries, feedback, suggestions, or concerns, while the latter focuses on securing the member's account with two-factor authentication.

Limitations

However, the proposed system project had several limitations that may affect its ability to function and be realized, including:

1. The system was built for the Gensan Youth Orchestra and Choir Corporation, so it might not be compatible with other organizations offering the same services as GY OCC.
2. The system's reports and analytics module could not provide or generate reports regarding instructor feedback data, event revenue, and income, as the organization's expenses are confidential. This hindered the proposed system from calculating the final revenue and income after expenses.
3. The lesson session module could not create recurring or sequence lessons and only allowed one session to be scheduled at a time.
4. The audition appointment had no callback after an audition appointment as this was managed by the organization itself, which was not part of the scope of the proposed system, as the system was specifically designed to handle only the appointment booking functionality and does not extend to post-audition processes or membership management within the organization.

Review of Related Literature

The development of the web management system for GYOCC was closely inspired by the insights and best practices observed in existing studies about ticketing systems, appointment reservations, or booking systems; by reviewing the current research and case studies in this category, the proponents could gain a deeper understanding of the key factors that contributed to the success and widespread implementation of such technology-driven solutions. Published research adds to the efficiency of appointment reservation systems by stating that using the Technology Acceptance Model (TAM) and System Usability Scale (SUS), the study sought to evaluate the deployed system's efficiency. The results, which indicated a notable level of client satisfaction with an overall percentage of 90.2, validate the efficacy of the web-based appointment system. Because of this, this innovative strategy may significantly cut down on customer wait times at the Office of the President and facilitate more convenient employee appointment scheduling, both of which could be significant steps toward more effective administrative processes (Ampuan & Deleña, 2024).

This was furthered reinforced by this journal article, which stated that as the dependent variable, the study examined the variables that influence a customer's intention to utilize the online reservation system. The independent variables that may influence a customer's intention to use the organization's online reservation system include the level of service provided, the ease of access to the internet, the method of payment, and the lack of information provided on the website. The operators who allowed online reservations said that their customers were happy with the way this approach works. They consistently received compliments on how convenient their online reservation was (Generoso, Illagan, Evangelista, Mendoza, & Pulhin, 2019).

Furthermore, another literature article further solidified that e-commerce websites could support an increasingly popular method of booking or reservation, as demonstrated by the emergence of e-services. On this subject, online reservation systems could be developed for bus terminals, airports, lodging facilities, movie theatres, and other reservation-based establishments (Akinyede, Balogun, & Iwasokun, 2017)

Lastly, the utilization of online ticket booking systems and the perception of control were significant moderators of the relationship between socioeconomic position and customer well-being (Guan, Wu, & Jia, 2020).

One piece of literature stated that fans of popular music have switched from purchasing records to purchasing concert tickets in the era of streaming. This change begs the question of how concertgoers evaluated their experiences. In response, this article made the case that three important financial, emotional, and experiential factors shape the conversation because of their impact on ticket purchases (Westgate, 2019).

These research articles highlighted the critical role of the ticketing system component within the integrated solution for GYOCC. By incorporating the main factors and best practices gathered from these studies, the proposed system could be designed to deliver a more comprehensive and impactful experience for the organization's customers, ultimately contributing to the overall success and sustainability of its concert offerings.

Related Studies/Systems

In developing a web management system for GYOCC, it was important to investigate existing systems and solutions that address similar functionalities and requirements. By analyzing the features, strengths, and limitations of related ticketing and appointment reservation systems currently available in the market or implemented by other organizations, the project team could gain valuable insights to inform the design and implementation of the proposed integrated system.

A system named BOOKAZOR implements its own system through the use of a web stack, where they further elaborated by stating that Bookazor was a web-based application for scheduling and making appointments that were used to make appointments in certain locations with parlors, hospitals, and architects. It was an open-source SDK for creating hybrid mobile applications that made use of JavaScript, HTML, and CSS. Firebase provided features including analytics, database, messaging, and crash reporting, which were utilized for data fetching. A server that made appointments available at predetermined times was part of the system, and NodeJS was used to store requests. The scheduler uses a simulated annealing technique to create a new set of appointments every time it updates routes to avoid repetition (Akshay, Kumar, Alagappan, & Gnanavel, 2019).

One system also experimented with the idea of implementing such related systems cross-platform. Their article stated that the study found that the SDLC may be used to create a cross-platform scheduling and appointment booking system for clinics, with end users and IT specialists participating in the assessment phase. The ISO 25010 requirements must be incorporated to guarantee that the system could be of the highest calibre and satisfies end users' needs (Fuente, et al., 2023).

This one foreign system/study indicated that online movie streaming had grown significantly in popularity because of technological advancements and digitization. It offered a quick and simple option to order tickets online, saving time from standing in line at the movie theatre. It was really fascinating to peruse the newly released films at home, read the reviews, and then reserve the seats based on our budget. In the suggested study paper, we resolved some of the most important problems with the online movie

management system. Due to an increase in traffic, server issues and transaction failure were the major problems with online movie booking systems with ticketing functionality built-in (Archit, Vinit, & Rajesh, 2019).

Another system made at the University of Alabama, TickIt, stated that customers dissatisfied with ticketing might benefit from the creation of a new platform, given the tensions that have recently arisen between the present ticketing platforms. This was the reason for TickIt's design and development. This working prototype was intended for ticketing events in Huntsville, Alabama's Von Braun Center Playhouse and Concert Hall (Holbrook, Lukins, Berzins, Hudson, & Smith, 2023).

In addition to the system above, another system also suggested that to offer a quick and easy ticketing process, we suggested a smart application that could digitally reserve tickets, assign seats to passengers automatically, and accept cashless payments, all of which will support digitization and smart city projects (Kazi, Bagasrawala, Shaikh, & Sayyed, 2018).

The analysis of existing ticketing and booking appointment systems provides valuable insights for the development of A WEB MANAGEMENT SYSTEM FOR GY OCC. The insights concluded that the ever-growing demand for online ticketing solutions was driven by technological advancements, the need for smooth and user-friendly experiences, and the importance of addressing the limitations of current systems, such as server issues and transaction failures. The proposed system could be designed to deliver a comprehensive, efficient, and innovative solution that sets GY OCC apart in the arts and cultural sector by strategically incorporating these learnings.

Synthesis

The review of related literature and existing systems provided a strong foundation for the development of the web management system for the Gensan Youth Orchestra and Choir Organization.

The insights gathered from the literature and systems could be synthesized as follows:

The available literature revealed a significant trend toward e-commerce and e-services, particularly in booking and reservation systems; as such, this growing demand underscores the importance for GYOCC to implement a robust online ticketing and appointment scheduling platform that meets evolving customer expectations.

Within this context, research and analysis of existing systems highlighted several critical factors that influence customer adoption of online reservation platforms, including service quality, accessibility, payment options, and information availability; these elements play a vital role in creating a web management system that ensured a seamless and satisfying user experience.

Research has consistently shown the effectiveness of web-based appointment systems. Studies reported high client satisfaction rates and improved organizational efficiency, and these findings supported the viability of implementing an integrated solution for GYOCC's needs.

The ticketing system emerged as a particularly significant component, reflecting evolving consumer preferences in concert ticket purchasing behaviour. Online ticketing has been shown to positively impact customer well-being and their sense of control over the process.

By examining related platforms such as online movie booking systems, smart ticketing applications, and cross-platform appointment scheduling solutions, we could draw valuable lessons about innovative features, technical considerations, and potential challenges; as such, these insights informed the development approach for the GYOCC system.

By synthesizing these findings, the project team could comprehensively understand the

current landscape, industry best practices, and critical success factors. This knowledge could then be strategically incorporated into the design and implementation of the Integrated Ticketing and Appointment Reservation System, ensuring that it delivered a cutting-edge and operationally efficient solution that sets GYOCC apart in the arts and cultural sector.

METHODOLOGY

The journey of developing the capstone project for the Gensan Youth Orchestra and Choir Corporation (GYOCC) was meticulously planned and executed through a series of well-defined phases, each enriched with targeted activities to ensure comprehensive progress and successful completion.

In the early days of February, the course began with an orientation session, setting the stage for what was to come. The project coordinator elaborated on the course structure, expectations, and key milestones, and this initial phase was crucial as it laid the groundwork for the entire project.

Shortly after the orientation, students form groups, selecting their teammates to share this challenging yet rewarding journey. Each group then dives into the topic selection process, exploring various ideas and aligning on a project that resonates with their interests and the course objectives. In the meantime, students also sought guidance and approval from potential capstone project advisers, initiating consultations that would continue throughout February and March.

With the project topics and advisers in place, the proponents move into the risk analysis phase. This phase was characterized by in-depth discussions and planning sessions aimed at identifying potential risks and challenges. The proponents met with their adviser, presenting their draft of Chapter 1, which outlined the project's context, objectives, and preliminary scope. These consultations were pivotal, as they provided early feedback and highlighted areas needing refinement.

During these months, the proponents were also busy revising Chapter 1 and beginning the groundwork for Chapter 2. They were prepared for their first significant milestone: a meeting with the host company, GYOCC; this meeting served to align the project's goals with the company's needs, ensuring that the proposed solutions were both relevant and feasible.

The engineering phase spanned several months, reflecting the complexity and depth of work involved. In February and March, the proponents continued their consultations with advisers and the host company, fine-tuning Chapters 1 and 2 based on the feedback received. They then dive deep into problem-solving sessions with the host company, identifying specific issues and brainstorming potential solutions.

As April approached, the focus shifted to finalizing Chapters 1 through 3. This period was marked by intense writing, revising, and adviser consultations aimed at producing a polished, comprehensive document. Simultaneously, the proponents began creating wireframes, translating their conceptual ideas into visual representations that guided the subsequent development phases.

By late April and early May, the proponents were ready to present their capstone project proposal. These presentations are not merely academic exercises but critical junctures where ideas were scrutinized and constructive feedback was gathered. The period then culminated with the Capstone Project Proposal Defense, a rigorous evaluation that tested the robustness and feasibility of the proposed system.

The evaluation was an ongoing process that started in February and extended through September, encompassing both formative and summative assessments. Weekly consultations were scheduled to ensure continuous feedback and iterative improvements starting from the outset. As the project evolved, the drafting of Chapter 4 began, capturing the development and initial implementation details.

Throughout the summer months, the proponents engaged in alpha testing and mock defenses. These activities were designed to simulate real-world usage and identify any functional gaps or areas for enhancement, and by the end of August, the proponent's system should have a 60% to 80% working system ready for the first mock defense.

In September, the focus shifted to refining Chapters 4 and 5, incorporating insights from alpha testing; the signing of endorsement letters signified that projects were ready for final defense, a culmination of months of hard work and dedication.

As the project nears its conclusion, final preparations were made, including addressing any last-minute feedback, ensuring all documents were complete, and making final adjustments

to the system based on the final round of testing. The project was then presented for the final defense and evaluated by a panel of experts.

Throughout this meticulous process, the project was continually shaped and refined, ensuring that the final output was a robust, well-documented system that met the needs of GYOCC and stands as a testament to the student's hard work and ingenuity.

Technical Background

Overview of Current Technologies to be Used in the System

Integrating modern technologies into the proposed system for the Gensan Youth Orchestra and Choir Corporation emphasized the use of online platforms and internet services to streamline ticket sales for concerts, appointment scheduling for music lessons, and the organization hiring for private or public concerts and performances. The system's components included a user-friendly ticketing system and an appointment reservation system, both of which leverage internet services for accessibility and efficiency. Integration with Callendar functionality to enhance scheduling while a secured payment gateway ensured smooth transactions. Despite the benefits, challenges such as technical constraints, security concerns, and user adoption issues must be addressed for successful implementation. Overall, integrating modern technologies aimed to enhance operational efficiency, improve customer experience, and maintain GYOCC's reputation as a leading arts organization.

Online Platforms and Internet Services integration of online platforms was central to addressing the challenges faced by GYOCC. The organization effectively promoted and published information, streamlined ticket sales, and managed appointment scheduling by utilizing internet services, particularly online platforms. This aligned with the broader trend of increasing reliance on online services for accessing information and conducting transactions.

Ticketing System the implementation of a ticketing system revolutionized how GYOCC managed ticket sales for its concerts. The organization efficiently handled ticket inventory, pricing, and sales tracking by leveraging modern technology, such as user-friendly online

platforms and physical sales outlets. This simplified the ticket purchasing process for customers and ensured accountability and transparency in ticket management.

Appointment Reservation System, where the appointment reservation component addressed the scheduling challenges encountered by GYOCC in managing music lessons. Students easily accessed instructor availability, scheduled lessons, and made payments by using an online platform to book appointments with instructors. This improved the efficiency of scheduling and enhanced the experience for aspiring music students.

Payment Gateway ensured secure and reliable payment processing, which was essential for the system's success. GYOCC guarantees smooth and secured online and offline transactions by implementing a robust payment gateway that supports multiple payment methods, thereby enhancing customer trust and satisfaction.

Django was a Python-based web framework that simplified the development of web applications. It provided built-in tools for handling user authentication, content management, and database interactions, speeding up the development process. For the GYOCC system, Django managed the backend operations to ensure that features such as ticket sales and appointment scheduling worked smoothly.

Apache 2 was a web server software that hosted GYOCC's online platforms. Its primary role was storing, processing, and delivering requested information or webpages to users. Apache 2 was used for the GYOCC system due to its compatibility with various modules and meeting the specific requirements for the system's ticketing and reservation systems.

PostgreSQL was an open-source database system used to store and manage the data for GYOCC's operations. PostgreSQL handles large amounts of information, such as ticket inventory and appointment schedules, and supports complex queries, large datasets, and JSON data types. PostgreSQL integrated smoothly with Django, which enabled the system to store, retrieve, and manipulate data efficiently while maintaining data integrity and security.

Hostinger's Virtual Private Server (VPS) was a hosting platform that offered dedicated resources for the GYOCC system. Hostinger provided an isolated environment for

deploying and managing the GYOCC web management system and offered a user-friendly control panel.

Technical Infrastructure: The system's development and implementation depend on a range of hardware and software components, including laptops, PostgreSQL, GitHub, Figma, Visual Studio Code, and Microsoft Office. These tools enabled the system's creation, testing, and documentation, ensuring its functionality and usability.

Limitations and Challenges: despite the benefits of integrating modern technologies, the project faced several limitations and challenges, including technical constraints, security concerns, adoption and training issues, and operational considerations.

Addressing these challenges was crucial for ensuring the successful implementation and operation of the proposed system. The proponent's chosen approach to system development, named the Inverted Spiral Model, was a strategic framework for system development that emphasizes continuous progress and risk management; the model was initiated with thorough planning, where project goals and constraints were clearly defined; following this, the risk analysis was conducted to identify potential obstacles that could affect the project's success. Prototyping and design lead into the engineering phases of coding, integration, and testing; this continuous cycle of implementation and evaluation ensured that each iteration of the project was assessed for quality and effectiveness, allowing for timely adjustments and refinements. This structured yet flexible approach enabled a responsive development process and fostered a robust and efficient path toward successful system deployment.

In summary, integrating modern technologies into the proposed system for GYOCC represented a significant step towards enhancing operational efficiency, improving customer experience, and maintaining the organization's reputation as a leading arts and cultural institution. GYOCC streamlined its operations and provided exceptional services to its customers and students by leveraging online platforms, advanced scheduling systems, and secure payment gateways.

Calendar of Activities

Course orientation at the beginning of February, the project coordinator held a comprehensive orientation session for all students. This session was designed to provide an overview of the capstone project, outlining the flow of activities, key requirements, and critical milestones. It was an essential kickoff that set the tone for the months ahead, ensuring everyone was aligned on the expectations and goals.

Following the orientation, the proponents embarked on forming their project groups, each consisting of four members. This initial phase was crucial as it involved selecting the right teammates and brainstorming potential capstone project titles. The proponents engaged in detailed discussions to choose topics that not only interested them but also met the course requirements. Concurrently, they began preparing for their title presentation, a pivotal step that guided the direction of their project.

With a clear direction in mind, the proponents sought out advisers whose expertise aligned with their chosen topics. They carefully selected advisers who provided valuable insights and guidance throughout the project. This selection process was pivotal in ensuring the proponents had the support needed to navigate the complexities of their capstone project.

The proponents presented the initial draft of Chapter 1 to their advisers. The advisers provided critical feedback, suggesting revisions to the objectives, area of study, project context, and purpose and description. This iterative process of review and revision helped refine the project's foundation, ensuring a strong start.

After receiving feedback, the proponents revised Chapter 1 accordingly. Simultaneously, they began drafting Chapter 2, which focused on literature from the year 2015 to the present. The advisers' guidance during this phase was instrumental in shaping the direction and depth of the research.

The proponents continued to consult with their advisers, who discussed the revisions made to Chapter 1 and the progress on Chapter 2. These consultations were crucial for validating their approach and ensuring the chapters met academic and project standards.

Then, meeting with the host company came where the proponents met with the CEO of their host company, Gensan Youth Orchestra and Choir Corporation. This meeting was

aimed at understanding the company's operations and identifying specific challenges that the capstone project could address. It provided real-world context and ensured the project would be practically relevant.

Following their initial meeting, the proponents spent time verifying the identified problems and brainstorming potential solutions. They discovered that the company lacked an efficient system for booking members and distributing tickets, which hindered smooth operations. The proponents then proposed solutions to address these issues, aligning their project objectives with the company's needs.

As the project progressed, the proponents worked diligently to finalize Chapters 1 through 3. This involved continuous revision and refinement, guided by their advisers' feedback. The goal was to prepare these chapters thoroughly before the upcoming title proposal defense.

Then, in preparation for the title proposal defense, the proponents consulted extensively with their advisers. They reviewed the chapters that presented and discussed strategies for effectively communicating their project's significance and feasibility.

The advisers suggested further revisions, including adding an Entity Relationship Diagram (ERD) of the proposed system; the proponents incorporated these suggestions, ensuring their documentation was comprehensive and ready for the defense.

The proponents then presented their capstone project proposal. During this presentation, panelists provided valuable feedback and suggestions on various aspects, such as hosting options and diagrammatic representations like ERD, DFD, CFD, and Use Case Diagrams. This feedback was crucial for refining their project.

Following the proposal presentation, the proponents sought further advice from their advisers. They discussed the panelists' recommendations and received guidance on effectively implementing the suggested changes. Armed with the panelists' and advisers' feedback, the proponents met again with the host company. They informed the company of the proposed changes and discussed how these impacted the system, ensuring alignment with the company's expectations.

The proponents then focused on finalizing all the chapters in preparation for the project proposal defense. This involved meticulous editing and ensuring that all elements were cohesive and well-documented. In parallel with finalizing the documentation, the proponents brainstormed and created wireframes for the proposed system. These visual representations were crucial for the project proposal defense, illustrating the system's design and functionality.

On the day of the capstone project proposal defense, the proponents successfully defended their project proposal with a verdict of major revision, showcasing their system's potential and readiness for the development stage as well as improvements that they can make; this defense marked a significant milestone, allowing them to proceed with development.

As the development stage commenced, the project coordinator conducted another orientation to outline expectations and milestones for this phase. Weekly consultations with advisers ensured the proponents received ongoing feedback and support, guiding their development efforts.

During this period, the proponents began drafting Chapter 4, documenting the system's development process. With their advisers' help, they detailed each step, from initial design to coding and testing.

The proponents conducted a mock defense, presenting a partially functional system to their advisers, the project coordinator, and an external panel. This exercise provided invaluable feedback, helping them identify areas for improvement. In response to the mock defense feedback, the proponents revised Chapter 4 and began drafting Chapter 5. These chapters documented the testing phases and detailed the final stages of system development.

Then, the proponents conducted alpha testing with the host company and its members. This real-world testing phase was critical for identifying and resolving any remaining issues, ensuring the system was robust and user-friendly.

Finally, the proponents processed the endorsement letter, and formal approval was required to proceed to the final defense. This endorsement signified that all requirements had been met, paving the way for the project's successful completion. Refer to Appendix H for the

Gantt Chart of Activities.

Resources

Hardware

This section of the study presented the hardware requirements and components that the system needs. It listed every important tool to the system's development.

Personal Computer

In developing the system, the personal computer was one of the essential hardware components that allowed the developers to design, develop, and test it. Refer to Appendix H for the table of minimum specifications.

Smart Phones

Developers tested and accessed the system using smart devices such as smartphones and tablets. These devices must connect to the system via the Wi-Fi network. Refer to Appendix H for the table of minimum specifications.

Software

Integrated Development Environment (IDE)

1. Visual Studio Code

Developers build, edit, and debug code with the help of Microsoft's integrated development environment before publishing an app. It was a Microsoft-made code editor that was employed in the development of websites.

2. Django with Apache2 Webserver

Developers employ Django with Apache2 web server. Leveraging Django's robust features like authentication, ORM, and templating, creating a user-friendly platform. With Django's

scalability and security, the developers ensured safe interactions and accommodated growth. Using third-party packages, the developers added features like payment APIs, SMS APIs, email APIs, etc., delivering an efficient solution tailored to the organization's members, while Apache2 provided a platform for the Django application to be deployed and published in the chosen VPS.

Version Control

3. GitHub

It was a tool for storing, monitoring, and teamwork on software projects. It made it simple for developers to cooperate on open-source projects, share code files, and pitch their work.

User-Interface Design

4. Figma

It was a collaborative web application for interface design with additional offline features enabled by desktop applications for macOS and Windows that are used for creating wireframes and user-interface design.

Database

5. PostgreSQL Database

The developers had decided to use the PostgreSQL database along with its management software pgAdmin4 in connection with the Django Web Server.

Hosting

6. Hostinger (VPS)

Leveraging Hostinger's infrastructure and user-friendly control panel, the proponents ensured seamless performance and scalability for the system's platform. With Hostinger's secured environment and optimized server resources, the developers expected reliable uptime and data integrity, which was crucial for handling transactions and user-sensitive

data information.

Requirements Analysis

The requirements analysis phase was crucial for understanding how users interact with the system and identifying the needs that the system must address. This section provided detailed insights into the user needs and the environment in which the system operates, ensuring that the proposed web management system for the Gensan Youth Orchestra and Choir Corporation (GYOCC) effectively meets those needs.

User Interaction with the System

The primary users of the GYOCC system include students or members, instructors, administrators, organizers, superusers, and staff. Each group interacts with the system differently to fulfill specific roles and responsibilities.

Students/Members use the system to access learning materials, join lesson sessions, book appointments for events, and purchase tickets. They need a seamless experience that allows them to manage their activities efficiently.

Instructors manage and schedule lessons, track attendance, and provide feedback on student progress. They require a system that simplifies administrative tasks, allowing them to focus on teaching.

Administrators oversee the entire system, managing user roles and permissions, organizing events, generating reports, and ensuring smooth operations. They needed robust tools to handle these diverse responsibilities effectively.

Organizers coordinate events, manage bookings, and handle logistical details. They relied on the system to streamline event management processes, ensuring successful event execution.

Superusers were advanced users with extended privileges who assisted administrators and provided support to other users. Their role was to ensure the system functions optimally and troubleshoot any issues.

Staff assist with various administrative tasks, including customer support and handling inquiries. They needed access to accurate information to perform their duties efficiently.

The system's business activities included ticketing, appointment booking, generating reports and analytics, user management, and payment management. These activities were performed within a web-based environment, making the system accessible from any location with internet connectivity. Users accessed the system from home, educational institutions, the GYOCC office, or event venues, ensuring flexibility and convenience.

The system operated on a real-time basis, providing users with up-to-date information on ticket availability, booking statuses, and schedule changes. It was designed for 24/7 access, allowing users to perform tasks at any time. The timing of scheduling and management activities were aligned with event dates and lesson timings to ensure efficient operations.

Prior to the implementation of the system, many processes at GYOCC were handled manually, including ticket sales, appointment bookings, and report generation. These manual processes led to inefficiencies and potential errors. Through interviews with the CEO, staff, and customers, as well as observations of current workflows, several areas were identified where automation significantly improves efficiency. Organizational processes were categorized and grouped into related modules and submodules, forming the foundation of the proposed system.

The proposed system addressed these needs through various computing solutions:

Ticketing Module: The need for efficient ticket sales and distribution management was met with an online platform that allows users to purchase tickets with real-time availability updates and automated sales reporting.

Appointment Booking Module: The need for simplified scheduling and management of lessons and events was addressed by a booking system that provides real-time updates on availability and allows users to view and manage their schedules.

Payment Management Module: The need for secure and efficient handling of online payments was met through integration with secure payment gateways, enabling real-time transaction processing and generation of receipts and confirmation emails.

Reports and Analytics Module: Tools that generated detailed reports on ticket sales, event attendance, and lesson participation addressed the need for comprehensive insights into various operations aspects, providing valuable analytics.

User Management Module: An administrative interface that allows administrators to manage user registrations, profiles, and permissions meets the need for effective user role and permission management, ensuring secure and organized user management.

By addressing these needs with appropriate computing solutions, the proposed system aimed to enhance the efficiency, accuracy, and user experience for all stakeholders involved with GYOCC. This comprehensive approach ensured that the system met current needs and was adaptable to future requirements, positioning GYOCC as a music education and performance leader.

Requirements Documentation

The purpose of this document was to outline the functional and non-functional requirements for the development of a web management system for the Gensan Youth Orchestra and Choir Corporation (GYOCC); this document served as a comprehensive guide for the project team ensured that the system met the specific needs of GYOCC and its diverse user base, the proponents had meticulously compiled these requirements to create a system that enhanced the operational efficiency and user experience for all stakeholders involved.

The web management system aimed to significantly improve GYOCC's operational processes and user experience. The system encompassed several key modules, including ticketing, appointment booking, and comprehensive reporting. By leveraging cloud computing, data reports, and secure payment gateways, the system was designed to make the services offered by GYOCC efficient.

The system provided real-time updates and allowed easy organization of bookings and lessons. It enabled staff, superusers, organizers, and instructors to check schedules to avoid conflicts, thus improving coordination and planning. Additionally, the system offered online ticket sales, real-time booking updates, and detailed reports on ticket sales, event attendance, and lesson participation. This comprehensive functionality aimed to enhance the GYOCC's overall management and operational capabilities.

The user roles within the system included Administrators, Instructors, Members, Organizers, and Superusers. The ticketing module handled an online sale of tickets, provided real-time updates on availability, and generated sales reports. Then, the appointment booking module managed bookings for events and lessons, provided real-time updates on booking statuses, and allowed users to view and manage schedules. In contrast, the reports and analytics module generated comprehensive reports on event attendance, lesson participation, sales, and losses, providing insights and analytics. Lastly, the member management module allowed administrators and superusers to manage user roles, permissions, and user profiles.

In terms of specific functional requirements, the system allowed users to purchase tickets online, provide real-time updates on ticket availability, and generate reports on ticket sales. It enabled users to book appointments for events and lessons, provided real-time updates on booking statuses and allowed users to view and manage their appointment schedules. Additionally, the system generated comprehensive reports on event attendance, provided analytics on lesson participation and user engagement, and generated sales and losses reports based on ticket sales, lesson transactions, and booking appointment transactions. Then, administrators were enabled to manage user roles and permissions. At the same time, users updated their profiles and contact information, logged in with a username and password, and changed account details, including password, name, mobile phone number, and email address. The system detected schedule conflicts and prevented the creation of conflicting events, providing real-time updates using a calendar and displaying events appropriately as a visual aid. Furthermore, the system allowed users to make payments online, integrated with secure payment gateways, processed transactions in real time and generated receipts and confirmation emails for completed transactions.

For non-functional requirements, the system could be accessible via web browsers on Windows, macOS, and Linux operating systems. It would support integration with secure payment gateways, process ticket purchases and bookings in real time, and retrieve reports and analytics within five seconds. The system used encryption to protect sensitive user data and implement role-based access control to restrict access to administrative functions. Additionally, it supported the English language for all user interfaces and complied with data privacy laws applicable in the Philippines.

The ticketing module was designed to handle all aspects of online ticket sales for GYOCC events. This module allowed users to purchase tickets online (REQ001), provided real-time updates on ticket availability (REQ002), and generated detailed reports on ticket sales (REQ003). By integrating this module, GYOCC made the ticketing process more efficient and user-friendly.

The appointment booking module manages the scheduling and booking of events and lessons. As such, it enabled users to book appointments for events and lessons (REQ004),

provided real-time updates on booking statuses (REQ005), and allowed users to view and manage their appointment schedules (REQ006). This module ensured that scheduling conflicts were minimized and users had a clear view of their commitments.

The reports and analytics module generated comprehensive insights into various aspects of GYOC's operations; it generated reports on event attendance (REQ007), provided analytics on lesson participation and user engagement (REQ008), and generated sales and losses reports based on ticket sales, lesson transactions, and booking appointment transactions (REQ009), this module was crucial for strategic decision-making and operational efficiency.

The member management module provided robust tools for managing user roles and permissions, allowing administrators to manage user roles and permissions (REQ010), enabling users to update their profiles and contact information (REQ011), and supporting user login with a username and password (REQ012). Additionally, it allowed users to change their account details (REQ013); this module ensured that user data was securely managed and that users had appropriate access to the system's features.

The event management module focused on preventing scheduling conflicts and providing real-time updates; it would detect conflicts of schedules and prevent events from being created by warning the appropriate staff, instructor, or superuser (REQ014); it would also provide real-time updates of these events using a calendar, displaying each range of events appropriately as a visual aid (REQ015), this module is essential for maintaining a well-organized event schedule.

The payment integration module handled all aspects of online payments, ensuring secure and efficient transaction processing. It allowed users to make payments online (REQ016), integrate with secure payment gateways (REQ017), process transactions in real-time (REQ018), and generate receipts and confirmation emails for completed transactions (REQ019). This module enhanced the convenience and security of financial transactions within the system.

Through careful research and consultation, the proponents ensured that these requirements

aligned with GYOCC's operational needs and strategic goals. By adhering to these requirements, the web management system provided a robust, secure, and user-friendly platform for managing the diverse activities and services of the Gensan Youth Orchestra and Choir Corporation.

The current system had the following pages that fulfilled the above requirements:

At the landing page shown in Figure 7.1, we saw what the user saw regarding the user interface of the system the first they visited the website; users that already had an account could directly log in to the website.

On the sign-up page shown in Figure 7.2, the system sign-up page was shown; this was where first-time users signed up for an account.

On the landing page, a user also clicked the FAQ button within the navigation bar to see the frequently asked questions regarding the system; we saw how it looks in Figure 7.3 below.

When a user had successfully logged in, it enabled to see Figure 7.4, which was the user interface of the user dashboard; it contained a hamburger menu consisting of a services tab, join GYOCC tab, history button, learning materials button, your calendar button, and notifications button.

At the Services Tab > Book an Appointment button, the user booked an appointment for an event. Refer to Figure 7.5 for how it looked and what relevant fields the user filled out to successfully book an appointment.

At the Services Tab > Purchase a Ticket button, the user purchased tickets for an ongoing or upcoming event; refer to Figure 7.6 for how it looked and what relevant fields the user filled out to successfully purchase a ticket.

At the Join GYOCC Tab > Audition button, the user could choose which part of GYOCC the user wanted to audition; refer to Figure 7.7 for how it looked and what choice the user had in submitting an audition.

At the Join GYOCC Tab > Audition > Audition for an Orchestra button, the user could submit an audition entry to GYOCC for their orchestra; refer to figure 7.8 for how it looked.

At the Join GYOCC Tab > Audition > Audition for a Choir button, the user could submit an audition entry to GYOCC for their choir; refer to figure 7.9 for how it looks and what fields the user needs to fulfill for a successful audition submission.

At the History button, the user saw their history regarding their transactions within the website; it included ticket history, booking history, lesson history, audition history, and private event payment history; refer to Figure 7.10 for how the history page looked.

At the Learning Materials button, the user saw a drop-down of different instruments; if an instrument was selected, articles available to that instrument were available for viewing; refer to Figure 7.11 for the learning materials page.

At the Your Calendar button, the user saw their schedule unique to the events, bookings, and tickets the user had purchased or submitted, which served as a visual aid for the user; refer to figure 7.12 for how the “Your Calendar” page looked.

At the Notifications button, the user saw a list of notifications unique to them, which the user clicked and viewed for more information, as well as buttons for marking selected notifications as read or unread. Refer to Figure 7.13 for how the notifications page looked.

My Profile button under the user’s profile picture at the top right was where their profile could be seen; they were allowed to change their details by clicking the edit button as well as changing their password; refer to figure 7.14 for the profile page’s look.

At the staff dashboard, the administrator, superuser, instructor, and organizer saw their own set of options in the hamburger menu at the left; refer to Figure 7.15 for how the staff dashboard looked.

At the staff dashboard > analytics page, the administrator and superuser saw reports regarding the data generated, gathered, and collected by the system using visual aids; refer to Figure 7.16 for how the analytics page looked.

At the staff dashboard > site configuration page, the administrator and superuser saw site settings that the system used as well as configuring them as the administrators and superuser saw fit; refer to Figure 7.17 for the analytics page's look.

At the staff dashboard > scan a ticket, administrator, superuser, and organizer could scan ticket QR codes for entry passing validation of users that attended an event; refer to Figure 7.18 for how the scan of a ticket looked.

At the staff dashboard > scan for attendance, administrator, superuser, and instructors scanned lesson attendance QR codes for lesson attendance recording of a specific lesson; refer to Figure 7.19 for how the scan a lesson attendance looked.

On the payment page, users could fulfill payments regarding services that they wanted; this included but was not limited to ticket purchasing and booking appointments that were public and private; refer to Figure 7.20 for how the payment page would look.

At the start of the 2fa verification page, users selected options to verify their identity; this was usually divided into two buttons: SMS OTP authentication or email OTP authentication. Refer to Figure 7.21 for how the 2fa verification page looked.

After selecting a 2fa verification, users could then input the OTP sent to their selected option and submit it for verification; refer to Figure 7.22 for the OTP input page's look. Refer to Appendix H. for the figures.

Design of Software, System, Product, and/or Processes

The Context Flow Diagram (Figure 3) showed the high-level interactions between the system and external entities, such as users and external systems; it also provided an overview of the system's limitations and the flow of information between the system and its environment; this diagram was essential for understanding the scope of the system to be developed. For a visual representation, refer to the Appendix H, Figure 3.

The Use Case Diagram (Figure 4) captured the functional requirements of the system by depicting the interactions between users (actors) and the system, where each use case represented a specific function that the system performs, such as booking appointments, purchasing tickets, or generating reports, this diagram helped in understanding the system's functionalities from the user's perspective and was important for guiding the development of user interfaces and business logic. For a visual representation, refer to Appendix H, Figure 4.

The Entity Relationship Diagram (Figure 5) modeled the data structure of the system, showing the relationships between different entities such as Members, Events, Tickets, and Booking (Appointments); it defined how data was organized and related with each other in the system which was vital for database design and ensuring data integrity, this diagram also helped in the logical design of the database and serve as a blueprint for implementing the database schema. For a visual representation, refer to Appendix H, Figure 5.

The Data Flow Diagram (Figure 6) details the flow of data within the system showed how data moves between processes, data stores, and external entities; it also helped in understanding the internal workings of the system, may it be identifying data inputs, processing steps, and outputs, this diagram was important for designing the system's processes and ensuring efficient data handling and processing. For a visual representation, refer to Appendix H, Figure 6.

By following this structured design approach and utilizing these diagrams, the development team ensured that the GYOCC web management system was well-organized, efficient, and met the needs of its users; these diagrams provided a clear guide for development, which

facilitated communication among individuals involved in the project and guiding the implementation process.

Development

The development of the Gensan Youth Orchestra and Choir Corporation (GYOCC) web management system was a meticulous process, adhering to industry standards and best practices to ensure the creation of a robust, scalable, and user-friendly application. The development team employed the Inverted Spiral model, which facilitated iterative enhancements while continuously addressing risks and incorporating user feedback. These narratives detailed the practices, tools, and techniques utilized during the development process.

The journey began with a strong emphasis on adhering to established industry standards. The team followed guidelines set by IEEE and ISO standards for software development, ensuring that all deliverables met high-quality benchmarks. The Inverted Spiral model provided the flexibility needed to make iterative improvements based on ongoing feedback and evaluation. Throughout the project, comprehensive documentation was maintained, ensuring transparency and traceability.

Moreover, a well-defined architectural design was crucial for the project, which was why the system was built using a multi-tier architecture, which separated the presentation layer, business logic layer, and data access layer, and this separation to enhance scalability and maintainability, allowing each component to be developed, tested, and maintained independently, using this modular approach ensured that changes in one part of the system did not adversely affect others.

The choice of technology stack played a vital role in the system's development; the front end was developed using HTML5, CSS3, and JavaScript, technologies known for creating responsive and interactive user interfaces. Django and Python were selected for their robustness and scalability on the backend. Django provided a secure framework for developing business logic and APIs, while PostgreSQL was chosen for the database due to its reliability and robust feature set. SSL/TLS protocols were implemented, and integration with secure payment gateways ensured safe online transactions to secure data transmission.

During development, the team placed a strong emphasis on code quality. Code reviews and static analysis tools were used to ensure the application of coding standards, minimize bugs, and enhance maintainability. GitHub was employed for version control, facilitating effective collaboration and enabling the team to seamlessly manage different codebase versions.

Testing and validation were integral to the development process. Each module underwent rigorous unit testing to verify individual functionality. Integration testing followed, ensuring that modules worked properly together. System testing was conducted in environments simulating real-world usage, verifying that the system met all functional and non-functional requirements. User Acceptance Testing (UAT) or Alpha Testing involved GYOCC members, instructors, and administrators, whose feedback was instrumental in refining the system to meet user expectations.

Documentation and training were also prioritized. Detailing was created for both technical and non-technical users, including system architecture diagrams, API documentation, user manuals, and troubleshooting guides. Training sessions were conducted for GYOCC's staff and users to ensure they could effectively use the system. These sessions covered system functionalities, best practices, and common troubleshooting steps.

Deployment and maintenance were the final stages of development. This was where the system was deployed using a cloud-based infrastructure, ensuring high availability and scalability. Automated deployment scripts streamlined the deployment process, minimizing downtime. Post-deployment, the system was continuously monitored for performance and security, and regular updates and patches were applied to address any issues and enhance functionalities.

In conclusion, by rigorously adhering to development standards and best practices, the GYOCC web management system was successfully developed as a reliable, efficient, and user-friendly platform. This system significantly enhances GYOCC's operational capabilities, positioning it as a leader in music education and performance.

RESULTS AND DISCUSSION

Testing

After thoroughly scrutinizing the system, the proponents decided to implement some of the features recommended by the panelists, which were then tested through alpha testing by the members of the Gensan Youth Orchestra and Choir Corporation. Members were asked to register accounts and explore the system and its features. Afterward, they were given a form to provide their feedback and evaluate the system's quality and functionality.

The feedback collected through various surveys, including the member questionnaire, staff & superuser questionnaire, instructor questionnaire, and organizer questionnaire, was valuable in enhancing and ensuring the system meets the organization's needs and improved operational efficiency and processed using performance testing. This comprehensive testing approach was crucial for achieving an efficient user experience and optimizing the system for broader deployment using the Likert scale, where respondents were given five (5) options to rate the system based on the given question per section; which options are these legends: Strongly Agree was five (5) at the interval of 4.21 to 5.00, Agree was four (4) at the interval of 3.41 to 4.20, Neutral was three (3) at the interval of 2.61 to 3.40, Disagree was at (2) at the interval of 1.81 – 2.60, and Strongly Disagree was at one (1) at the interval of 1.00 to 1.80.

section 1 of the UI survey in Table 1 in the Appendix, the overall feedback on the user interface (UI) was overwhelmingly positive, with an overall mean of 4.67, which is 93%, with most mean scores falling in the "Strongly Agree" range (4.47 to 4.88), indicating that users generally find the system easy to use and effective. The only exception was the question about viewing upcoming events in the calendar, which scored slightly lower at 4.18 and was interpreted as "Agree," suggesting that while this aspect was still positively rated, there may be room for improvement in displaying the integrated calendar in the proposed system.

As for the ease of navigation (both the dashboard and the audition section), displaying

information (like tickets, lessons, and history), and accessing materials received high ratings, with users mostly "Strongly Agreeing" that these features are functioning well.

In conclusion, users had a very favorable view of the UI, particularly in areas of navigation, displaying information, and accessing content, while criticisms of the calendar section may benefit from some enhancements to improve user satisfaction.

Based on Section 2 of table 2 on appendix H: Process evaluation results, the system demonstrated an exceptional performance with an overall mean of 4.72 out of 5 (94.4% satisfaction rate). The highest-rated aspects, scoring 4.88, were the ticket confirmation/generation system after purchase, joining a lesson satisfaction, and the systematic procedure for joining a music lesson session. The booking and purchasing functionalities received notably strong scores above 4.7, indicating highly efficient implementation. While all these aspects were rated "Strongly Agree," the automated audition confirmation and notification component, though still very positive at 4.35, presents an opportunity for minor potential enhancement. Overall, these metrics strongly validated the effectiveness of the proposed system, with users consistently expressing strong satisfaction across all ten evaluated criteria.

Table 3 of section 3 on Appendix H: conclusion gave exceptional results with an overall mean of 4.76 out of 5 (95.2% satisfaction rate). The ticket purchase experience emerged as the highest-rated aspect, with a score of 4.88, closely followed by both the audition appointment process and overall website experience, each scoring 4.82. The system's user-friendliness was solidified by the high score of 4.76 for both music lesson session accessibility and the likelihood of recommendation to others. While all aspects received "Strongly Agree" ratings, the ticket purchase option through the calendar was still very positive at 4.53, indicating a slight potential for improvement. The consistently high scores across all metrics and users' strong willingness to recommend the website prove the system's success in meeting user needs and expectations.

Description of Prototype

The prototype consists of two main components: The administrator panel and the user panel.

The Administrator Panel was designed to manage system operations. It includes a search bar, sidebar, and user dashboard. Administrators land on a login page with a simple form for entering their username and password. After logging in, they are taken to the dashboard, which contains all the administrator modules and their submodules. The dashboard showed recent actions made within the panel, and a sidebar allowed easy navigation across the site without returning to the main dashboard.

The Main menu in the sidebar included tools for managing various aspects of the system. Administrators handled attachments like images and videos for announcements, managed members by adding groups and assigning specific permissions, checked whether users had read notifications, and posted or edited announcements. They could also configure site settings, such as toggling SMS notifications, setting fees, enabling OTP for emails and SMS, and changing the domain URL.

The Message menu allowed administrators to view, search, write, reply to, and manage messages that were sent.

The Authentication and Authorization menu provided tools for managing user groups, adding or removing them, and adjusting permissions. It also included analytics displayed in charts, covering ticket sales, music lessons, popular instruments, yearly bookings, and preferred instructors. Administrators filtered this data by specific date ranges.

The Event menu lets administrators view booked events and their statuses. They could set event schedules, manage seating and costs, and start booking events to allow public event ticket purchases. A calendar view provided an overview of all events and bookings.

The Tickets menu enabled the administrators to track scanned tickets, check details of purchased tickets, and scan tickets during events using a camera.

The Lesson menu offered tools for managing available instruments, tracking attendance for lessons, and creating and scheduling lessons. Administrators and instructors could set lesson costs, locations, and maximum slots. They could edit their profiles, publish repertoire articles, view lesson schedules on a calendar, and scan QR codes to track attendance.

The Resource menu focused on inventory management. Administrators could view the history of borrowed items and update their condition upon return. They could also manage the availability of items like instruments and tools.

The Audition menu allowed administrators to view and manage submitted auditions. They could schedule in-person audition dates and notify users of their schedules.

The Payment menu provided an overview of payments made for auditions, bookings, events, lessons, and tickets.

The Feedback menu was where administrators could view user feedback submitted to instructors.

The Member 2FA menu tracks two-factor authentication codes sent through email or SMS and shows their statuses.

The Newsletter menu displayed a list of email addresses subscribed to the GYOCC newsletter.

The User Panel was designed to allow users to access the services offered by GYOCC. On the landing page, users were greeted with a login form, a navigation bar, promotional text and images, and a footer. From this page, they could log in, register, read FAQs, and subscribe to the newsletter.

The FAQ page provided answers to common questions about GYOCC, including appointments, tickets, joining GYOCC, music lessons, and other services.

The Signup page contained a form where users were asked to provide their personal information, including username, name, email, contact number, and password. The contact

details entered were later used for notifications and transaction confirmations.

After logging in, users were directed to the Dashboard, which included a sidebar for navigating GYOCC services, a recent activities section to track their transactions, an announcement wall for updates and articles, and a general event calendar. Clicking on an event in the calendar redirects users to the ticket purchase page.

The Book an Appointment page allowed users to request GYOCC's services for their events. The form pre-filled their personal details from registration, and they manually added details like event name, description, type, and schedule. After submission, users were redirected to a payment page, though they could also opt to pay later and access this through their history.

The Purchase a Ticket page enabled users to buy tickets for GYOCC events by selecting an event and a ticket type (e.g., VIP, Gold, Silver, or General Admission). After choosing, they proceeded to the payment page.

The Join GYOCC page provided options for auditioning for the orchestra or choir. Users filled out forms pre-filled with their registration details, with the option to upload a video entry for the choir.

The History page showed users a record of their transactions, such as tickets, bookings, lessons, auditions, and private event payments. Clicking on specific items provides additional details, such as ticket status or lesson information.

The Learning Materials page allowed users to access public resources by selecting an instrument from a dropdown menu. Relevant articles were then displayed for them to read.

The Your Calendar page visually represented the user's bookings and public events. Clicking on an event redirects them to the ticket purchase page.

The Notification page updated users on new events, posts, auditions, lessons, and other system activities.

Lastly, the My Profile page displayed personal information, which users could edit. They

could also change their password by selecting an authentication method (email or SMS) and entering the received code before creating a new password.

Implementation Plan

The system was to be deployed cooperatively and rightly planned in such a manner that it proved helpful for the organization to make a smooth shift. After the development of the system, incorporating every feature according to the specific needs of the company, the developers introduced data migration planning. This could be the decisive step in replacing the current hand-written documents of the organization on physical or traditional media with a more streamlined, easily accessible computerized system. The rollout to the sites was done a week before the final deployment to test the system properly and make all necessary changes.

To ensure the system was effectively adopted, the organization conducted comprehensive training sessions for its staff. These sessions focused on equipping employees with the necessary skills and knowledge to operate the system efficiently, fostering confidence, and reducing any potential disruptions during the transition.

The implementation began with the developers working closely with the organization to ensure all functionalities were ready and aligned with operational needs.

Training could be conducted in parallel, providing hands-on guidance to familiarize staff with the system's features and workflow. This collaborative effort ensured that employees were prepared to use the system effectively from day one.

Once the staff were trained and the system was ready, the deployment proceeded in a phased manner. This approach allowed for real-world testing and gradual integration into daily operations. During this period, the developers monitored the system closely, addressing any issues and making necessary adjustments to optimize performance.

Through this combination of thorough training and a carefully managed rollout, the system could be successfully implemented, ensuring both immediate usability and long-term operational benefits. The eventual implementation was gradual, beginning with a phased

rollout to test the workability functions. For this phase, the development team closely monitored the system, addressing any issues that may arise and committing other changes that could improve efficiency. In this deliberate approach, the system could reach full production status to transform the clerical data procedure of a company into a smooth digital workflow. It was in prioritizing collaboration, accuracy, and flexibility that the implementation strategy ensured change was effective but also in a position to prepare an organization for future expansion and improvement.

Conclusion

This study set out to develop and implement a web management system for the Gensan Youth Orchestra and Choir Corporation (GYOCC), addressing operational inefficiencies and enhancing user experience across various organizational functions. The comprehensive alpha testing phase involved members, staff, superusers, instructors, and organizers, ensuring a holistic evaluation of the system's features and usability.

The alpha testing phase gathered significant insights into the system's performance and user satisfaction. Participants across different demographics provided valuable feedback through structured questionnaires, which highlighted areas of success and opportunities for improvement. The findings indicated high levels of satisfaction with the system's ease of navigation, booking and appointment processes, ticket purchasing, audition applications, lesson enrollments, and overall user experience.

Most members found the system easy to navigate, with a user-friendly dashboard that effectively displays relevant information. This statement was based on the high percentage of positive responses regarding the dashboard's usability and content helpfulness. Additionally, the booking and ticketing modules were highly rated for their straightforward processes and effective automation, where members appreciated the efficient and smooth experience from booking appointments to purchasing tickets, confirming the system's ability to organize these operations efficiently. Overall, members reported a positive experience with the system, citing the ease of accessing learning materials, managing events, and viewing the history of transactions. The feedback highlighted the system's success in providing a cohesive and satisfying user experience, which aligned with GYOCC's operational goals.

The successful implementation of this system was expected to significantly improve GYOCC's operational efficiency, member engagement, and overall service within the organization, where this proposed study set a benchmark for similar organizations in this sector.

REFERENCES

- Akinyede, R., Balogun, T., & Iwasokun, G. (2017). Design and implementation of an online booking system for a cinema house. International Journal of Artificial Intelligence Research, Abstract.
- Akshay, V., Kumar, A., Alagappan, M., & Gnanavel, S. (2019). BOOKAZOR - an Online Appointment Booking System. <https://doi.org/10.1109/ViTECoN.2019.8899460>, Abstract.
- Ampuan, A., & Deleña, R. (2024). A Quantitative Evaluation of Online Appointment System at Mindanao State University–Main Campus: Employing the System Usability Scale (SUS) and Technology Acceptance Model (TAM). <https://doi.org/10.1109/ICDXA61007.2024.10470770>, Abstract.
- Archit, R., Vinit, S., & Rajesh, K. (2019). <https://www.indianjournals.com/ijor.aspx?target=ijor:rjet&volume=10&issue=1&article=004>. <http://dx.doi.org/10.5958/2321-581X.2019.00004.7>, Abstract.
- Fuente, M., Facunla, J., Jacinto, E., J, H., Olipas, C., & Cunanan, A. (2023). Project Clinik: A Cross-Platform Scheduling and Appointment Reservation System. <https://doi.org/10.55927/fjcis.v2i1.4077>, Abstract.
- Generoso, E., Illagan, R., Evangelista, J., Mendoza, J., & Pulhin, J. (2019). Resort Online Reservation: Basis for Action Plan. Asia Pacific Journal of Education, Arts and Sciences, 30.

Guan, Y., Wu, B., & Jia, J. (2020). Does online ticket booking system make people better off? An empirical study on railway service. <https://doi.org/10.1016/j.trf.2020.03.014>, Abstract.

Holbrook, C. J., Lukins, B., Berzins, P., Hudson, V., & Smith, N. (2023). Development of Tickit, A Ticket Reservation System.

Kazi, S., Bagasrawala, M., Shaikh, F., & Sayyed, A. (2018). Smart E-Ticketing System for Public Transport Bus. <https://doi.org/10.1109/ICSCET.2018.8537302>, Abstract.

Morris, L., & McDermott, L. (2021). Enhancing academic skills appointments through a new booking system. Journal of Learning Development in Higher Education, Abstract.

Sutabri, T., Wijaya, A., Seprina, I., & Amalia, R. (2023). Ticket Reservation System Design with Web-Based. International Journal of Artificial Intelligence Research, Abstract.

Westgate, C. (2019). Popular Music Fans and the Value of Concert Tickets.
<https://doi.org/10.1080/03007766.2019.1601152>, Abstract.

Gensan Youth Orchestra and Choir Corporation. (2024, August 13). GenSan Youth Choir shines at the 13th BICF. Retrieved from <https://gyocc.org/gensan-youth-choir-shines-at-the-13th-bicf/>

Dennis, A., Wixom, B.H., & Tegarden, D. (2015). Systems Analysis & Design: An Object-Oriented Approach with UML (5th ed.). Hoboken: John Wiley & Sons, Inc.

Epstein, D., & Maltzman, R. (2014). Project Workflow Management: A Business Process Approach. Retrieved from https://books.google.com.ph/books?id=_ePJCgAAQBAJ

Nailing Your Software Requirements Documentation (2018). In Lucidchart. Retrieved from <https://www.lucidchart.com/blog/software-requirements-documentation>

O'Regan, G. (2017). Concise Guide to Software Engineering: From Fundamentals to Application Methods. Cham, Switzerland: Springer.

Seidl, M., Scholz, M., Huemer, C., & Kappel, G. (2015). UML @ Classroom: An Introduction to Object-Oriented Modeling. Cham, Switzerland: Springer International Publishing.

Use-case Diagrams Reference (n.d.). In UML Diagrams. Retrieved from
<https://www.uml-diagrams.org/use-case-reference.html>

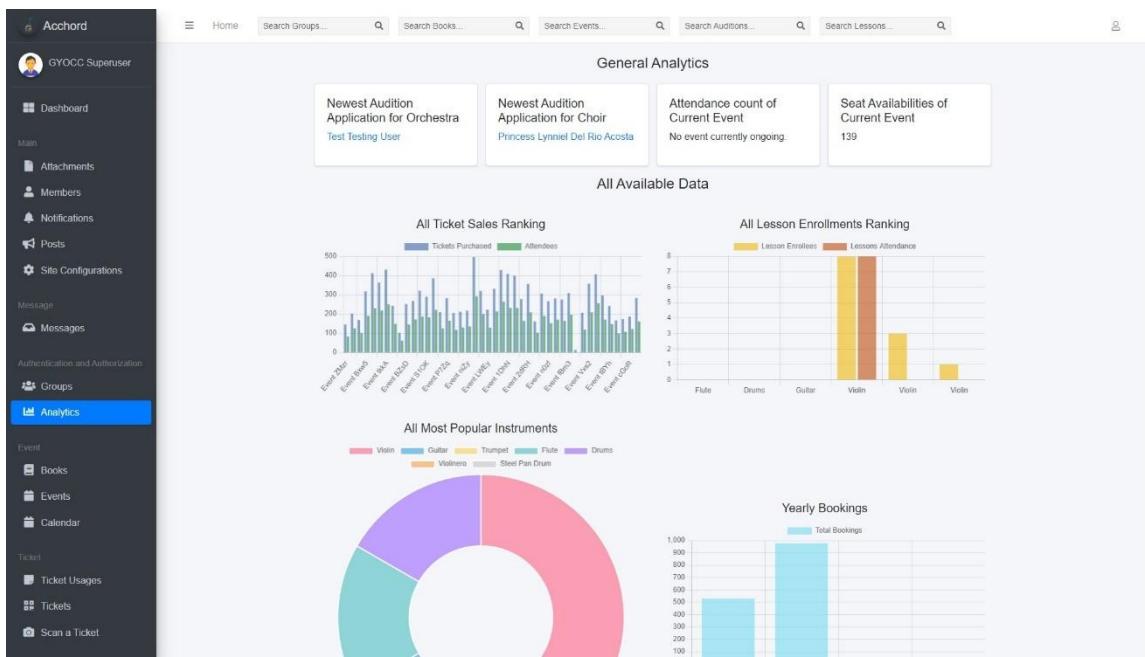
APPENDICES

APPENDIX A. RESOURCE PERSONS

The following person or personalities had a great contribution to the proponent's study:

- Gensan Youth Orchestra & Choir Corporation - the host company of the proponent is located at Ground Floor, Jumabo Building, 23 Kudalasan, General Santos City, 9500 South Cotabato.
- Mr. Fernan Globen Talonding – the Chief Executive Officer of the Gensan Youth Orchestra & Choir Corporation and was one of the proponent's main sources of information in developing the proposed system.
- Ma'am Allyssa Kaye A. Parcon – the proponent's adviser who rendered her expertise and insights for the development of the proposed study.
- Ma'am Kristel Joy C. Tulagan, MIT – as one of the proponent's panel members during the final defense of Capstone 1.
- Sir Darwin, Berja – a former faculty member of STI College General Santos who shared his expertise and insights in the early stages of development of the proposed study.
- Sir Dan Henley Sales - as one of the proponent's panel members during the final defense of Capstone 2.
- Sir Nur Ali Padasan, MIT - as one of the proponent's panel members during the final defense of Capstone 1.
- Dr. Ann Gilyn B. Premarion - as one of the proponent's panel members during the final defense of Capstone 1 and 2.
- Sir Anthony Mark F. Silong, MIT - as one of the proponent's panel members during the final defense of Capstone 1 and 2.
- Ma'am Ivy Grace C. Laurente – as the capstone project coordinator.

APPENDIX B. SAMPLE INPUT/OUTPUT/REPORTS



Lesson Information

Instructor

GYOCC Superuser

Lesson Information

Lesson Instrument: Violin

Lesson Status: Finished

Lesson Start: Oct. 12, 2024, 2 p.m.

Lesson End: Oct. 12, 2024, 5 p.m.

Personal Details

First Name: GYOCC

Last Name: Superuser

Phone Number: +639081600631

Email: calvindraktesezt9@gmail.com

QR Attendance



Make sure to have your QR scanned by staff to be listed in the attendance.

Ticket Information

Ticket Number

12210

Event Information

Event Name: Event SMCmBTq2

Ticket Seat: GA

Ticket Status: AVAILABLE

Ticket Used on: None

Personal Details

First Name: Dorothy

Last Name: Dunlap

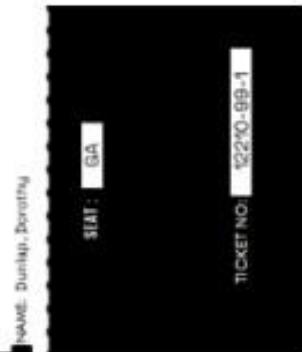
Phone Number: 09803142444

Email: garciajeffrey@example.org

QR Image



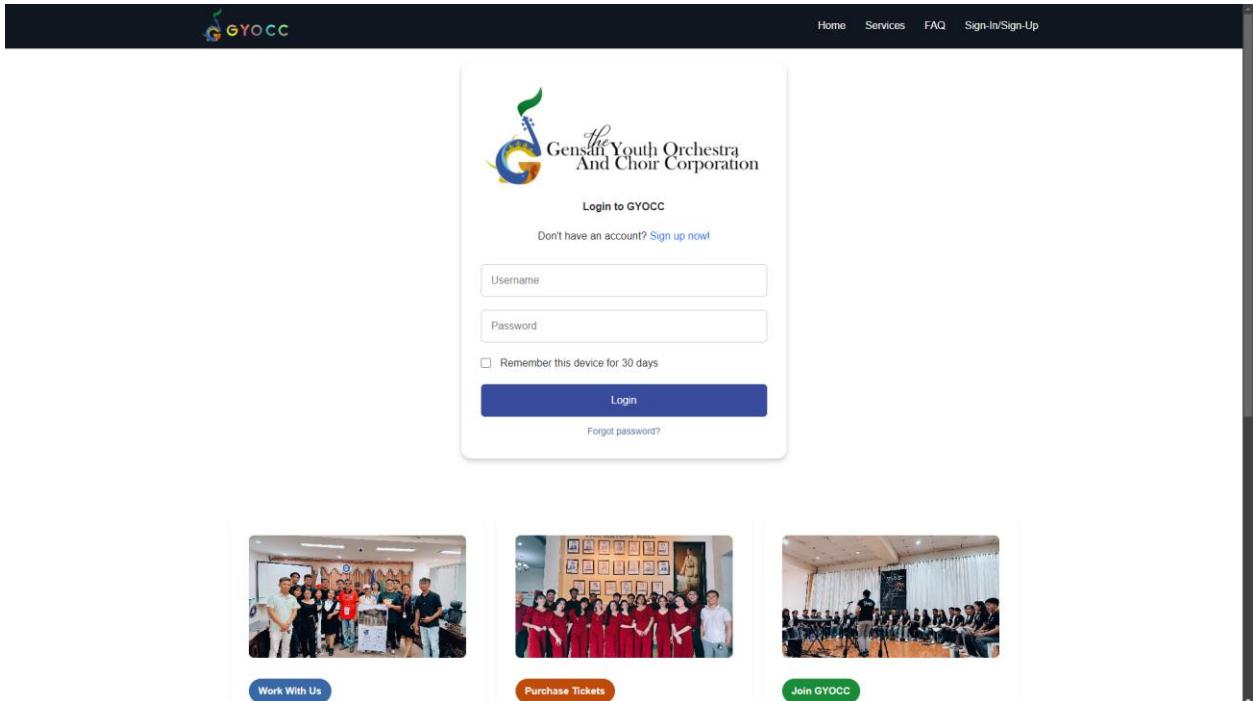
DATE: December 28, 2024, 06:00 PM to January 05, 2025, 06:00 PM
VENUE: Location Iko, SHSE



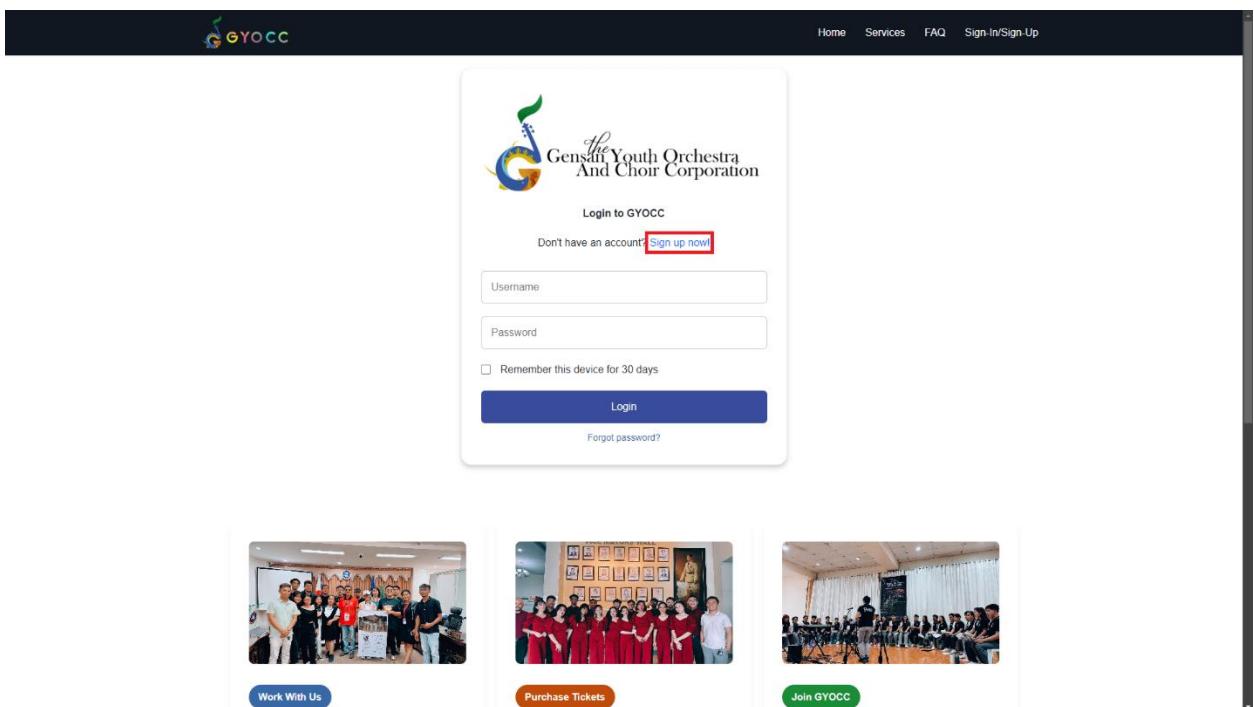
APPENDIX C. USER'S GUIDE

Member Manual

1. Account Registration
 - Home Page



- Click the “Sign up now!” link to go to the registration page.



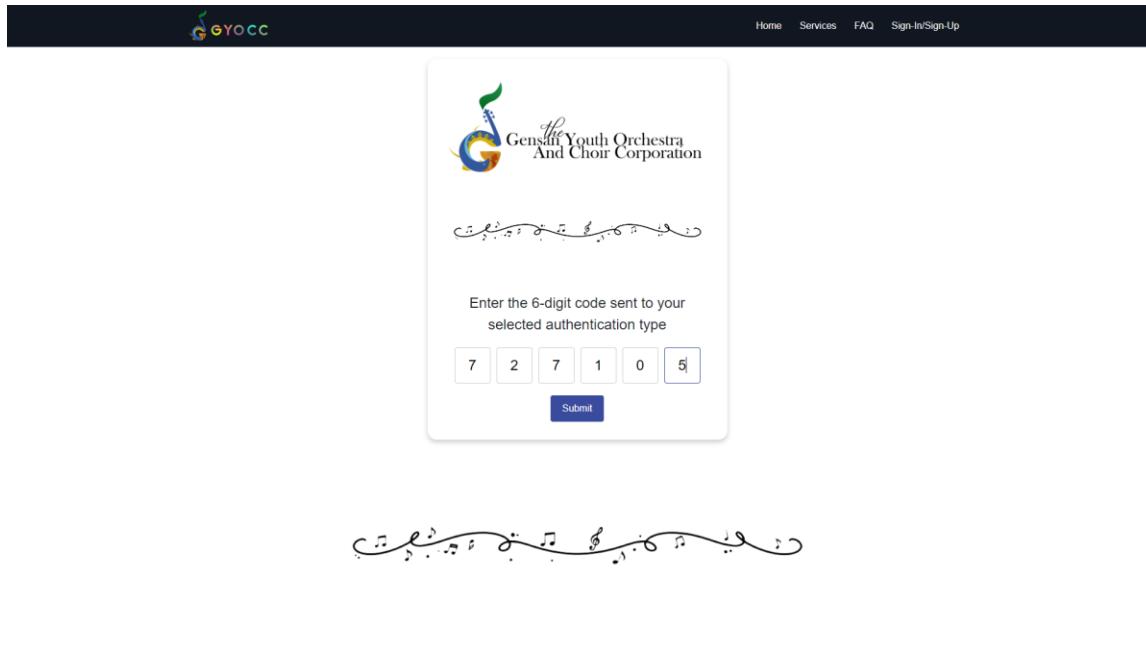
- Filled up registration details

The screenshot shows the registration form for GYOCC. It includes fields for first name ('testuser25'), last name ('Test'), middle name ('User'), email ('testuser25@gmail.com'), phone number ('+63 9512498751'), and password ('*****'). There is also a checkbox for accepting the Terms of Service and a large blue 'Register' button.

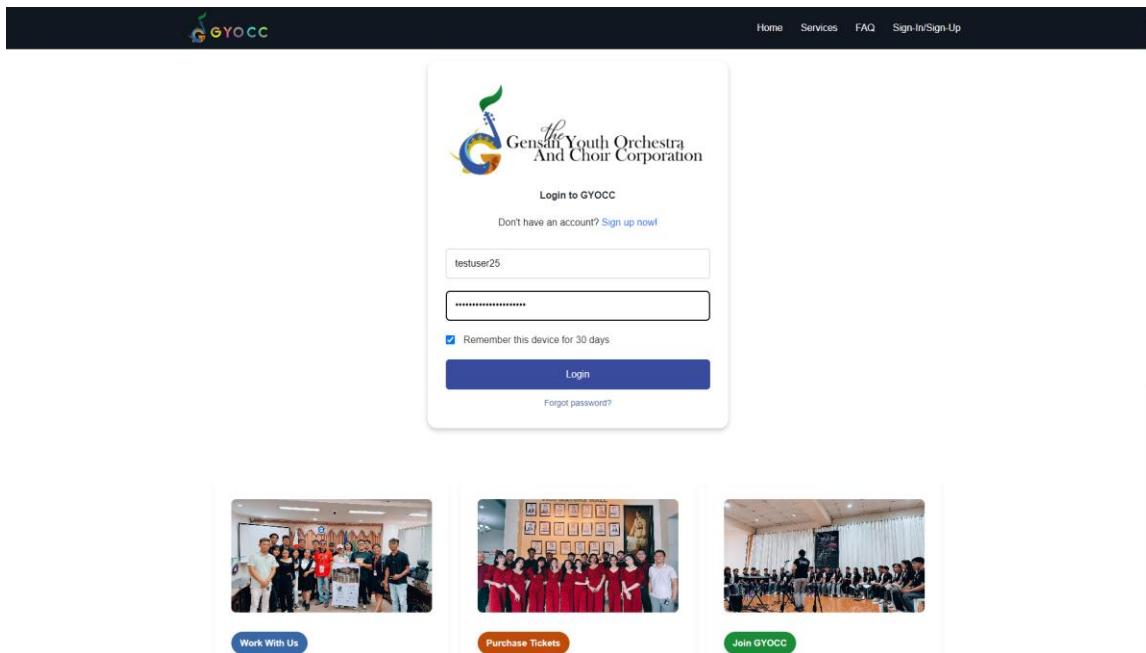
- Select an authentication method to register the account.

The screenshot shows the homepage of GYOCC after registration. It features a 'Select Authentication Method' section with a blue 'Authenticate via Email' button. Below this are three main sections: 'Work With Us' (showing a group photo and a 'Work With Us' button), 'Purchase Tickets' (showing a group photo and a 'Purchase Tickets' button), and 'Join GYOCC' (showing a group photo and a 'Join GYOCC' button).

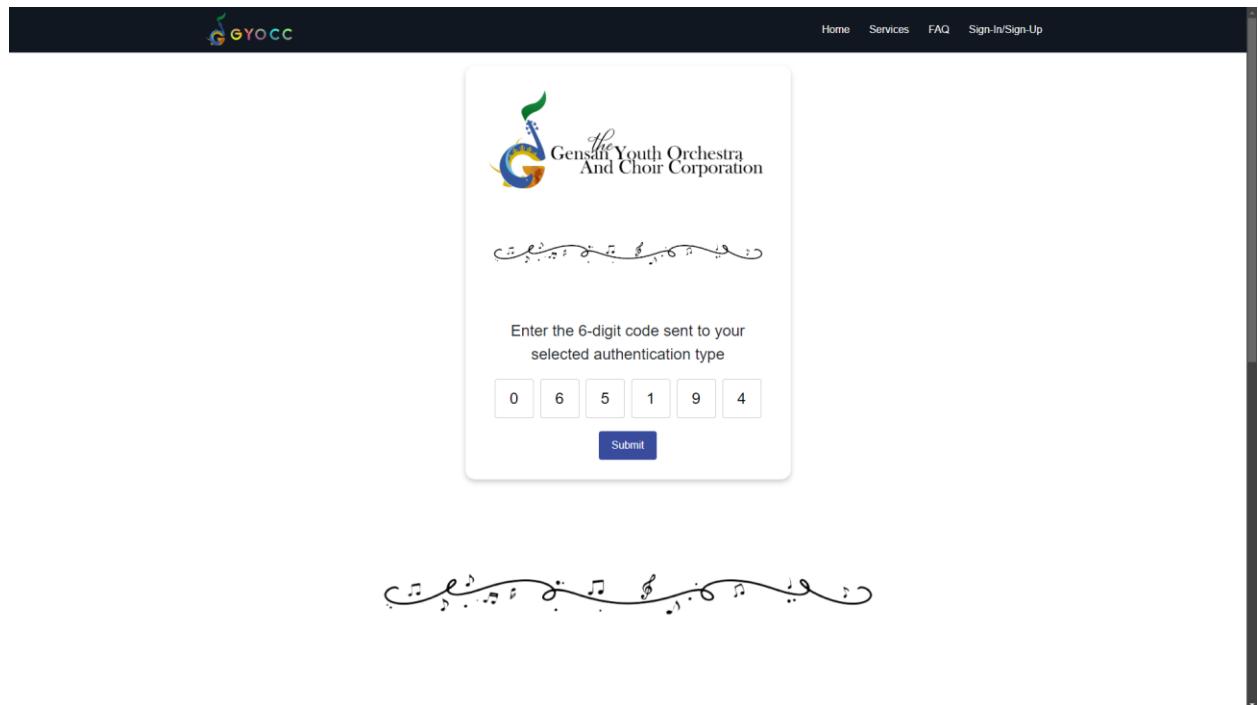
- To complete your registration, enter the 2FA code sent to your chosen authentication method.



- After completing your registration, you are automatically redirected to the login page. Sign in using your newly registered account.



- To login, enter the 2FA code sent to your chosen authentication method.



- After logging in, you were redirected to the dashboard.

- ## 2. Booking an appointment
- First, you had to log in into your account and proceed to the dashboard.

The screenshot shows the GYOCC (General Santos Youth Chorus) dashboard. On the left, a sidebar titled 'CONTENTS' includes links for Services, Join GYOCC, History, Learning Materials, Your Calendar, and Notifications. The main area features a 'Welcome Test User!' message and four status boxes: 'No Recent Purchase' (Recently Purchased Ticket), 'No Recent Booking' (Recently Booked Appointment), 'No Recent Lesson' (Recent Lesson), and 'No Recent Audition' (Recent Audition). Below these is an 'Announcement Wall' section with a post from 'GYOCC Superuser' about the GenSan Youth Choir's victory at the 13th BICF. To the right is a 'General Event Calendar' for October 2024, showing various events like 'World Chorale Day' and 'Kibabag Day'.

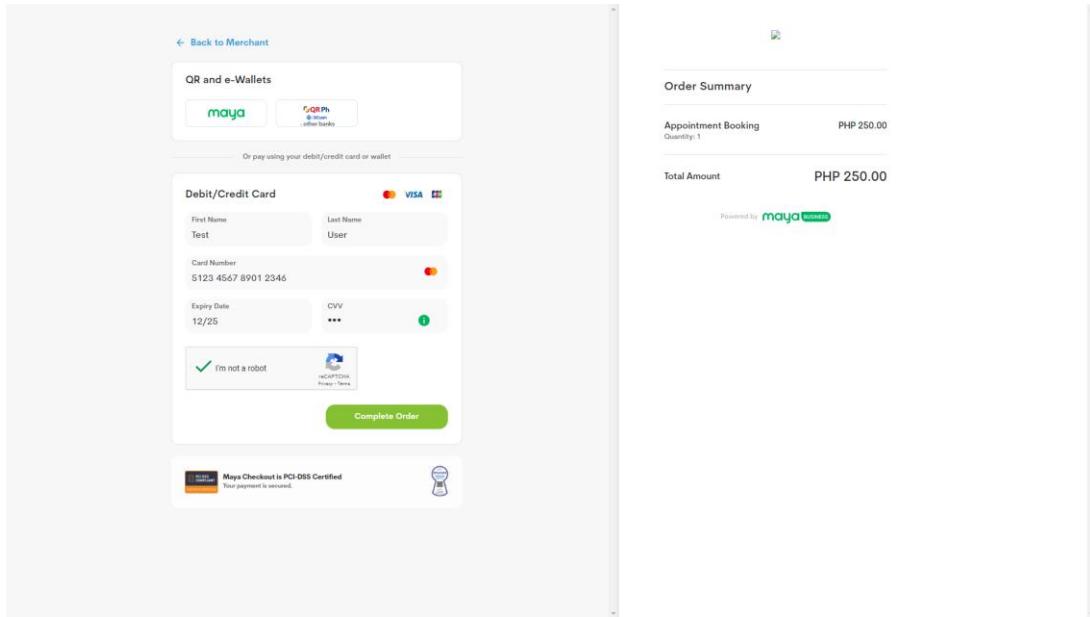
- To book an appointment, go to the sidebar, select 'Services,' and choose 'Book an Appointment.'

This screenshot is similar to the previous one but highlights the 'Book an Appointment' link in the sidebar under the 'Services' dropdown. The rest of the interface remains the same, showing the announcement wall and event calendar.

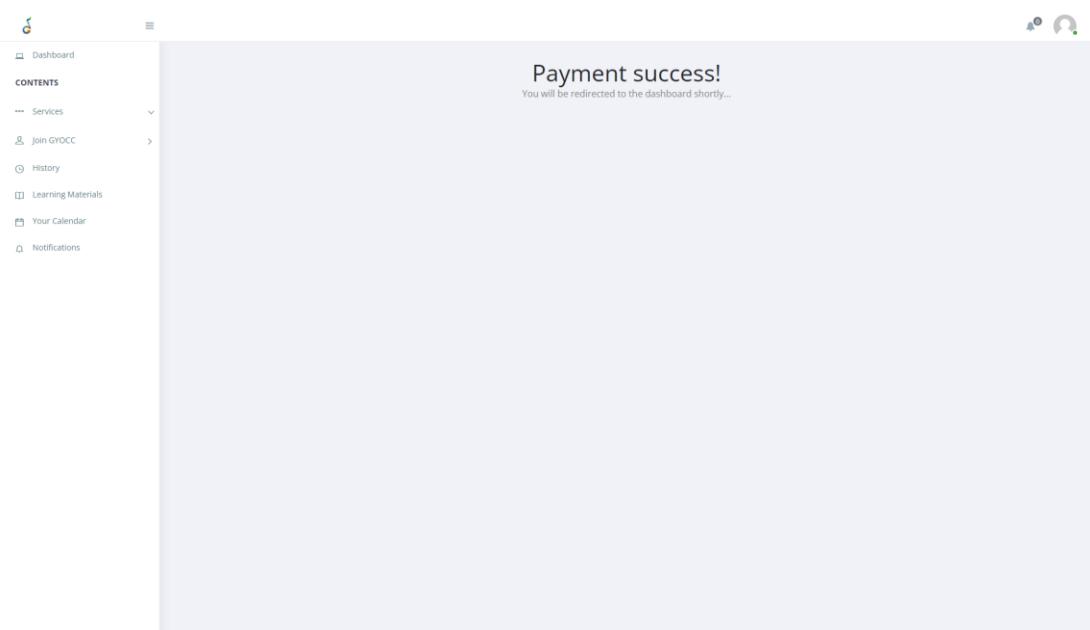
- Enter the necessary details for your event to proceed with the booking then click submit.

- After clicking submit, you were prompted to pay now or pay later.

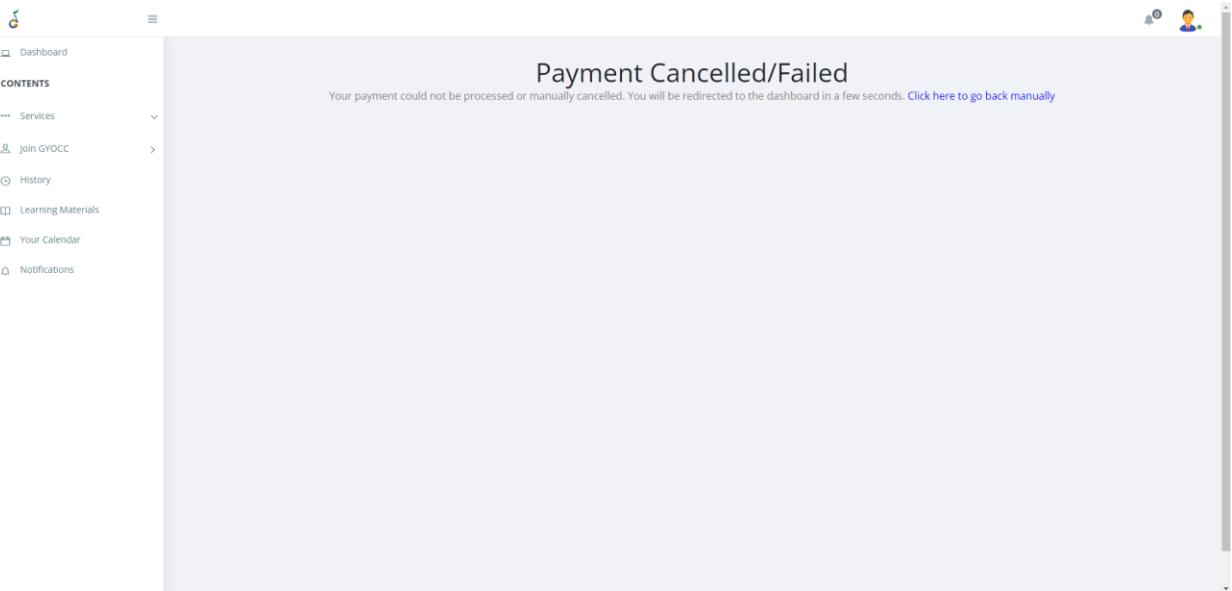
- By clicking “Pay Now,” you were redirected to the payment page. You had to enter your payment details on this page to complete the transaction.



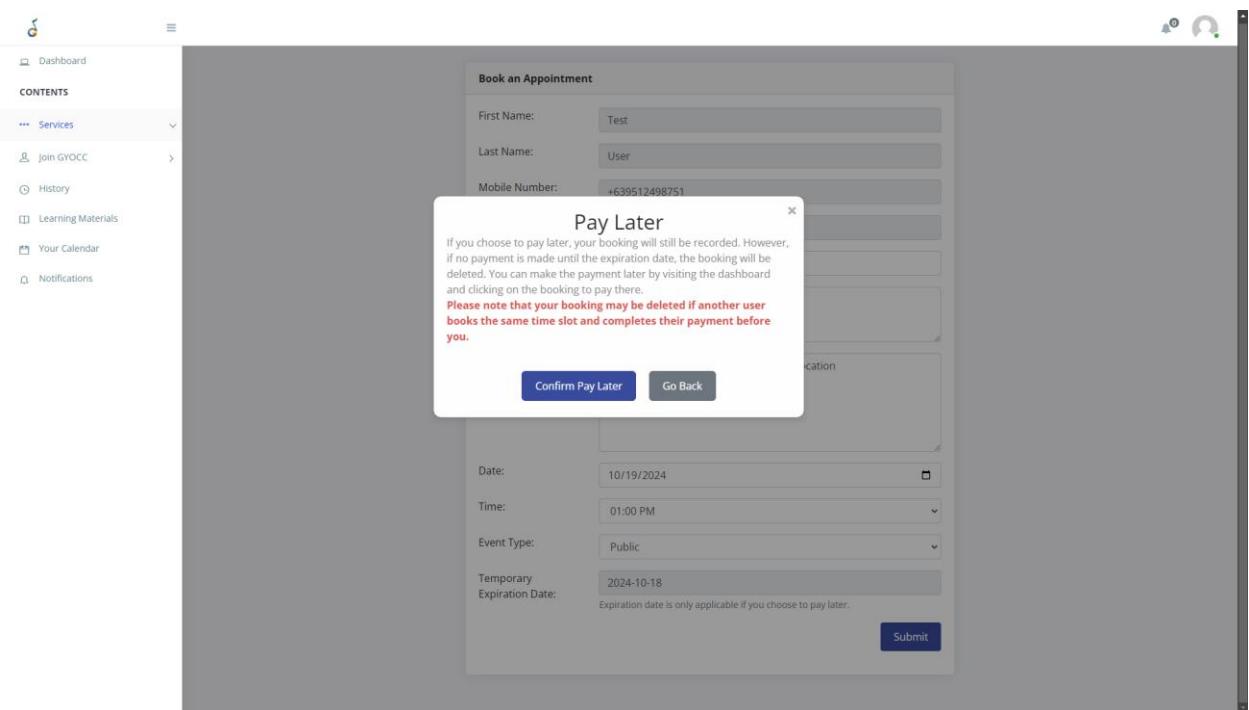
- Upon completing the transaction, you were notified if the payment succeeded or failed.
- Payment Success



- Payment Failed



- Clicking pay later could record your planned appointment and could still be paid later. However, the booking could be deleted if no payment was made on the expiration date or if another user booked the same time slot and completed their payment before you.



- ### 3. Purchasing ticket
- First, you had to log in to your account and proceed to the dashboard.

The screenshot shows the dashboard of a software application. On the left, there is a sidebar with a 'Dashboard' button and a 'CONTENTS' section containing links to 'Services' (with 'Purchase a Ticket' highlighted), 'Join GYOC', 'History', 'Learning Materials', 'Your Calendar', and 'Notifications'. The main area features a 'Welcome Test User!' message. Below it is an 'Announcement Wall' section with a post from 'GYOCC Superuser' dated Aug. 17, 2024, at 9:58 a.m., titled 'GenSan Youth Choir Shines at the 13th BICF'. The post includes a photo of a group of people on stage at the BICF, holding a trophy and a banner. To the right is a 'General Event Calendar' for October 2024, showing various events like 'World Chorale Day' and 'Event Vxs2WfJW' with specific times like 8:55a and 7:18p.

- To purchase a ticket, go to the sidebar, select 'Services,' and choose 'Purchase a Ticket'.

This screenshot is similar to the one above, but the 'Purchase a Ticket' link in the sidebar's 'Services' section is highlighted with a red box. The rest of the interface is identical, showing the announcement wall and the general event calendar.

- The fields were automatically filled with your name, contact number, and email. You could modify these details if you wanted to buy a ticket for someone else. Select the event you'd like to attend and choose the ticket type on this page, then click submit.

Purchase a Ticket

First Name: Test

Last Name: User

Mobile Number: +63 9512498751

Email: testuser25@gmail.com

Event: World Chorale Day

Select Ticket: VIP

If there are no ticket types available it means the tickets for that event is sold out.

Submit

- Upon clicking the submit button, you were redirected to the payment page. On this page, you had to enter your payment details to complete the transaction.

← Back to Merchant

QR and e-Wallets

maya GGR Ph other banks

Or pay using your debit/credit card or wallet

Debit/Credit Card

First Name Test	Last Name User
Card Number 4123 4501 3100 1522	VISA
Expiry Date 12/25	CVV ***
<input checked="" type="checkbox"/> I'm not a robot	

Complete Order

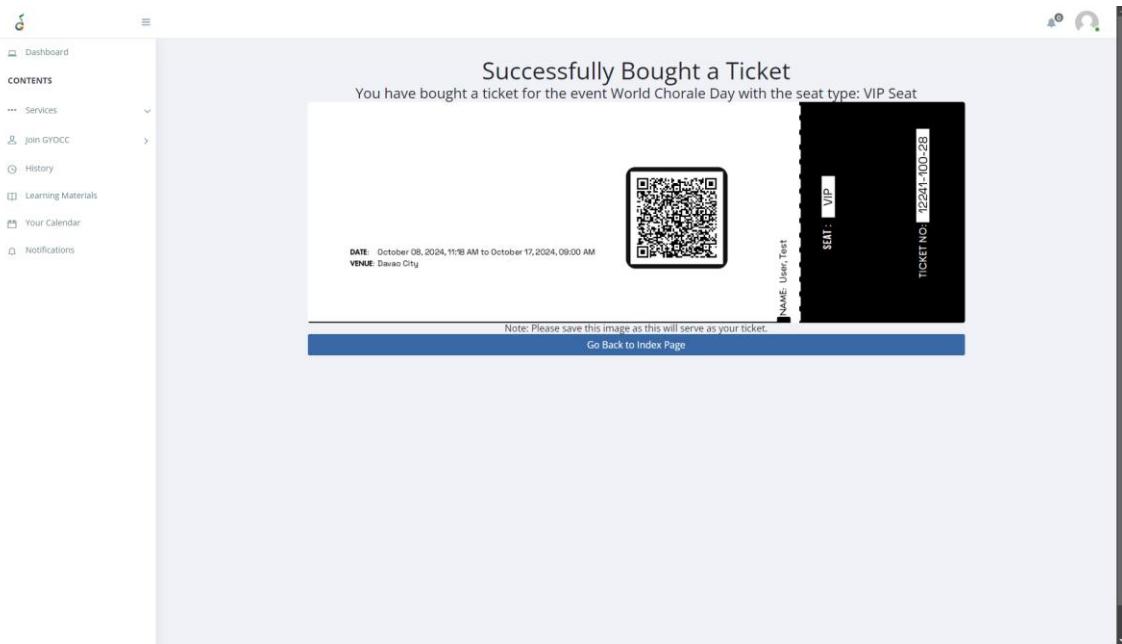
Order Summary

VIP Seat Ticket for World Ch...	PHP 1,000,000.00
Quantity: 1	
Total Amount	PHP 1,000,000.00

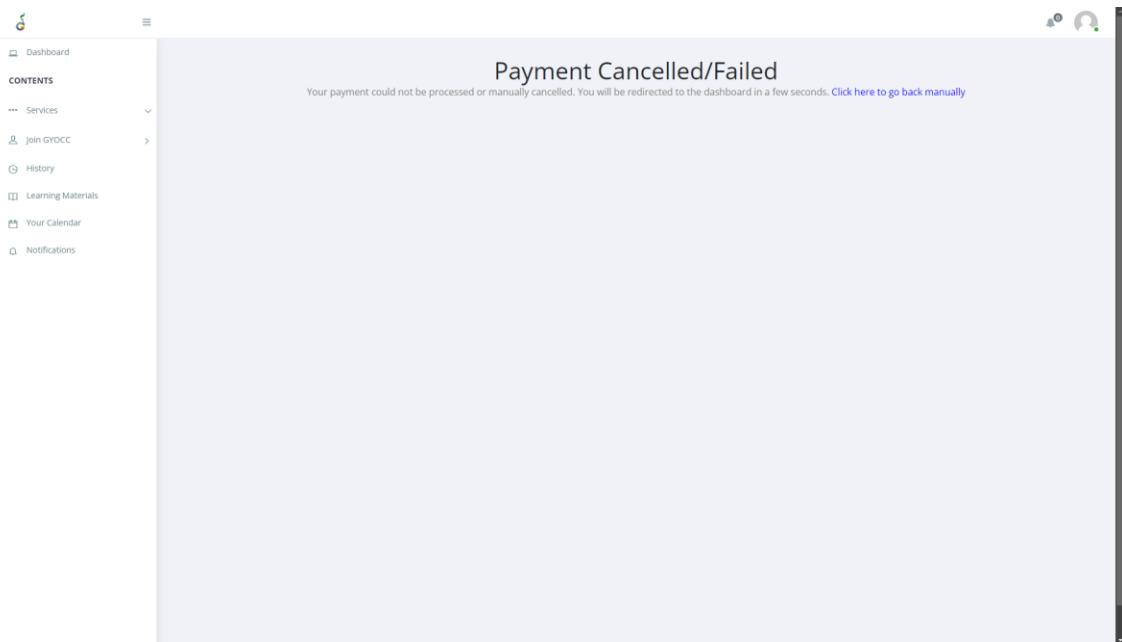
Powered by maya

Maya Checkout is PCI-DSS Certified Your payment is secured.

- Upon completing the transaction, you were notified if the payment succeeded or failed.
 - Payment Success – A ticket generated for you, which you used to enter the event venue.



- Payment Fail – You were notified that your payment had failed and redirected to the dashboard.



4. Submitting an Audition

- First, you had to log in to your account and proceed to the dashboard.

The screenshot shows the GYOCC (Global Youth Choral Competition) dashboard. On the left, a sidebar titled 'CONTENTS' includes links for Services, Join GYOCC, Audition, Enroll in a Music Lesson, History, Learning Materials, Your Calendar, and Notifications. The main area features a 'Welcome Test User!' message. Below it is the 'Announcement Wall' which displays a post from 'GYOCC Superuser' (Superuser) dated Aug. 17, 2024, at 9:58 a.m., with the title 'GenSan Youth Choir Shines at the 13th BICF'. The post includes a photo of a group of people on stage at the BICF, holding a trophy and a banner. To the right is the 'General Event Calendar' for October 2024, showing various events like '7:18p World Chorale Day' and '12:45p Event K80M0gMR'.

- To submit an audition, go to the sidebar, select 'Join GYOCC,' and choose 'Audition'.

Welcome Test User!

Announcement Wall

GYOCC Superuser
Superuser
Aug. 17, 2024, 9:58 a.m.
[VIEW POST](#)

Title: GenSan Youth Choir Shines at the 13th BICF

General Event Calendar

October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W40 29	30	1	2	3	4	5
8:55a Event Vxs2WfdW						
W41 6	7	8	9	10	11	12
8:55a Event Vxs2WfdW						7:18p World Chorale Day
W42 13	14	15	16	17	18	19
7:18p World Chorale Day						
W43 20	21	22	23	24	25	26
						12:43p Event KRBmBdgM6
W44 27	28	29	30	31	1	2
						12: Event I
W45 3	4	5	6	7	8	9

GenSan Youth Choir Shines at the 13th BICF

The GenSan Youth Choir from General Santos City, Philippines, achieved a remarkable victory at the 13th Bali

- Upon clicking the audition page, you were redirected to a selection page where you could choose whether you want to join the Orchestra or Choir.

<https://gyocc.online/audition/orchestra/>

Audition for our Orchestra

Orchestra

Audition for our Choir

Choir

- Auditioning for an Orchestra required you to enter only your address; the rest were pre-filled with the information you entered during registration .

Audition for an Orchestra

First Name: Test

Last Name: User

Mobile Number: +639512498751

Email: testuser25@gmail.com

Audition Type: Orchestra

Address:

Audition Expiration: 2024-10-21 23:05:57

Submit

- Auditioning for the choir also only requires your address; however, you could also submit a video audition by uploading your entry.

Audition for Choir

First Name: Test

Last Name: User

Mobile Number: +639512498751

Email: testuser25@gmail.com

Audition Type: Choir

Address:

Video Entry: Choose File No file chosen
Accepted video formats: .mp4 | .avi | .mov
You can attach a video file here showcasing your entry point to your audition, there is also a slight chance that you won't need to do a live audition and get accepted only with the video you attached.

Audition Expiration: 2024-10-21 23:06:52

Submit

5. Enrolling for a Music Lesson

- First, you have to log in into your account and proceed to the dashboard .

The screenshot shows the GYOCC dashboard. On the left, there's a sidebar with 'Dashboard' selected and a 'CONTENTS' section containing links like 'Services', 'Join GYOCC', 'History', 'Learning Materials', 'Your Calendar', and 'Notifications'. The main area features a 'Welcome Test User!' message. Below it is an 'Announcement Wall' with a post from 'GYOCC Superuser' about the GenSan Youth Choir's victory at the 13th BICF. To the right is a 'General Event Calendar' for October 2024, showing various events like 'World Chorale Day' and 'Test Event'.

- To enroll in a music lesson, go to the sidebar, select 'Join GYOCC,' and choose 'Enroll in a Music Lesson'.

This screenshot is similar to the one above, but the 'Join GYOCC' link in the sidebar is highlighted with a red box. The rest of the interface, including the announcement wall and event calendar, remains the same.

- Upon clicking the ‘Enroll in a Music Lesson’ button, you were redirected to the page where you could select an instrument, instructor, and schedule.

Join a Lesson

Instrument:

Select Instructor: Select an instrument first
If there are no instructors available it means that **there is currently no lessons for that instrument.**

Select Schedule: Select an instructor first
Lesson Cost: ...
If there are no schedules available it means that **you have already joined it or there are no more slots.**

Submit

- Select your desired instrument, and then available instructors and their available schedules appear. Then click submit.

Join a Lesson

Instrument: Violin

Select Instructor: Calvin Drakke Improgo Rulete
If there are no instructors available it means that **there is currently no lessons for that instrument.**

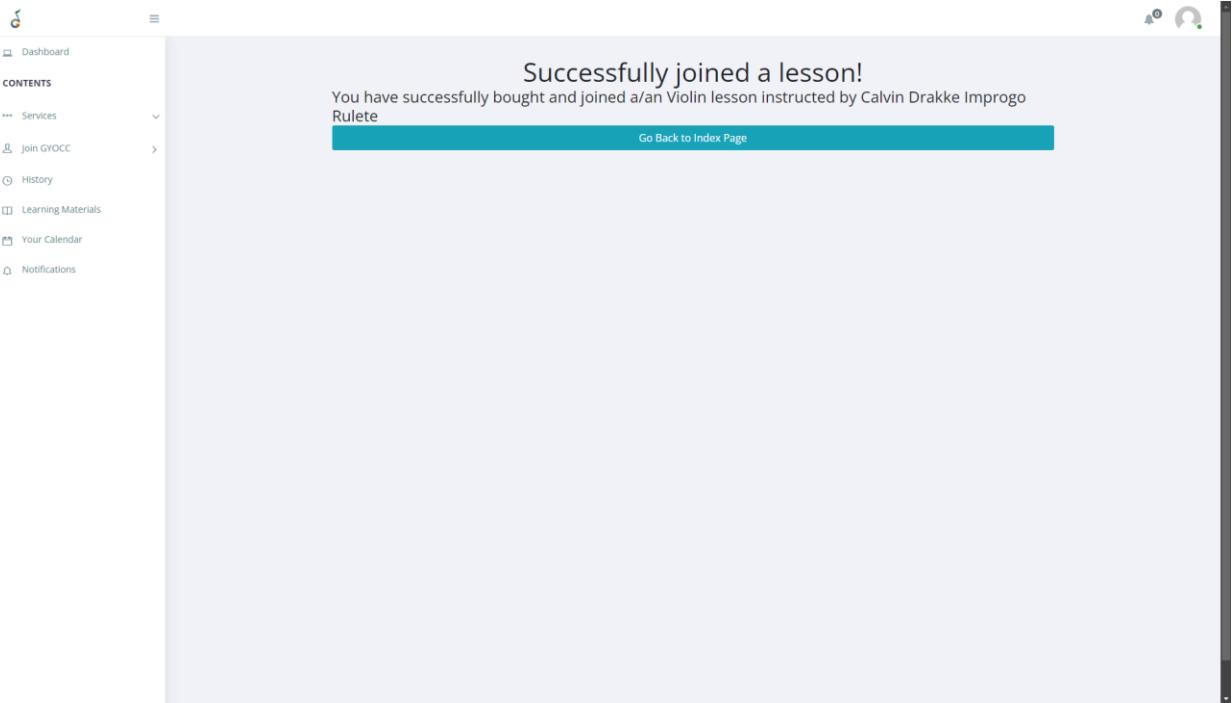
Select Schedule: Oct 13, 2024, 11:55 a.m. - Oct 15, 2024, 11:55 a.m.
Lesson Cost: 600.00 PHP
If there are no schedules available it means that **you have already joined it or there are no more slots.**

Submit

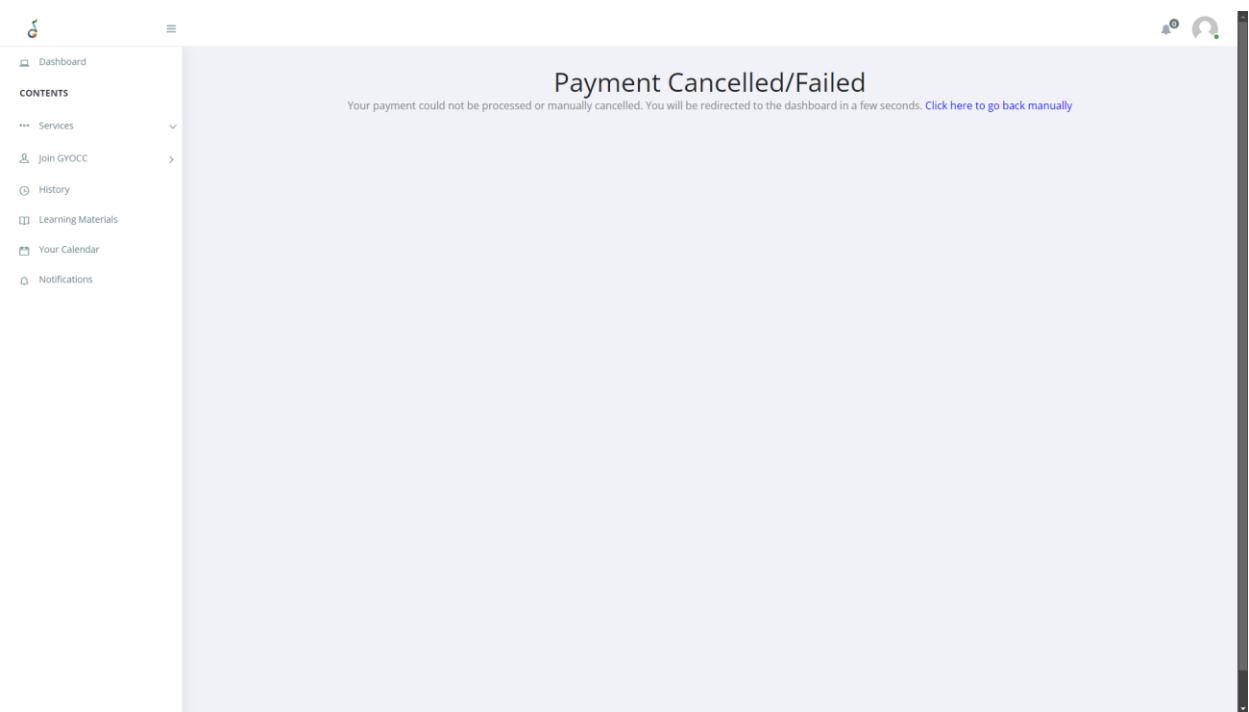
- Fill in your payment details to complete the transaction.

The screenshot shows a payment page for a merchant named "Calvin Drakke Impregro Rulet...". The total amount is PHP 600.00. The payment method selected is a Debit/Credit Card. The card details entered are: First Name: Test, Last Name: User, Card Number: 4123 4501 3100 1381, Expiry Date: 12/25, and CVV: 321. A CAPTCHA verification "I'm not a robot" is checked. The payment is completed successfully, indicated by a green "Complete Order" button and a message: "Maya Checkout is PCI-DSS Certified. Your payment is secured." The page is powered by "maya BUSINESS".

- Upon completing the transaction, you were notified if the payment succeeded or failed.
- Payment Success



- Payment Failed



6. Viewing Transaction History

- First, you have to log in to your account and proceed into the dashboard

The screenshot shows the GYOCC Online Dashboard. On the left, a sidebar lists 'Dashboard', 'CONTENTS' (Services, Join GYOCC, History, Learning Materials, Your Calendar, Notifications), and a 'History' button which is highlighted. The main content area features a 'Welcome Test User!' message. Below it is the 'Announcement Wall' section, which displays a post from 'GYOCC Superuser' about the GenSan Youth Choir's victory at the 13th BICF. The post includes a photo of the choir members on stage with a trophy. To the right is the 'General Event Calendar' for October 2024, showing various events like 'World Chorale Day' and 'Violin Recent Lesson'.

- To view your transaction history, go to the sidebar, then select History.

This screenshot is identical to the one above, but the 'History' button in the sidebar is explicitly highlighted with a red box, emphasizing the step to take to view transaction history.

- Upon clicking the History button, you were redirected to the page where you could view your transaction history.

Ticket History
Click a Ticket to see more information about it.

NAME OF EVENT	SEAT TYPE	DATE USED	STATUS
World Chorale Day	VIP	Unused	AVAILABLE

Booking History
Click a non-paid booking to start a payment for it.
Only bookings that have not passed can initiate payment.

BOOK NAME	SCHEDULE STATUS	MEETING DATE	PAYMENT STATUS
Test Event	Upcoming	Oct. 19, 2024, 1 P.M.	Paid

Lesson History
Click a lesson row to see more details about it.

INSTRUCTOR	INSTRUMENT	START DATE	END DATE	STATUS
Calvin Drakke Improgo Rulete	Violin	Oct. 13, 2024, 11:55 A.M.	Oct. 15, 2024, 11:55 A.M.	ONGOING

Audition History
Click an audition to see more information about it.

AUDITION TYPE	AUDITION DATE	AUDITION STATUS
Orchestra	To Be Set By GYOC	PENDING

- Clicking on an item could give you more detailed information about the transaction.
 - Tickets

Ticket Information

Ticket Number
12241

Event Information
Event Name: World Chorale Day
Ticket Seat: VIP
Ticket Status: AVAILABLE
Ticket Used on: None

Personal Details
First Name: Test
Last Name: User
Phone Number: +639512498751
Email: testuser25@gmail.com

QR Image

The QR image shows a digital representation of a ticket stub with the following text:
GYOCO General Santos City Branch
Ticket No. 12241
Event: World Chorale Day
Seat: VIP
Date: Oct. 19, 2024, 1 P.M.
Status: Available

Lesson

The screenshot shows the GYOCC platform interface. On the left is a sidebar with a navigation menu:

- Dashboard
- CONTENTS**
 - Services
 - Join GYOCC
 - History
 - Learning Materials
 - Your Calendar
 - Notifications

The main content area has a dark green header bar with the title "Lesson Information". Below it, there are two sections separated by horizontal lines:

- Instructor**

Calvin Drakke Rulete
- Lesson Information**

Lesson Instrument: Violin
Lesson Status: Ongoing
Lesson Start: Oct. 13, 2024, 11:55 a.m.
Lesson End: Oct. 15, 2024, 11:55 a.m.

Below these sections is another section titled "Personal Details" with the following information:

First Name: Test
Last Name: User
Phone Number: +639512498751
Email: testuser25@gmail.com

At the bottom of the main content area is a section titled "QR Attendance" containing two QR codes. A small note below the QR codes reads: "Make sure to have your QR scanned by staff to be listed in the attendance."

○ Audition

The screenshot shows the GYOCC platform interface. On the left is a sidebar with a navigation menu:

- Dashboard
- CONTENTS**
 - Services
 - Join GYOCC
 - History
 - Learning Materials
 - Your Calendar
 - Notifications

The main content area has a yellow header bar with the title "Audition Information". Below it, there is one section titled "Personal Details" with the following information:

First Name: Test
Last Name: User
Phone Number: +639512498751
Email: testuser25@gmail.com
Audition Type: Orchestra
Audition Status: Pending
Audition Schedule: To be set by GYOCC

7. Viewing the Calendar

- First, you have to log in into your account and proceed to the dashboard.

The screenshot shows the GYOCC online dashboard. On the left, a sidebar titled "CONTENTS" includes links for Services, Join GYOCC, History, Learning Materials, Your Calendar (which is highlighted with a red box), and Notifications. The main area features a "Welcome Test User!" message. Below it is an "Announcement Wall" section with a post from "GYOCC Superuser" about the GenSan Youth Choir's victory at the 13th BICF. To the right is a "General Event Calendar" for October 2024, showing various events like "World Chorale Day" and "Violin Recent Lesson".

- To view your calendar, go to the sidebar, the click Your Calendar.

This screenshot is identical to the one above, but the "Your Calendar" link in the sidebar is explicitly highlighted with a red rectangular box, emphasizing the step to take to view the calendar.

- Upon clicking the Calendar button, you were redirected to the page where you could view the General Events Calendar. On this page, you can also click on an event to purchase a ticket for the event.

Your Calendar
October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W40 29	30	1	2	3	4	5
W41 6	7	8	9	10	11	12
W42 13	14	15	16	17	18	19
W43 20	21	22	23	24	25	26
W44 27	28	29	30	31	1	2
W45 3	4	5	6	7	8	9

8. Editing your profile and changing passwords

- First, you must log into your account and proceed to the dashboard.

Welcome Test User!

Announcement Wall

GYOCC Superuser
Superuser
Aug. 17, 2024, 9:58 a.m.
VIEW POST

Title: GenSan Youth Choir Shines at the 13th BICF

General Event Calendar
October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W40 29	30	1	2	3	4	5
W41 6	7	8	9	10	11	12
W42 13	14	15	16	17	18	19
W43 20	21	22	23	24	25	26
W44 27	28	29	30	31	1	2
W45 3	4	5	6	7	8	9

- Clicked on the user profile icon located on the top right of the screen, then select ‘My Profile’.

Welcome Test User!

Announcement Wall

GYOCC Superuser
Superuser
Aug. 17, 2024, 9:58 a.m.
[VIEW POST](#)
Title: GenSan Youth Choir Shines at the 13th BICF

General Event Calendar
October 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
W40 29	30	1	2	3	4	5
8:55a Event VisZWr-dW						
W41 6	7	8	9	10	11	12
8:55a Event VisZWr-dW 7:18p World Chorale Day						
W42 13	14	15	16	17	18	19
7:18p World Chorale Day						
W43 20	21	22	23	24	25	26
12:43p Event K8bMBqMR						
W44 27	28	29	30	31	1	2
12 Event N						
W45 3	4	5	6	7	8	9

GenSan Youth Choir Shines at the 13th BICF
The GenSan Youth Choir from General Santos City, Philippines, achieved a remarkable victory at the 13th BICF.

- Clicking the ‘My Profile’ button redirected you to the User Profile page, where you could edit your information and change your password.

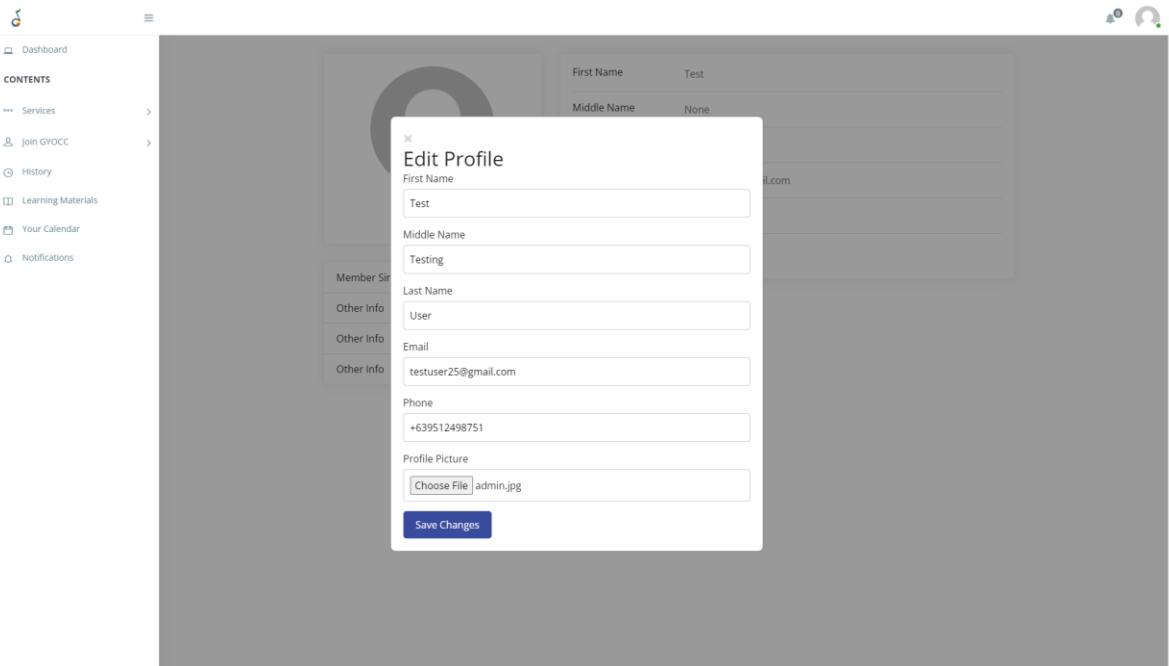
Test User Member

First Name: Test
Middle Name: None
Last Name: User
Email: testuser25@gmail.com
Phone: +639512498751

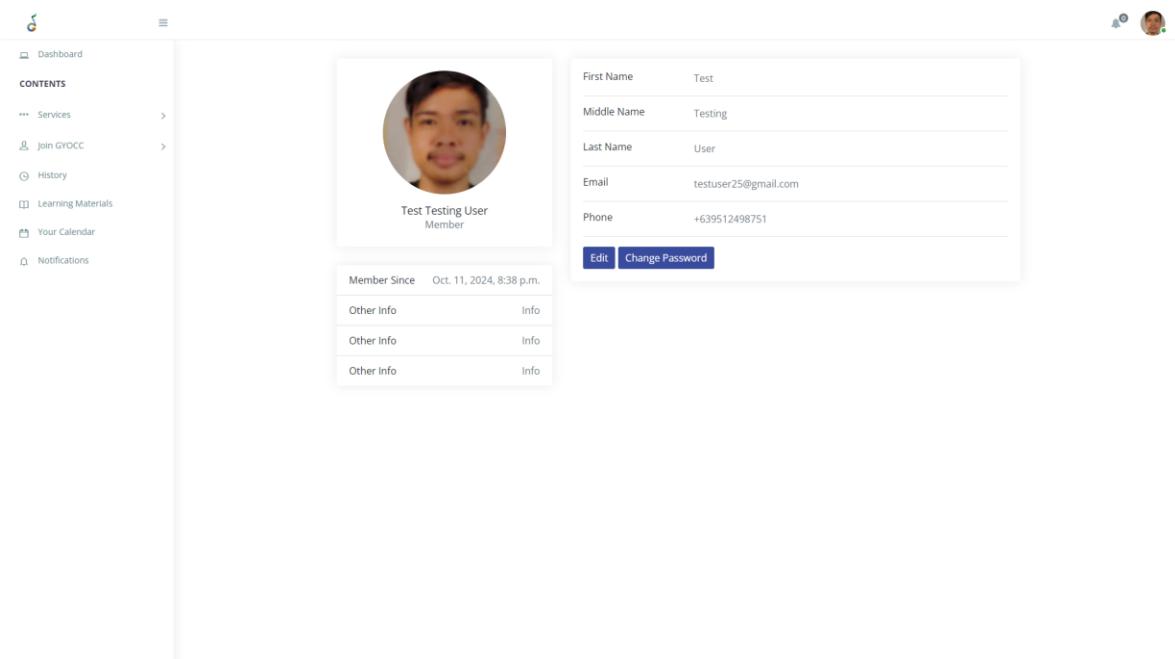
[Edit](#) [Change Password](#)

Member Since	Oct. 11, 2024, 8:38 p.m.
Other Info	Info
Other Info	Info
Other Info	Info

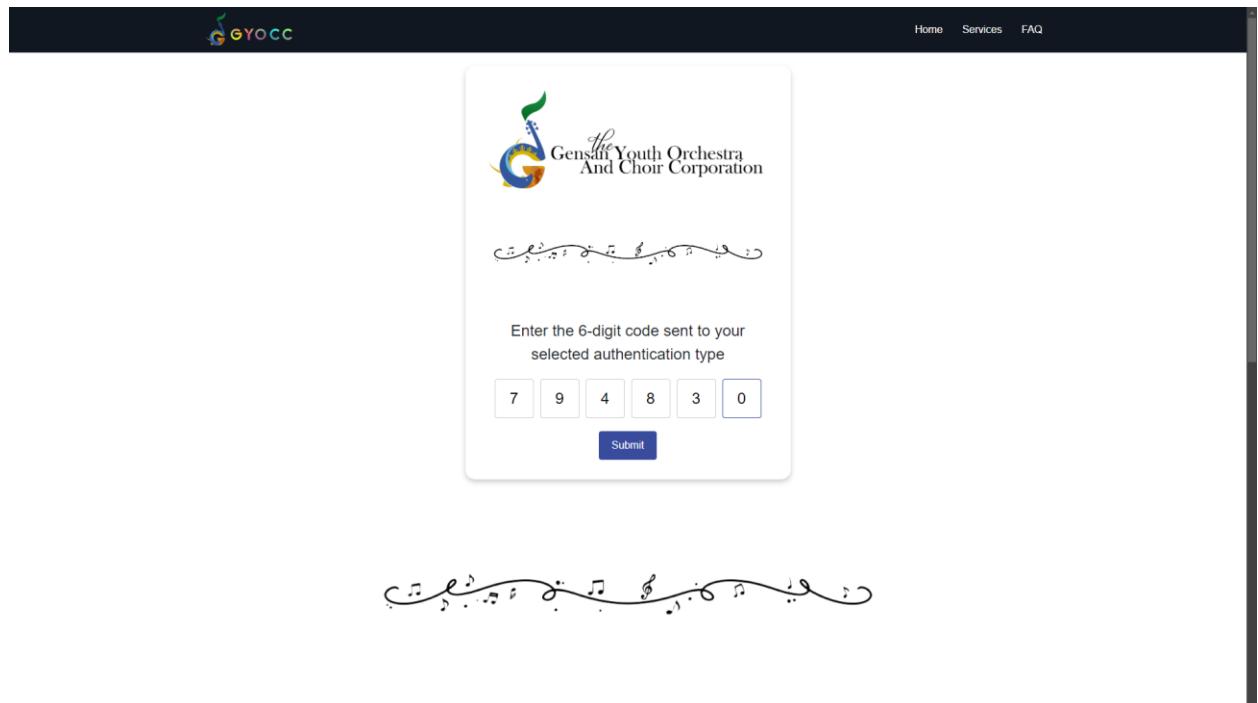
- Click on Edit to edit your personal information, then click Save Changes to save your changed information.



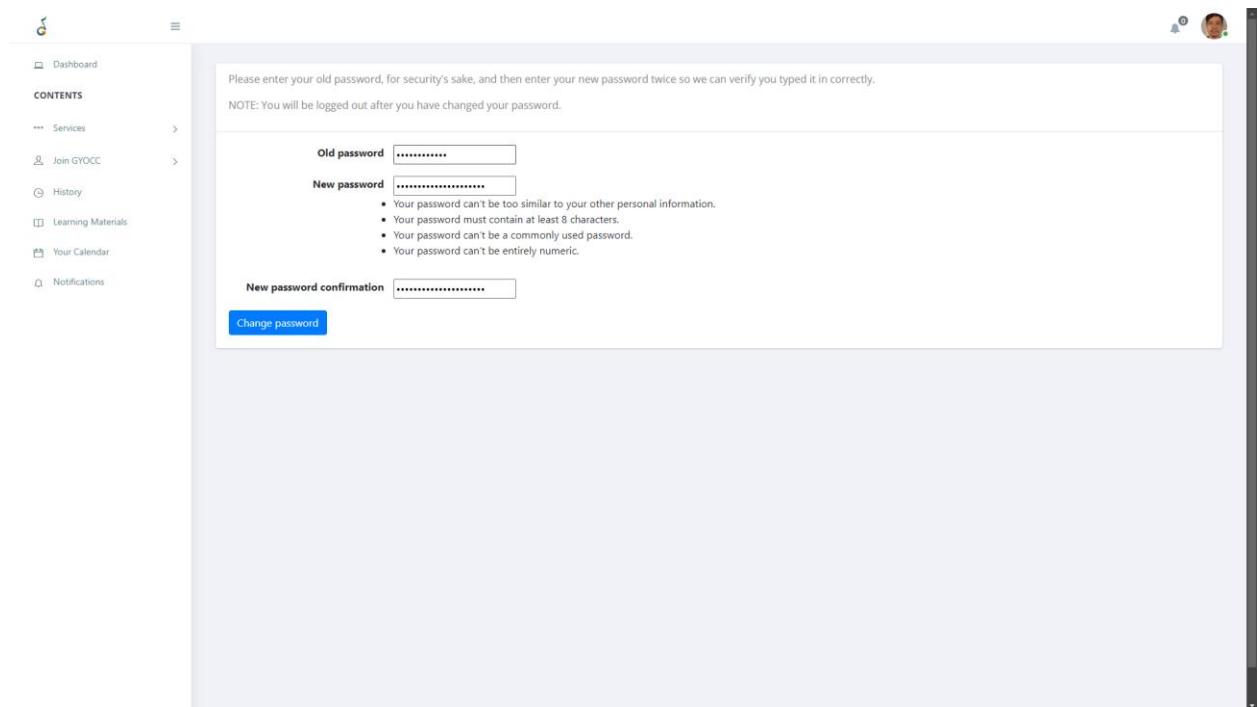
- After saving the changed information, the page reloaded and reflected the newly changed information.



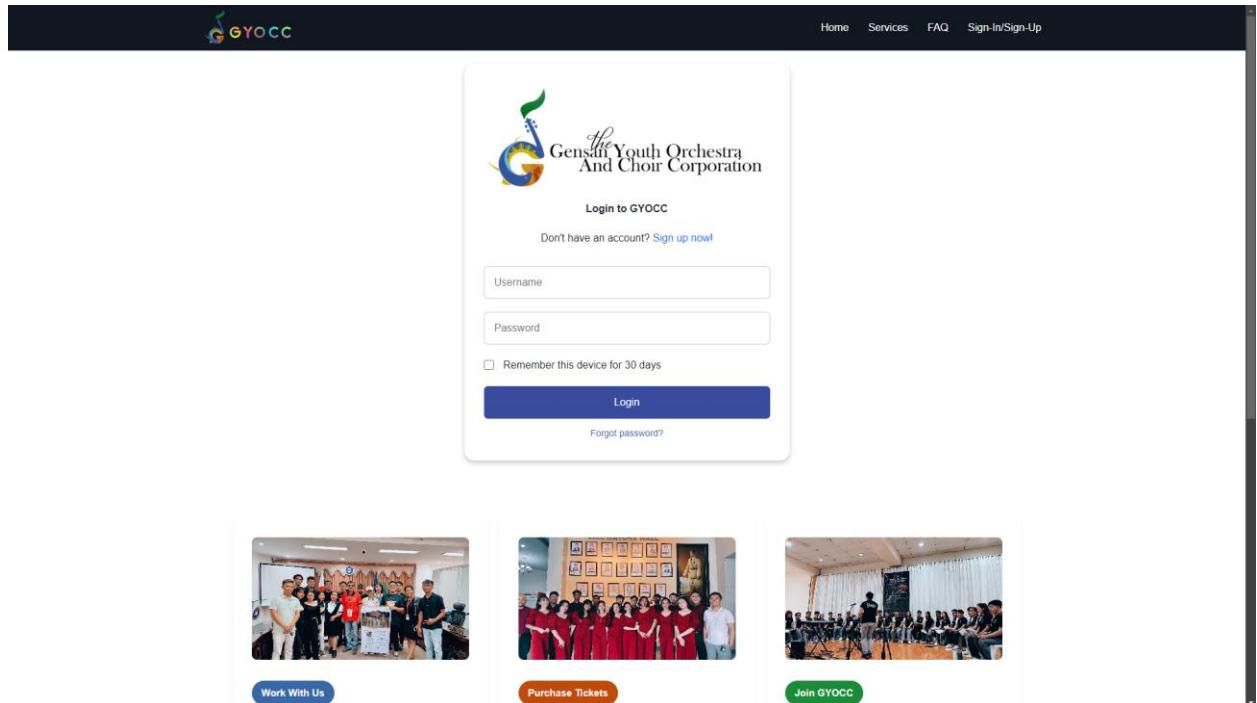
- To change your password, click the ‘Change Password’ button, then select an authentication method to verify your identity before setting a new password.



- After confirming the authentication, enter your old password as well as the new one in their respective fields.



- After confirming the new password, you were logged out and asked to log in again.
Log in again to access GYOCC's services.



9. Learning Materials

- First, you have to login to your account and proceed into the dashboard

The screenshot shows the GYOCC dashboard interface. On the left, there is a sidebar with navigation links: Dashboard, Services, Join GYOCC, History, Learning Materials, Your Calendar, and Notifications. The main content area starts with a welcome message "Welcome Test Testing User!". Below it is an "Announcement Wall" section displaying two posts from "GYOCC Superuser". The first post is from Oct 15, 2024, at 12:52 a.m., titled "Test Post", and the second is from Aug 17, 2024, at 9:58 a.m., titled "GenSan Youth Choir Shines at the 13th BiCF". To the right of the announcements is a "General Event Calendar" for October 2024. The calendar grid shows dates from W40 to W45, with specific events highlighted in different colors: grey for "ESSa Event Visz2WfJW", blue for "8:55a Event Visz2WfJW" and "7:18p World Chorale Day", yellow for "7:18p World Chorale Day", green for "12:43p Event K8bMBqMR", and red for "12 Event I".

- To view available lesson materials, go to the sidebar, the click Lesson Materials.

The screenshot shows the GYOCC dashboard with the 'Learning Materials' option selected in the sidebar. The main content area displays a 'Welcome Test Testing User!' message and an 'Announcement Wall' section. The announcement wall shows two posts from 'GYOCC Superuser' (Superuser) dated Aug. 17, 2024, and Oct. 15, 2024. The post from Aug. 17 includes a photo of a stage setup. To the right, there is a 'General Event Calendar' for October 2024, showing various events like 'World Chorale Day' and 'GenSan Youth Choir Shines at the 13th BICF'.

- You were then redirected to the Learnings Materials Page, where you could select articles written by GYOCC's instructors.

The screenshot shows the 'Learning Materials from GYOCC' page. The sidebar still shows 'Learning Materials' as the selected option. The main content area features a grid of instrument categories: Violin, Guitar, Trumpet, Flute, Drums, Violinero, and Steel Pan Drum. Below the grid is a large, empty rectangular area, likely a placeholder for a list or grid of materials.

- Click on an instrument to browse available articles

The screenshot shows the GYOCC Learning Materials interface. On the left is a sidebar with a Google logo, a dashboard icon, and a menu titled "CONTENTS" containing links to "Dashboard", "Services", "Join GYOCC", "History", "Learning Materials" (which is highlighted in blue), "Your Calendar", and "Notifications". The main content area has a title "Learning Materials from GYOCC" and a subtitle "Access available learning materials from our instructors at GYOCC". Below this is a horizontal navigation bar with tabs for "Violin" (highlighted in blue), "Guitar", "Trumpet", "Flute", and "Drums". Under the "Violin" tab, there is a sub-section titled "Available articles for Violin" with a sub-sub-section titled "Violin Lesson". It features an image of a violin and a bow. Below the image is the title "The Symphony of the Violin" and the author "Calvin Drakke Rulete". A short description follows: "Let the music guide your arms."

- Then, click on an item to view the article.

The screenshot shows the GYOCC Learning Materials interface displaying a specific article. The sidebar and top navigation are identical to the previous screenshot. The main content area shows an article by "Calvin Drakke Rulete", an administrator for Violin. It features a profile picture of Calvin Drakke Rulete. The article title is "The Symphony of the Violin". Below the title are two large images of a violin: one showing the front view and another showing the back/soundboard view. At the bottom of the article area, there is a caption: "Let the flow of the music guide you."

APPENDIX D. PHOTO DOCUMENTATION

Login and Signup Demonstration



Alpha Test Participants Signing Up



Book Appointment Ticket, and Lesson Testing



Payment Testing



Responding to Queries, Problems and Concerns

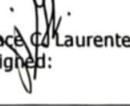


APPENDIX E. ACCOMPLISHMENT AND CONSULTATION FORM

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
Week Number: 1

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
Course Orientation	
Prepared by:  Calvin Jackie Rulete  Owen Lourd Ravina	 John Michael Panizales  Andy Jay Borre
Checked by:  Allyssa Kaye A. Parcon Date Signed:	Noted by:  Ivy Grace C. Laurente Date Signed:

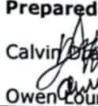
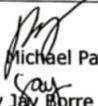
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 1 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GYOCC
 Week Number: 2-4

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
Creating groups, topics, discussion, and brainstorming	
Prepared by:  Calvin Brakke Rulete  Owen Lourd Ravina	
Checked by:  Allyssa Kaye A. Parcon Date Signed:	
Noted by:  John Michael Panizales  Andy Jay Borre	
Noted by:  Ivy Grace C. Laurente Date Signed:	

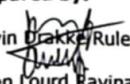
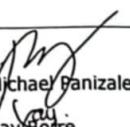
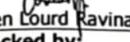
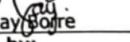
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

 ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 2 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GYOCC
 Week Number: 7

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
<ul style="list-style-type: none"> • Revising of Chapter 1 • Preparation of Chapter 2 RRL, RRS, and Synthesis 	<ul style="list-style-type: none"> ➢ Checking Revised Chapter 1 ➢ Advising of preparation of Chapter 2, year need 2015 – present
Prepared by: <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1; text-align: left;">  Calvin Drakke Rulete </div> <div style="flex: 1; text-align: right;">  John Michael Panizales </div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1; text-align: left;">  Owen Lourd Ravina </div> <div style="flex: 1; text-align: right;">  Andy Jay Bonre </div> </div>	
Checked by: <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1; text-align: left;">  Allyssa Kaye A. Parcon </div> <div style="flex: 1; text-align: right;">  Ivy Grace Laurente </div> </div> Date Signed: <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1; text-align: left;"> (Signature) </div> <div style="flex: 1; text-align: right;"> (Signature) </div> </div>	
Noted by: <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="flex: 1; text-align: left;"> (Signature) </div> <div style="flex: 1; text-align: right;"> (Signature) </div> </div>	

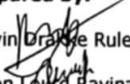
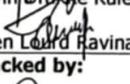
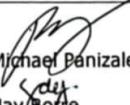
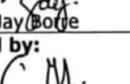
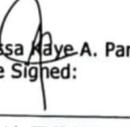
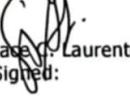
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

 ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 5 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
 Week Number: 8

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
<ul style="list-style-type: none"> • Consultation of revised chapters • Consultation for chapter 2 • Meeting of host company 	<ul style="list-style-type: none"> ➢ Checking of revised chapters ➢ Problems stated by the host company <ul style="list-style-type: none"> • Ticketing • Booking
Prepared by:  Calvin Drake Rulete  Owen Lourd Ravina	 John Michael Panizales  Andy Jay Borne
Checked by:  Allyssa Maye A. Parcon Date Signed:	 Ivy Grace C. Laurente Date Signed:

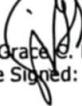
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

 ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 6 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
Week Number: 9

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
<ul style="list-style-type: none">• Chapter 1 90% complete with consultation of adviser• Verifying and providing of problems of the host company to be fixed	<ul style="list-style-type: none">• Checked if the chapters stated have been revised as advice
Prepared by:  Calvin Drake Rulete  Owen Lourd Havina	 John Michael Panizales  Andy Jay Bayre
Checked by:  Allyssa Kaye A. Parcon Date Signed:	Noted by:  Ivy Grace C. Laurente Date Signed:

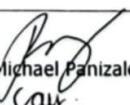
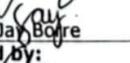
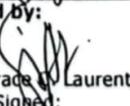
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 7 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
 Week Number: 10

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
<ul style="list-style-type: none"> • Finalizing all chapters from 1-3 • Consultation of the adviser from the upcoming title proposal 	<ul style="list-style-type: none"> ➢ Final checking before title proposal defense ➢ Website responsiveness <ul style="list-style-type: none"> • ERD • Geomapping
Prepared by:  Calvin Drake Rulete  Owen Leurd Ravina	 John Michael Panizales  Andy Jay Boire
Checked by:  Allyssa Kaye A. Parcon Date Signed:	 Ivy Grade M. Laurente Date Signed:

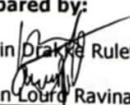
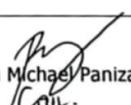
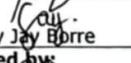
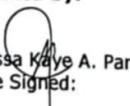
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 8 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
 Week Number: 11

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
Capstone Project Proposal Presentation <ul style="list-style-type: none"> ➢ The panelist provided insights into the host company and the complexity needed for the proposal ➢ Adviser consultation for the final title to be presented on the final defense of Capstone 1 	<ul style="list-style-type: none"> ➢ Hosting ➢ Chapter 1-3 ➢ Diagrams <ul style="list-style-type: none"> • ERD • CFD • DFD • Use Case
Prepared by:  Calvin Drake Rulete  Owen Louie Ravina	 John Michael Panizales  Andy Jay Borre
Checked by:  Allyssa Kaye A. Parcon Date Signed:	 Ivy Grace C. Laurente Date Signed:

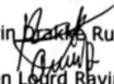
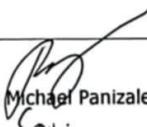
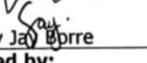
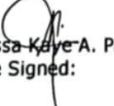
Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

 ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 9 OF 14

ACCOMPLISHMENT AND CONSULTATION FORM

INSTRUCTION: List all the activities, improvements, or accomplishments in your Thesis/Capstone Project Documentation and System/Prototype. This form may be reproduced as you go along with your thesis/capstone project. This form should be submitted to your Thesis/Capstone Project Adviser every week.

Thesis/Capstone Project Title: A Web Management System for GY OCC
 Week Number: 12

ACTIVITY/ ACCOMPLISHMENT	REMARKS/ COMMENTS/ SUGGESTIONS/ DELIVERABLES and DUE DATE
<ul style="list-style-type: none"> • Finalize checking chapter 1-3 • Diagrams Checking • Wireframe Checking 	<ul style="list-style-type: none"> ➢ Analytical Reports Module should include all the reports and logs ➢ CFD and DFD element suggestions and guidelines ➢ Temporary Event Booking Changes (Fixed Expiration Date)
Prepared by:  Calvin Markie Rulete  Owen Lourd Ravina	 John Michael Panizales  Andy Jay Borre
Checked by:  Allyssa Kaye A. Parcon Date Signed:	Noted by:  Ivy Grace Laurente Date Signed:

Copyright STI EDUCATION SERVICES GROUP, INC. All rights reserved. May not be reproduced without permission. Records of all approved quality documents are maintained and controlled in the Quality Document Catalog. Distribution and use of quality documents shall be in accordance to the Control of Documents Procedure.

 ACCOMPLISHMENT AND CONSULTATION FORM | FT-CRD-187-00 | PAGE 10 OF 14

APPENDIX F. TRANSCRIPTION



October 9, 2024

Mr. Fernan Globen Talonding

Chief Executive Officer

Gensan Youth Orchestra and Choir Corporation

Ground Floor, Jumabo Bldg, 23 Kadulasang, General Santos City, 9500 South Cotabato

Dear Sir/Ma'am,

Mabuhay!

The undersigned is currently working on research titled "**A Web Management System for Gensan Youth Orchestra and Choir Corporation**". This study aims to gather data related to the development of the management system for the corporation.

In connection with this, I would like to ask your permission and endorsement to allow us to survey/interview the CEO, Staff, Instructors, and Organizers. Rest assured that whatever data will be gathered from this study will be treated with utmost confidentiality.

I am hoping for your favorable response to this undertaking.

Respectfully yours,

John Michael S. Panizales

Calvin Drakke I. Rulete

Owen Lourd L. Ravina

Andy Jay Borre

Noted by:

Allyssa Kaye A. Parcon

Adviser

John Michael S. Panizales
michaelpanizales1203@gmail.com

9/1/2024

Gensan Youth Orchestra and Choir Corporation
+639084091939
Ground Floor, Jumabo Bldg, 23 Kadulasang, General Santos City, 9500 South Cotabato

Dear, Sir Fernan Globen Talonding,

We are writing to confirm the details of the services and features that have been requested by your organization as part of our collaboration.

1. Auditions:

- i. **Applicant Information:** We will collect the following details from individuals interested in joining the orchestra or choir:
 - o Full Name
 - o Address
 - o Contact Number
- ii. **Choir Audition Options:**
 - o Applicants may either:
 1. Submit an online video of their singing performance.
 2. Book an appointment for a live audition.
- iii. **Orchestra Audition:**
 - o Applicants will need to book an appointment to audition live.

2. Music Lessons Enrollment:

- i. **Paid Service:** This service will require payment to avail.
- ii. **Instruments Offered:** Various instruments are available for lessons, such as voice, violin, piano, etc. Additionally, there is the flexibility to add new services.
- iii. **Teacher Availability and Pricing:**
 - o **One-on-One Sessions:**
 - Personalized and exclusive teaching for the student.
 - Pricing is higher due to individual attention.
 - Scheduling options: Once a week, twice a week, or thrice a week.
The fee will scale accordingly.
 - The ability to edit pricing for these sessions.
 - o **Group Sessions:**
 - Sessions can accommodate 3 or more students per hour.

- The fee is lower compared to one-on-one due to shared time among students.
 - Scheduling options: Once a week, twice a week, or thrice a week. The fee will scale accordingly.
 - The ability to set the fee for these sessions.
- **Long-Term Commitment:**
 - These lessons can extend over months or even years, with payments being made monthly.

3. Workshop Services:

- i. **Duration:**
 - Workshops are short-term, ranging from 2 days to a week, depending on your preference.
 - Example: A recent 2-day voice workshop with the Madrigals, followed by a concert on the third day.
- ii. **Offerings:**
 - Registration for the workshop (one-time payment).
 - Ticket sales for the concert.

This letter serves as an official record of the features and services your company has requested. Please review the details provided above to ensure they meet your expectations.

Sincerely,

Acknowledgement by GY OCC:



Mr. Fernan Globen Talonding
Chief Executive Officer
Gensan Youth Orchestra and Choir Corporation

John Michael S. Panizales
michaelpanizales1203@gmail.com
5/13/2024

Gensan Youth Orchestra and Choir Corporation
+639084091939
Ground Floor, Jumabo Bldg, 23 Kadulasang, General Santos City, 9500 South Cotabato

Certification of Interview(s)
Date of Interview(s): April 30, 2024, May 9, 2024

Dear Mr. Fernan Globen Talonding

This letter is to certify that an interview was conducted by our group on dates April 30, 2024 and May 9, 2024. The following information was provided during the interview:

Following our recent discussions and interview regarding the development of a new system tailored specifically for GY OCC, we are writing to summarize and confirm the key points and requirements as we understood them. This will ensure that our final deliverables align perfectly with your needs and expectations.

Approved Suggested Functionalities:

Repertoire Management System: To effectively manage and organize your musical pieces and arrangements.

Member Management System: To streamline the management of member information and activities.

Additional Features Requested:

No Appointment Fees: Ensuring that booking and appointments are handled without additional charges.

Calendar Automation: To facilitate the automatic updating and management of your event schedules.

Clarification on Terms and Functionalities:

Pencil Booking: You have described this as a temporary booking status where commitments are considered provisional until confirmed within a stipulated timeframe (one week). We would like to confirm that this term and its definition are correctly captured as it will directly influence the booking module's design.

Seat Mapping Concerns:

You have requested a feature that allows the input of available seats without a fixed seating arrangement. The system should also alert the staff when only a limited number of seats remain available. We understand that this functionality is crucial due to the lack of ownership over performance venues.

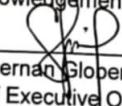
Justification for System Development:

GYOCC's unique position in the region as a pioneer in offering structured music education and performance opportunities justifies the need for a bespoke system. The system will support the organization's goals to provide grants and scholarships, manage finances transparently, and comply with governmental reporting requirements. This will not only streamline administrative processes but also support strategic decision-making to further your mission.

These information needs clarification by the organization to allow us to make a detailed planning and regular consultations with your team to ensure that the development of this system aligns with GYOCC's operational workflows and long-term objectives and to validate the accuracy of the information shared during the interview.

Sincerely,

Acknowledgement by GYOCC:



Mr. Fernan Globen Talonding
Chief Executive Officer
Gensan Youth Orchestra and Choir Corporation

John Michael S. Panizales
michaelpanizales1203@gmail.com
10/12/2024

Gensan Youth Orchestra and Choir Corporation
+639084091939
Ground Floor, Jumabo Bldg, 23 Kadulasan, General Santos City, 9500 South Cotabato

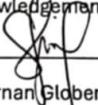
Validation of data gathered after the Alpha Testing

Dear Mr. Fernan Globen Talonding

We are writing to request your validation of the data gathered after the alpha test we conducted in your organization on October 9, 2024. The information collected is genuine and shows the evaluation in terms of user experience, functionality and overall performance of our system. During the alpha test, we would also like to confirm that 60% of the daily average rehearsal participants participated in this test. Your confirmation of these details will be greatly appreciated as we continue to develop the system. Thank you for your cooperation.

Sincerely,

Acknowledgement by GY OCC:



Mr. Fernan Globen Talonding

Chief Executive Officer
Gensan Youth Orchestra and Choir Corporation

John Michael S. Panizales
michaelpanizales1203@gmail.com

9/25/2024

Gensan Youth Orchestra and Choir Corporation
+639084091939
Ground Floor, Jumabo Bldg, 23 Kadulasang, General Santos City, 9500 South Cotabato

Dear, Sir Ferman Globen Talonding,

We are writing to confirm the details of the services and features that have been requested by your organization as part of our collaboration.

The Gensan Youth Orchestra and Choir Corporation (GYOCC) operates by organizing multiple events and offering various music-related services. They sell between 80% to 85% of their tickets for events, with prices ranging from 500 PHP for general admission to 2,500 PHP for VIP tickets, resulting in estimated revenue between 50,000 to 100,000 PHP. GYOCC's services include gig and event bookings, music workshops and lessons starting at 1,500 PHP, recording services, music production, an annual music camp, and participation in both national and international competitions and events. Additionally, they participated recently a Choral Celebration. The organization consists of a 35-member choir and a 12-member orchestra, with about 30 to 40 public and private events annually. GYOCC is managed by five organizers.

Annual Data:

- Tickets available – 2000
- Ticket sales rate – 80%-85% sold
- Ticket prices – 500 PHP (general admission) to 2,500 PHP (VIP)
- Revenue ranges – 25,000-75,000 PHP
- Organizers – 1

Services provided:

- Gig and event bookings
- Music workshops and lessons (starting from 1,500 PHP)
- Recording services
- Music production
- Annual music camp
- Participation in national & international events and competitions

Active members:

- Choir – 35
- Orchestra – 12

Annual events – 30 - 40 (public and private)

This data will serve as the relevance of the functionality of the system

This letter serves as an official record of the features and services your company has requested. Please review the details provided above to ensure they meet your expectations.

Sincerely,

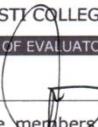
Acknowledgement by GY OCC:



Mr. Fernan Globen Talonding
Chief Executive Officer
Gensan Youth Orchestra and Choir Corporation

APPENDIX G. REVISION'S LIST

REVISION LIST FORM

CAPSTONE PROJECT TITLE		
ACCHORD: AN OPERATIONAL INFORMATION SYSTEM FOR GYOC'S SERVICES		
NAME OF GROUP MEMBERS		
Andy Jay Borre		John Michael S. Panizales
Owen Lourd L. Ravina		Calvin Drakke I. Rulete
DATE OF DEFENSE	TIME OF DEFENSE	STI CAMPUS
May 9, 2024	9:00 AM	STI COLLEGE - GENERAL SANTOD
NAME OF EVALUATOR		SIGNATURE OF EVALUATOR
Nur Ali Padasan, MIT		

INSTRUCTIONS: Indicate the questions you intend to ask the members of the Capstone Project group. Questions may include but are not limited to the Capstone Project document contents, requirements, analysis, design and programming of the software. Please indicate short comments, which will serve as your evaluation of the response(s) given to the question.

LIST OF REVISIONS	
Questions	Comments
Web-Based Management System for GYOC	* reconstruct the specific objectives * proposed modules for a Web-Based Mgt. System for GYOC
1. Member Mgt.	- registration, membership, etc.
2. Event Mgt.	- booking, ticketing, etc.
3. Repertoire Mgt.	- upload music sheets, audio files, etc.
4. Resource Mgt.	- inventory mgt. for musical instruments, equipment, uniforms - borrowing
5. Data Reporting and Analytics	
6. Messaging (internal)	
	* social media integration

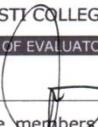
REVISION LIST FORM

CAPSTONE PROJECT TITLE		
ACCHORD: AN OPERATIONAL INFORMATION SYSTEM FOR GYOC's SERVICES		
NAME OF GROUP MEMBERS		
Andy Jay Borre		John Michael S. Panizales
Owen Lourd L. Ravina		Calvin Drakke I. Rulete
DATE OF DEFENSE	TIME OF DEFENSE	STI CAMPUS
May 9, 2024	9:00 AM	STI COLLEGE – GENERAL SANTOD
NAME OF EVALUATOR		SIGNATURE OF EVALUATOR
Kristel Joy C. Tulagan, MIT, LPT		

INSTRUCTIONS: Indicate the questions you intend to ask the members of the Capstone Project group. Questions may include but are not limited to the Capstone Project document contents, requirements, analysis, design and programming of the software. Please indicate short comments, which will serve as your evaluation of the response(s) given to the question.

LIST OF REVISIONS	
Questions	Comments
-change title	Features: - booking (2FA) → - - ticket → 1 - lessons 2 - → + Payment
Tourism & Reservation Management: ↳ Member Management ↳ Booking Management ↳ by Ticketholder Management ↳ Scheduling ↳ Seat management ↳ Payment Management ↳ Resource Management ↳ Reports ↳ Analytics of events ↳ Announcement & Post of events ↳ 2FA applied ↳ Use Power Graph API integrated in the system	+ Customer Relation (Marketing) ↳ Newsletter - → Scheduling (Event) ↳ Processing of Event requested by client ↳ Processing of ↳ Document: + Attach supporting documents

REVISION LIST FORM

CAPSTONE PROJECT TITLE		
ACCHORD: AN OPERATIONAL INFORMATION SYSTEM FOR GYOC'S SERVICES		
NAME OF GROUP MEMBERS		
Andy Jay Borre		John Michael S. Panizales
Owen Lourd L. Ravina		Calvin Drakke I. Rulete
DATE OF DEFENSE	TIME OF DEFENSE	STI CAMPUS
May 9, 2024	9:00 AM	STI COLLEGE - GENERAL SANTOD
NAME OF EVALUATOR		SIGNATURE OF EVALUATOR
Nur Ali Padasan, MIT		

INSTRUCTIONS: Indicate the questions you intend to ask the members of the Capstone Project group. Questions may include but are not limited to the Capstone Project document contents, requirements, analysis, design and programming of the software. Please indicate short comments, which will serve as your evaluation of the response(s) given to the question.

LIST OF REVISIONS	
Questions	Comments
Web-Based Management System for GYOC	* reconstruct the specific objectives * proposed modules for a Web-Based Mgt. System for GYOC
1. Member Mgt.	- registration, membership, etc.
2. Event Mgt.	- booking, ticketing, etc.
3. Repertoire Mgt.	- upload music sheets, audio files, etc.
4. Resource Mgt.	- inventory mgt. for musical instruments, equipment, uniforms - borrowing
5. Data Reporting and Analytics	
6. Messaging (internal)	
	* social media integration

APPENDIX H. FIGURES AND TABLES

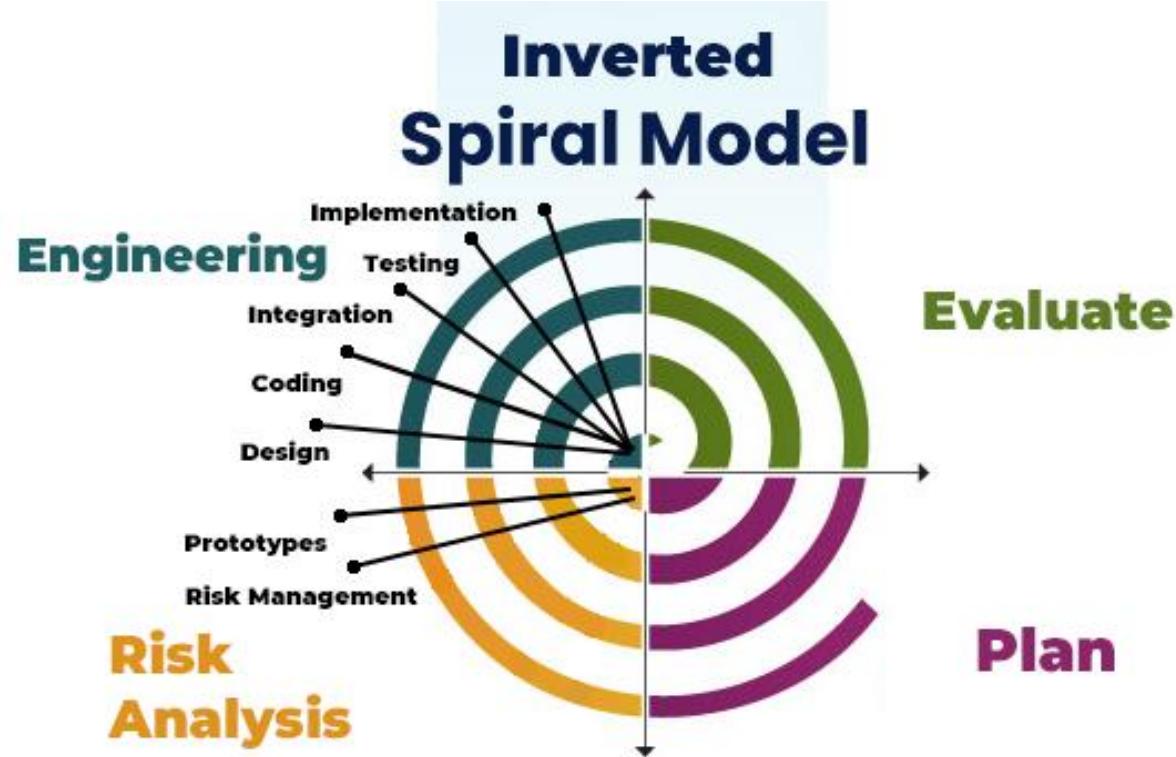


Figure 1: Inverted Spiral Model

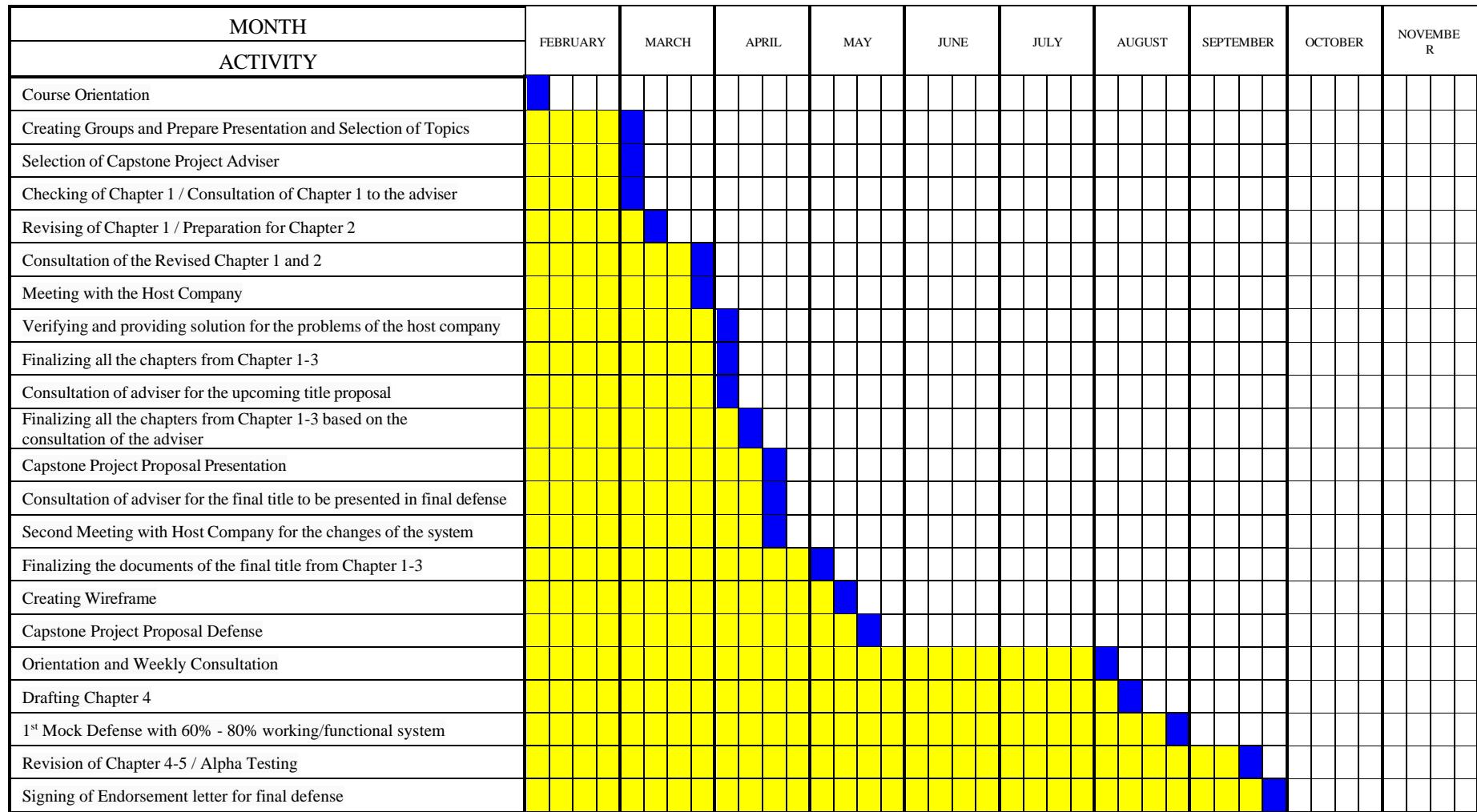


Figure 2: Gantt Chart of Activities

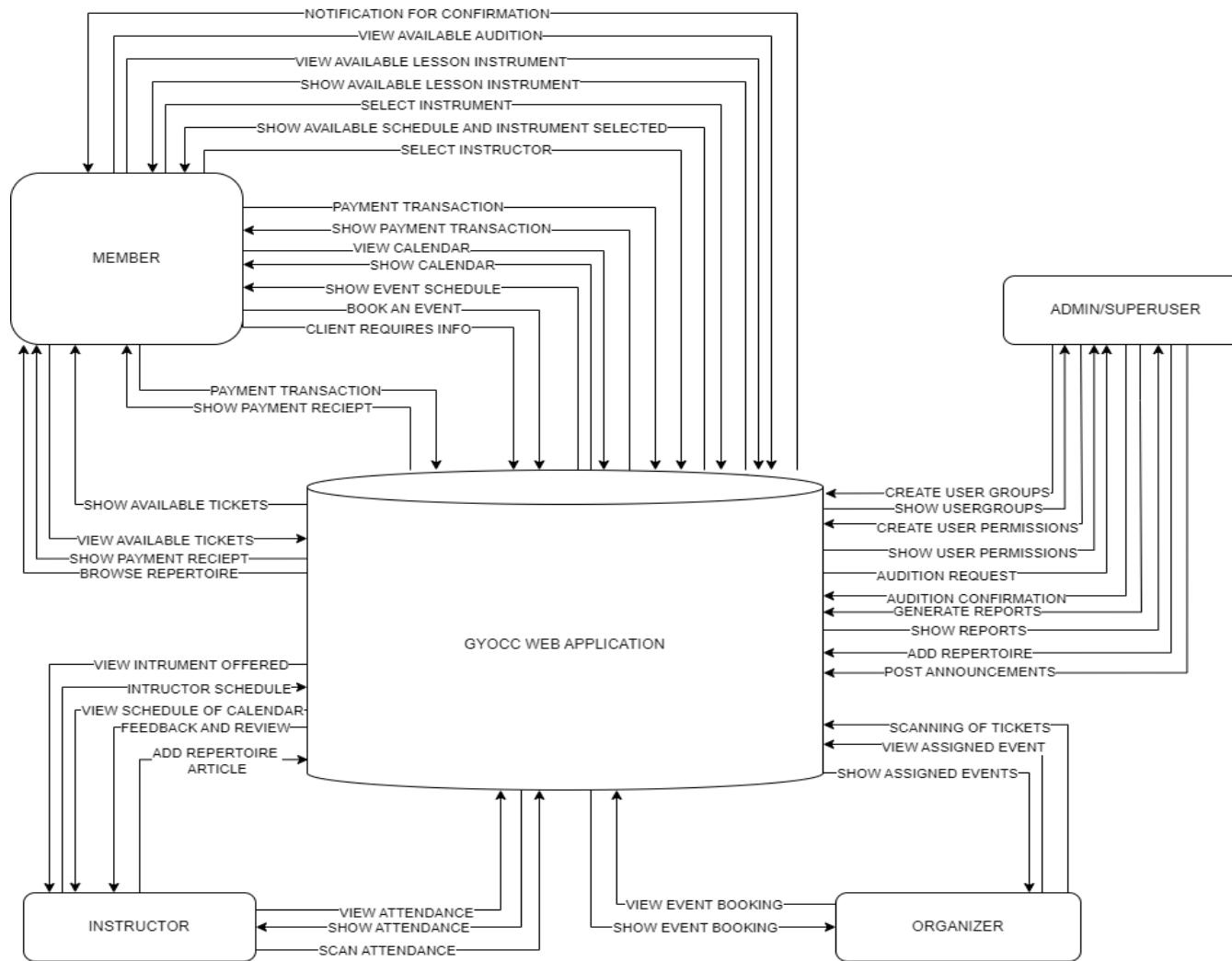


Figure 3. Context Flow Diagram

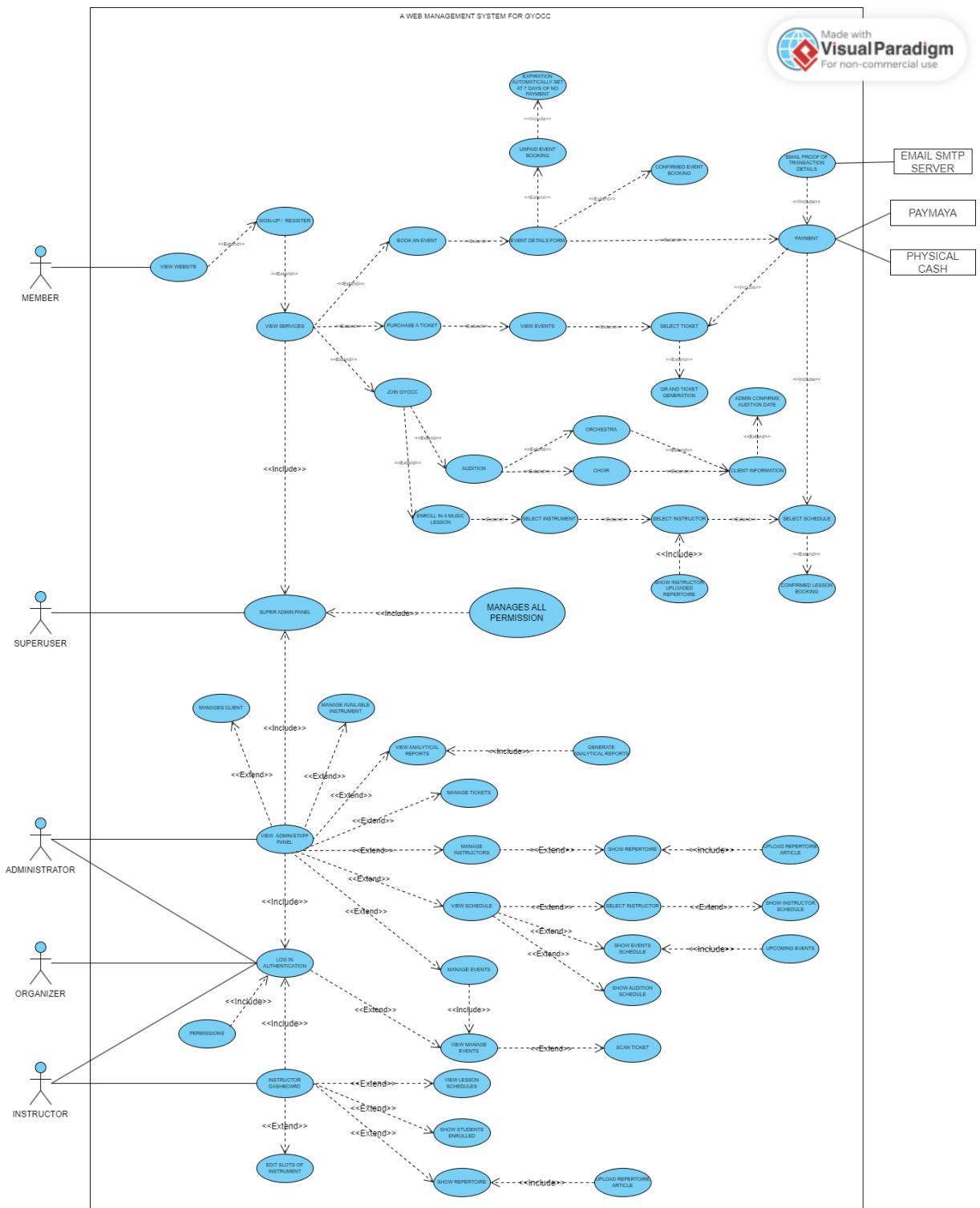


Figure 4: Use Case Diagram

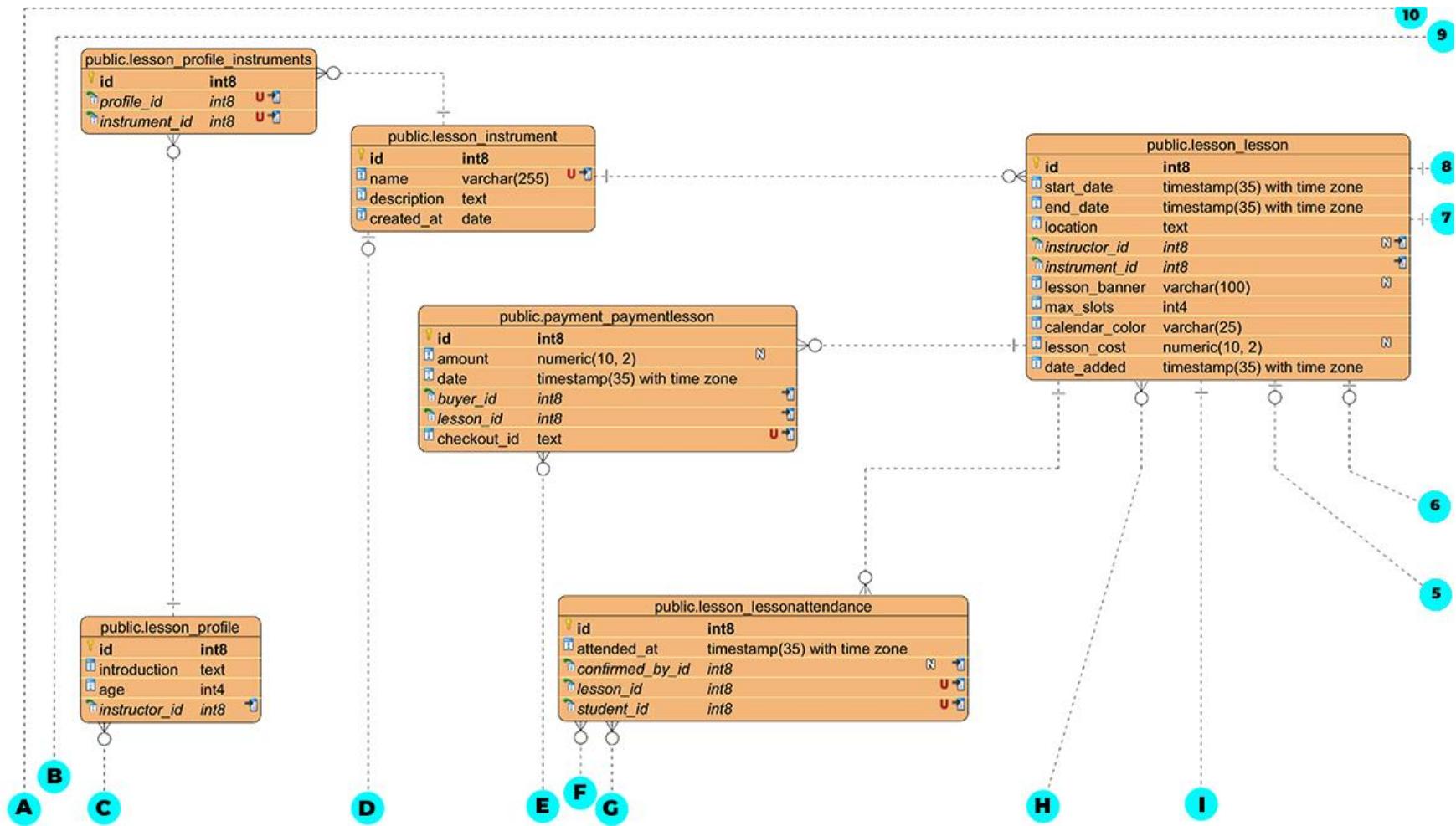


Figure 5: Entity Relationship Diagram

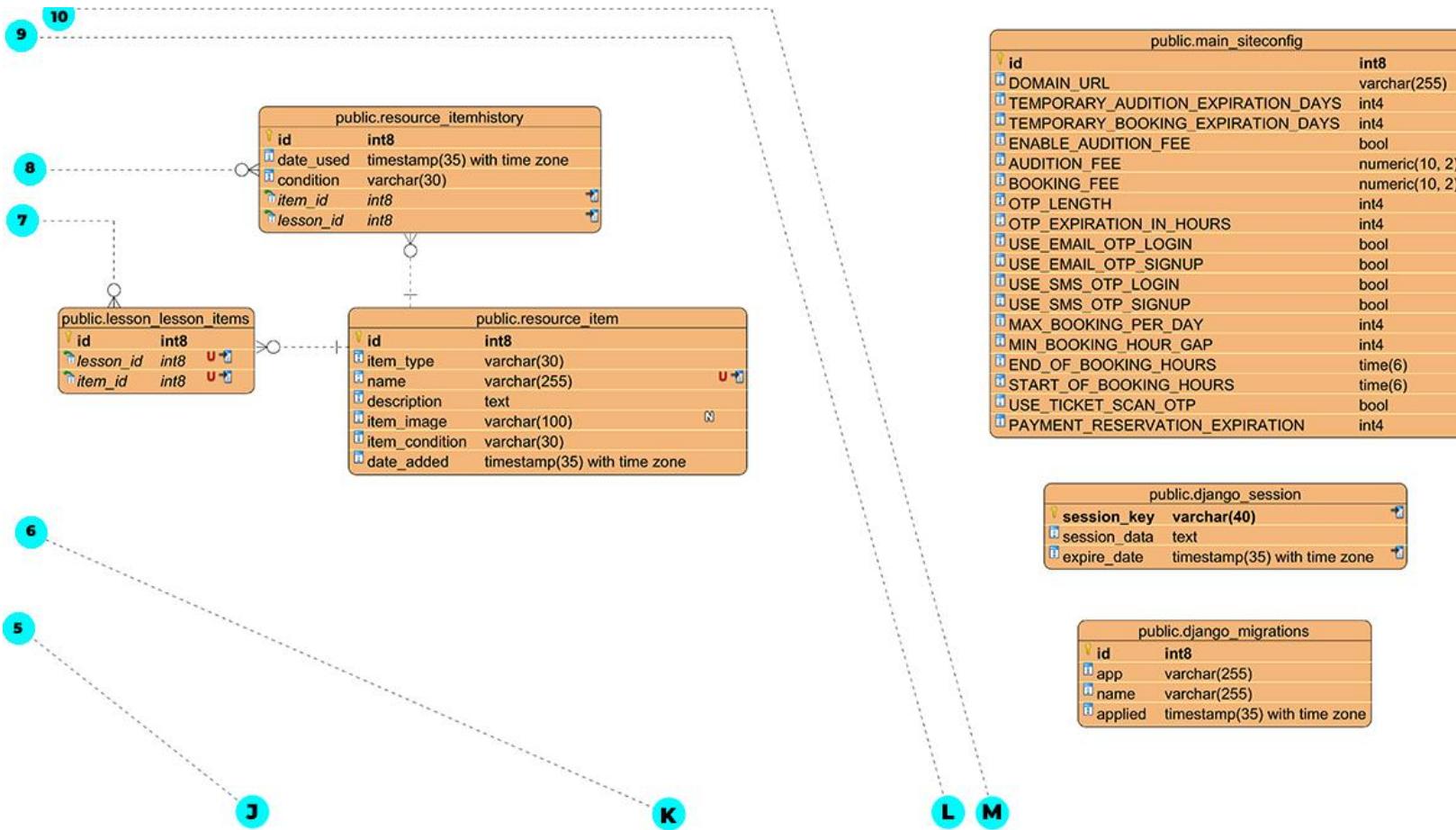


Figure 5.1: Entity Relationship Diagram

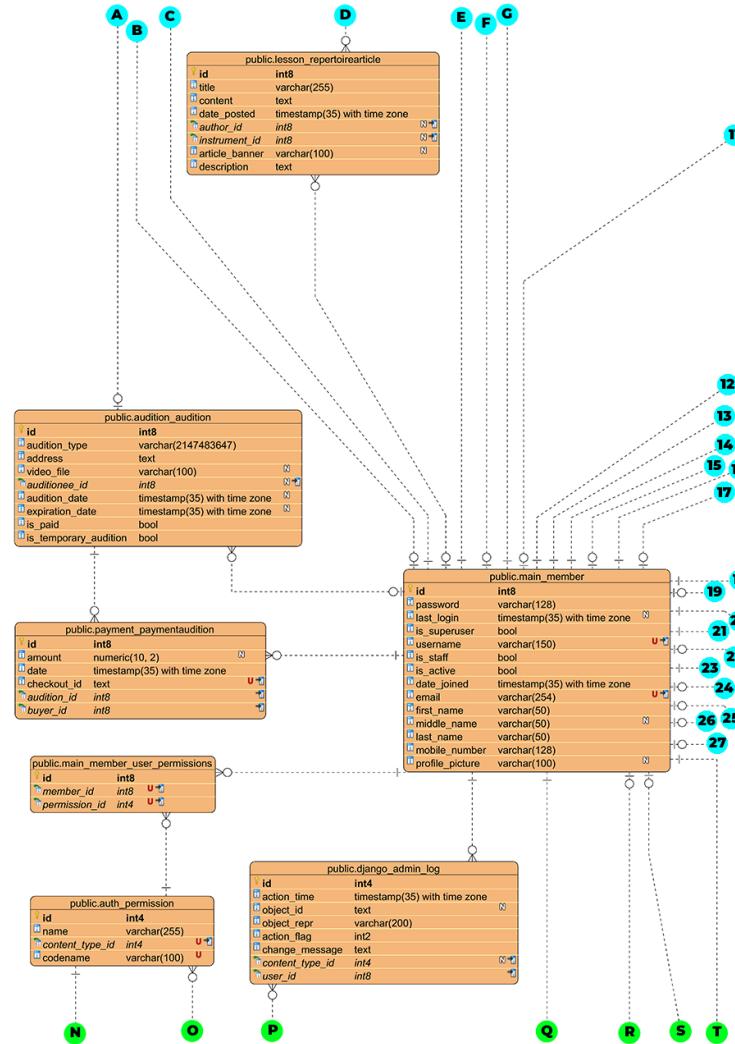


Figure 5.2: Entity Relationship Diagram

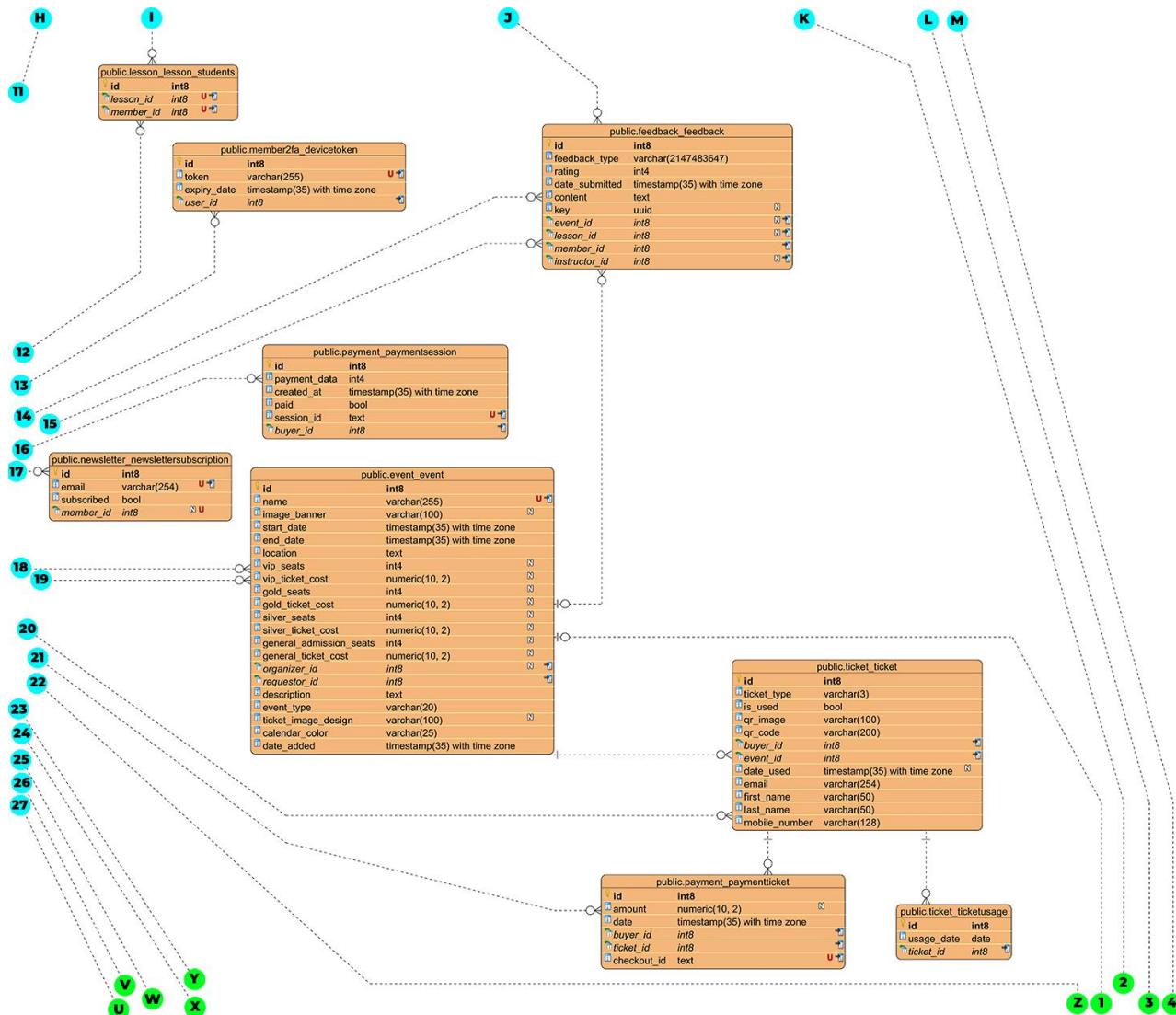


Figure 5.3: Entity Relationship Diagram

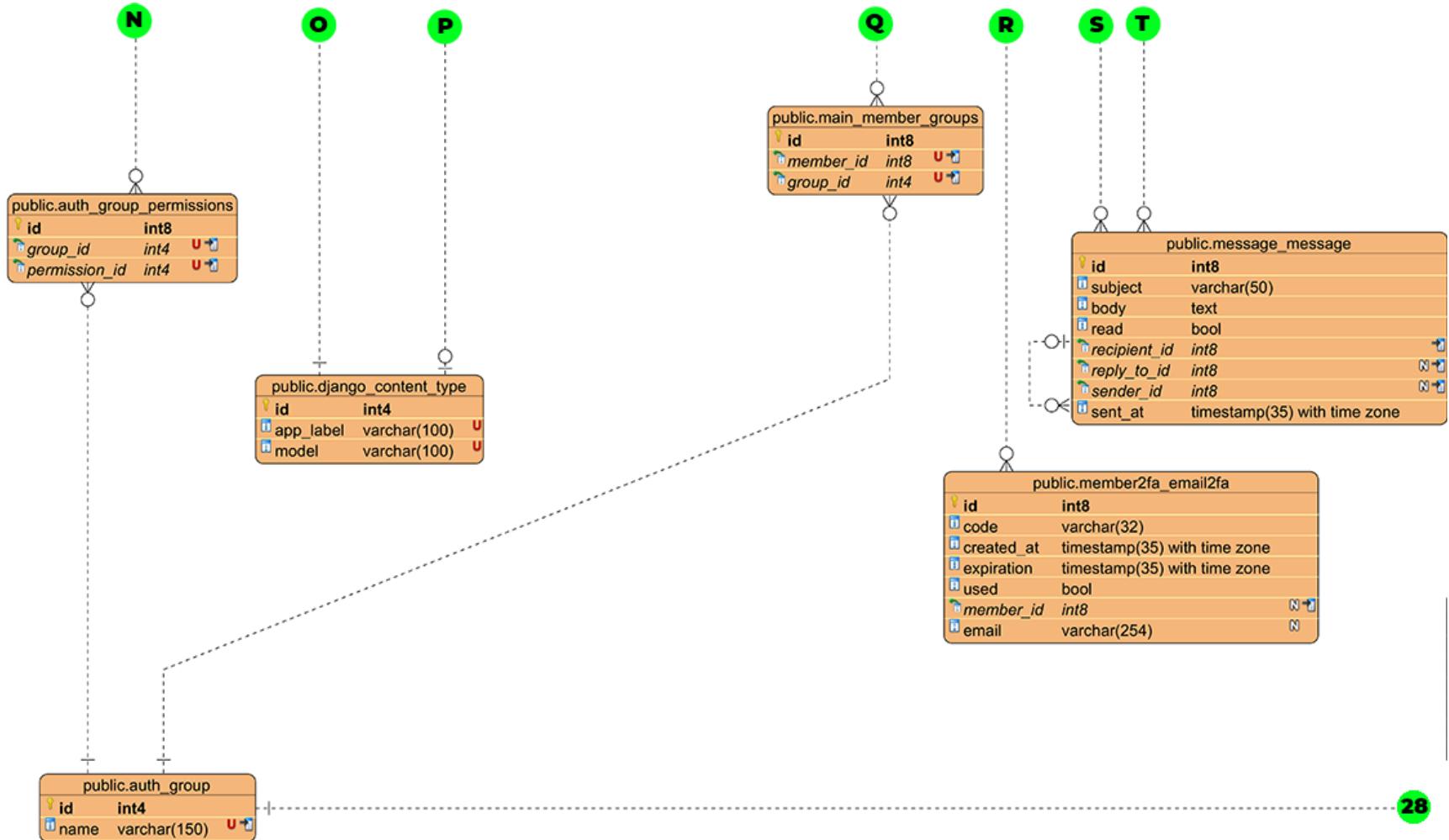


Figure 5.4: Entity Relationship Diagram



Figure 5.5: Entity Relationship Diagram

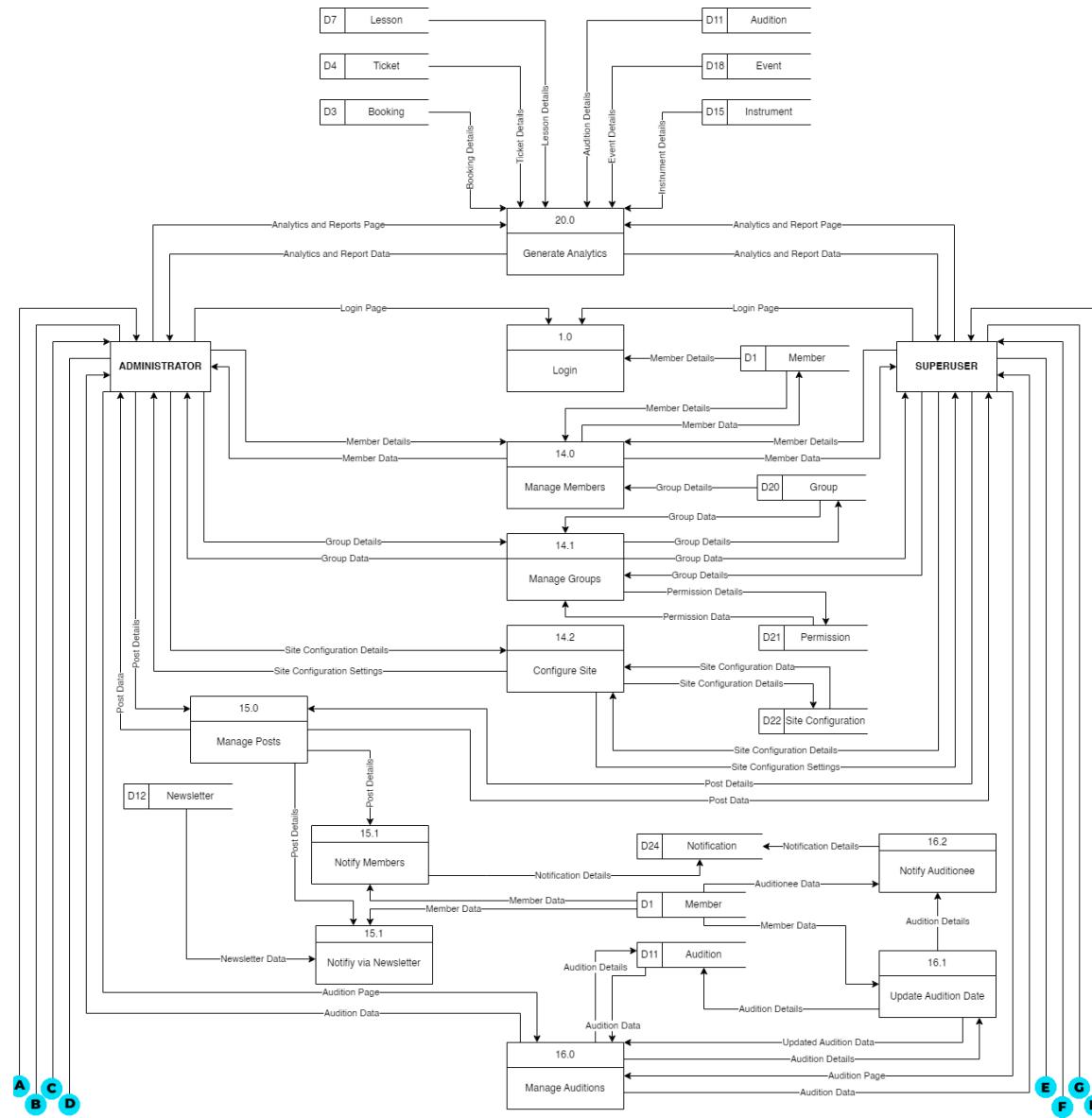


Figure 6: Data Flow Diagram

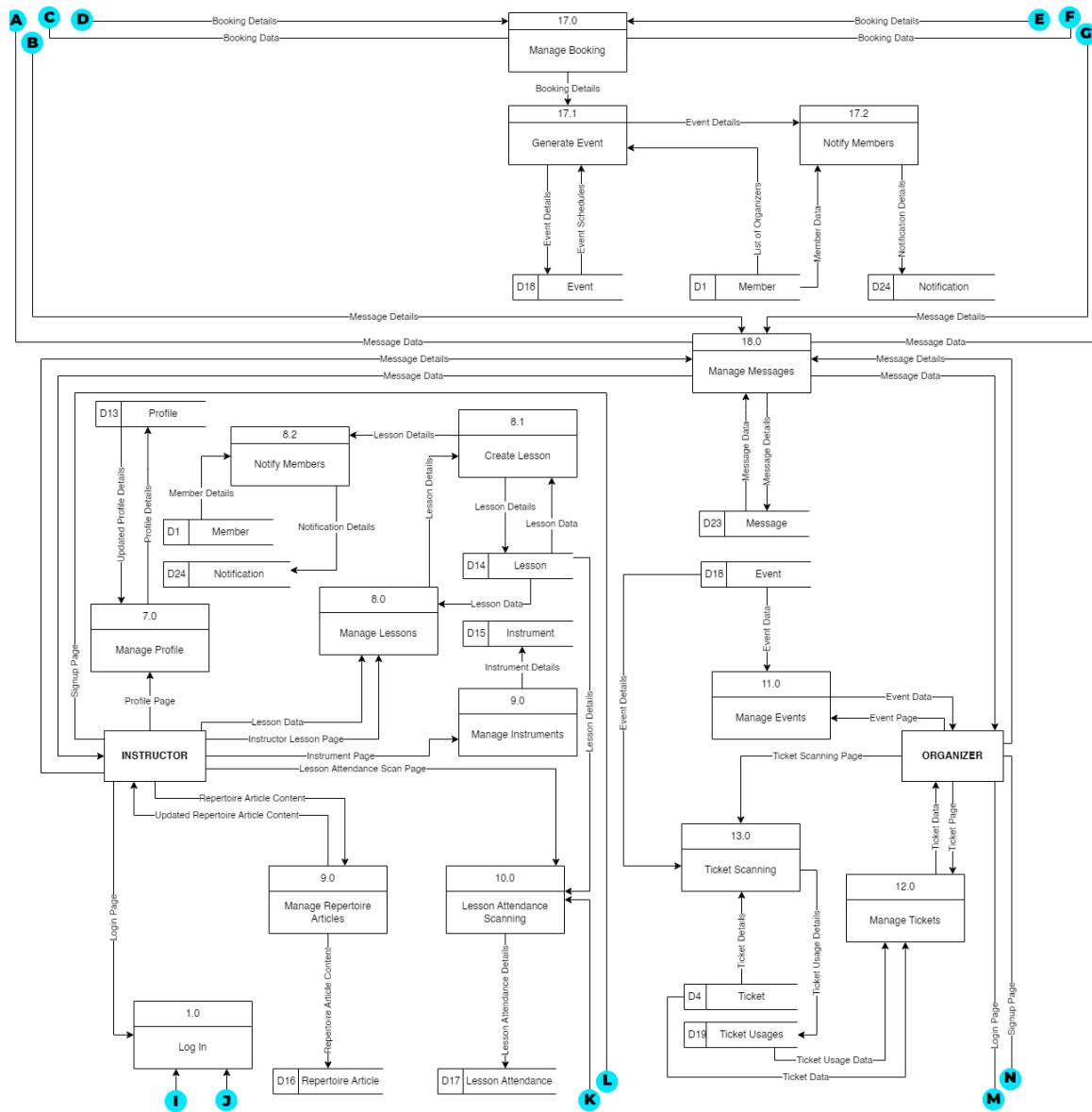


Figure 6.1: Data Flow Diagram

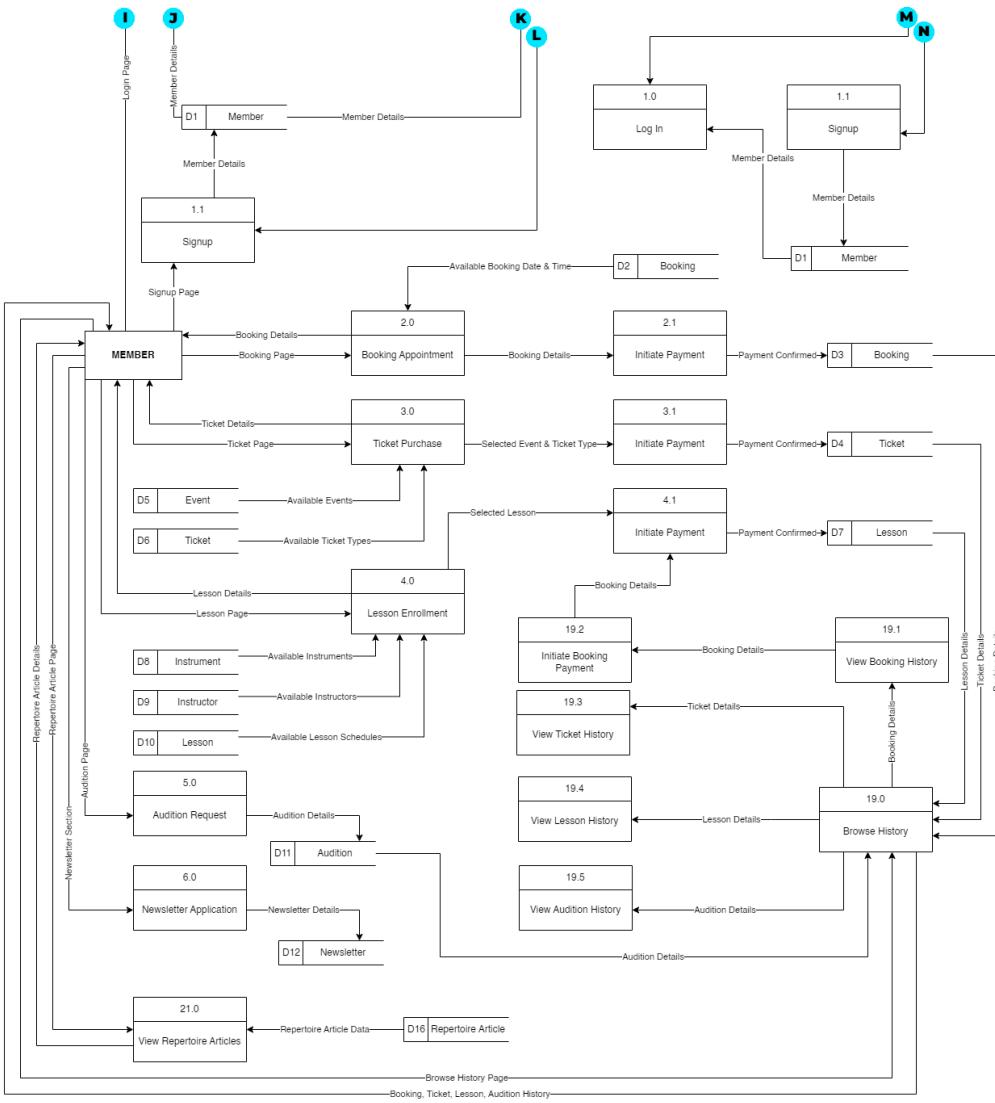


Figure 6.2: Data Flow Diagram

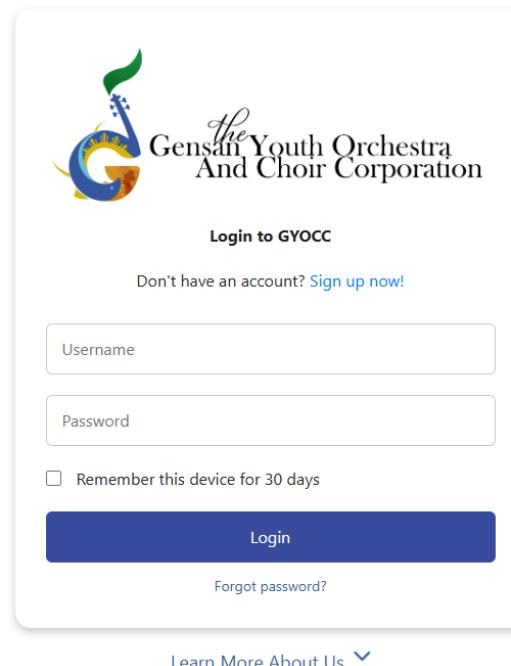


Figure 7.1 Landing Page

Home Services ▾ FAQ Sign-In/Sign-Up

Gensan Youth Orchestra And Choir Corporation

Register to GYOC

Already have an account? [Login instead](#)

Username*	First Name*
Enter username*	Enter first name*
Middle Name	Last Name*
Enter middle name	Enter last name*
Email*	Mobile Number*
Enter email*	+63 0000000000*
Password*	Confirm Password*
Enter password*	Confirm password*

Password Guidelines

- Your password cannot be too similar to your personal information.
- Your password must contain at least 8 characters.
- Your password cannot be a commonly used password.
- Your password can't be entirely numeric.

I agree to the [Terms of Service](#)

Register

Gensan Youth Orchestra Choir Corporation
The biggest orchestra program in Mindanao coupled with an internationally competing choir program.

[FAQ](#) [Terms of Service](#) [Privacy Policy](#)

Follow Us:

#23 Jomabao Building
Kadulasan Street,
Barangay Dadiangas East

0908 409 1939
[Call us](#)

[Subscribe](#)
Subscribe to our news letter to be update about future events

Your email address

All Rights Reserved, Gensan Youth Orchestra and Choir Corporation

Figure 7.2 Sign-up Page

Figure 7.3 Frequently Asked Questions Page

Welcome GYOCC Superuser!



VIP - Birthday ni JMP
Recently Purchased Ticket



No Recent Booking
Recently Booked Appointment



Violin
Recent Lesson



Choir
Recent Audition

Announcement Wall

GYOCC Superuser

 Superuser
 Aug. 17, 2024, 9:58 a.m.
[VIEW POST](#)

Title: GenSan Youth Choir Shines at the 13th BICF



General Event Calendar

		November 2024							month	week	day	list
27	28	29	30	31	1	2						
1 Oct												
3	4	5	6	7	8	9						
10	11	12	13	14	15	16						
							4:36a Event tByhPzoc					
17	18	19	20	21	22	23						
							2:29p Event nBjQ48ph					
24	25	26	27	28	29	30						
							2: Even					
1	2	3	4	5	6	7						
							9:31p Event KNugsWnV					
												5:45p Event YQdYie2q

Figure 7.4 User Dashboard

The screenshot shows a web-based application interface for booking an appointment. On the left, there's a sidebar with a navigation menu titled 'CONTENTS' containing links like 'Dashboard', 'Services', 'Join GYOCC', 'History', 'Learning Materials', 'Your Calendar', and 'Notifications'. The main area is titled 'Book an Appointment for an Event'. It contains the following fields:

- First Name: GYOCC
- Last Name: Superuser
- Mobile Number: +639081600631
- Email: calvindrakktesezt9@gmail.com
- Event Name: (empty input field)
- Event Description: (large text area)
- Event Type: Public
- Location: (large text area)
- Date: dd/mm/yyyy (with a calendar icon)
- Time: (dropdown menu)
- Temporary Expiration Date: 2024-12-06

Below the date field, there's a note: "If time selection is empty it means that the selected date is fully booked." Another note below the expiration date says: "Expiration date is only applicable if you choose to pay later." At the bottom right is a blue 'Submit' button.

Figure 7.5 Book an Appointment for an Event

The screenshot shows a user interface for purchasing a ticket. On the left, there is a sidebar with a logo at the top, followed by a 'Dashboard' link. Below it, under 'CONTENTS', are several items: 'Services' (with a dropdown arrow), 'Join GYOC', 'History', 'Learning Materials', 'Your Calendar', and 'Notifications'. The main area is titled 'Purchase a Ticket'. It contains the following fields:

First Name:	GYOCC
Last Name:	Superuser
Mobile Number:	+63 9081600631
Email:	calvindrakkteszt9@gmail.com
Event:	-----
Select Ticket:	Select an event first

A note below the ticket selection field states: "If there are no ticket types available it means the tickets for that event is **sold out**". At the bottom right is a blue 'Submit' button.

Figure 7.6 Purchase a Ticket

The image shows a screenshot of the GYOCC website's dashboard. On the left, there is a sidebar with a logo, a dashboard icon, and a menu titled "CONTENTS" containing links to "Services", "Join GYOCC" (which is highlighted in blue), "History", "Learning Materials", "Your Calendar", and "Notifications". The main content area features two large cards. The first card, titled "Audition for our Orchestra", shows a group photo of people gathered in a room with a whiteboard and a banner. It includes a green "Orchestra" button and a right-pointing arrow. The second card, titled "Audition for our Choir", shows a group of people in red choir gowns standing in front of a wall with framed portraits. It includes a green "Choir" button and a right-pointing arrow.

Figure 7.7 Join GYOCC by Submitting an Audition

The screenshot shows a web-based application interface. At the top left is a logo with a stylized orange and green leaf-like icon. To its right are three small icons: a bell with a '0', a user profile, and a downward arrow. On the far right is a vertical dark sidebar.

The main area has a light gray header bar with the text "Audition for an Orchestra". Below this is a form with the following fields:

- First Name: GYOCC
- Last Name: Superuser
- Mobile Number: +639081600631
- Email: calvindrakketeszt9@gmail.com
- Audition Type: Orchestra
- Address: (A large empty text area)
- Audition Expiration: 2024-12-06 02:19:25

At the bottom right of the form is a blue "Submit" button.

Figure 7.8 Audition for an Orchestra

The screenshot shows a web-based application interface. On the left, there is a sidebar with a navigation menu. The menu items include 'Dashboard', 'CONTENTS' (which is expanded), 'Services', 'Join GY OCC' (which is selected and highlighted in blue), 'History', 'Learning Materials', 'Your Calendar', and 'Notifications'. At the top right, there are icons for notifications (0) and user profile.

The main content area is titled 'Audition for Choir'. It contains several input fields:

- First Name: GY OCC
- Last Name: Superuser
- Mobile Number: +639081600631
- Email: calvindrakkeset9@gmail.com
- Audition Type: Choir
- Address: (A large text input field)

Below these fields is a section for 'Video Entry':

- 'Choose File' button: No file chosen
- Accepted video formats: .mp4 | .avi | .mov
- Description: You can attach a video file here showcasing your entry point to your audition, there is also a slight chance that you won't need to do a live audition and get accepted only with the video you attached.

At the bottom of the form, there is a date field labeled 'Audition Expiration: 2024-12-06 02:19:25' and a blue 'Submit' button.

Figure 7.9 Audition for a Choir

The screenshot shows the History page of a software application, likely a management system for events or bookings. The left sidebar contains navigation links: Dashboard, CONTENTS (Services, Join GYOC, History), Learning Materials, Your Calendar, and Notifications. The main content area is divided into several sections:

- Booking History:** A table showing bookings with columns: BOOK NAME, SCHEDULE STATUS, MEETING DATE, and PAYMENT STATUS. Entries include "Birthday Ni JMP" (Unused, Available), "Capstone Title Defense" (G, Oct. 22, 2024, Available), "Event NBJQ48ph" (GA, Nov. 24, 2024, Used), and "Event NBJQ48ph" (GA, Nov. 24, 2024, Used).
- Lesson History:** A table showing lessons with columns: INSTRUCTOR, INSTRUMENT, START DATE, END DATE, and STATUS. An entry for "GYOCC Superuser" (Violin) is listed as "Booking Has Passed" from Oct. 20, 2024, 4 P.M. to Oct. 24, 2024, 9 A.M., marked as "NOT PAID" and "PENDING".
- Audition History:** A table showing auditions with columns: AUDITION TYPE, AUDITION DATE, and AUDITION STATUS. Three entries for "Choir" are listed: "To Be Set By GYOC" (Pending), "To Be Set By GYOC" (Pending), and "Aug. 22, 2024, 6:24 P.M." (Accepted).
- Private Event Payment History:** A table showing payments with columns: EVENT, PAID, TOTAL, PAYMENT DATE, RECEIPT, and PAY BALANCE. A message at the bottom states "NO RECENT PAYMENTS FOR PRIVATE EVENTS".

Figure 7.10 History Page

The screenshot shows a web-based learning platform interface. On the left is a sidebar with a logo, a dashboard link, contents, services, join, history, learning materials (which is selected and highlighted in blue), calendar, and notifications. The main content area has a title "Learning Materials from GYOCC" and a subtitle "Access available learning materials from our instructors at GYOCC". A search bar contains the text "Trumpet". Below this, a section titled "Available articles for Trumpet" lists one article: "The Honks of the Trumpet" by Calvin Drakke Rulete, with the subtitle "Is mayonnaise an instrument?". There is also a thumbnail image of a trumpet.

Learning Materials from GYOCC

Access available learning materials from our instructors at GYOCC

Trumpet

Available articles for Trumpet

The trumpet

The Honks of the Trumpet
Calvin Drakke Rulete
Is mayonnaise an instrument?

Figure 7.11 Learning Materials Page

The screenshot shows the 'Your Calendar' page from a digital platform. At the top, there's a navigation bar with icons for Dashboard, Services, Join GY OCC, History, Learning Materials, Your Calendar (which is highlighted in blue), and Notifications. To the right of the navigation is a user profile icon with a notification count of 0.

The main area is titled 'Your Calendar' and displays a monthly calendar for November 2024. The calendar grid shows days from Sunday to Saturday. Several events are listed with icons and descriptions:

- Sunday, Nov 27: ● 1a Father som
- Tuesday, Nov 29: ● 1a Send today
- Friday, Nov 30: ● 4a Health role
- Wednesday, Dec 4: ● 1a American d
- Wednesday, Dec 4: ● 3a Break part
- Wednesday, Dec 4: 4:36a Event tBYhPzoc (represented by a long orange bar)
- Wednesday, Dec 4: ● 3a Form return 2:29p Event nBJQ48ph (represented by a long blue bar)
- Wednesday, Dec 4: 2:29p Event nBJQ48ph (represented by a short blue bar)
- Friday, Dec 6: ● 3a Data analy...

At the bottom of the calendar, there are buttons for 'month', 'week', 'day', and 'list' view modes.

Figure 7.12 Your Calendar Page

The screenshot shows a notifications page with a sidebar on the left containing links like Dashboard, Services, Join GYOC, History, Learning Materials, Your Calendar, and Notifications. The Notifications section is highlighted. The main content area lists nine events, each with a checkbox and two buttons at the top right: 'Mark as Read' (green) and 'Mark as Unread' (blue). The events are:

- Event** A new event was announced: Evans, Blake and Davies
- Event** A new event was announced: Zavala-James
- Event** A new event was announced: Bullock and Sons
- Event** A new event was announced: Fisher PLC
- Event** A new event was announced: Martin LLC
- Event** A new event was announced: Schmitt, Holden and Porter
- Event** A new event was announced: Mitchell-Mata
- Event** A new event was announced: Patterson Inc
- Event** A new event was announced: Romero, Avery and Wolfe

Figure 7.13 Notifications Page

The screenshot shows a user profile page. At the top right, there are notification and user icons. On the left, a sidebar titled "CONTENTS" lists "Services", "Join GY OCC", "History", "Learning Materials", "Your Calendar", and "Notifications". The main area features a profile card with a placeholder image, the text "GYOCC Superuser" and "Superuser", and a "Member Since" timestamp of "Aug. 12, 2024, 12:37 p.m.". Below this are three "Other Info" sections. To the right is a detailed profile form with fields for First Name (GYOCC), Middle Name (None), Last Name (Superuser), Email (calvindrakteset9@gmail.com), and Phone (+639081600631). There are "Edit" and "Change Password" buttons at the bottom of this form.

First Name	GYOCC
Middle Name	None
Last Name	Superuser
Email	calvindrakteset9@gmail.com
Phone	+639081600631

Member Since Aug. 12, 2024, 12:37 p.m.

Other Info Info

Other Info Info

Other Info Info

Edit Change Password

Figure 7.14 Profile Page

The screenshot shows the Staff Dashboard of the Acchord application. The left sidebar contains navigation links for Main, Message, Authentication and Authorization, and Event categories. The main content area is divided into several sections:

- Main**: Includes Attachments, Members, Notifications, Posts, and Site Configurations, each with Add and Change buttons.
- Message**: Includes Messages, with an Add button.
- Authentication and Authorization**: Includes Groups and Analytics, each with Add and Change buttons.
- Event**: Includes Books, Events, and Calendar, each with Add, Change, and View buttons.
- Resource**: Includes Item Histories and Items, each with Add and Change buttons.
- Audition**: Includes Auditions, with an Add button.
- Payment**: Includes Payment Auditions, Payment Bookings, Payment Events, Payment Lessons, Payment Sessions, and Payment Tickets, each with Add and Change buttons.
- Feedback**: Includes Feedbacks, with a View button.

At the top, there are search bars for Groups, Books, Events, Auditions, and Lessons. On the right, a "Recent actions" section lists recent events with status icons, timestamps, and descriptions.

Figure 7.15 Staff Dashboard

Acchord

GYOCC Superuser

- Dashboard
- Main
- Attachments
- Members
- Notifications
- Posts
- Site Configurations
- Message
- Messages
- Authentication and Authorization
- Groups
- Analytics**
- Event
- Books
- Events
- Calendar

Home Search Groups... Search Books... Search Events... Search Auditions... Search Lessons... 

General Analytics

Newest Audition Application for Orchestra
Test Testing User

Newest Audition Application for Choir
Princess Lynniel Del Rio Acosta

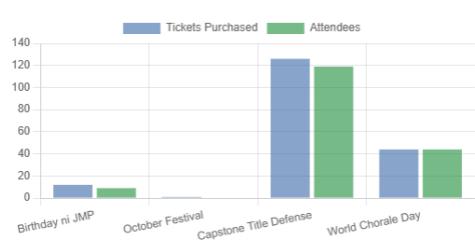
Attendance count of Current Event
No event currently ongoing.

Seat Availabilities of Current Event
No event currently ongoing.

Toggle Charts

All Available Data

All Ticket Sales Ranking
Click on an event to view analytics
Tickets Purchased Attendees



Event	Tickets Purchased	Attendees
Birthday ni JMP	~10	~10
October Festival	~10	~10
Capstone Title Defense	~125	~120
World Chorale Day	~45	~45

All Lesson Enrollments Ranking
Lesson Enrollees Lessons Attendance



Instrument/Cat	Lesson Enrollees	Lessons Attendance
Flute	0	0
Drums	0	0
Violin	8	8
Violin	2	2
Violin - Finished	1	1
Violin - Upcoming	1	1
Sample - Scanned	1	1
Sample - Ongoing	1	1
Guitar	1	1
Guitar	1	1
Violin - Ongoing	1	1
Flute	1	1

All-Time Most Popular Instruments



Instrument	Total Bookings
Violin	5
Guitar	4
Trumpet	2
Flute	1
Drums	1

Yearly Bookings
Total Bookings



Instrument	Total Bookings
Violin	10
Guitar	8
Trumpet	4
Flute	2
Drums	1

Figure 7.16 Reports & Analytics Page

The screenshot shows the 'Site Configurations' page in the Acchord application. The left sidebar contains a navigation menu with items like Home, Dashboard, Main (Attachments, Members, Notifications, Posts), Site Configurations (highlighted in blue), Message (Messages), Authentication and Authorization (Groups, Analytics), Event (Books, Events, Calendar), and a user profile for GYOCC Superuser.

The main content area displays the 'Booking' tab of the Site Configurations. It includes the following configuration options:

- Use SMS Notificator for Booking Date Changes**: A checked checkbox with a note: "Caution: 1 sms operation is equal to 0.17€".
- Temporary Booking Expiration Days ***: A text input field containing "7".
- Booking Fee ***: A text input field containing "250.00".
- Maximum Bookings per Day ***: A text input field containing "5".
- Minimum Hour Gap per Booking (in Hours) ***: A text input field containing "1".
- Start of booking hours ***: A text input field containing "09:00:00" with a time picker icon below it.
- End of booking hours ***: A text input field containing "21:00:00" with a time picker icon below it.

On the right side, there are three action buttons: **Save** (green), **Save and continue editing** (blue), and **History** (grey).

Figure 7.17 Site Configurations Page

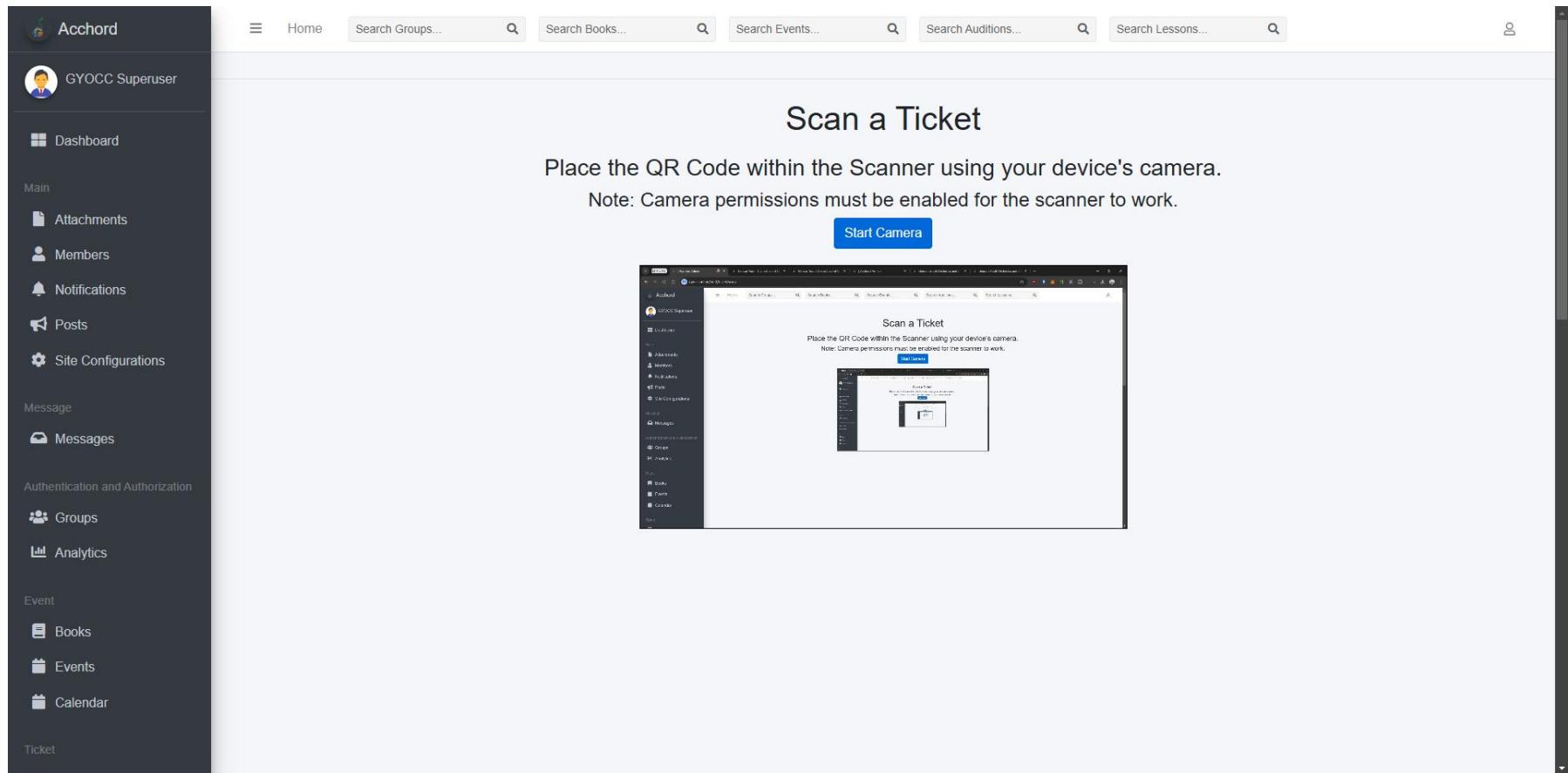


Figure 7.18 Scan a Ticket

The screenshot shows the Acchord application interface. On the left is a dark sidebar with a user profile for 'GYOCC Superuser' and a navigation menu. The main content area has a title 'Scan a QR Code for Student Attendance' and instructions: 'Place the QR Code within the Scanner using your device's camera.' It also includes a note: 'Note: Camera permissions must be enabled for the scanner to work.' A blue button labeled 'Start Camera' is visible. Below the note is a small preview window showing a QR code.

- Home
- Search Groups...
- Search Books...
- Search Events...
- Search Auditions...
- Search Lessons...

Scan a QR Code for Student Attendance

Place the QR Code within the Scanner using your device's camera.

Note: Camera permissions must be enabled for the scanner to work.

Start Camera

Acchord

GYOCC Superuser

Dashboard

Main

- Attachments
- Members
- Notifications
- Posts
- Site Configurations

Message

- Messages

Authentication and Authorization

- Groups
- Analytics

Event

- Books
- Events
- Calendar

Ticket

Scan a QR Code for Student Attendance

Place the QR Code within the Scanner using your device's camera.

Note: Camera permissions must be enabled for the scanner to work.

Start Camera

Figure 7.19 Scan for Lesson Attendance

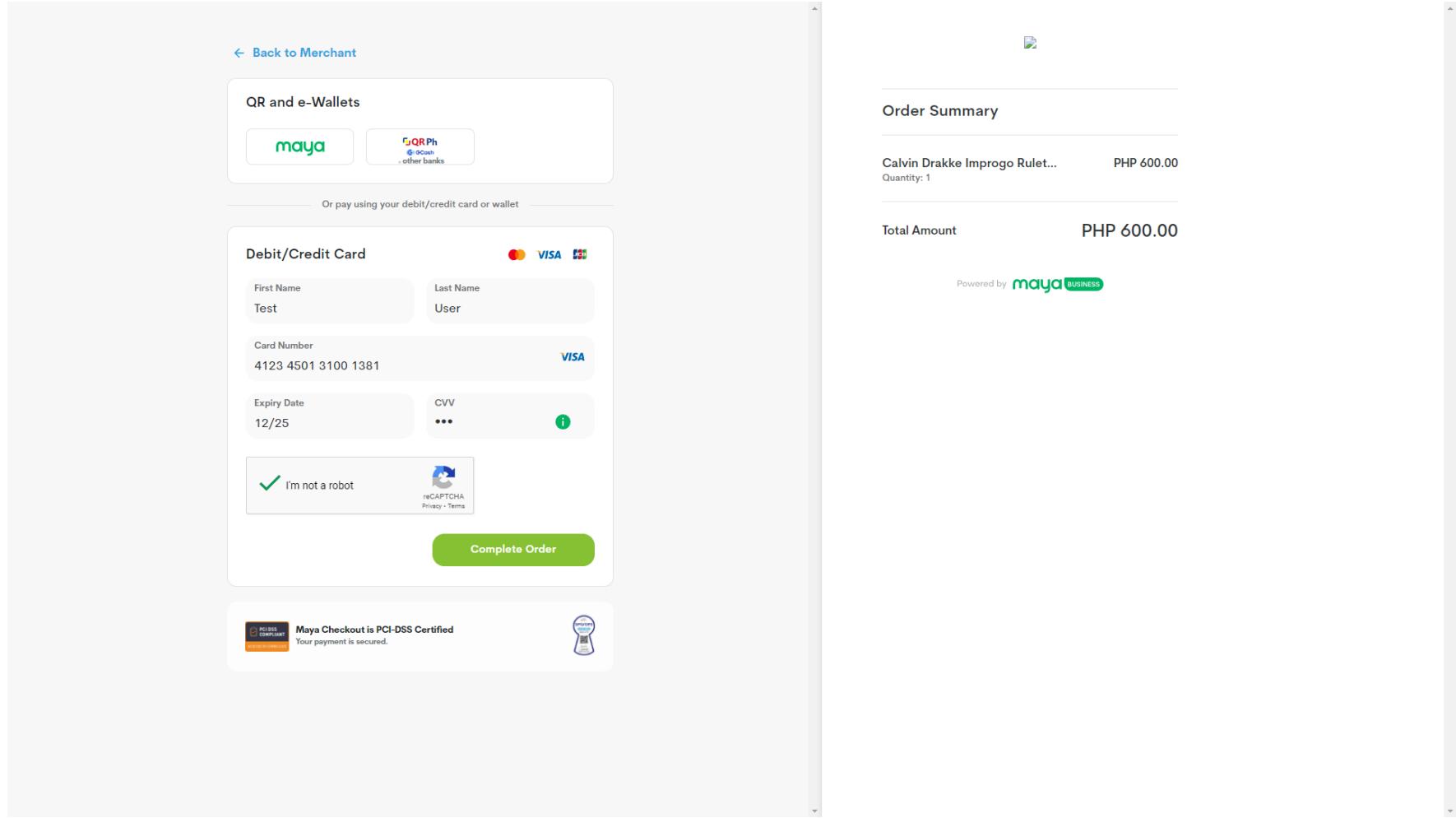


Figure 7.20 Payment Page



The Gensan Youth Orchestra
And Choir Corporation

Select Authentication Method

[Authenticate via Email](#)

[Authenticate via SMS](#)



[Work With Us](#)



[Purchase Tickets](#)



[Join GYOCC](#)

Figure 7.21 2FA Verification Page



The Gensan Youth Orchestra And Choir Corporation

Enter the 6-digit code sent to your selected authentication type

[Submit](#)



Figure 7.22 2FA OTP Input Page

Operating System	x64 based Windows or Linux
CPU	A8 7650K or Athlon 200GE
RAM	8GB
Storage	SATA SSD 500GB

Table 1. Personal Computer minimum specifications

Operating System	Android 8.0 Oreo or iOS 10.0
RAM	4GB

Table 2. Smart Phone minimum specifications

QUESTIONS	MEAN	INTERPRETATION
SECTION 1: UI		
1. The dashboard is easy to navigate.	4.47	Strongly Agree
2. The information displayed in the dashboard are helpful.	4.59	Strongly Agree
3. The audition section was easy to navigate.	4.88	Strongly Agree
4. Clicking on an item correctly displays my ticket, lesson and audition information.	4.82	Strongly Agree
5. The system correctly displays my ticket, booking, lesson, and audition history.	4.82	Strongly Agree
6. The learning materials are easy to access and	4.65	Strongly Agree

read.		
7. The learning materials and its content were displayed correctly.	4.71	Strongly Agree
8. Viewing upcoming events in the calendar was easy.	4.18	Agree
9. The learning materials provided by GYOCC were useful.	4.71	Strongly Agree
10. Accessing my history (tickets, bookings, lessons, auditions) was easy.	4.82	Strongly Agree
OVERALL MEAN	4.67	Strongly Agree

Table 3. The results of evaluation done by the users regarding the UI. The survey results indicate that users find the system's user interface highly user-friendly, with strong agreement across most aspects. Navigation within the dashboard and audition sections is easy, and information such as tickets, bookings, lessons, and history is accurately displayed and accessible. Learning materials are well-presented, easy to access, and considered useful. While viewing events in the calendar received slightly lower agreement (4.18), it is still rated positively. Overall, with an average score of 4.67, users strongly agree that the system is intuitive, efficient, and meets their needs effectively.

QUESTIONS	MEAN	INTERPRETATION
SECTION 2: Process		
1. Booking an appointment was straightforward.	4.76	Strongly Agree
2. I was satisfied with the booking process.	4.59	Strongly Agree
3. Purchasing a ticket for an event was easy and straightforward.	4.71	Strongly Agree
4. The automation for appointment confirmation works as expected.	4.59	Strongly Agree
5. The automated ticket confirmation and generation worked as expected.	4.88	Strongly Agree
6. The ticket purchasing system follows the correct steps from selection to payment	4.82	Strongly Agree
7. The audition process follows a correct procedure for submission and scheduling.	4.71	Strongly Agree
8. The automated audition confirmation and notification were implemented correctly.	4.35	Strongly Agree
9. I was satisfied with the joining of a lesson session process.	4.88	Strongly Agree
10. The system follows a correct procedure of joining a lesson session.	4.88	Strongly Agree
OVERALL MEAN	4.72	Strongly Agree

Table 4. The results of evaluation done by the users regarding the process. The survey

results for the Process section show that users strongly agree that the system's processes are well-designed, efficient, and easy to follow, with an impressive overall mean of 4.72. Booking appointments and purchasing tickets are straightforward, and the automated confirmation systems for appointments, tickets, and auditions work as expected. The system follows correct procedures for scheduling auditions and joining lesson sessions, with high satisfaction in these areas (4.88). While the audition confirmation process received slightly lower agreement (4.35), it remains positive.

QUESTIONS	MEAN	INTERPRETATION
SECTION 3: Conclusion		
1. I was satisfied with the ticket purchase experience.	4.88	Strongly Agree
2. I was satisfied with the audition process.	4.82	Strongly Agree
3. Joining in a lesson session was easy.	4.76	Strongly Agree
4. I was satisfied with the ticket purchase option through the calendar.	4.53	Strongly Agree
5. I had a positive experience using the website.	4.82	Strongly Agree
6. I am likely to recommend the website to others.	4.76	Strongly Agree
OVERALL MEAN	4.76	Strongly Agree

Table 5. The results of evaluation done by the users regarding the overall system. The survey results for the Conclusion section reveal strong user satisfaction, with an overall mean of 4.76 indicating that users strongly agree with the statements. Ticket purchasing, the audition process, and joining lesson sessions were smooth and satisfactory experiences. Users also appreciated the ticket purchase option via the calendar (4.53) and reported

having a positive overall experience with the website (4.82). Additionally, the high likelihood of recommending the website (4.76) underscores the platform's effectiveness and user-friendliness.

APPENDIX I. CERTIFICATES

A Web Management System for Gensan Youth Orchestra and Choir Corporation.docx, new.edited

by DO NOT DELETE OTHER FILES OR ELSE WARRANTY WILL BE VOID

General metrics

91,949	13,141	812	52 min 33 sec	1 hr 41 min
characters	words	sentences	reading time	speaking time

Score



Writing Issues

317	14	303
Issues left		Critical

This text scores better than 92% of all texts checked by Grammarly

Plagiarism



This text seems 100% original. Grammarly found no matching text on the Internet or in ProQuest's databases.



November 30, 2024

CERTIFICATION

This certification confirms that an expert in the English language edited the manuscript detailed below. The undersigned grammarian guarantees the manuscript's language accuracy without altering the authors' intent and content of the undergraduate research paper.

A Web Management System for Gensan Youth Orchestra and Choir Corporation

By

**Andy Jay Borre
John Michael S. Panizales
Owen Lourd L. Ravina
Calvin Drakke I. Rulete**

October 2024


SUNSHINE M. DIAGAN, LPT
Grammarians

(083) 554-3038 J. Catolico Avenue, General Santos City

www.sti.edu



STATISTICIAN'S CERTIFICATE

To whom it may concern

This is to certify that the undersigned has review the undergraduate thesis of **Andy Jay Borre, John Michael S. Panizales, Owen Lourd L. Ravina, and Calvin Drakke I. Rulete** entitled: "**A Web Management System for Gensan Youth Orchestra and Choir Corporation**" in terms of statistical analysis.

This certification issued upon the request of **Andy Jay Borre, John Michael S. Panizales, Owen Lourd L. Ravina, and Calvin Drakke I. Rulete** for whatever purpose this may serve them best.

Given this 8th of December 7, 2024.

A handwritten signature in black ink, appearing to read "Handro Cesar B. Arlegui".

Handro Cesar B. Arlegui, LPT
Capstone Statistician



December 4, 2024

CERTIFICATION

This is to certify that the team from STI College of General Santos has successfully completed their capstone project entitled below for Gensan Youth Orchestra and Choir Corporation during the school year 2024 – 2025.

The project team, consisting of Rulete, Calvin Drakke, Ravina, Owen Lourd, Panizales, John Michael, and Borre, Andy Jay has demonstrated exceptional professionalism, technical skill, and commitment in developing a solution that meets the specific requirements and objectives outlined by our organization.

A Web Management System for Gensan Youth Orchestra and Choir Corporation

By

**Andy Jay Borre
John Michael S. Panizales
Owen Lourd L. Ravina
Calvin Drakke I. Rulete**

October 2024

Acknowledgement by GYOCC:


Mr. Fernan Globen Talonding
Chief Executive Officer
Gensan Youth Orchestra and Choir Corporation

(083) 554-3038 J. Catolico Avenue, General Santos City

www.sti.edu

APPENDIX J. PERSONAL TECHNICAL VITAE

Curriculum Vitae of
ANDY JAY BORRE
042 Manansala Street, Dadiangas North, General Santos City, South Cotabato
andyborre26@gmail.com
+639488864116



EDUCATIONAL BACKGROUND		
Level	Inclusive Dates	Name of school/ Institution
Tertiary	2021-Present	STI College General Santos City
High School	2014-2019	Dadiangas North High School
Elementary	2009-2014	Mintal Elementary School

PROFESSIONAL OR VOLUNTEER EXPERIENCE

Inclusive Dates	Nature of Experience/ Job Title	Name and Address of Company or Organization
November 2023	Volunteer	Batang Star Philippines
November 2021	Volunteer	STI College General Santos City

Listed in reverse chronological order (most recent first).

AFFILIATIONS

Inclusive Dates	Name of Organization	Position
November 2023	Batang Star	Member
November 2021	KKDAT	PIO 2

Listed in reverse chronological order (most recent first).

SKILLS

SKILLS	Level of Competency	Date Acquired
Computer Encoder	Advanced	2016-Present
Annotator	Beginner	2024

TRAININGS, SEMINARS, OR WORKSHOPS ATTENDED

Inclusive Dates	Title of Training, Seminar, or Workshop
November 2023	Leadership Engagement and Program (LEAP)

Listed in reverse chronological order (most recent first).

Curriculum Vitae of
CALVIN DRAKKE I. RULETE
Purok Maliwanag, Calumpang, General Santos City, South Cotabato
calvindrakkez9@gmail.com
+639081600631



EDUCATIONAL BACKGROUND

Level	Inclusive Dates	Name of school/ Institution
Tertiary	2021-Present	STI College General Santos
Vocational/Technical	2019-2021	General Santos City National High School
High School	2015-2019	General Santos City National High School
Elementary	2009-2013	Brentwood Academy of Dasmarinas Incorporated Upper Calumpang Seventh Day Adventist Elementary School

PROFESSIONAL OR VOLUNTEER EXPERIENCE

Inclusive Dates	Nature of Experience/ Job Title	Name and Address of Company or Organization
2017	Class President	Upper Calumpang Seventh Day Adventist Elementary School

Listed in reverse chronological order (most recent first).

SKILLS

SKILLS	Level of Competency	Date Acquired
Programming in C#	Intermediate	2020-Present
Programming in Python	Advanced	2016-Present
Photoshop Experience	Intermediate	2021-2023
Premiere Pro Experience	Beginner	2022-2023

TRAININGS, SEMINARS, OR WORKSHOPS ATTENDED

Inclusive Dates	Title of Training, Seminar, or Workshop
2018-2019	Freecodecamp online Python press-conference.

Listed in reverse chronological order (most recent first).



Curriculum Vitae of
OWEN LOURD L. RAVINA
3rd Road Sta. Cruz, Barangay Cannery Site, Polomolok, South Cotabato
Owen.mael14@gmail.com
09994950948

EDUCATIONAL BACKGROUND

Level	Inclusive Dates	Name of school/ Institution
Tertiary	2021-Present	STI General Santos
Vocational/Technical	2019-2021	Polomolok National High School
High School	2015-2019	Polomolok National High School
Elementary	2009-2015	Perfecto B. Salada Elementary School

PROFESSIONAL OR VOLUNTEER EXPERIENCE

Inclusive Dates	Nature of Experience/ Job Title	Name and Address of Company or Organization
2023-Present	Freelancing/Floor Planning	Upwork
2019-2021	NDEP/Peer Educator Sex	ADAPT-H PNHS

Listed in reverse chronological order (most recent first).

SKILLS

SKILLS	Level of Competency	Date Acquired
Floor Planning	Intermediate	2022-Present
Programming in Python	Intermediate	2021-Present
Server Maintenance and Set	Advance	2019-Present

TRAININGS, SEMINARS, OR WORKSHOPS ATTENDED

Inclusive Dates	Title of Training, Seminar, or Workshop
2021	DICT Mach Programming Seminar

Listed in reverse chronological order (most recent first).



Curriculum Vitae of
John Michael S. Panizales
Purok Malok, Barangay Labangal, General Santos City, South Cotabato
michaelpanizales1203@gmail.com
+639518196687

EDUCATIONAL BACKGROUND

Level	Inclusive Dates	Name of school/ Institution
Tertiary	2021-Present	STI College General Santos
Senior High School	2018-2019	ACLC College of General Santos
High School	2014-2018	General Santos City National Highschool
Elementary	2008-2014	Saavedra Saway Central Elementary

PROFESSIONAL OR VOLUNTEER EXPERIENCE

Inclusive Dates	Nature of Experience/ Job Title	Name and Address of Company or Organization
2019-2020	On-The-Job Training	Mabuhay IT Park Ced Avenue, National Highway, General Santos City, South Cotabato

Listed in reverse chronological order (most recent first).

SKILLS

SKILLS	Level of Competency	Date Acquired
HTML/CSS	Beginner	2024
MS SQL	Beginner	2024
C#	Beginner	2019
Oracle SQL	Beginner	2019
Java	Beginner	2018
Computer Hardware Servicing	Intermediate	2016

TRAININGS, SEMINARS, OR WORKSHOPS ATTENDED

Inclusive Dates	Title of Training, Seminar, or Workshop
2019	Fire Prevention and Safety Training
2018	Gensan ICT Summit 2018

Listed in reverse chronological order (most recent first).

APPENDIX K. RELEVANT SOURCE CODE

acchord > views.py

```
from decimal import Decimal
from urllib.parse import urlencode
import uuid
from django.conf import settings
from django.http import HttpResponseRedirect, Http404
from django.shortcuts import get_object_or_404, redirect, render
from django.contrib.auth.decorators import permission_required
from django.contrib.auth.decorators import login_required
from django.contrib.admin.sites import site
from django.urls import reverse
import requests
from audition.models import Audition
from feedback.models import Feedback
from main.models import Member, SiteConfig
from member2fa.forms import Auth2FAForm, ScanAuth2FAForm
from member2fa.models import SMS2FA
from payment.models import PaymentEvent, PaymentLesson, PaymentSession
from ticket.models import Ticket, TicketUsage
from django.utils import timezone
from event.models import Book, Event
import json, datetime, random
from django.http import JsonResponse
from lesson.models import Instrument, Lesson, LessonAttendance
from django.contrib.admin.views.decorators import
staff_member_required
from django.db.models import Count
from django.db.models import Q, F, Sum, Count
from django.contrib import messages
from django.http import Http404
from django.db.models.functions import ExtractMonth, ExtractYear
from django.db.models import Avg
from django.utils import timezone
from django.utils.dateparse import parse_date
from django.utils.timezone import make_aware
from collections import defaultdict
from utilities.maya.paymentdata import PaymentData, PaymentBuyer,
PaymentContact, PaymentItem, PaymentItemAmount, PaymentRedirectUrl,
PaymentTotalAmount
import csv

@login_required()
@staff_member_required()
def get_event_id(request, name):
```

```

event = get_object_or_404(Event, name=name)
return JsonResponse({'event_id': event.id})

@login_required()
@staff_member_required()
def generate_booking_csv(request):
    if request.method == 'GET':
        start_date_str = request.GET.get('start_date', None)
        end_date_str = request.GET.get('end_date', None)

        start_date = parse_date(start_date_str) if start_date_str else
None
        end_date = parse_date(end_date_str) if end_date_str else None

        response = HttpResponse(content_type='text/csv')
        response['Content-Disposition'] = f'attachment;
filename="booking_analytics_{start_date.strftime("%Y%m%d")}_{end_date.
strftime("%Y%m%d")}.csv"'

        writer = csv.writer(response)

        # Basic Export Information
        writer.writerow(['Generated By', request.user])
        writer.writerow(['Date Generated', timezone.now().strftime("%B
%d, %Y, %I:%M %p")])
        writer.writerow(['Start Date', start_date])
        writer.writerow(['End Date', end_date])
        writer.writerow([])

        # Get all bookings within date range
bookings = Book.objects.filter(
            meet_date__gte=start_date,
            meet_date__lte=end_date
).order_by('meet_date')

        # Total bookings
total_bookings = bookings.count()
writer.writerow(['Total Bookings', total_bookings])
writer.writerow([])

        # Calculate most booked month and average bookings per month
monthly_bookings = defaultdict(list)
for booking in bookings:
    month_key = booking.meet_date.strftime("%Y-%m")
    monthly_bookings[month_key].append(booking)

```

```

        # Find most booked month
        if monthly_bookings:
            most_booked_month = max(monthly_bookings.items(),
key=lambda x: len(x[1]))
            most_booked_month_date =
datetime.datetime.strptime(most_booked_month[0], "%Y-%m")
            writer.writerow(['Most Booked Month',
most_booked_month_date.strftime("%B %Y")])
            writer.writerow(['Bookings in Most Booked Month',
len(most_booked_month[1])])
            writer.writerow([])

            # Calculate average bookings per month
            total_months = len(monthly_bookings)
            avg_bookings = total_bookings / total_months if
total_months > 0 else 0
            writer.writerow(['Average Bookings per Month',
f"{avg_bookings:.2f}'])
            writer.writerow([])

            # Write monthly booking details
            writer.writerow(['Monthly Booking Details'])
            writer.writerow([])

            for month, month_bookings in
sorted(monthly_bookings.items()):
                month_date = datetime.datetime.strptime(month, "%Y-
%m")
                writer.writerow([f"Bookings for
{month_date.strftime('%B %Y')}"])
                writer.writerow(['Booker', 'Date', 'Booking Name'])

                for booking in sorted(month_bookings, key=lambda x:
x.meet_date):
                    writer.writerow([
                        booking.booker.get_full_name() if
booking.booker else 'No Booker',
                        booking.meet_date.strftime("%B %d, %Y, %I:%M
%p"),
                        booking.name
                    ])
                writer.writerow([])

        return response
    
```

```

@login_required()
@staff_member_required()
def analytics_specific_date_range_private_event(request):
    if request.method == 'GET':
        start_date_str = request.GET.get('start_date', None)
        end_date_str = request.GET.get('end_date', None)

        start_date = parse_date(start_date_str) if start_date_str else
None
        end_date = parse_date(end_date_str) if end_date_str else None

        if start_date and end_date:
            paymentEvents =
PaymentEvent.objects.filter(date_paid__range=(start_date, end_date),
event__isnull=False)

            if not paymentEvents or paymentEvents.count() <= 0:
                messages.add_message(request, messages.ERROR,
"Analytics page generation failed: selected specific date range has no
private event payment data.")
                return redirect('base_analytics_page')

            payment_event_list = [
{
                "paymentEventId": payment_event.id,
                "paymentAmount":
float(payment_event.payment_amount),
                "event": payment_event.event.name if
payment_event.event else None,
                "date_paid": payment_event.date_paid.strftime("%Y-
%m-%d")
}
            for payment_event in paymentEvents
]

        if 'download_csv' in request.GET:
            return generate_payment_events_csv(
                request.user,
                paymentEvents
            )

        context = {
            'available_apps': site.get_app_list(request),
            'user': request.user,

```

```

        'paymentData': json.dumps(payment_event_list)
    }

    return render(request,
'admin/admin_analytics_specific_private_event.html', context)
else:
    return render(request, '404.html')

def generate_payment_events_csv(user, payment_events):
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment;
filename="payment_events.csv"'

    writer = csv.writer(response)

    # Basic Information
    writer.writerow(['Exported By', user.username])
    writer.writerow(['Time Generated', timezone.now().strftime("%B %d,
%Y, %I:%M %p")])
    writer.writerow([])

    # Totals
    total_payment_events = payment_events.count()
    total_sales_sum =
    payment_events.aggregate(total_sum=Sum('payment_amount'))['total_sum']
    or 0
    writer.writerow(['Total Number of Payment Events',
total_payment_events])
    writer.writerow(['Total Sum of Sales from All Payment Events',
f"PHP {total_sales_sum:.2f}"])
    writer.writerow([])

    # Payment Events Table Header
    writer.writerow(['Reference Number', 'Payment Amount', 'Total
Amount', 'Date Paid', 'Remarks'])

    # Payment Events Data
    for event in payment_events:
        remarks = "Fully Paid" if event.payment_amount ==
event.total_amount else "Not Fully Paid"
        writer.writerow([
            event.reference_number,
            f"PHP {event.payment_amount:.2f}",
            f"PHP {event.total_amount:.2f}",
            event.date_paid.strftime("%B %d, %Y, %I:%M %p"),
            remarks
        ])

```

```

        remarks
    ])

    return response

@login_required()
@staff_member_required()
def analytics_specific_date_range(request):
    if request.method == 'GET':
        start_date_str = request.GET.get('start_date', None)
        end_date_str = request.GET.get('end_date', None)

        start_date = parse_date(start_date_str) if start_date_str else
None
        end_date = parse_date(end_date_str) if end_date_str else None

        eventData = {}
        availableTicketsPerEvent = {}
        ticketPricesPerEvent = {}
        total_unsold = 0
        total_tickets = 0

        ticket_types = ["VIP", "Gold", "Silver", "Gen Ad"]

        if start_date and end_date:
            events = Event.objects.filter(
                start_date__lte=end_date,
                end_date__gte=start_date,
                event_type=Event.EventType.PUBLIC
            )

            for event in events:
                # eventData
                eventData[event.name] = {}
                eventData[event.name]['start_date'] =
event.start_date.strftime("%Y-%m-%d")
                eventData[event.name]['end_date'] =
event.end_date.strftime("%Y-%m-%d")
                eventData[event.name]['data'] = []

                tickets = Ticket.objects.filter(event=event)

                start_date = event.start_date.date()
                end_date = event.end_date.date()
                event_days = (end_date - start_date).days + 1

```

```

        days = [start_date + timezone.timedelta(days=i) for i
in range(event_days)]

        for ticket in tickets:
            usage_dates =
TicketUsage.objects.filter(ticket=ticket).values_list('usage_date',
flat=True)
            attended = {f"day{i+1}": "☑" if day in
usage_dates else "✗" for i, day in enumerate(days)}

            ticket_type = ticket.get_ticket_type_display()
            if ticket_type == 'VIP':
                ticket_type = 'VIP'
            if ticket_type == 'GOLD':
                ticket_type = 'Gold'
            if ticket_type == 'SILVER':
                ticket_type = 'Silver'
            if ticket_type == 'GENERAL ADMISSION':
                ticket_type = 'Gen Ad'

            eventData[event.name]['data'].append({
                "name": f"{ticket.first_name}"
{ticket.last_name}",
                "ticketType": ticket_type,
                "attended": attended
            })

            if eventData[event.name]['data']:
                total_unsold += event.vip_seats
                total_unsold += event.gold_seats
                total_unsold += event.silver_seats
                total_unsold += event.general_admission_seats

                total_tickets +=
Ticket.objects.filter(event=event).count()

                availableTicketsPerEvent[event.name] = {}
                ticketPricesPerEvent[event.name] = {}

                for tick_type in ticket_types:
                    if tick_type == "VIP":
                        availableTicketsPerEvent[event.name][tick_
type] = event.vip_seats
                        ticketPricesPerEvent[event.name][tick_type]
] = float(event.vip_ticket_cost)

```

```

                elif tick_type == "Gold":
                    availableTicketsPerEvent[event.name][tick_
type] = event.gold_seats
                    ticketPricesPerEvent[event.name][tick_type]
= float(event.gold_ticket_cost)
                elif tick_type == "Silver":
                    availableTicketsPerEvent[event.name][tick_
type] = event.silver_seats
                    ticketPricesPerEvent[event.name][tick_type]
= float(event.silver_ticket_cost)
                elif tick_type == "Gen Ad":
                    availableTicketsPerEvent[event.name][tick_
type] = event.general_admission_seats
                    ticketPricesPerEvent[event.name][tick_type]
= float(event.general_ticket_cost)

        if not start_date or not end_date:
            return render(request, '404.html')

        hasData = False
        keysToDelete = []
        for key, value in eventData.items():
            if value['data']:
                hasData = True

            if not value['data']:
                keysToDelete.append(key)

        for key in keysToDelete:
            del eventData[key]

        if not hasData:
            messages.add_message(request, messages.ERROR, "Analytics
page generation failed: selected specific date range has no event
data.")
            return redirect('base_analytics_page')

        if 'download_csv' in request.GET:
            return generate_multi_event_csv(
                request,
                eventData,
                availableTicketsPerEvent,
                ticketPricesPerEvent,
                start_date,

```

```

        end_date
    )

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'total_unsold': total_unsold,
        'ticketPricesPerEvent': json.dumps(ticketPricesPerEvent),
        'availableTicketsPerEvent':
            json.dumps(availableTicketsPerEvent),
        'eventData': json.dumps(eventData)
    }

    return render(request,
    'admin/admin_analytics_specific_range.html', context)

def generate_multi_event_csv(request, events_data, available_tickets,
ticket_prices, start_date, end_date):
    """
    Generate CSV report for multiple events with consolidated summary
    and individual event analytics

    Args:
        user: User generating the report
        events_data: Dictionary with event attendance data
        available_tickets: Dictionary with available tickets per type
per event
        ticket_prices: Dictionary with ticket prices per type per
event
    """
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment;
filename="multi_event_analytics_{start_date}_to_{end_date}.csv"'
    writer = csv.writer(response)

    # Report Header
    writer.writerow(['Multi-Event Analytics Report'])
    writer.writerow(['Generated By:', request.user])
    writer.writerow(['Date Generated:',
timezone.localtime().strftime("%B %d, %Y, %I:%M %p")])
    writer.writerow([])

    # Initialize consolidated statistics
    total_stats = {
        'total_available': 0,

```

```

'total_sold': 0,
'by_type': {
    'VIP': {'available': 0, 'sold': 0, 'sales': 0,
'missed_sales': 0},
    'Gold': {'available': 0, 'sold': 0, 'sales': 0,
'missed_sales': 0},
    'Silver': {'available': 0, 'sold': 0, 'sales': 0,
'missed_sales': 0},
    'Gen Ad': {'available': 0, 'sold': 0, 'sales': 0,
'missed_sales': 0}
}
}

for event_name, event_data in events_data.items():
    for ticket_type in ['VIP', 'Gold', 'Silver', 'Gen Ad']:
        sold = sum(1 for attendee in event_data['data'] if
attendee['ticketType'] == ticket_type)

        unsold = available_tickets[event_name][ticket_type]
        price = ticket_prices[event_name][ticket_type]

        sales = sold * price
        missed_sales = unsold * price

        stats = total_stats['by_type'][ticket_type]
        stats['available'] += unsold
        stats['sold'] += sold
        stats['sales'] += sales
        stats['missed_sales'] += missed_sales

        total_stats['total_available'] += unsold
        total_stats['total_sold'] += sold

# Write Overall Summary
total_tickets = total_stats['total_available'] +
total_stats['total_sold']
writer.writerow(['OVERALL SUMMARY'])
writer.writerow(['Total Available + Sold Tickets:', total_tickets])
writer.writerow([])

# Tickets Sold Summary
writer.writerow(['TICKETS SOLD'])
for ticket_type, stats in total_stats['by_type'].items():

```

```

        percentage = (stats['sold'] / total_stats['total_sold'] * 100)
if total_stats['total_sold'] > 0 else 0
    writer.writerow([
        f"{ticket_type}:",
        f"{stats['sold']} tickets ({percentage:.1f}%)"
    ])
writer.writerow(['Total Tickets Sold:',
total_stats['total_sold']])
writer.writerow([])

# Unsold (Available) Tickets Summary
writer.writerow(['UNSOLD TICKETS (AVAILABLE)'])
for ticket_type, stats in total_stats['by_type'].items():
    percentage = (stats['available'] /
total_stats['total_available'] * 100) if
total_stats['total_available'] > 0 else 0
    writer.writerow([
        f"{ticket_type}:",
        f"{stats['available']} tickets ({percentage:.1f}%)"
    ])
writer.writerow(['Total Unsold Tickets:',
total_stats['total_available']])
writer.writerow([])

# Calculate total sales and missed sales
total_sales = sum(stats['sales'] for stats in
total_stats['by_type'].values())
total_missed_sales = sum(stats['missed_sales'] for stats in
total_stats['by_type'].values())
total_possible_sales = total_sales + total_missed_sales

# Sales Summary
writer.writerow(['TICKET SALES'])
for ticket_type, stats in total_stats['by_type'].items():
    percentage = (stats['sales'] / total_sales * 100) if
total_sales > 0 else 0
    writer.writerow([
        f"{ticket_type}:",
        f"PHP {stats['sales']:.2f} ({percentage:.1f}%)"
    ])
writer.writerow(['Total Sales:', f"PHP {total_sales:.2f}"])
writer.writerow(['Total Tickets Worth:', f"PHP
{total_possible_sales:.2f}"])
writer.writerow([])

```

```

# Ticket Loss Summary
writer.writerow(['TICKET LOSS'])
for ticket_type, stats in total_stats['by_type'].items():
    percentage = (stats['missed_sales'] / total_missed_sales * 100)
    if total_missed_sales > 0 else 0
    writer.writerow([
        f'{ticket_type}:',
        f'PHP {stats["missed_sales"]:.2f} ({percentage:.1f}%)'
    ])
writer.writerow(['Total Missed Sales:', f'PHP {total_missed_sales:.2f}'])
writer.writerow([])

# Final Summary
writer.writerow(['SUMMARY'])
writer.writerow(['Total Sales:', f'PHP {total_sales:.2f}'])
writer.writerow(['Missed Sales:', f'PHP {total_missed_sales:.2f}'])
writer.writerow(['Total Possible Sales:', f'PHP {total_possible_sales:.2f}'])
writer.writerow([])

# Individual Event Analytics
for event_name, event_data in events_data.items():
    writer.writerow(['=' * 50])
    writer.writerow([f'EVENT: {event_name}'])
    writer.writerow(['Start Date:', event_data['start_date']])
    writer.writerow(['End Date:', event_data['end_date']])
    writer.writerow([])

    event_stats = {
        'total_available': 0,
        'total_sold': 0,
        'by_type': {
            'VIP': {'available': 0, 'sold': 0, 'sales': 0},
            'missed_sales': 0,
            'Gold': {'available': 0, 'sold': 0, 'sales': 0},
            'missed_sales': 0,
            'Silver': {'available': 0, 'sold': 0, 'sales': 0},
            'missed_sales': 0,
            'Gen Ad': {'available': 0, 'sold': 0, 'sales': 0},
            'missed_sales': 0
        }
    }

```

```

# Calculate event statistics
for ticket_type in ['VIP', 'Gold', 'Silver', 'Gen Ad']:
    sold = sum(1 for attendee in event_data['data'] if
attendee['ticketType'] == ticket_type)
    unsold = available_tickets[event_name][ticket_type]
    price = ticket_prices[event_name][ticket_type]

    sales = sold * price
    missed_sales = unsold * price

    stats = event_stats['by_type'][ticket_type]
    stats['available'] = unsold
    stats['sold'] = sold
    stats['sales'] = sales
    stats['missed_sales'] = missed_sales

    event_stats['total_available'] += unsold
    event_stats['total_sold'] += sold

    event_total_tickets = event_stats['total_available'] +
event_stats['total_sold']
    event_total_sales = sum(stats['sales'] for stats in
event_stats['by_type'].values())
    event_total_missed_sales = sum(stats['missed_sales'] for stats
in event_stats['by_type'].values())
    event_total_possible_sales = event_total_sales +
event_total_missed_sales

    # Write event ticket distribution
    writer.writerow(['TICKETS SOLD'])
    for ticket_type, stats in event_stats['by_type'].items():
        percentage = (stats['sold'] / event_stats['total_sold']) *
100) if event_stats['total_sold'] > 0 else 0
        writer.writerow([
            f"{ticket_type}:",
            f"{stats['sold']} tickets ({percentage:.1f}%)"
        ])
    writer.writerow(['Total Tickets Sold:',
event_stats['total_sold']])
    writer.writerow([])

    writer.writerow(['TICKETS UNSOLD'])
    for ticket_type, stats in event_stats['by_type'].items():

```

```

        percentage = (stats['available'] /
event_stats['total_available'] * 100) if
event_stats['total_available'] > 0 else 0
        writer.writerow([
            f"{ticket_type}:",
            f"{stats['available']} tickets ({percentage:.1f}%)"
        ])
    writer.writerow(['Total Tickets Unsold:',
event_stats['total_available']])
    writer.writerow([])

    writer.writerow(['TICKET SALES'])
    for ticket_type, stats in event_stats['by_type'].items():
        percentage = (stats['sales'] / event_total_sales * 100) if
event_total_sales > 0 else 0
        writer.writerow([
            f"{ticket_type}:",
            f"PHP {stats['sales']:.2f} ({percentage:.1f}%)"
        ])
    writer.writerow(['Total Sales:', f"PHP
{event_total_sales:.2f}"])
    writer.writerow(['Total Tickets Worth:', f"PHP
{event_total_possible_sales:.2f}"])
    writer.writerow([])

    writer.writerow(['TICKET LOSS'])
    for ticket_type, stats in event_stats['by_type'].items():
        percentage = (stats['missed_sales'] /
event_total_missed_sales * 100) if event_total_missed_sales > 0 else 0
        writer.writerow([
            f"{ticket_type}:",
            f"PHP {stats['missed_sales']:.2f}
({percentage:.1f}%)"
        ])
    writer.writerow(['Total Missed Sales:', f"PHP
{event_total_missed_sales:.2f}"])
    writer.writerow([])

    writer.writerow(['SUMMARY'])
    writer.writerow(['Total Sales:', f"PHP
{event_total_sales:.2f}"])
    writer.writerow(['Missed Sales:', f"PHP
{event_total_missed_sales:.2f}"])
    writer.writerow(['Total Possible Sales:', f"PHP
{event_total_possible_sales:.2f}"])

```

```

writer.writerow([])

return response

@login_required()
@staff_member_required()
def analytics_specific_event(request, pk):
    event = get_object_or_404(Event, id=pk)
    tickets = Ticket.objects.filter(event=event)
    ticket_count = tickets.count()
    attendee_data = []

    start_date = event.start_date.date()
    end_date = event.end_date.date()
    event_days = (end_date - start_date).days + 1
    days = [start_date + timezone.timedelta(days=i) for i in
            range(event_days)]

    for ticket in tickets:
        usage_dates =
TicketUsage.objects.filter(ticket=ticket).values_list('usage_date',
flat=True)
            attended = {f"day{i+1}": "☑" if day in usage_dates else "✗"
for i, day in enumerate(days)}

        ticket_type = ticket.get_ticket_type_display()
        if ticket_type == 'VIP':
            ticket_type = 'VIP'
        if ticket_type == 'GOLD':
            ticket_type = 'Gold'
        if ticket_type == 'SILVER':
            ticket_type = 'Silver'
        if ticket_type == 'GENERAL ADMISSION':
            ticket_type = 'Gen Ad'

        attendee_data.append({
            "name": f'{ticket.first_name} {ticket.last_name}',
            "ticketType": ticket_type,
            "attended": attended
        })

    attendee_count = tickets.filter(is_used=True).count()
    no_show_count = tickets.filter(is_used=False).count()

```

```

        total_vip_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_VIP,
event=event).count()
        total_gold_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_GOLD,
event=event).count()
        total_silver_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_SILVER,
event=event).count()
        total_general_admission_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_GENERAL_ADMISSION,
event=event).count()
        total_unsold_tickets = event.gold_seats + event.vip_seats +
event.silver_seats + event.general_admission_seats

    if 'download_csv' in request.GET:
        return generate_csv(
            request.user,
            event,
            attendee_count,
            no_show_count,
            ticket_count,
            attendee_data,
            days
        )

context = {
    'event': event,
    'available_apps': site.get_app_list(request),
    'user': request.user,
    'attendee_data': json.dumps(attendee_data),
    'event_name': event.name,
    'attendee_count': attendee_count,
    'no_show_count': no_show_count,
    'total_tickets': ticket_count,
    'total_unsold_tickets': total_unsold_tickets,
    "total_vip_tickets": total_vip_tickets + event.vip_seats,
    "total_gold_tickets": total_gold_tickets + event.gold_seats,
    "total_silver_tickets": total_silver_tickets +
event.silver_seats,
    "total_general_admission_tickets":
total_general_admission_tickets + event.general_admission_seats,
    "ticket_VIP_price": float(event.vip_ticket_cost),
    "ticket_gold_price": float(event.gold_ticket_cost),
    "ticket_silver_price": float(event.silver_ticket_cost),
}

```

```

        "ticket_general_admission_price":  

    float(event.general_ticket_cost)  

    }  
  

    return render(request,  

    'admin/admin_analytics_specific_event.html', context)  
  

def generate_csv(user, event: Event, attendee_count, no_show_count,  

total_tickets, attendee_data, days):  

    response = HttpResponse(content_type='text/csv')  

    response['Content-Disposition'] = f'attachment;  

filename="{event.name}_analytics.csv"'  
  

    writer = csv.writer(response)  

# Basic Event Information  

    writer.writerow(['Exported By', user.username])  

    writer.writerow(['Time Generated', timezone.now().strftime("%B %d,  

%Y, %I:%M %p")])  

    writer.writerow(['Event Name', event.name])  

    writer.writerow(['Event Start Date', event.start_date.strftime("%B  

%d, %Y, %I:%M %p")])  

    writer.writerow(['Event End Date', event.end_date.strftime("%B %d,  

%Y, %I:%M %p")])  

    writer.writerow([])  
  

# Attendance Statistics  

    writer.writerow(['Attendance Statistics'])  

    writer.writerow(['Total Tickets Purchased', total_tickets])  

    writer.writerow(['Number of Attendees', attendee_count])  

    writer.writerow(['No Shows', no_show_count])  

    writer.writerow([])  

    writer.writerow(['Ticket Costs'])  

    writer.writerow(['VIP', f"PHP {event.vip_ticket_cost}"])  

    writer.writerow(['Gold', f"PHP {event.gold_ticket_cost}"])  

    writer.writerow(['Silver', f"PHP {event.silver_ticket_cost}"])  

    writer.writerow(['General Admission', f"PHP  

{event.general_ticket_cost}"])  

    writer.writerow([])  
  

# Ticket Sales Statistics  

    writer.writerow(['Ticket Sales Statistics'])  
  

# Sold Tickets by Type

```

```

        total_vip_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_VIP,
event=event).count()
        total_gold_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_GOLD,
event=event).count()
        total_silver_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_SILVER,
event=event).count()
        total_general_admission_tickets =
Ticket.objects.filter(ticket_type=Ticket.SEAT_GENERAL_ADMISSION,
event=event).count()

        writer.writerow(['Ticket Type', 'Sold', 'Available', 'Total',
'Sales Price', 'Total Sales', 'Total Loss'])
        writer.writerow(['VIP', total_vip_tickets, event.vip_seats,
total_vip_tickets + event.vip_seats,
f"PHP {event.vip_ticket_cost:.2f}", f"PHP
{total_vip_tickets * event.vip_ticket_cost:.2f}",
f"PHP {event.vip_seats *
event.vip_ticket_cost:.2f}"])

        writer.writerow(['Gold', total_gold_tickets, event.gold_seats,
total_gold_tickets + event.gold_seats,
f"PHP {event.gold_ticket_cost:.2f}", f"PHP
{total_gold_tickets * event.gold_ticket_cost:.2f}",
f"PHP {event.gold_seats *
event.gold_ticket_cost:.2f}"])

        writer.writerow(['Silver', total_silver_tickets,
event.silver_seats, total_silver_tickets + event.silver_seats,
f"PHP {event.silver_ticket_cost:.2f}", f"PHP
{total_silver_tickets * event.silver_ticket_cost:.2f}",
f"PHP {event.silver_seats *
event.silver_ticket_cost:.2f}"])

        writer.writerow(['General Admission',
total_general_admission_tickets, event.general_admission_seats,
total_general_admission_tickets +
event.general_admission_seats,
f"PHP {event.general_ticket_cost:.2f}", f"PHP
{total_general_admission_tickets * event.general_ticket_cost:.2f}",
f"PHP {event.general_admission_seats *
event.general_ticket_cost:.2f}"])

```

```

writer.writerow([])

# Total Revenue and Lost Revenue
total_sales = (total_vip_tickets * event.vip_ticket_cost +
               total_gold_tickets * event.gold_ticket_cost +
               total_silver_tickets * event.silver_ticket_cost +
               total_general_admission_tickets *
               event.general_ticket_cost)

lost_sales = (event.vip_seats * event.vip_ticket_cost +
              event.gold_seats * event.gold_ticket_cost +
              event.silver_seats *
              event.silver_ticket_cost +
              event.general_admission_seats *
              event.general_ticket_cost)

writer.writerow(['Sales Summary'])
writer.writerow(['Total Sales', f"PHP {total_sales:.2f}"])
writer.writerow(['Lost Sales', f"PHP {lost_sales:.2f}"])
writer.writerow(['Total Tickets Worth', f"PHP
{lost_sales+total_sales:.2f}"])
writer.writerow([])

# Attendance Details
writer.writerow(['Attendance Details'])
header = ['Name', 'Ticket Type'] + [f"Day {i+1}" for i in
range(len(days))]
writer.writerow(header)

for attendee in attendee_data:
    row = [
        attendee['name'],
        attendee['ticketType']
    ]
    for i in range(len(days)):
        day_key = f"day{i+1}"
        attendance = "Attended" if attendee['attended'][day_key] ==
        "☑" else "No Show"
        row.append(attendance)
    writer.writerow(row)

return response

@login_required()

```

```

@staff_member_required()
def analytics_specific_lesson(request):
    if request.method == 'GET':
        start_date_str = request.GET.get('start_date', None)
        end_date_str = request.GET.get('end_date', None)

        start_date = parse_date(start_date_str) if start_date_str else
None
        end_date = parse_date(end_date_str) if end_date_str else None

        instruments = Instrument.objects.all()

        lessons =
Lesson.objects.annotate(num_students=Count('students')).filter(
            start_date__gte=start_date,
            end_date__lte=end_date,
            num_students__gt=0
        )

        available_slots = {}
        for lesson in lessons:
            available_slots[lesson.id] = lesson.max_slots

        instruments_data = []
        lessonCostData = {}
        for lesson in lessons:
            lessonCostData[lesson.id] = float(lesson.lesson_cost)
            for student in lesson.students.all():
                attendance_str = ""
                current_date = lesson.start_date
                while current_date <= lesson.end_date:
                    if LessonAttendance.objects.filter(lesson=lesson,
student=student, attended_at__date=current_date.date()).exists():
                        attendance_str += "✓"
                    else:
                        attendance_str += "✗"
                    current_date += timezone.timedelta(days=1)
            instruments_data.append({
                'instrument': lesson.instrument.name,
                'user': student.get_full_name(),
                'instructor': lesson.instructor.get_full_name() if
lesson.instructor else 'N/A',
                'startDate': lesson.start_date.strftime('%Y-%m-
%d'),
                'endDate': lesson.end_date.strftime('%Y-%m-%d'),
            })

```

```

        'attendance': attendance_str,
        'lesson_id': lesson.id
    })

    if 'download_csv' in request.GET:
        return generate_lesson_analytics_csv(
            request,
            lessons,
            start_date,
            end_date
        )

    print(json.dumps(instruments_data, indent=4))
    print(json.dumps(available_slots, indent=4))
    print(json.dumps(lessonCostData, indent=4))

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'availableSlots': json.dumps(available_slots),
        'instrumentsData': json.dumps(instruments_data),
        'lessonCostData': json.dumps(lessonCostData)
    }

    return render(request,
        'admin/admin_analytics_specific_lesson.html', context)

    return Http404("Invalid request method.")

def generate_lesson_analytics_csv(request, lessons_data, start_date,
end_date):
    """
    Generate CSV report for lesson analytics with consolidated summary
    and details grouped by instrument

    Args:
        request: The HTTP request object
        lessons_data: QuerySet of Lesson objects to analyze
    """
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment;
filename="lesson_analytics_{start_date}_to_{end_date}.csv"'
    writer = csv.writer(response)

    # Report Header

```

```

writer.writerow(['Lesson Analytics Report'])
writer.writerow(['Generated By:', request.user])
writer.writerow(['Date Generated:',
timezone.localtime().strftime("%B %d, %Y, %I:%M %p")])
writer.writerow([])

# Initialize consolidated statistics
total_stats = {
    'total_revenue': Decimal('0.00'),
    'total_potential_revenue': Decimal('0.00'),
    'total_students': 0,
    'total_slots': 0,
    'total_attendance': 0
}

# Group lessons by instrument
instrument_groups = {}
for lesson in lessons_data:
    instrument_name = lesson.instrument.name
    if instrument_name not in instrument_groups:
        instrument_groups[instrument_name] = {
            'lessons': [],
            'revenue': Decimal('0.00'),
            'potential_revenue': Decimal('0.00'),
            'enrolled_students': 0,
            'total_slots': 0,
            'attendance_count': 0,
            'instructors': set()
        }

    # Add lesson to group and calculate statistics
    group = instrument_groups[instrument_name]
    group['lessons'].append(lesson)

    enrolled_count = lesson.students.count()
    revenue = enrolled_count * lesson.lesson_cost if
lesson.lesson_cost else Decimal('0.00')
    potential_revenue = lesson.max_slots * lesson.lesson_cost if
lesson.lesson_cost else Decimal('0.00')

    group['revenue'] += revenue
    group['potential_revenue'] += potential_revenue
    group['enrolled_students'] += enrolled_count
    group['total_slots'] += lesson.max_slots
    group['attendance_count'] += lesson.attendances.count()

```

```

if lesson.instructor:
    group['instructors'].add(lesson.instructor.id)

    # Update total statistics
    total_stats['total_revenue'] += revenue
    total_stats['total_potential_revenue'] += potential_revenue
    total_stats['total_students'] += enrolled_count
    total_stats['total_slots'] += lesson.max_slots
    total_stats['total_attendance'] += lesson.attendances.count()

# Write Overall Summary
writer.writerow(['OVERALL SUMMARY'])
writer.writerow(['Total Instruments:', len(instrument_groups)])
writer.writerow(['Total Lessons:', len(lessons_data)])
writer.writerow([])

# Overall Revenue Summary
total_missed_revenue = total_stats['total_potential_revenue'] -
total_stats['total_revenue']
revenue_percentage = (total_stats['total_revenue'] / 
total_stats['total_potential_revenue'] * 100) if
total_stats['total_potential_revenue'] > 0 else 0
loss_percentage = (total_missed_revenue / 
total_stats['total_potential_revenue'] * 100) if
total_stats['total_potential_revenue'] > 0 else 0

writer.writerow(['OVERALL SALES SUMMARY'])
writer.writerow(['Total Sales:', f"PHP
{total_stats['total_revenue']:.2f} ({revenue_percentage:.1f}%)"])
writer.writerow(['Missed Sales:', f"PHP
{total_missed_revenue:.2f} ({loss_percentage:.1f}%)"])
writer.writerow(['Total Potential Sales:', f"PHP
{total_stats['total_potential_revenue']:.2f}"])
writer.writerow([])

# Overall Enrollment Summary
enrollment_percentage = (total_stats['total_students'] /
total_stats['total_slots'] * 100) if total_stats['total_slots'] > 0
else 0
empty_slots = total_stats['total_slots'] -
total_stats['total_students']
empty_percentage = (empty_slots / total_stats['total_slots'] * 
100) if total_stats['total_slots'] > 0 else 0

writer.writerow(['OVERALL SUMMARY'])

```

```

        writer.writerow(['Total Students Enrolled:',
f"{{total_stats['total_students']}} ({enrollment_percentage:.1f}%)"])
        writer.writerow(['Empty Slots:', f"{{empty_slots}}
({empty_percentage:.1f}%)"])
        writer.writerow(['Total Available Slots:',
total_stats['total_slots']])
        writer.writerow([])

        # Overall Attendance Summary
        attendance_percentage = (total_stats['total_attendance'] /
total_stats['total_students'] * 100) if total_stats['total_students'] > 0 else 0
        writer.writerow(['OVERALL ATTENDANCE SUMMARY'])
        writer.writerow(['Total Attendances:',
f"{{total_stats['total_attendance']}} ({attendance_percentage:.1f}%)"])
        writer.writerow([])

        # Instrument-wise Analytics
        for instrument_name, stats in instrument_groups.items():
            writer.writerow(['=' * 50])
            writer.writerow([f'INSTRUMENT: {instrument_name}'])
            writer.writerow(['Total Lessons:', len(stats['lessons'])])
            writer.writerow(['Total Unique Instructors:',
len(stats['instructors'])])
            writer.writerow([])

            # Instrument Revenue Analysis
            missed_revenue = stats['potential_revenue'] - stats['revenue']
            revenue_percentage = (stats['revenue'] /
stats['potential_revenue'] * 100) if stats['potential_revenue'] > 0 else 0
            loss_percentage = (missed_revenue / stats['potential_revenue'] *
100) if stats['potential_revenue'] > 0 else 0

            writer.writerow(['SALES ANALYSIS'])
            writer.writerow(['Total Sales:', f"PHP {stats['revenue']:.2f}
({revenue_percentage:.1f}%)"])
            writer.writerow(['Missed Sales:', f"PHP {missed_revenue:.2f}
({loss_percentage:.1f}%)"])
            writer.writerow(['Potential Sales:', f"PHP
{stats['potential_revenue']:.2f}"])
            writer.writerow([])

        # Instrument Enrollment Analysis

```

```

        empty_slots = stats['total_slots'] -
stats['enrolled_students']
            enrollment_percentage = (stats['enrolled_students'] /
stats['total_slots'] * 100) if stats['total_slots'] > 0 else 0
            empty_percentage = (empty_slots / stats['total_slots'] * 100)
if stats['total_slots'] > 0 else 0

            writer.writerow(['INSTRUMENT ANALYSIS'])
            writer.writerow(['Total Students Joined:',
f'{stats['enrolled_students']} ({enrollment_percentage:.1f}%)'])
            writer.writerow(['Empty Slots:', f'{empty_slots}'
({empty_percentage:.1f}%)'])
            writer.writerow(['Total Available Slots:',
stats['total_slots']])
            writer.writerow([])

            # Instrument Attendance Analysis
            attendance_percentage = (stats['attendance_count'] /
stats['enrolled_students'] * 100) if stats['enrolled_students'] > 0
else 0

            writer.writerow(['ATTENDANCE ANALYSIS'])
            writer.writerow(['Total Attendances:',
f'{stats['attendance_count']} ({attendance_percentage:.1f}%)'])
            writer.writerow([])

            # Individual Lessons in this Instrument Group
            writer.writerow(['INDIVIDUAL LESSONS'])
            writer.writerow(['Date', 'Instructor', 'Students', 'Sales',
'Attendance'])

            for lesson in sorted(stats['lessons'], key=lambda x:
x.start_date):
                enrolled_count = lesson.students.count()
                revenue = enrolled_count * lesson.lesson_cost if
lesson.lesson_cost else Decimal('0.00')
                attendance_count = lesson.attendances.count()

                writer.writerow([
                    lesson.start_date.strftime("%B %d, %Y, %I:%M %p"),
                    str(lesson.instructor) if lesson.instructor else 'No
Instructor',
                    f'{enrolled_count}/{lesson.max_slots}',
                    f"PHP {revenue:.2f}",
                    f'{attendance_count}/{enrolled_count}'
```

```

        ])
writer.writerow([])

return response

@login_required()
@staff_member_required()
def analytics(request):
    # Cards Data
    newest_orchestra_audition =
Audition.objects.filter(audition_type=Audition.AuditionType.ORCHESTRA)
.order_by('-audition_date').first()
    newest_choir_audition =
Audition.objects.filter(audition_type=Audition.AuditionType.CHOIR).ord
er_by('-audition_date').first()

    events =
Event.objects.filter(event_type=Event.EventType.PUBLIC).order_by('-
start_date')
    current_event = None
    for event in events:
        if event.is_current():
            current_event = event
            break
    attendance_count = Ticket.objects.filter(event=current_event,
is_used=True).count() if current_event else 0

    available_seats = 0
    if current_event:
        total_seats = sum([getattr(current_event, f"{seat}_seats") for
seat in ['vip', 'gold', 'silver', 'general_admission']])
        sold_tickets =
Ticket.objects.filter(event=current_event).exclude(ticket_type='GA').c
ount()
        available_seats = total_seats - sold_tickets if current_event
else 0

    # Charts Data
    events_ticket_sales = (
        Event.objects.filter(event_type=Event.EventType.PUBLIC)
.annotate(ticket_sales=Count('ticket_event'))
.order_by('-ticket_sales')[:7]
)
    attended_events = (
        Event.objects.filter(event_type=Event.EventType.PUBLIC)

```

```

        .annotate(attendance_count=Count('ticket_event',
filter=Q(ticket_event__is_used=True)))
        .order_by('-attendance_count')[:7]
    )
    lessons_attendance =
Lesson.objects.annotate(attendance_count=Count('attendances')).order_b
y('-attendance_count')[:7]
    lessons_enrollment =
Lesson.objects.annotate(enrollment_count=Count('students')).order_by('
-enrollment_count')[:7]

instruments = Instrument.objects.all()

popular_instruments_data = []

for instrument in instruments:
    popularity =
Lesson.objects.filter(instrument=instrument).count()

    popular_instruments_data.append({
        'instrument': instrument.name,
        'popularity': popularity,
        'enrollmentDate': instrument.created_at.strftime('%Y')
    })

events =
Event.objects.filter(event_type=Event.EventType.PUBLIC).order_by('
-start_date')

ticket_sales_data = []

for event in events:
    tickets_purchased = Ticket.objects.filter(event=event).count()
    attendees = Ticket.objects.filter(event=event,
is_used=True).count()

    ticket_sales_data.append({
        'event': event.name,
        'ticketsPurchased': tickets_purchased,
        'attendees': attendees,
        'startDate': event.start_date.strftime('%Y-%m-%d'),
        'endDate': event.end_date.strftime('%Y-%m-%d'),
    })

lessons = Lesson.objects.all()

```

```

lessons_data = []

for lesson in lessons:
    lessons_enrolled = lesson.students.count()
    lessons_attended =
LessonAttendance.objects.filter(lesson=lesson).count()

    lessons_data.append({
        'event': lesson.instrument.name,
        'lessonsEnrolled': lessons_enrolled,
        'lessonsAttended': lessons_attended,
        'startDate': lesson.start_date.strftime('%Y-%m-%d'),
        'endDate': lesson.end_date.strftime('%Y-%m-%d'),
    })

bookings = Book.objects.annotate(
    year=ExtractYear('meet_date'),
    month=ExtractMonth('meet_date')
).values('year',
'month').annotate(count=Count('id')).order_by('year', 'month')

booking_data = {}
for booking in bookings:
    year = str(booking['year'])
    month = booking['month']
    count = booking['count']

    if year not in booking_data:
        booking_data[year] = {
            "January": 0,
            "February": 0,
            "March": 0,
            "April": 0,
            "May": 0,
            "June": 0,
            "July": 0,
            "August": 0,
            "September": 0,
            "October": 0,
            "November": 0,
            "December": 0,
        }

    if month == 1:

```

```

        booking_data[year]["January"] = count
    elif month == 2:
        booking_data[year]["February"] = count
    elif month == 3:
        booking_data[year]["March"] = count
    elif month == 4:
        booking_data[year]["April"] = count
    elif month == 5:
        booking_data[year]["May"] = count
    elif month == 6:
        booking_data[year]["June"] = count
    elif month == 7:
        booking_data[year]["July"] = count
    elif month == 8:
        booking_data[year]["August"] = count
    elif month == 9:
        booking_data[year]["September"] = count
    elif month == 10:
        booking_data[year]["October"] = count
    elif month == 11:
        booking_data[year]["November"] = count
    elif month == 12:
        booking_data[year]["December"] = count

instructor_ratings = (
    Feedback.objects
    .filter(feedback_type=Feedback.FeedbackType.Instructor)
    .values('instructor')
    .annotate(averageRating=Avg('rating'))
    .order_by('-averageRating')
)

feedback_data = [
{
    "name": f'{Member.objects.get(id=rating['instructor']).get_full_name()}',
    "averageRating": round(rating['averageRating'], 1)
}
for rating in instructor_ratings
]

context = {
    'newest_orchestra_audition': newest_orchestra_audition,
    'newest_choir_audition': newest_choir_audition,
    'attendance_count': attendance_count,
}

```

```

        'available_seats': available_seats,
        'events_ticket_sales': events_ticket_sales,
        'attended_events': attended_events,
        'lessons_attendance': lessons_attendance,
        'lessons_enrollment': lessons_enrollment,
        'popular_instruments': popular_instruments_data,
        'available_apps': site.get_app_list(request),
        'booking_data': json.dumps(booking_data),
        'ticket_data': json.dumps(ticket_sales_data),
        'lesson_data': json.dumps(lessons_data),
        'feedback_data': json.dumps(feedback_data[:5]),
        'user': request.user,
    }

    return render(request, 'admin/admin_analytics.html', context)

@login_required()
@permission_required(["ticket.change_ticket", "ticket.view_ticket",
"ticket.delete_ticket", "ticket.add_ticket"])
def scan_2fa(request):
    if request.method == 'GET':
        ticket_id = request.session.get('scanned_ticket_id', None)
        if not ticket_id:
            return Http404("Object not found.")

        ticket = Ticket.objects.get(id=ticket_id)
        member_id = ticket.buyer.id
        user = Member.objects.get(id=member_id)
        sms2fa = SMS2FA.objects.filter(member=user).order_by('-
created_at').first()

        if not sms2fa:
            sms2fa = SMS2FA.objects.create(member=user)
        else:
            if sms2fa.used or timezone.now() >= sms2fa.expiration:
                sms2fa = SMS2FA.objects.create(member=user)

        mobile_number = str(ticket.mobile_number)
        site_config = SiteConfig.load()
        sms2fa.send_specific(mobile_number, f"Ticket #{ticket.id} for
event {ticket.event.name}'s 2FA code is: {sms2fa.code}\n\nDo not share
this OTP code as you may compromise your ticket integrity.\nThis 2FA
code will expire in {site_config.OTP_EXPIRATION_IN_HOURS} hours.\nThis
OTP is only applicable in the GY OCC website.")

```

```

form = ScanAuth2FAForm()
content = {
    'form': form,
    'available_apps': site.get_app_list(request),
    'fullname': ticket.buyer.get_full_name()
}
return render(request, 'admin/admin_scan_2fa.html', content)

elif request.method == "POST":
    form = ScanAuth2FAForm(request.POST)
    if form.is_valid():
        ticket_id = request.session.get('scanned_ticket_id', None)
        ticket = Ticket.objects.get(id=ticket_id)
        otp = form.cleaned_data['code']
        otp = int(otp)
        member_id = ticket.buyer.id
        member = Member.objects.filter(id=member_id).first()
        sms2fa = SMS2FA.objects.filter(member=member).order_by('-created_at').first()

        if otp and member_id:
            if int(sms2fa.code) == otp and not
sms2fa.is_expired():
                sms2fa.used = True
                sms2fa.save()

            today = timezone.now().date()

            TicketUsage.objects.create(
                ticket=ticket,
                usage_date=today
            )

            if ticket.event.end_date.date() == today:
                ticket.is_used = True
                ticket.date_used = timezone.now()
                ticket.save()

            del request.session['scanned_ticket_id']

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'ticket': ticket
    }

```

```

        return render(request,
"admin/admin_ticket_scan_success.html", context)
    else:
        if sms2fa.is_expired():
            messages.error(request, "Expired OTP.")
            content = {
                'available_apps':
site.get_app_list(request),
                'user': request.user,
                'form': form
            }
            return render(request,
'admin/admin_scan_2fa.html', content)

            messages.error(request, "Invalid OTP.")
            content = {
                'available_apps': site.get_app_list(request),
                'user': request.user,
                'form': form
            }
            return render(request,
'admin/admin_scan_2fa.html', content)
        else:
            messages.error(request, "OTP verification failed.")
            content = {
                'available_apps': site.get_app_list(request),
                'user': request.user,
                'form': form
            }
            return render(request, 'admin/admin_scan_2fa.html',
content)
    else:
        content = {
            'available_apps': site.get_app_list(request),
            'user': request.user,
            'form': form
        }
        return render(request, 'admin/admin_scan_2fa.html',
content)

@login_required()
@permission_required(["ticket.change_ticket", "ticket.view_ticket",
"ticket.delete_ticket", "ticket.add_ticket"])
def scan(request):

```

```

if request.method == 'GET':
    data = request.GET.get('data', None)
    if data:
        try:
            js_data = json.loads(data)
        except json.decoder.JSONDecodeError:
            context = {
                'available_apps': site.get_app_list(request),
                'user': request.user,
                'message': "Invalid Ticket"
            }

            return render(request,
"admin/admin_ticket_scan_fail.html", context)

ticket = Ticket.objects.get(id=js_data['ticket_id'])

if not ticket.event.is_current():
    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'message': "Can't scan tickets yet, the event is
currently not ongoing or current."
    }

    return render(request,
"admin/admin_ticket_scan_fail.html", context)

today = timezone.now().date()
if TicketUsage.objects.filter(ticket=ticket,
usage_date=today).exists():
    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'message': "This ticket has already been used and
scanned before."
    }

    return render(request,
"admin/admin_ticket_scan_fail.html", context)

success, result = ticket.validate_qr(data)
if success and result == "Match":
    request.session['scanned_ticket_id'] = ticket.id
    site_config = SiteConfig.load()

```

```

        if not site_config.USE_TICKET_SCAN OTP:
            today = timezone.now().date()

            TicketUsage.objects.create(
                ticket=ticket,
                usage_date=today
            )

        if ticket.event.end_date.date() == today:
            ticket.is_used = True
            ticket.date_used = timezone.now()
            ticket.save()

        del request.session['scanned_ticket_id']

        context = {
            'available_apps': site.get_app_list(request),
            'user': request.user,
            'ticket': ticket
        }

        return render(request,
        "admin/admin_ticket_scan_success.html", context)

    return redirect('/staff/ticket/scan/2fa/')

elif not success and result == "Mismatch":
    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'message': "This ticket does not match the
security hash."
    }

    return render(request,
    "admin/admin_ticket_scan_fail.html", context)

elif not success and result == "Used":
    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'message': "This ticket has already been used and
scanned before."
    }

```

```

                return render(request,
"admin/admin_ticket_scan_fail.html", context)

        context = {
            'available_apps': site.get_app_list(request),
            'user': request.user
        }
        return render(request, "admin/admin_ticket_scan.html", context)

@login_required()
@permission_required(["event.change_event", "event.view_event",
"event.delete_event", "event.add_event"])
def event_calendar(request):
    event_queryset = Event.objects.all()
    book_queryset = Book.objects.filter(is_paid=True)
    if not request.user.groups.filter(name__in=['Administrator',
'Superuser']).exists():
        event_queryset = event_queryset.filter(organizer=request.user)
    events = []
    for event in event_queryset:

        if event.event_type == Event.EventType.PRIVATE:
            if PaymentEvent.objects.filter(event=event).exists():
                paymentEvent =
PaymentEvent.objects.filter(event=event).first()
                if paymentEvent.payment_amount <
paymentEvent.total_amount:
                    continue

        events.append(
{
    'id': event.id,
    'title': event.name,
    'start': event.start_date.isoformat(),
    'end': event.end_date.isoformat(),
    'description': event.description,
    'location': event.location,
    'url': f"/staff/event/event/{event.id}/change/",
    'color': str(event.calendar_color)
}
)
    for book in book_queryset:
        events.append(
{

```

```

        'id': book.id,
        'title': book.name,
        'start': book.meet_date.isoformat(),
        'description': book.description,
        'location': book.location,
        'url': f"/staff/event/book/{book.id}/change/",
        'color': "#404854"
    }
)

context = {
    'available_apps': site.get_app_list(request),
    'user': request.user,
    'events': json.dumps(events)
}
return render(request, "admin/admin_event_calendar.html", context)

@login_required()
@permission_required(["lesson.change_lesson", "lesson.view_lesson",
"lesson.delete_lesson", "lesson.add_lesson"])
def lesson_calendar(request):
    lessons_queryset = Lesson.objects.filter(instructor=request.user)
    lessons = []
    for lesson in lessons_queryset:
        lessons.append(
            {
                'id': lesson.id,
                'title': lesson.instrument.name,
                'start': lesson.start_date.isoformat(),
                'end': lesson.end_date.isoformat(),
                'location': lesson.location,
                'url': f"/staff/lesson/lesson/{lesson.id}/change/",
                'color': str(lesson.calendar_color)
            }
        )

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'lessons': json.dumps(lessons)
    }
    return render(request, "admin/admin_lesson_calendar.html",
    context)

@login_required()

```

```

@permission_required(["lesson.change_lesson", "lesson.view_lesson",
"lesson.delete_lesson", "lesson.add_lesson"])
def lesson_user_specific_overview_redirect(request):
    user_id = request.user.id
    return redirect('instructor_dashboard_specific', pk=user_id)

@login_required()
@staff_member_required()
def attendance_scan(request):
    if request.method == 'GET':
        data = request.GET.get('data', None)
        if data:
            member_id = data.split("-")[0]
            lesson_id = data.split("-")[1]
            try:
                lesson = Lesson.objects.get(id=lesson_id)
                member = Member.objects.get(id=member_id)
            except Lesson.DoesNotExist or Member.DoesNotExist:
                context = {
                    'available_apps': site.get_app_list(request),
                    'user': request.user,
                    'title': 'Scan Failed',
                    'message': 'This QR Code is invalid and does not
exist.',
                    'failed': True
                }
                return render(request,
                            "admin/lesson_scan_attendance_after.html", context)

            if not lesson.is_current():
                context = {
                    'available_apps': site.get_app_list(request),
                    'user': request.user,
                    'title': 'Scan Failed',
                    'message': 'This lesson can\'t take attendances
yet as it is not currently ongoing.',
                    'failed': True
                }
                return render(request,
                            "admin/lesson_scan_attendance_after.html", context)

            if not member in lesson.students.all():
                context = {

```

```

        'available_apps': site.get_app_list(request),
        'user': request.user,
        'title': 'Scan Failed',
        'message': 'This student is NOT A PART of this
lesson.',
        'failed': True
    }
    return render(request,
"admin/lesson_scan_attendance_after.html", context)

    if LessonAttendance.objects.filter(lesson=lesson,
student=member).exists():
        context = {
            'available_apps': site.get_app_list(request),
            'user': request.user,
            'title': 'Scan Complete',
            'message': 'This student has already attended this
lesson.',
            'failed': True
        }
        return render(request,
"admin/lesson_scan_attendance_after.html", context)

    LessonAttendance.objects.create(
        lesson=lesson,
        student=member,
        confirmed_by=request.user
    )

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
        'title': 'Scan Complete',
        'message': 'Added student to the list of attendees for
this lesson.',
        'failed': False
    }
    return render(request,
"admin/lesson_scan_attendance_after.html", context)

    context = {
        'available_apps': site.get_app_list(request),
        'user': request.user,
    }

```

```

        return render(request, "admin/lesson_scan_attendance.html",
context)

@login_required()
@staff_member_required()
def start_payment_for_private_event(request, book_id, member_id,
amount, total_amount):
    if request.method == 'GET':
        amount = float(amount)
        total_amount = float(total_amount)

    book = Book.objects.get(id=book_id)
    member = Member.objects.get(id=member_id)

    site_config = SiteConfig.load()
    domain_url = site_config.DOMAIN_URL
    maya_encrypted_key = settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

    random_uuid = uuid.uuid4()
    payment_data = PaymentData(
        totalAmount = PaymentTotalAmount(value=float(amount),
currency="PHP"),
        buyer = PaymentBuyer(
            contact = PaymentContact(email=member.email),
            firstName = member.first_name,
            lastName = member.last_name
        ),
        redirectUrl = PaymentRedirectUrl(
            success = domain_url +
f'event/private/success?session_id={str(random_uuid)}',
            failure = domain_url + f'event/private/failed',
            cancel = domain_url + f'event/private/failed'
        ),
        items = [
            PaymentItem(
                totalAmount=PaymentItemAmount(value=float(amount))

                name=f"Payment for Private Event: {book.name}",
                quantity="1",
                code=str(book.id),
                description="Online payment procedure for private
events."
            )
        ],
        requestReferenceNumber=f"{str(random_uuid)}"
    )

```

```

        )

    p_data = {
        "type": "private_event",
        "target": book.id,
        "amount_paid": float(payment_data.totalAmount.value),
        "amount_total": float(total_amount)
    }

    PaymentSession.objects.create(
        buyer = member,
        payment_data = p_data,
        paid = False,
        session_id = str(random_uuid)
    )

    headers = {
        "accept": "application/json",
        "authorization": f"Basic {str(maya_encrypted_key)}",
        "content-type": "application/json"
    }

    response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
    return redirect(json.loads(response.text)["redirectUrl"])

def private_event_payment_successful(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession and not paymentSession.paid:
                paymentSession.paid = True
                paymentSession.save()
                target_event = paymentSession.payment_data["target"]
                amount_paid =
paymentSession.payment_data["amount_paid"]
                amount_total =
paymentSession.payment_data["amount_total"]
                book = Book.objects.get(id=target_event)

                PaymentEvent.objects.create(
                    reference_number = checkout_id,
                    payment_amount = amount_paid,

```

```

        total_amount = amount_total
    )

    base_name = book.name
    counter = 1
    new_name = base_name
    while Event.objects.filter(name=new_name).exists():
        new_name = f"{base_name} ({counter})"
        counter += 1

    base_url = reverse('admin:event_event_add')
    params = {
        'requestor': book.booker.id,
        'name': new_name,
        'description': book.description,
        'location': book.location,
        'event_type': book.event_type,
    }

    if
request.user.groups.filter(name__in=['Organizer']).exists():
        params['organizer'] = request.user.id

    url = f'{base_url}?{urlencode(params)}'

    messages.success(request, f"Payment for private event:
{book.name} was successful, please proceed to create the event.")
    return redirect(url)
elif paymentSession and paymentSession.paid:
    return render(request, "payment_success.html")
else:
    return HttpResponse("Stop trying to blindly guess
session ids..")
else:
    messages.success(request, f"Payment for private event
unsuccessful.")
    return redirect("/staff/")

def private_event_payment_failed_or_cancelled(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if not checkout_id == "N/A":
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession:

```

```

        paymentSession.paid = False
        paymentSession.save()
        target_event = paymentSession.payment_data["target"]
        book = Book.objects.get(id=target_event)
        messages.error(request, f"Payment for private event: {book.name} was unsuccessful.")
        return redirect("/staff/")
    else:
        return HttpResponse("Stop trying to blindly guess session ids..")
    else:
        messages.error(request, f"Payment for private event unsuccessful.")
        return redirect("/staff/")

```

audition > models.py

```

from audioop import reverse
from django.conf import settings
from django.db import models
from django.forms import ValidationError
from main.models import Member, Notification, SiteConfig
from django.core.validators import FileExtensionValidator
from django.utils import timezone
from django.core.mail import send_mail

from utilities.sms import send_sms

# Create your models here.
class Audition(models.Model):
    class AuditionType(models.TextChoices):
        ORCHESTRA = "Orchestra", "Orchestra"
        CHOIR = "Choir", "Choir"

        auditionee = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name='auditionee', null=True, blank=True)
        audition_type = models.CharField(choices=AuditionType.choices,
default=AuditionType.ORCHESTRA)
        audition_date = models.DateTimeField(blank=True, null=True)
        address = models.TextField(null=False, blank=False)
        video_file = models.FileField(
            upload_to='auditions/videos/',

```

```

        validators=[FileExtensionValidator(allowed_extensions=['mp4',
        'avi', 'mov'])],
        null=True,
        blank=True
    )
    is_temporary_audition = models.BooleanField(default=True)
    is_paid = models.BooleanField(default=False)
    expiration_date = models.DateTimeField(blank=True, null=True)

    def clean(self):
        super().clean()

        if self.audition_date and self.audition_date <=
timezone.now():
            raise ValidationError({'audition_date': 'Audition date
must be in the future.'})

    def save(self, *args, **kwargs):
        self.full_clean()
        user = kwargs.pop('user', None)

        if self.pk and user:
            old_audition = Audition.objects.get(pk=self.pk)
            old_audition_date =
timezone.localtime(old_audition.audition_date)
            new_audition_date = timezone.localtime(self.audition_date)
            if old_audition_date != new_audition_date:
                self.send_date_change_email(old_audition_date,
new_audition_date)
                self.send_date_change_notification(old_audition_date,
new_audition_date, user)
                site_config = SiteConfig.load()
                if
site_config.USE_SMS_NOTIFICATION_FOR_AUDITION_CHANGE:
                    self.send_date_change_sms(new_audition_date)

        super().save(*args, **kwargs)

    def send_date_change_email(self, old_date, new_date):
        if old_date:
            old_date = old_date.strftime("%B %d, %Y, %I:%M %p")

        subject = 'GYOCC Audition Date Changed/Confirmed'
        message = f"""
        Dear {self.auditionee.get_full_name()},

```

```

Your audition date has been changed or confirmed.
Old audition date was: {old_date}
Your audition date is now: {new_date.strftime("%B %d, %Y,
%I:%M %p")}

If you have any questions, please contact us.

Best regards,
GYOCC
"""
from_email = settings.EMAIL_HOST_USER
recipient_list = [self.auditionee.email]

send_mail(subject, message, from_email, recipient_list,
fail_silently=False)

def send_date_change_notification(self, old_date, new_date, user):
    if old_date:
        old_date = old_date.strftime("%B %d, %Y, %I:%M %p")

    message = f"Your audition date has been changed or confirmed
to {new_date.strftime('%B %d, %Y, %I:%M %p')}."
    long_message = f"""
    Old audition date was: {old_date} - Your audition date is now:
{new_date.strftime("%B %d, %Y, %I:%M %p")}
"""

    Notification.objects.create(
        notification_type=Notification.NotificationType.AUDITION,
        member=self.auditionee,
        notifier=user,
        audition=self,
        message=message,
        long_message=long_message,
        url=f'audition/{self.pk}/'
    )

def send_date_change_sms(self, new_date):
    to = self.auditionee.mobile_number
    body = f"Hello {self.auditionee.last_name}, \n\nYour audition
date has been changed or confirmed to {new_date.strftime('%B %d, %Y,
%I:%M %p')}."

    send_sms(to, body)

```

```

def has_passed(self):
    return self.audition_date < timezone.now()

def is_current(self):
    now = timezone.now()
    return self.audition_date == now

def is_upcoming(self):
    return self.audition_date > timezone.now()

def payment_status(self):
    if self.is_paid:
        return "Accepted"
    else:
        return "Pending"

def schedule_status(self):
    if self.has_passed():
        return "Audition Has Passed"
    elif self.is_current():
        return "Currently Ongoing"
    elif self.is_upcoming():
        return "Upcoming"

def __str__(self):
    return self.auditionee.get_full_name()

```

audition > views.py

```

import json
import uuid
from django.http import HttpResponseRedirect
from django.shortcuts import redirect, render
import requests
from payment.models import PaymentAudition, PaymentSession
from utilities.maya.paymentdata import PaymentBuyer, PaymentContact,
PaymentData, PaymentItem, PaymentItemAmount, PaymentRedirectUrl,
PaymentTotalAmount
from .forms import OrchestraAuditionForm, ChoirAuditionForm
from .models import Audition
from django.utils import timezone
from django.conf import settings

```

```

from main.models import SiteConfig

# Create your views here.
def audition_index(request):
    return render(request, 'audition_index.html')

def audition_orchestra(request):
    if not request.user.is_authenticated:
        return redirect("/login/")

    if request.method == 'GET':
        site_config = SiteConfig.load()
        expiration_date = timezone.now() +
    timezone.timedelta(days=site_config.TEMPORARY_AUDITION_EXPIRATION_DAYS)
    )
    form = OrchestraAuditionForm(
        initial={
            'audition_type': Audition.AuditionType.ORCHESTRA,
            'firstname': request.user.first_name,
            'lastname': request.user.last_name,
            'email': request.user.email,
            'mobile_number': request.user.mobile_number,
            'is_temporary_audition': True,
            'expiration_date': expiration_date
        }
    )

    content = {
        'form': form
    }
    return render(request, 'audition_orchestra.html', content)
elif request.method == 'POST':
    form = OrchestraAuditionForm(request.POST)
    if form.is_valid() and form.clean():
        audition = form.save()
        audition.auditionee = request.user
        audition.save()

        site_config = SiteConfig.load()

        if not site_config.ENABLE_AUDITION_FEE:
            audition.is_paid = True
            audition.save()

```

```

        return render(request,
"audition_book_payment_success.html")

    pay_now = request.GET.get('payNow', "N/A")
    pay_now = pay_now.lower() == 'true' if pay_now != "N/A"
else False

    if not pay_now:
        return redirect("/dashboard/")

    domain_url = site_config.DOMAIN_URL
    maya_encrypted_key =
settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

    random_uuid = uuid.uuid4()
    payment_data = PaymentData(
        totalAmount =
PaymentTotalAmount(value=site_config.AUDITION_FEE, currency="PHP"),
        buyer = PaymentBuyer(
            contact =
PaymentContact(email=request.user.email),
                firstName = request.user.first_name,
                lastName = request.user.last_name
            ),
        redirectUrl = PaymentRedirectUrl(
            success = domain_url +
f'audition/success?session_id={str(random_uuid)}',
            failure = domain_url + f'audition/failed',
            cancel = domain_url + f'audition/failed'
        ),
        items = [
            PaymentItem(
                totalAmount=PaymentItemAmount(value=site_config.AUDITION_FEE),
                name="Audition Fee",
                quantity="1",
                code=str(audition.pk),
                description="Audition fee for Gensan Youth
Orchestra Choir Corporation, cancellation of payment means your
audition will be labelled as temporary and will not be prioritized to
other auditions that are paid."
            )
        ],
        requestReferenceNumber=f"{str(random_uuid)}"
    )

```

```

p_data = {
    "type": "audition",
    "target": audition.pk,
    "amount_total": payment_data.totalAmount.value
}

PaymentSession.objects.create(
    buyer = request.user,
    payment_data = p_data,
    paid = False,
    session_id = str(random_uuid)
)

headers = {
    "accept": "application/json",
    "authorization": f"Basic {str(maya_encrypted_key)}",
    "content-type": "application/json"
}

response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
    return redirect(json.loads(response.text)["redirectUrl"])

content = {
    'form': form
}
    return render(request, 'audition_orchestra.html', content)
else:
    return redirect("/")
}

def audition_choir(request):
    if request.method == 'GET':
        site_config = SiteConfig.load()
        expiration_date = timezone.now() +
timezone.timedelta(days=site_config.TEMPORARY_AUDITION_EXPIRATION_DAYS)
    )
        form = ChoirAuditionForm(
            initial={
                'audition_type': Audition.AuditionType.CHOIR,
                'firstname': request.user.first_name,
                'lastname': request.user.last_name,
                'email': request.user.email,
                'mobile_number': request.user.mobile_number,
                'is_temporary_audition': True,

```

```

        'expiration_date': expiration_date
    }
)

content = {
    'form': form
}
return render(request, 'audition_choir.html', content)
elif request.method == 'POST':
    form = ChoirAuditionForm(request.POST, request.FILES)
    if form.is_valid() and form.clean():
        audition = form.save()
        audition.auditionee = request.user
        audition.save()

    site_config = SiteConfig.load()

    if not site_config.ENABLE_AUDITION_FEE:
        audition.is_paid = True
        audition.save()

    return render(request,
"audition_book_payment_success.html")

pay_now = request.GET.get('payNow', "N/A")
pay_now = pay_now.lower() == 'true' if pay_now != "N/A"
else False

if not pay_now:
    return redirect("/dashboard/")

domain_url = site_config.DOMAIN_URL
maya_encrypted_key =
settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

random_uuid = uuid.uuid4()
payment_data = PaymentData(
    totalAmount =
PaymentTotalAmount(value=site_config.AUDITION_FEE, currency="PHP"),
    buyer = PaymentBuyer(
        contact =
PaymentContact(email=request.user.email),
            firstName = request.user.first_name,
            lastName = request.user.last_name
        ),

```

```

        redirectUrl = PaymentRedirectUrl(
            success = domain_url +
f'audition/success?session_id={str(random_uuid)}',
            failure = domain_url + f'audition/failed',
            cancel = domain_url + f'audition/failed'
        ),
        items = [
            PaymentItem(
                totalAmount=PaymentItemAmount(value=site_config
g.AUDITION_FEE),
                name="Audition Fee",
                quantity="1",
                code=str(audition.pk),
                description="Audition fee for Gensan Youth
Orchestra Choir Corporation, cancellation of payment means your
audition will be labelled as temporary and will not be prioritized to
other auditions that are paid."
            )
        ],
        requestReferenceNumber=f"{str(random_uuid)}"
    )

    p_data = {
        "type": "audition",
        "target": audition.pk,
        "amount_total": payment_data.totalAmount.value
    }

    PaymentSession.objects.create(
        buyer = request.user,
        payment_data = p_data,
        paid = False,
        session_id = str(random_uuid)
    )

    headers = {
        "accept": "application/json",
        "authorization": f"Basic {str(maya_encrypted_key)}",
        "content-type": "application/json"
    }

    response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
    return redirect(json.loads(response.text)["redirectUrl"])

```

```

        content = {
            'form': form
        }
        return render(request, 'audition_choir.html', content)
    else:
        return redirect("/")
}

def audition_payment_successful(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession and not paymentSession.paid:
                paymentSession.paid = True
                paymentSession.save()
                target_audition =
paymentSession.payment_data["target"]
                audition = Audition.objects.get(id=target_audition)
                if audition:
                    audition.is_paid = True
                    audition.save()

                    PaymentAudition.objects.create(
                        buyer = audition.auditionee,
                        audition = audition,
                        amount =
paymentSession.payment_data["amount_total"],
                        checkout_id=checkout_id
                    )

                    return render(request,
"audition_book_payment_success.html")
                else:
                    return HttpResponse(f"Audition with id
{target_audition} not found!")
            elif paymentSession and paymentSession.paid:
                return render(request,
"audition_book_payment_success.html")
            else:
                return HttpResponse("Stop trying to blindly guess
session ids..")
        else:
            return redirect("/dashboard/")

```

```

def audition_payment_failed(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
                PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession:
                paymentSession.paid = False
                paymentSession.save()
                target_audition =
                    paymentSession.payment_data["target"]
                audition = Audition.objects.get(id=target_audition)
                if audition:
                    audition.is_paid = False
                    audition.is_temporary_audition = True
                    audition.save()

                return render(request, "payment_cancelled.html")
            else:
                return HttpResponse(f"Audition with id {target_audition} not found!")
        else:
            return HttpResponse("Stop trying to blindly guess session ids..")
    else:
        return redirect("/dashboard/")

```

event > models.py

```

from django.conf import settings
from django.db import models
from django.forms import ValidationError
from main.models import Member, Notification, SiteConfig
from django.core.validators import FileExtensionValidator
from django.utils import timezone
from colorfield.fields import ColorField
from django.db.models import Q
from django.core.mail import send_mail
from django.utils.translation import gettext_lazy as _

from utilities.sms import send_sms

# Create your models here.
class Event(models.Model):
    # Django auto generates id = AutoField(primary_key=True

```

```

class EventType(models.TextChoices):
    PRIVATE = "Private", "Private"
    PUBLIC = "Public", "Public"

    organizer = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name='organized_events', blank=True, null=True)
    requestor = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name='requested_events')
    name = models.CharField(max_length=255, unique=True)
    event_type = models.CharField('Event Type',
choices=EventType.choices, max_length=20, default=EventType.PUBLIC)
    description = models.TextField()
    image_banner = models.ImageField(
        upload_to='event_banners/',
        validators=[FileExtensionValidator(['jpg', 'jpeg', 'png'])],
        blank=True,
        null=True,
        verbose_name="Image Banner"
    )
    ticket_image_design = models.ImageField(
        upload_to='ticket_image_designs/',
        validators=[FileExtensionValidator(['jpg', 'jpeg', 'png'])],
        blank=True,
        null=True,
        verbose_name="Ticket Image Design"
    )
    start_date = models.DateTimeField()
    end_date = models.DateTimeField()
    location = models.TextField()

    # Seats
    vip_seats = models.IntegerField(default=0, verbose_name="VIP Seats",
blank=True, null=True)
    gold_seats = models.IntegerField(default=0, verbose_name="Gold Seats",
blank=True, null=True)
    silver_seats = models.IntegerField(default=0, verbose_name="Silver Seats",
blank=True, null=True)
    general_admission_seats = models.IntegerField(default=0,
verbose_name="General Admission Seats", blank=True, null=True)

    # Costs
    vip_ticket_cost = models.DecimalField(max_digits=10,
verbose_name="VIP Cost", decimal_places=2, blank=True, null=True)
    gold_ticket_cost = models.DecimalField(max_digits=10,
verbose_name="Gold Cost", decimal_places=2, blank=True, null=True)

```

```

silver_ticket_cost = models.DecimalField(max_digits=10,
verbose_name="Silver Cost", decimal_places=2, blank=True, null=True)
general_ticket_cost = models.DecimalField(max_digits=10,
verbose_name="General Admission Cost", decimal_places=2, blank=True,
null=True)

calendar_color = ColorField(default='#3868A6',
verbose_name="Calendar Color")
date_added = models.DateTimeField(auto_now_add=True, blank=True)

def has_passed(self):
    return self.end_date < timezone.now()

def is_current(self):
    now = timezone.now()
    return self.start_date <= now <= self.end_date

def is_upcoming(self):
    return self.start_date > timezone.now()

def __str__(self):
    if len(self.name) <= 0:
        return "No set name"
    return self.name

def clean(self):
    super().clean()
    if self.start_date and self.end_date:

        if self._state.adding:
            if self.start_date < timezone.now() or self.end_date <
timezone.now():
                raise ValidationError('Start date or end date must
be greater than current time.')

            if self.start_date >= self.end_date:
                raise ValidationError('Start date must be before the
end date.')

        overlapping_events = Event.objects.filter(
            Q(start_date__lt=self.end_date) &
            Q(end_date__gt=self.start_date)
        ).exclude(pk=self.pk)

        if overlapping_events.exists():

```

```

        raise ValidationError('This event overlaps with
existing events.')

    elif bool(self.start_date) != bool(self.end_date):
        raise ValidationError('Both start date and end date must
be set.')

@classmethod
def get_events_with_vip_occupancy(cls):
    return Event.upcoming_or_ongoing().filter(vip_seats__gt=0)

@classmethod
def get_events_with_gold_occupancy(cls):
    return Event.upcoming_or_ongoing().filter(gold_seats__gt=0)

@classmethod
def get_events_with_silver_occupancy(cls):
    return Event.upcoming_or_ongoing().filter(silver_seats__gt=0)

@classmethod
def get_events_with_general_admission_occupancy(cls):
    return
Event.upcoming_or_ongoing().filter(general_admission_seats__gt=0)

@classmethod
def upcoming_or_ongoing(cls):
    now = timezone.now()
    return cls.objects.filter(models.Q(start_date__gte=now) |
models.Q(end_date__gte=now))

class Book(models.Model):
    # Django auto generates id = AutoField(primary_key=True)
    booker = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="member_book", blank=True, null=True)
    name = models.CharField(max_length=255, unique=True)
    description = models.TextField()
    location = models.TextField()
    meet_date = models.DateTimeField()
    event_type = models.CharField('Event Type',
choices=Event.EventType.choices, max_length=20,
default=Event.EventType.PUBLIC)
    is_temporary_booking = models.BooleanField(default=True)
    is_paid = models.BooleanField(default=False)
    is_time_reserved = models.BooleanField(default=False)
    temporary_expiration_date = models.DateTimeField()

```

```

date_submitted = models.DateTimeField(auto_now_add=True,
blank=True)

def has_passed(self):
    return self.meet_date < timezone.now()

def is_current(self):
    now = timezone.now()
    return self.meet_date == now

def is_upcoming(self):
    return self.meet_date > timezone.now()

def payment_status(self):
    if self.is_paid:
        return "Paid"
    else:
        return "Not Paid"

def schedule_status(self):
    if self.has_passed():
        return "Booking Has Passed"
    elif self.is_current():
        return "Currently Ongoing"
    elif self.is_upcoming():
        return "Upcoming"

def clean(self):
    super().clean()

site_config = SiteConfig.load()

meet_date = timezone.localtime(self.meet_date)

start_of_day = meet_date.replace(hour=0, minute=0, second=0,
microsecond=0)
end_of_day = meet_date.replace(hour=23, minute=59, second=59,
microsecond=999999)

bookings_for_day =
Book.objects.filter(meet_date__range=(start_of_day, end_of_day),
is_paid=True).exclude(pk=self.pk)
if bookings_for_day.count() >=
site_config.MAX_BOOKING_PER_DAY:

```

```

        raise ValidationError(_("The maximum number of 5 bookings
for this day has been reached."))

    for booking in bookings_for_day:
        existing_meet_date = timezone.localtime(booking.meet_date)
        time_diff = abs((existing_meet_date -
meet_date).total_seconds()) / (site_config.MIN_BOOKING_HOUR_GAP * 60 * 60)
        if time_diff < 1:
            raise ValidationError(_("There must be at least a 1-
hour gap between bookings."))

    def save(self, *args, **kwargs):
        self.full_clean()
        user = kwargs.pop('user', None)

        if self.pk and user:
            old_booking = Book.objects.get(pk=self.pk)
            old_booking_date =
timezone.localtime(old_booking.meet_date)
            new_booking_date = timezone.localtime(self.meet_date)
            if old_booking_date != new_booking_date:
                self.send_date_change_email(old_booking_date,
new_booking_date)
                self.send_date_change_notification(old_booking_date,
new_booking_date, user)
                site_config = SiteConfig.load()
                if
site_config.USE_SMS_NOTIFICATION_FOR_BOOKING_CHANGE:
                    self.send_date_change_sms(new_booking_date)

        super().save(*args, **kwargs)

    def send_date_change_email(self, old_date, new_date):
        if old_date:
            old_date = old_date.strftime("%B %d, %Y, %I:%M %p")

        subject = 'GYOCC Booking Meeting Date Changed'
        message = f"""
Dear {self.booker.get_full_name()},

Your booking: {self.name}'s meeting date has been changed.
Old meeting date was: {old_date}
Your new meeting date is now: {new_date.strftime("%B %d, %Y,
%I:%M %p")}

"""

```

If you have any questions, please contact us.

Best regards,
GYOCC

```

from_email = settings.EMAIL_HOST_USER
recipient_list = [self.booker.email]

    send_mail(subject, message, from_email, recipient_list,
fail_silently=False)

def send_date_change_notification(self, old_date, new_date, user):
    if old_date:
        old_date = old_date.strftime("%B %d, %Y, %I:%M %p")

        message = f"Your booking {self.name}'s meeting date has been
changed to {new_date.strftime('%B %d, %Y, %I:%M %p')}."
        long_message = f"""
        Old meeting date was: {old_date} - Your meeting date is now:
{new_date.strftime("%B %d, %Y, %I:%M %p")}
"""

        Notification.objects.create(
            notification_type=Notification.NotificationType.BOOKING,
            member=self.booker,
            notifier=user,
            book=self,
            message=message,
            long_message=long_message,
            url=f'history/'
        )

    def send_date_change_sms(self, new_date):
        to = self.booker.mobile_number
        body = f"Hello {self.booker.last_name}, \n\nYour booking
\"{self.name}\"'s meeting date has been changed to
{new_date.strftime('%B %d, %Y, %I:%M %p')}."

        send_sms(to, body)

    def __str__(self):
        if len(self.name) <= 0:
            return "No set name"
        return self.name

```

event > views.py

```
from decimal import Decimal
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect, JsonResponse
from .forms import EventBookForm
from .models import Book
from django.conf import settings
from main.models import Member, Notification, SiteConfig
from phonenumber_field.phonenumber import PhoneNumber
from utilities.maya.paymentdata import PaymentData, PaymentBuyer,
PaymentContact, PaymentItem, PaymentItemAmount, PaymentRedirectUrl,
PaymentTotalAmount
from payment.models import PaymentBook, PaymentEvent, PaymentSession
import datetime, requests, json, uuid
from django.utils.timezone import localtime, make_aware
from django.utils import timezone

# Create your views here.
def event_index(request):
    if request.method == "GET":

        if not request.user.is_authenticated:
            return redirect("/login/")

        now = datetime.date.today()
        site_config = SiteConfig.load()
        expiration_date = now +
            datetime.timedelta(days=site_config.TEMPORARY_BOOKING_EXPIRATION_DAYS)
        ebf = EventBookForm(initial={
            'firstname': request.user.first_name,
            'lastname': request.user.last_name,
            'email': request.user.email,
            'mobile_number': request.user.mobile_number,
            'is_temporary_booking': True,
            'temporary_expiration_date': expiration_date
        })
        content = {
            "form": ebf
        }
        return render(request, 'event_book.html', content)

    elif request.method == "POST":
        if not request.user.is_authenticated:
            return redirect("/login/")
```

```

        form = EventBookForm(request.POST)
        selected_date = request.POST.get('date')
        selected_time = request.POST.get('time')
        available_times = get_available_times(request, selected_date)
        form.set_time_choices(json.loads(available_times.content)[
            'available_times'])

        time_obj = datetime.datetime.strptime(selected_time, "%I:%M
%p").time()
        date_obj = datetime.datetime.strptime(selected_date, "%Y-%m-
%d").date()
        meet_datetime =
timezone.make_aware(datetime.datetime.combine(date_obj, time_obj))
        form.fields['meet_date'].initial = meet_datetime

        # Check if meet_datetime is currently reserved
        if Book.objects.filter(meet_date=meet_datetime,
is_time_reserved=True).exists():
            book = Book.objects.filter(meet_date=meet_datetime,
is_time_reserved=True).first()
            paymentSession =
PaymentSession.objects.filter(payment_data__contains={"type": "book",
"target": book.id}).first()

            if not paymentSession.is_expired():
                content = {
                    "wait_hours": paymentSession.get_time_remaining()
                }
                return render(request,
'event_book_failed_already_reserved.html', content)

        else:
            book.is_time_reserved = False
            book.save()

        if form.is_valid() and form.clean():
            new_book = form.save()

            new_book.booker = request.user
            new_book.save()

            pay_now = request.GET.get('payNow', "N/A")
            pay_now = pay_now.lower() == 'true' if pay_now != "N/A"
else False
    
```

```

if not pay_now:
    return redirect("/dashboard/")

site_config = SiteConfig.load()
domain_url = site_config.DOMAIN_URL
maya_encrypted_key =
settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

random_uuid = uuid.uuid4()
payment_data = PaymentData(
    totalAmount =
PaymentTotalAmount(value=float(site_config.BOOKING_FEE),
currency="PHP"),
    buyer = PaymentBuyer(
        contact =
PaymentContact(email=request.user.email),
            firstName = request.user.first_name,
            lastName = request.user.last_name
        ),
    redirectUrl = PaymentRedirectUrl(
        success = domain_url +
f'event/success?session_id={str(random_uuid)}',
        failure = domain_url + f'event/failed',
        cancel = domain_url + f'event/failed'
    ),
    items = [
        PaymentItem(
            totalAmount=PaymentItemAmount(value=float(site
_config.BOOKING_FEE)),
            name="Appointment Booking",
            quantity="1",
            code=str(new_book.pk),
            description="Appointment booking fee for
Gensan Youth Orchestra Choir Corporation, cancellation of payment
means your booking will be labelled as pencil booking and will not be
prioritized to other bookings that are paid."
        )
    ],
    requestReferenceNumber=f"{str(random_uuid)}"
)

p_data = {
    "type": "book",
    "target": new_book.pk,

```

```

        "amount_total": float(payment_data.totalAmount.value)
    }

PaymentSession.objects.create(
    buyer = request.user,
    payment_data = p_data,
    paid = False,
    session_id = str(random_uuid)
)

new_book.is_time_reserved = True
new_book.save()

headers = {
    "accept": "application/json",
    "authorization": f"Basic {str(maya_encrypted_key)}",
    "content-type": "application/json"
}

response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
return redirect(json.loads(response.text)["redirectUrl"])

else:
    content = {
        "form": form,
    }
    return render(request, 'event_book.html', content)
return HttpResponse("Currently in development")

def event_booking_payment_successful(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession and not paymentSession.paid:
                paymentSession.paid = True
                paymentSession.save()
                target_event = paymentSession.payment_data["target"]
                book = Book.objects.get(id=target_event)
                if book:
                    book.is_paid = True
                    book.save()

```

```

        other_books =
Book.objects.filter(meet_date=book.meet_date).exclude(id=book.id)
        if other_books:
            for bk in other_books:
                Notification.objects.create(
                    member=book.booker,
                    notifier=book.booker,
                    notification_type=Notification.NotificationType.EVENT,
                    message=f'Your temporary booking: {book.name} on {book.meet_date} was deleted.',
                    long_message=f'The booking was deleted due to a payment made by someone for their booking with the same date and time as yours.',
                    url = "history/",
                    event = None
                )
                bk.delete()

        PaymentBook.objects.create(
            buyer = book.booker,
            book = book,
            amount =
paymentSession.payment_data["amount_total"],
            checkout_id=checkout_id
        )

        return render(request, "payment_success.html")
    else:
        return HttpResponse(f"Booking with id {target_event} not found!")
    elif paymentSession and paymentSession.paid:
        return render(request, "payment_success.html")
    else:
        return HttpResponse("Stop trying to blindly guess session ids..")
    else:
        return redirect("/dashboard/")

def event_booking_payment_failed_or_cancelled(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if not checkout_id == "N/A":
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)

```

```

        if paymentSession:
            paymentSession.paid = False
            paymentSession.save()
            target_event = paymentSession.payment_data["target"]
            book = Book.objects.get(id=target_event)
            if book:
                book.is_paid = False
                book.is_temporary_booking = True
                book.save()

                    return render(request, "payment_cancelled.html")
            else:
                return HttpResponse(f"Booking with id {target_event} not found!")
            else:
                return HttpResponse("Stop trying to blindly guess session ids..")
            else:
                return render(request, "payment_cancelled.html")

def get_available_times(request, date_selected=None):
    selected_date = None
    if not date_selected:
        selected_date = request.GET.get('date')
    else:
        selected_date = date_selected
    site_config = SiteConfig.load()

    if not selected_date:
        return JsonResponse({"error": "No date provided"}, status=400)

    try:
        selected_date = datetime.datetime.strptime(selected_date, "%Y-%m-%d").date()
    except ValueError:
        return JsonResponse({"error": "Invalid date format"}, status=400)

    start_time =
    datetime.time(site_config.START_OF_BOOKING_HOURS.hour,
    site_config.START_OF_BOOKING_HOURS.minute)
    end_time = datetime.time(site_config.END_OF_BOOKING_HOURS.hour,
    site_config.END_OF_BOOKING_HOURS.minute)
    time_delta =
    datetime.timedelta(hours=site_config.MIN_BOOKING_HOUR_GAP)

```

```

available_times = []

tz = timezone.get_current_timezone()

start_datetime =
make_aware(datetime.datetime.combine(selected_date, start_time), tz)
end_datetime = make_aware(datetime.datetime.combine(selected_date,
end_time), tz)

current_datetime = start_datetime
while current_datetime <= end_datetime:
    if Book.objects.filter(meet_date=current_datetime,
is_time_reserved=True).exists():
        book = Book.objects.filter(meet_date=current_datetime,
is_time_reserved=True).first()
        paymentSession =
PaymentSession.objects.filter(payment_data__contains={"type": "book",
"target": book.id}).first()
        if paymentSession.is_expired():
            book.is_time_reserved = False
            book.save()

        if not Book.objects.filter(meet_date=current_datetime,
is_paid=True).exists() and not
Book.objects.filter(meet_date=current_datetime,
is_time_reserved=True).exists():
            start_of_day =
make_aware(datetime.datetime.combine(selected_date,
datetime.time.min), tz)
            end_of_day =
make_aware(datetime.datetime.combine(selected_date,
datetime.time.max), tz)

            bookings_for_day =
Book.objects.filter(meet_date__range=(start_of_day, end_of_day))
            if bookings_for_day.count() >=
site_config.MAX_BOOKING_PER_DAY:
                return JsonResponse({"available_times": []})

            available_times.append(localtime(current_datetime).strftime
("%I:%M %p"))
            current_datetime += time_delta

return JsonResponse({"available_times": available_times})

```

```

def start_payment_for_private_event_follow_up(request, book_id,
member_id, paymentEventId, amount, total_amount):
    if request.method == 'GET':
        amount = float(amount)
        total_amount = float(total_amount)

        book = Book.objects.get(id=book_id)
        member = Member.objects.get(id=member_id)
        paymentEvent = PaymentEvent.objects.get(id=paymentEventId)

        site_config = SiteConfig.load()
        domain_url = site_config.DOMAIN_URL
        maya_encrypted_key = settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

        random_uuid = uuid.uuid4()
        payment_data = PaymentData(
            totalAmount = PaymentTotalAmount(value=float(amount),
currency="PHP"),
            buyer = PaymentBuyer(
                contact = PaymentContact(email=member.email),
                firstName = member.first_name,
                lastName = member.last_name
            ),
            redirectUrl = PaymentRedirectUrl(
                success = domain_url +
f'event/private/follow_up/success?session_id={str(random_uuid)}',
                failure = domain_url +
f'event/private/follow_up/failed',
                cancel = domain_url +
f'event/private/follow_up/failed'
            ),
            items = [
                PaymentItem(
                    totalAmount=PaymentItemAmount(value=float(amount))

,
                    name=f"Follow-up Payment for Private Event:
{book.name}",
                    quantity="1",
                    code=str(book.id),
                    description="Online payment procedure for private
events."
                )
            ],
            requestReferenceNumber=f"{str(random_uuid)}"
        )

```

```

p_data = {
    "type": "private_event_follow_up",
    "target": paymentEvent.id,
    "amount_paid": float(payment_data.totalAmount.value),
    "amount_total": float(total_amount)
}

PaymentSession.objects.create(
    buyer = member,
    payment_data = p_data,
    paid = False,
    session_id = str(random_uuid)
)

headers = {
    "accept": "application/json",
    "authorization": f"Basic {str(maya_encrypted_key)}",
    "content-type": "application/json"
}

response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
return redirect(json.loads(response.text)["redirectUrl"])

def private_event_payment_successful_follow_up(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession and not paymentSession.paid:
                paymentSession.paid = True
                paymentSession.save()
                target_paymentEvent =
paymentSession.payment_data["target"]
                amount_paid =
paymentSession.payment_data["amount_paid"]

                paymentEvent =
PaymentEvent.objects.get(id=target_paymentEvent)
                paymentEvent.payment_amount +=
Decimal(str(amount_paid))
                print(paymentEvent.payment_amount)
                paymentEvent.save()

```

```

        return render(request, "payment_success.html")
    elif paymentSession and paymentSession.paid:
        return render(request, "payment_success.html")
    else:
        return HttpResponse("Stop trying to blindly guess
session ids..")
    else:
        return redirect("/dashboard/")

def private_event_payment_failed_or_cancelled_follow_up(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if not checkout_id == "N/A":
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession:
                paymentSession.paid = False
                paymentSession.save()
                return render(request, "payment_cancelled.html")
            else:
                return HttpResponse("Stop trying to blindly guess
session ids..")
        else:
            return render(request, "payment_cancelled.html")

```

feedback > models.py

```

from django.db import models
from main.models import Member
from lesson.models import Lesson
from event.models import Event
from django.core.validators import MinValueValidator,
MaxValueValidator
import uuid

# Create your models here.
class Feedback(models.Model):
    class FeedbackType(models.TextChoices):
        Lesson = "Lesson", "Lesson"
        Event = "Event", "Event"
        Instructor = "Instructor", "Instructor"

```

```

member = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="feedback_member")
feedback_type = models.CharField(choices=FeedbackType.choices,
default=FeedbackType.Lesson)
lesson = models.ForeignKey(Lesson, on_delete=models.CASCADE,
related_name="feedback_lesson", blank=True, null=True)
event = models.ForeignKey(Event, on_delete=models.CASCADE,
related_name="feedback_event", blank=True, null=True)
instructor = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="feedback_event", blank=True, null=True)
rating = models.PositiveIntegerField(default=0,
validators=[MinValueValidator(0), MaxValueValidator(5)])
date_submitted = models.DateTimeField(auto_now_add=True)
content = models.TextField()
key = models.UUIDField(blank=True, null=True)

```

feedback > views.py

```

import json
from django.http import JsonResponse
from django.shortcuts import render
from .models import Feedback
from lesson.models import Lesson
from main.models import Member

# Create your views here.
def feedback_form(request, key):
    return render(request, "feedback_form.html")

def submit_instructor_review(request):
    if request.method == 'POST':
        data = json.loads(request.body)
        instructor_id = data.get('instructor_id')
        member_id = data.get('member_id')
        rating = data.get('rating')
        content = data.get('content')

        member = Member.objects.get(id=member_id)
        instructor = Member.objects.get(id=instructor_id)
        lessons = Lesson.objects.filter(instructor=instructor)
        found_history = False

        for lesson in lessons:

```

```

        if member in lesson.students.all():
            if lesson.has_passed():
                found_history = True

        if not found_history:
            return JsonResponse({
                'already_rated': True,
                'message': 'You have not attended any of this
instructor\'s lessons, as such you can\'t leave a rating.'
            })

        if Feedback.objects.filter(
            member = member,
            instructor = instructor
        ).exists():
            return JsonResponse({
                'already_rated': True,
                'message': 'You have already rated.'
            })

        if member == instructor:
            return JsonResponse({
                'already_rated': True,
                'message': 'You can\'t rate yourself.'
            })

        Feedback.objects.create(
            member = member,
            instructor = instructor,
            rating = rating,
            content = content,
            feedback_type = Feedback.FeedbackType.Instructor,
        )

        return JsonResponse({
            'already_rated': False
        })
    
```

lesson > models.py

```
from django.db import models
from django.forms import ValidationError
from main.models import Member
from django.core.validators import FileExtensionValidator
from django.utils import timezone
from django.db.models import Q
from colorfield.fields import ColorField
from event.models import Event, Book
from django_ckeditor_5.fields import CKEditor5Field

from resource.models import Item, ItemHistory

# Create your models here.

class Instrument(models.Model):
    name = models.CharField(max_length=255, unique=True)
    description = models.TextField(blank=True)
    created_at = models.DateField(auto_now_add=True)

    def __str__(self):
        return f"{self.name}"

class Profile(models.Model):
    instructor = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name='instructor_profile')
    introduction = models.TextField()
    age = models.IntegerField()
    instruments = models.ManyToManyField(Instrument,
related_name="instructor_instruments", blank=True)

class Lesson(models.Model):
    instrument = models.ForeignKey(Instrument,
on_delete=models.CASCADE, related_name='instruments')
    instructor = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name='instructor', blank=True, null=True)
    students = models.ManyToManyField(Member, related_name="students",
blank=True)
    items = models.ManyToManyField(Item, related_name="lesson_items",
blank=True)
    lesson_banner = models.ImageField(
        upload_to='lesson_banners/',
        validators=[FileExtensionValidator(['jpg', 'jpeg', 'png'])],
        blank=True,
```

```

        null=True,
        verbose_name="Image Banner"
    )
    start_date = models.DateTimeField(null=False)
    end_date = models.DateTimeField(null=False)
    date_added = models.DateTimeField(auto_now_add=True, blank=True)
    location = models.TextField(blank=True)
    max_slots = models.PositiveIntegerField(default=10)
    lesson_cost = models.DecimalField(max_digits=10,
        verbose_name="Lesson Cost", decimal_places=2, blank=True, null=True)
    calendar_color = ColorField(default='#3868A6',
        verbose_name="Calendar Color")

    def slots_used(self):
        return self.students.count()

    def has_passed(self):
        return self.end_date < timezone.now()

    def is_current(self):
        now = timezone.now()
        return self.start_date <= now <= self.end_date

    def is_upcoming(self):
        return self.start_date > timezone.now()

    def schedule_status(self):
        if self.has_passed():
            return "Finished"
        elif self.is_current():
            return "Ongoing"
        elif self.is_upcoming():
            return "Upcoming"

    def clean(self):
        super().clean()
        is_new = self._state.adding
        if not is_new:
            if self.instructor:
                if not self.instructor.is_in_instructor_group():
                    raise ValidationError("Selected user must be an
instructor.")

        if self.start_date and self.end_date:

```

```

        if self.start_date < timezone.now() or self.end_date <
timezone.now():
            raise ValidationError('Start date or end date must be
greater than current time.')

        if self.start_date >= self.end_date:
            raise ValidationError('Start date must be before the
end date.')

        overlapping_lessons = Lesson.objects.filter(
            Q(start_date__lt=self.end_date) &
            Q(end_date__gt=self.start_date) & Q(instructor=self.instructor)
        ).exclude(pk=self.pk)

        if overlapping_lessons.exists():
            raise ValidationError('This lesson schedule overlaps
with existing lessons.')

    elif bool(self.start_date) != bool(self.end_date):
        raise ValidationError('Both start date and end date must
be set.')

    def save(self, *args, **kwargs):
        is_new = self._state.adding
        super(Lesson, self).save(*args, **kwargs)

        if self.has_passed():
            return

        if not is_new:
            ItemHistory.objects.filter(lesson=self).delete()

        for item in self.items.all():
            ItemHistory.objects.create(
                item=item,
                date_used=timezone.now(),
                condition=item.item_condition,
                lesson=self
            )

    def __str__(self):
        return f'{self.instrument} - {self.instructor}'

    @classmethod
    def upcoming_or_ongoing(cls):

```

```

        now = timezone.now()
        return cls.objects.filter(models.Q(start_date__gte=now) |
models.Q(end_date__gte=now))

class LessonAttendance(models.Model):
    lesson = models.ForeignKey(Lesson, on_delete=models.CASCADE,
related_name="attendances")
    student = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="attendances")
    attended_at = models.DateTimeField(auto_now_add=True)
    confirmed_by = models.ForeignKey(Member,
on_delete=models.SET_NULL, null=True, blank=True,
related_name="attendance_confirmations")

    def __str__(self):
        return f"{self.student} attended {self.lesson} on
{self.attended_at}"

    class Meta:
        verbose_name = "Lesson Attendance"
        verbose_name_plural = "Lesson Attendances"
        constraints = [
            models.UniqueConstraint(fields=['lesson', 'student'],
name='unique_lesson_student')
        ]

class RepertoireArticle(models.Model):

    class Meta:
        verbose_name = "Repertoire Article"
        verbose_name_plural = "Repertoire Articles"

        author = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="repertoire_author", blank=True, null=True)
        instrument = models.ForeignKey(Instrument,
on_delete=models.CASCADE, related_name="repertoire_instrument",
blank=True, null=True)
        title = models.CharField(max_length=255)
        description = models.TextField(blank=True)
        content = CKEditor5Field('Content', config_name='default')
        date_posted = models.DateTimeField(auto_now_add=True, blank=True)
        article_banner = models.ImageField(
            upload_to='article_banners/',
            validators=[FileExtensionValidator(['jpg', 'jpeg', 'png'])],

```

```

        blank=True,
        null=True,
        verbose_name="Article Banner"
    )

    def __str__(self):
        return self.title

```

lesson > views.py

```

from django.utils import timezone
import json
import uuid
from django.conf import settings
from django.http import HttpResponseRedirect, JsonResponse
from django.shortcuts import redirect, render
import requests
from feedback.models import Feedback
from main.models import Member, SiteConfig
from payment.models import PaymentLesson, PaymentSession
from utilities.maya.paymentdata import PaymentBuyer, PaymentContact,
PaymentData, PaymentItem, PaymentItemAmount, PaymentRedirectUrl,
PaymentTotalAmount
from .models import Lesson, Instrument, Profile, RepertoireArticle
from .forms import InstrumentSelectForm
from django.contrib.auth.decorators import login_required
from django.utils.dateformat import DateFormat

# Create your views here.
@login_required
def get_available_instructors(request, pk):
    if request.method == 'GET':
        instrument = Instrument.objects.get(id=pk)
        upcoming_lessons = Lesson.upcoming_or_ongoing()
        lessons = upcoming_lessons.filter(instrument=instrument)
        seen = []

        data = {'data': []}

        for lesson in lessons:
            if lesson.instructor is None:
                continue

```

```

        if not lesson.instructor in seen:
            seen.append(lesson.instructor)

            instructor_name = lesson.instructor.__str__()

            data['data'].append({
                'id': lesson.instructor.id,
                'name': instructor_name
            })

    return JsonResponse(data)

return JsonResponse({'data': []})

@login_required
def get_available_schedules(request, pk, instructor_id):
    if request.method == 'GET':
        instrument = Instrument.objects.get(id=pk)
        instructor = Member.objects.filter(id=instructor_id).first()
        upcoming_lessons = Lesson.upcoming_or_ongoing()
        lessons = upcoming_lessons.filter(instrument=instrument,
instructor=instructor)

        data = {'data': []}

        for lesson in lessons:
            if lesson.instructor is None:
                continue

            if request.user in lesson.students.all():
                continue

            start_date_local = timezone.localtime(lesson.start_date)
            end_date_local = timezone.localtime(lesson.end_date)
            formatted_start_date =
DateFormat(start_date_local).format("M j, Y, g:i a")
            formatted_end_date = DateFormat(end_date_local).format("M
j, Y, g:i a")

            data['data'].append({
                'id': lesson.id,
                'time': f'{formatted_start_date} -
{formatted_end_date}',
                'cost': lesson.lesson_cost,
                'location': lesson.location
            })

```

```

        })

    return JsonResponse(data)

return JsonResponse({'data': []})

def index(request):
    if not request.user.is_authenticated:
        return redirect("/login/")

    if request.method == "GET":
        instrumentSelectForm = InstrumentSelectForm()

        instrument_id = request.GET.get('instrument')
        instructor_id = request.GET.get('instructor')
        lesson_id = request.GET.get('lesson')

        if instrument_id:
            instrumentSelectForm.fields['instrument'].initial =
instrument_id

            content = {
                "form": instrumentSelectForm
            }
            return render(request, 'lesson_index.html', content)

    elif request.method == "POST":
        form = InstrumentSelectForm(request.POST)
        if form.is_valid():
            instrument = form.cleaned_data['instrument']
            lesson_id = request.POST.get('schedule')
            lesson = Lesson.objects.get(id=lesson_id)

            site_config = SiteConfig.load()
            domain_url = site_config.DOMAIN_URL
            maya_encrypted_key =
settings.MAYA_ENCRYPTED_BASIC_CREDENTIALS

            random_uuid = uuid.uuid4()
            payment_data = PaymentData(
                totalAmount =
PaymentTotalAmount(value=float(lesson.lesson_cost), currency="PHP"),
                buyer = PaymentBuyer(

```

```

        contact =
PaymentContact(email=request.user.email),
            firstName = request.user.first_name,
            lastName = request.user.last_name
        ),
        redirectUrl = PaymentRedirectUrl(
            success = domain_url +
f'lesson/success?session_id={str(random_uuid)}',
            failure = domain_url + f'lesson/failed',
            cancel = domain_url + f'lesson/failed'
        ),
        items = [
            PaymentItem(
                totalAmount=PaymentItemAmount(value=float(less
on.lesson_cost)),
                name=f'{lesson.instructor.__str__()} - '
{lesson.instrument.name}',
                code=lesson_id,
                quantity="1",
                description=f"You are paying to join the
{lesson.instrument.name} lesson of {lesson.instructor.__str__()}!"
            )
        ],
        requestReferenceNumber=f'{str(random_uuid)}'
    )

p_data = {
    "type": "lesson",
    "data": {
        "instrument": lesson.instrument.name,
        "lesson_id": lesson.id
    },
    "amount_total": float(payment_data.totalAmount.value)
}

PaymentSession.objects.create(
    buyer = request.user,
    payment_data = p_data,
    paid = False,
    session_id = str(random_uuid)
)

headers = {
    "accept": "application/json",
    "authorization": f"Basic {str(maya_encrypted_key)}",
}

```

```

        "content-type": "application/json"
    }

    response = requests.post(settings.MAYA_CHECKOUT_URL,
json=json.loads(payment_data.to_json()), headers=headers)
    return redirect(json.loads(response.text)["redirectUrl"])

else:
    content = {
        "form": form
    }
    return render(request, 'lesson_index.html', content)

return render(request, 'lesson_index.html')

def instructor_dashboard_specific(request, pk):
    if not request.user.is_authenticated:
        return redirect("/login/")

    if request.method == 'GET':
        if pk:
            instructor = Member.objects.get(id=pk)
            if not instructor.groups.filter(name__in=['Administrator',
'Superuser', 'Instructor']).exists():
                content = {
                    'title': 'This user is not an Instructor.',
                    'message': 'You can\'t access this page as this
user is not an instructor and can\'t be rated.'
                }
                return render(request,
'db_new/side_bar/dashboard_message.html', content)
            ratings = []
            feedbacks =
Feedback.objects.filter(instructor=instructor).order_by('-
date_submitted')
            content = {}
            for feedback in feedbacks:
                ratings.append(
{
    'name': feedback.member.get_full_name(),
    'rating': feedback.rating,
    'text': feedback.content,
    'date': feedback.date_submitted.strftime("%b.
%d, %Y, %I:%M %p")
}
        }

```

```

        )

    content = {}
    content["instructor"] = instructor

    try:
        profile = Profile.objects.get(instructor=instructor)
    except Profile.DoesNotExist:
        profile = None
    if profile:
        favorite_instruments = ' | '.join([x.name for x in
profile.instruments.all()])
        content["profile"] = profile
        content["favorite_instruments"] = favorite_instruments

    if not ratings:
        content["has_ratings"] = False
    else:
        content["has_ratings"] = True
        content["data"] = json.dumps(ratings)
    return render(request, 'lesson_instructor_overview.html',
content)

    return render(request, 'lesson_instructor_overview.html')

def repertoire_index(request):
    if not request.user.is_authenticated:
        return redirect("/login/")

    instruments = Instrument.objects.all()
    articles = RepertoireArticle.objects.all()

    data = {}
    for instrument in instruments:
        data[instrument.name] = {}
        data[instrument.name]['title'] = f'Available articles for
{instrument.name}'
        data[instrument.name]['description'] =
f'{instrument.description}'
        data[instrument.name]['courses'] = []

    for article in articles:
        if article.instrument == instrument:
            try:
                data[instrument.name]['courses'].append(

```

```

        {
            'id': article.id,
            'image': article.article_banner.url,
            'title': article.title,
            'instructor':
article.author.get_full_name(),
                'description': article.description
            }
        )
    except ValueError:
        data[instrument.name]['courses'].append(
        {
            'id': article.id,
            'image': "/static/" +
settings.JAZZMIN_SETTINGS['site_logo'],
            'title': article.title,
            'instructor':
article.author.get_full_name(),
                'description': article.description
            }
        )
    )

content = {
    'instruments': instruments,
    'data': json.dumps(data),
}

```

return render(request, 'repertoire_index.html', content)

```

def repertoire_specific(request, pk):
    if not request.user.is_authenticated:
        return redirect("/login/")

    if pk:
        article = RepertoireArticle.objects.get(id=pk)
        if article:
            context = {
                'article': article
            }

            return render(request, 'repertoire_specific.html',
context)
        else:
            return HttpResponse('Article not found.')
    
```

```

        return render(request, 'repertoire_specific.html')

def lesson_payment_successful(request):
    if request.method == "GET":
        checkout_id = request.GET.get('session_id', "N/A")
        if checkout_id:
            paymentSession =
PaymentSession.objects.get(session_id=checkout_id)
            if paymentSession and not paymentSession.paid:
                paymentSession.paid = True
                paymentSession.save()
                lesson_id =
paymentSession.payment_data["data"]["lesson_id"]
                buyer = paymentSession.buyer
                lesson = Lesson.objects.get(id=lesson_id)
                lesson.students.add(buyer)
                lesson.save()

                paymentLesson = PaymentLesson.objects.create(
                    buyer = request.user,
                    lesson = lesson,
                    amount =
paymentSession.payment_data["amount_total"],
                    checkout_id = checkout_id
                )

                content = {
                    "lesson": paymentLesson.lesson,
                    "buyer": paymentLesson.buyer
                }
                return render(request, 'lesson_buy_success.html',
content)
            elif paymentSession and paymentSession.paid:
                paymentLesson =
PaymentLesson.objects.get(buyer=request.user, checkout_id=checkout_id)
                content = {
                    "lesson": paymentLesson.lesson,
                    "buyer": paymentLesson.buyer
                }
                return render(request, 'lesson_buy_success.html',
content)
            else:
                return HttpResponse("Stop trying to blindly guess
session ids..")
        else:

```

```

        return redirect("/dashboard/")

def lesson_payment_cancelled_or_failed(request):
    return render(request, "payment_cancelled.html")

```

main > models.py

```

from django.db import models
from django.contrib.auth.models import AbstractUser, Group
from django.urls import reverse
from phonenumber_field.modelfields import PhoneNumberField
from django.utils.safestring import mark_safe
from django.core.validators import FileExtensionValidator
from django_ckeditor_5.fields import CKEditor5Field
from django.contrib.admin import SimpleListFilter
import datetime

# Create your models here.
class Member(AbstractUser):
    class Meta:
        verbose_name = 'Member'

    profile_picture = models.ImageField(
        upload_to='profile_pictures/',
        validators=[FileExtensionValidator(['jpg', 'jpeg', 'png'])],
        blank=True,
        null=True,
        verbose_name="Profile Picture"
    )
    email = models.EmailField(unique=True)
    first_name = models.CharField(max_length=50)
    middle_name = models.CharField(max_length=50, blank=True,
null=True)
    last_name = models.CharField(max_length=50)
    mobile_number = PhoneNumberField()

    def __str__(self) -> str:
        full_name = ""
        if self.middle_name is None:
            full_name = f"{self.first_name} {self.last_name}"
        else:
            full_name = f"{self.first_name} {self.middle_name}
{self.last_name}"

```

```

        if not full_name.replace(" ", ""):
            return "Please set your first, middle and last name."

    return full_name

def image_tag(self):
    if not self.profile_picture:
        return mark_safe(f'<p>No set profile picture.</p>')
    return mark_safe(f'')

def get_first_group_name(self):
    return self.groups.first().name

def save(self, *args, **kwargs):
    is_new = self._state.adding
    super().save(*args, **kwargs)

    if is_new:
        try:
            member_group = Group.objects.get(name='Member')
            self.groups.add(member_group)
        except Group.DoesNotExist:
            pass

def is_in_instructor_group(self):
    return self.groups.filter(name__in=['Instructor',
'Administrator', 'Superuser']).exists()

image_tag.short_description = 'Current Profile Picture'

class Post(models.Model):
    author = models.ForeignKey(Member, on_delete=models.CASCADE,
related_name="author", blank=True, null=True)
    title = models.CharField(max_length=255)
    content = CKEditor5Field('Content', config_name='default')
    date_posted = models.DateTimeField(auto_now_add=True, blank=True)
    group_visibility = models.ManyToManyField(Group,
related_name='group_visibility', blank=True, verbose_name="Group
Visibility")

    def __str__(self):
        return self.title

```

```

    def get_absolute_url(self):
        return reverse("site_dashboard_announcements_specific",
                      kwargs={"pk": self.pk})
    class Attachment(models.Model):
        class AttachmentType(models.TextChoices):
            PHOTO = "Photo", "Photo"
            VIDEO = "Video", "Video"

        class Meta:
            verbose_name = 'Attachment'
            verbose_name_plural = 'Attachments'

        post = models.ForeignKey(Post, on_delete=models.CASCADE,
                               related_name="attachments")
        author = models.ForeignKey(Member, on_delete=models.CASCADE,
                               related_name="attachments", blank=True, null=True)
        date_created = models.DateTimeField(auto_now_add=True, blank=True)
        file = models.FileField(
            upload_to='post/attachments/',
            validators=[FileExtensionValidator(allowed_extensions=['jpg',
            'jpeg', 'png', 'gif', 'mp4', 'avi', 'mov'])],
            blank=True,
            null=True
        )
        file_type = models.CharField('File Type',
                                     choices=AttachmentType.choices, max_length=10)
        show = models.BooleanField(default=False)

    def __str__(self):
        return f"{self.file_type} attachment for {self.post.title}"

    class Notification(models.Model):
        class NotificationType(models.TextChoices):
            EVENT = "Event", "Event"
            LESSON = "Lesson", "Lesson"
            POST = "Post", "Post"
            AUDITION = "Audition", "Audition"
            BOOKING = "Booking", "Booking"

            notification_type =
models.CharField(choices=NotificationType.choices)
        member = models.ForeignKey(Member, on_delete=models.CASCADE,
                               related_name="notifications", blank=True, null=True)
        notifier = models.ForeignKey(Member, on_delete=models.CASCADE,
                               related_name="notifier", blank=True, null=True)

```

```

date_created = models.DateTimeField(auto_now_add=True, blank=True)
is_read = models.BooleanField(default=False)
post = models.ForeignKey(Post, on_delete=models.CASCADE,
related_name="notifications", blank=True, null=True)
event = models.ForeignKey('event.Event', on_delete=models.CASCADE,
related_name="notification_event", blank=True, null=True)
lesson = models.ForeignKey('lesson.Lesson',
on_delete=models.CASCADE, related_name="notification_lesson",
blank=True, null=True)
audition = models.ForeignKey('audition.Audition',
on_delete=models.CASCADE, related_name="notification_audition",
blank=True, null=True)
book = models.ForeignKey('event.Book', on_delete=models.CASCADE,
related_name="notification_book", blank=True, null=True)
message = models.CharField(max_length=255)
long_message = models.CharField(max_length=255, blank=True,
null=True)
url = models.CharField(max_length=150, blank=True, null=True)

def is_event(self):
    if self.event:
        return True
    return False

def is_post(self):
    if self.post:
        return True
    return False

def is_book(self):
    if self.book:
        return True
    return False

def is_audition(self):
    if self.audition:
        return True
    return False

def is_lesson(self):
    if self.lesson:
        return True
    return False

class SiteConfig(models.Model):

```

```

class Meta:
    verbose_name = 'Site Configuration'
    verbose_name_plural = 'Site Configurations'

    DOMAIN_URL = models.CharField(max_length=255,
default='http://127.0.0.1:8000/', verbose_name="Domain URL",
help_text="Make sure to follow the format of the default to avoid complications.")
    TEMPORARY_AUDITION_EXPIRATION_DAYS =
models.IntegerField(default=7, verbose_name="Temporary Audition
Expiration Days")
    TEMPORARY_BOOKING_EXPIRATION_DAYS = models.IntegerField(default=7,
verbose_name="Temporary Booking Expiration Days")
    ENABLE_AUDITION_FEE = models.BooleanField(default=False,
verbose_name="Enable Audition Fee")
    AUDITION_FEE = models.DecimalField(default=250, max_digits=10,
decimal_places=2, verbose_name="Audition Fee")
    BOOKING_FEE = models.DecimalField(default=250, max_digits=10,
decimal_places=2, verbose_name="Booking Fee")
    PAYMENT_RESERVATION_EXPIRATION = models.IntegerField(default=1,
verbose_name="Initiated Payment Reservation in Hours")
    MAX_BOOKING_PER_DAY = models.IntegerField(default=5,
verbose_name="Maximum Bookings per Day")
    MIN_BOOKING_HOUR_GAP = models.IntegerField(default=1,
verbose_name="Minimum Hour Gap per Booking (in Hours")
    START_OF_BOOKING_HOURS =
models.TimeField(default=datetime.time(9,00), verbose_name="Start of
Booking Open Hours")
    END_OF_BOOKING_HOURS =
models.TimeField(default=datetime.time(21,00), verbose_name="End of
Booking Open Hours")
    OTP_LENGTH = models.IntegerField(default=6, verbose_name="OTP
Length", editable=False)
    OTP_EXPIRATION_IN_HOURS = models.IntegerField(default=2,
verbose_name="OTP Expiration in Hours")
    USE_EMAIL OTP_LOGIN = models.BooleanField(default=True,
verbose_name="Use Email OTP Login")
    USE_EMAIL OTP SIGNUP = models.BooleanField(default=True,
verbose_name="Use Email OTP Signup")
    USE_SMS OTP_LOGIN = models.BooleanField(default=False,
verbose_name="Use SMS OTP Login", help_text="Caution: 1 sms operation
is equal to 0.17€")
    USE_SMS OTP SIGNUP = models.BooleanField(default=False,
verbose_name="Use SMS OTP Signup", help_text="Caution: 1 sms operation
is equal to 0.17€")

```

```

USE_TICKET_SCAN_OTP = models.BooleanField(default=False,
verbose_name="Use Ticket Scan OTP", help_text="Caution: 1 sms
operation is equal to 0.17€")
USE_SMS_NOTIFICATION_FOR_AUDITION_CHANGE =
models.BooleanField(default=False, verbose_name="Use SMS Notification
for Audition Date Changes", help_text="Caution: 1 sms operation is
equal to 0.17€")
USE_SMS_NOTIFICATION_FOR_BOOKING_CHANGE =
models.BooleanField(default=False, verbose_name="Use SMS Notificaton
for Booking Date Changes", help_text="Caution: 1 sms operation is
equal to 0.17€")

def __str__(self):
    return "Site Configuration"

def save(self, *args, **kwargs):
    self.pk = 1
    super().save(*args, **kwargs)

@classmethod
def load(cls):
    obj, created = cls.objects.get_or_create(pk=1)
    return obj

```