**A PROPOSED OFFERING OF WEB-BASED RESERVATION SYSTEM FOR THE ADDLIB DANCE STUDIO AT DATAMEX COLLEGE OF SAINT ADELINE VALENZUELA BRANCH**

A Thesis Project Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

By:

Bacani, Princess Mahallel

Delos Angeles, Arjay   
Mejila, Steven Carl   
Norte, Sakura Ann G.

**DESIGN DOCUMENT**

**INTRODUCTION**

This document presents a design document to provide a blueprint for the development of the Reservation System for The Adlib Dance Studio (TADS). This Reservation System for TADS is the document focuses on the software components necessary to support the reservation work flow, including user management, class scheduling, booking, payment, and notifications. It serves as a guide for the software development team, stakeholders as a clients and project managers ensure the both parties have a clear understanding of the system design components and functions. It ensures this is aligns directly with a goal to the original project proposal. This a reference point for a future maintenance and enhancements helping to preserve the future update of the system.

**Overview**

The Reservation System for The Adlib Dance Studio is a software application designed to make a web-based online booking and management of dance classes and other studio resources. The system will provide an interactive interface from multiple roles. It addresses the inefficiencies to reduce the manual workload, minimize bookings conflict and enhance the overall costumer experience.

The system includes user registration and authentication also the class scheduling, reservation management, payment processing and the notification services in real time. This system will support the multiple user roles such as the students, instructors and admins. System will maintain a database of users, classes and transactions. The design accommodates future growth, security to protect sensitive data and responsive to ensure accessibility across to any devices.

**Scope of the Design Document**

This scope defining the system behavior and core modules, system architecture, data flow diagrams, security features, performance, user interface layouts, deployment, error handling and maintenance and support. The system is a web-based any type of mobile devices can access to the website.

**SYSTEM ARCHITECTURE**

The Reservation System for TADS designed using a architecture that ensures maintainability, flexibility, and efficient performance.  The architecture follows a client-server model, to provide seamless interaction between users and the backend services.

**High-Level Components**

1. **Client Application** - Student and Instructors User Interface.
2. **Admin Application -** handles business logic and process request from the client.
3. **Database Server -** The stores user all the data, class schedule and reservations and feedback.
4. **Payment Method -** Secure all the payment of our client thru Face to face.
5. **Notification Server** - Sending bookings confirmations, reminders, cancellations and promotional messages.

**Deployment Architecture**

Client Side: Ensure high availability and scalability

Database: Ensure the reliability and backups.

**Communication Protocol and Interfaces**

The application server communicates with the database using standard SQL queries over secure connections

**DATABASE DESIGN**

**Students -** Users who book the classes

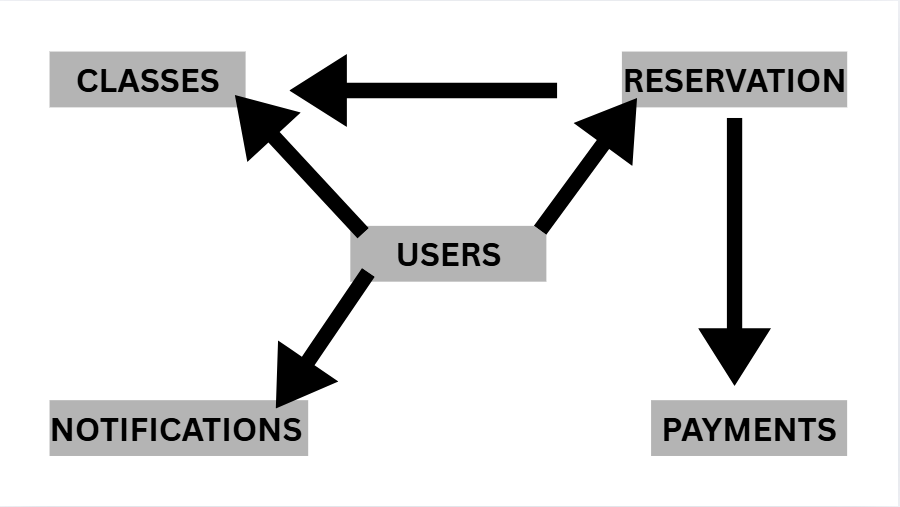
**Instructors** - Users who conduct classes

**Administrators** - Users who manage the system

**Payment -** External service for processing payments

**Notification Service** - External service for processing notifications

The Data Flow Diagram (DFD) for the database schema illustrates how data moves between different entities and the database within the system. The DFD focus on Users, Classes, Reservation, Payments and Notifications. This shows a data flowing from users to the system during registration and booking from the system to the payment from the system to the notification service sending a notification to users. Data stored and retrieved from the respective tables to support this flow.



*Figure 1. “The Data Flow Diagram”*

**Description of database tables, fields, and relationships.**

**USER TABLES -** The Users tables store user information and credentials. The role is to determine access permissions. Users to Reservation. Each record begins to the UserID and the First name and Last Name. Also, the unique way to the information is the Email and also the password. Identify what role the user (Student, Instructor, Administrator) then their PhoneNumber and DateCreated.

**Classes Table** - The Classes table contains detaails about each dance class, including the instructor and schedule. Classes to Reservation. We have ClassID to the ClassName and the Description of the Class. For InstructorID the ScheduleDateTime, Duration and MaxCapacity of the class.

**Reservation Table** - The Reservation Table tracks which users have reserved which class and the status or what type the reservation. Reservation to Payments. We need ReservationID and UserID and which ClassID that they belong. We give the information the ReservationDateTime and also the Status of the reservation.

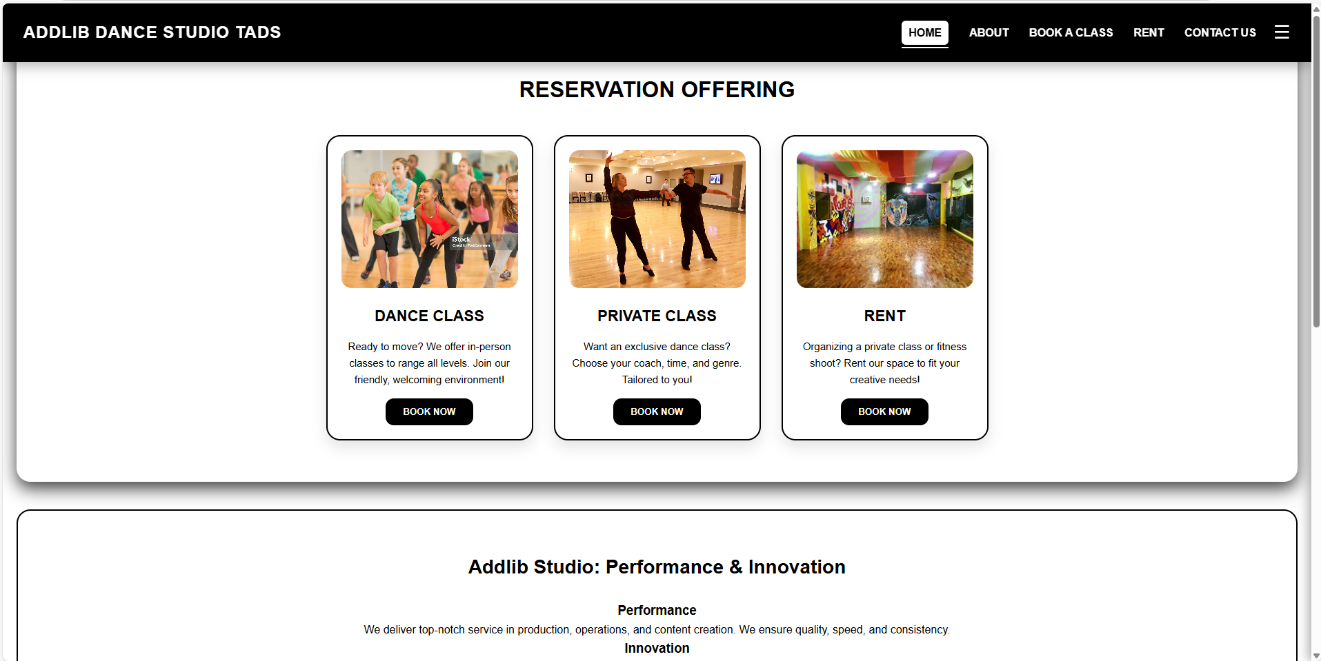
**Payments Table** - The Payment Table records the payment transaction linked to reservation. The PaymentID and ReservationID as a base for the PaymentDateTime. The Amount and PaymentStatus are reference to the PaymentMethod.

**Notification Table** - The Notification Table logs notification sent to the users.

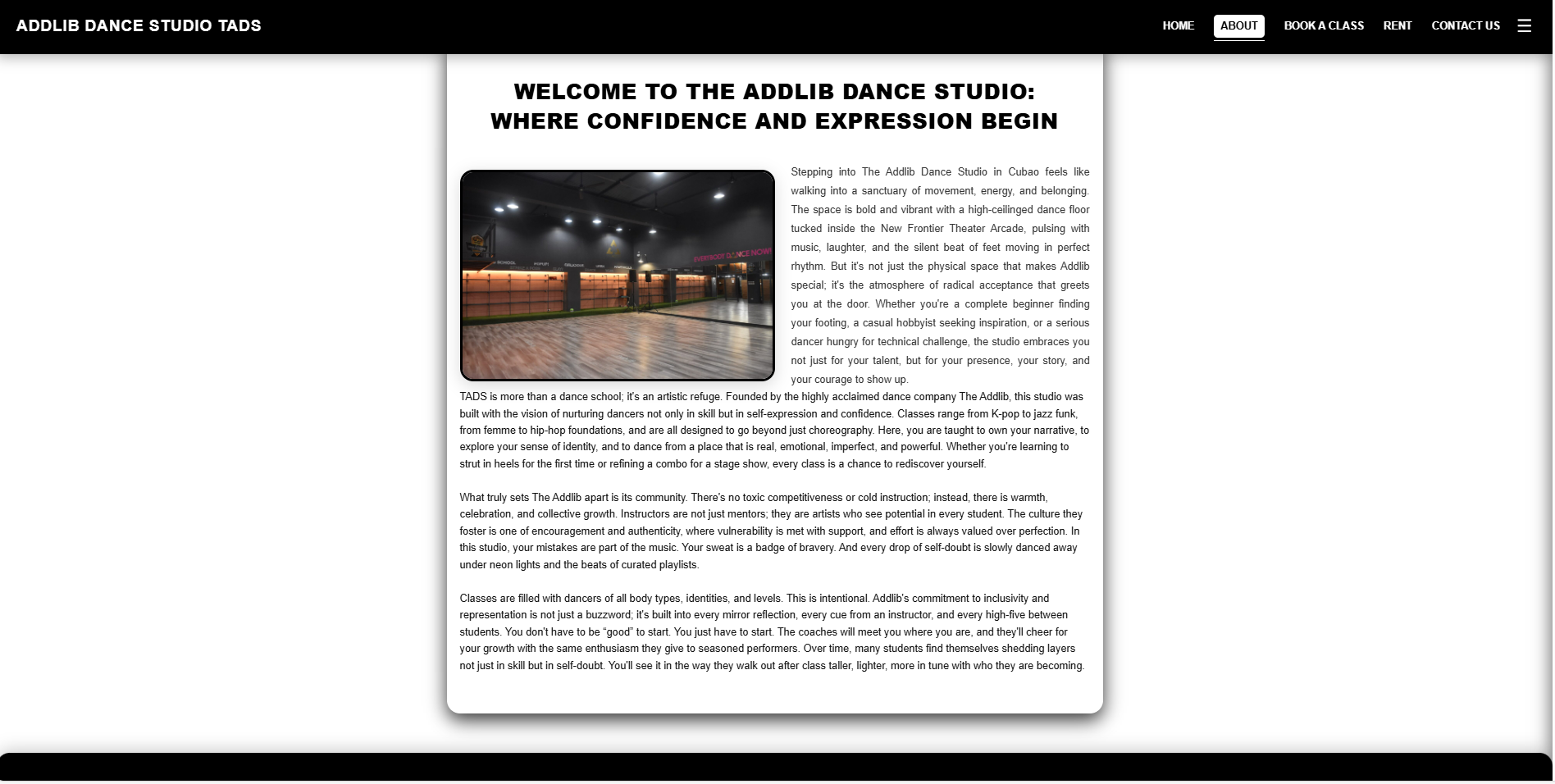
**Data Normalization**

To maintain data consistency and reduce redundancy, the database will apply **Third Normal Form (3NF)**. This less the instances data duplication, simplifies updates and improve the performance of the system.

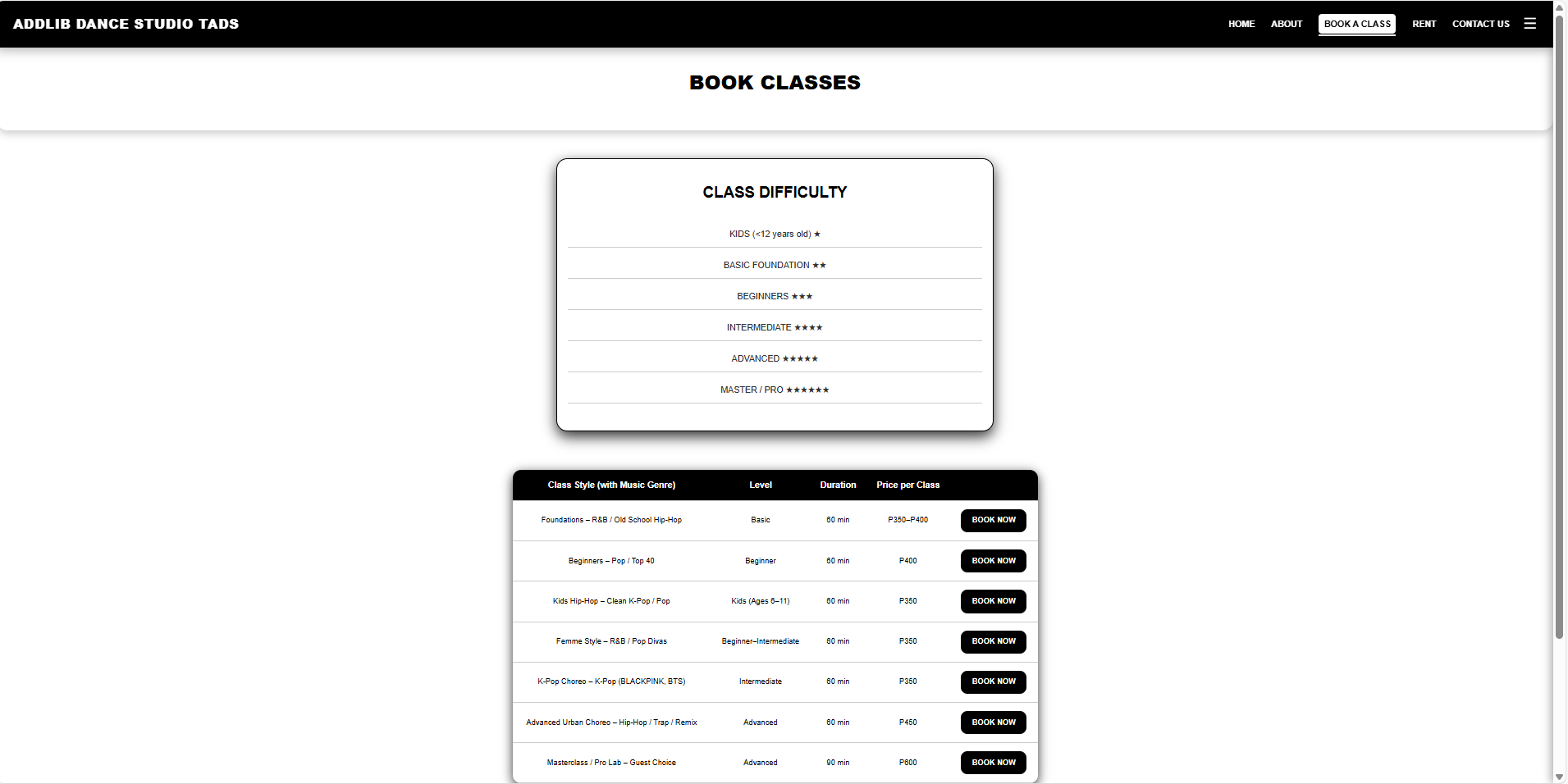
**USER INTERFACE DESIGN**



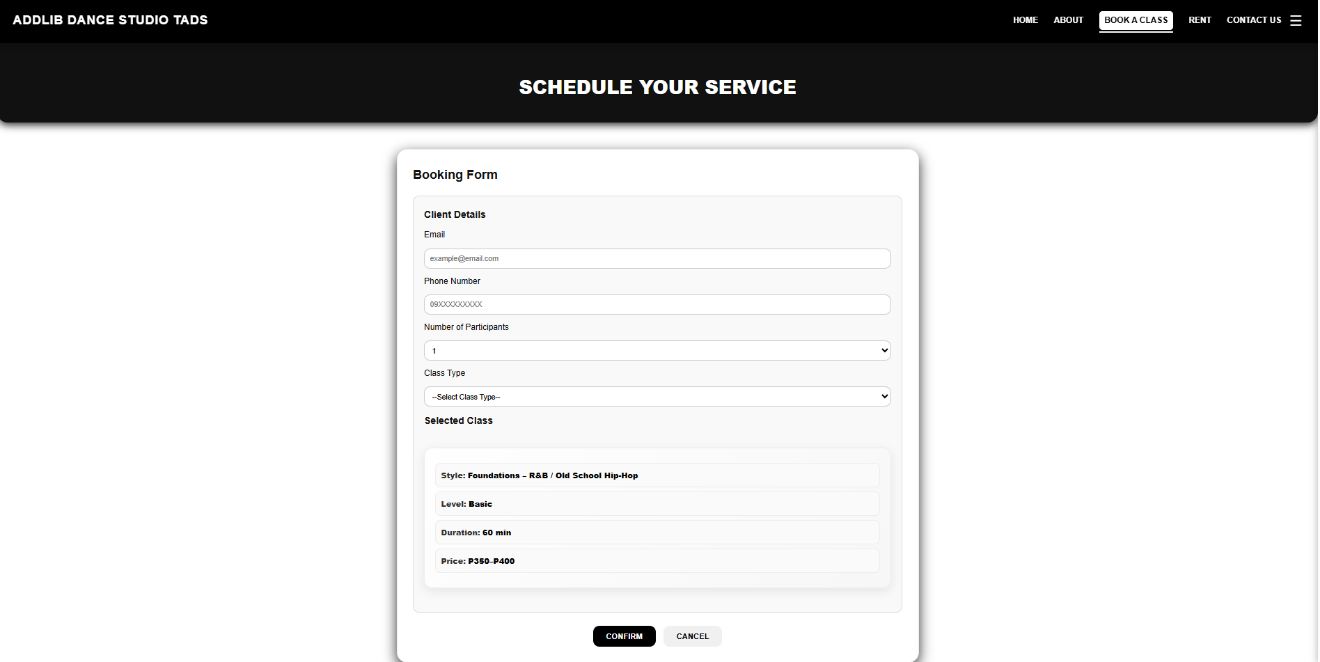
*Figure 2. “The Home Page”*

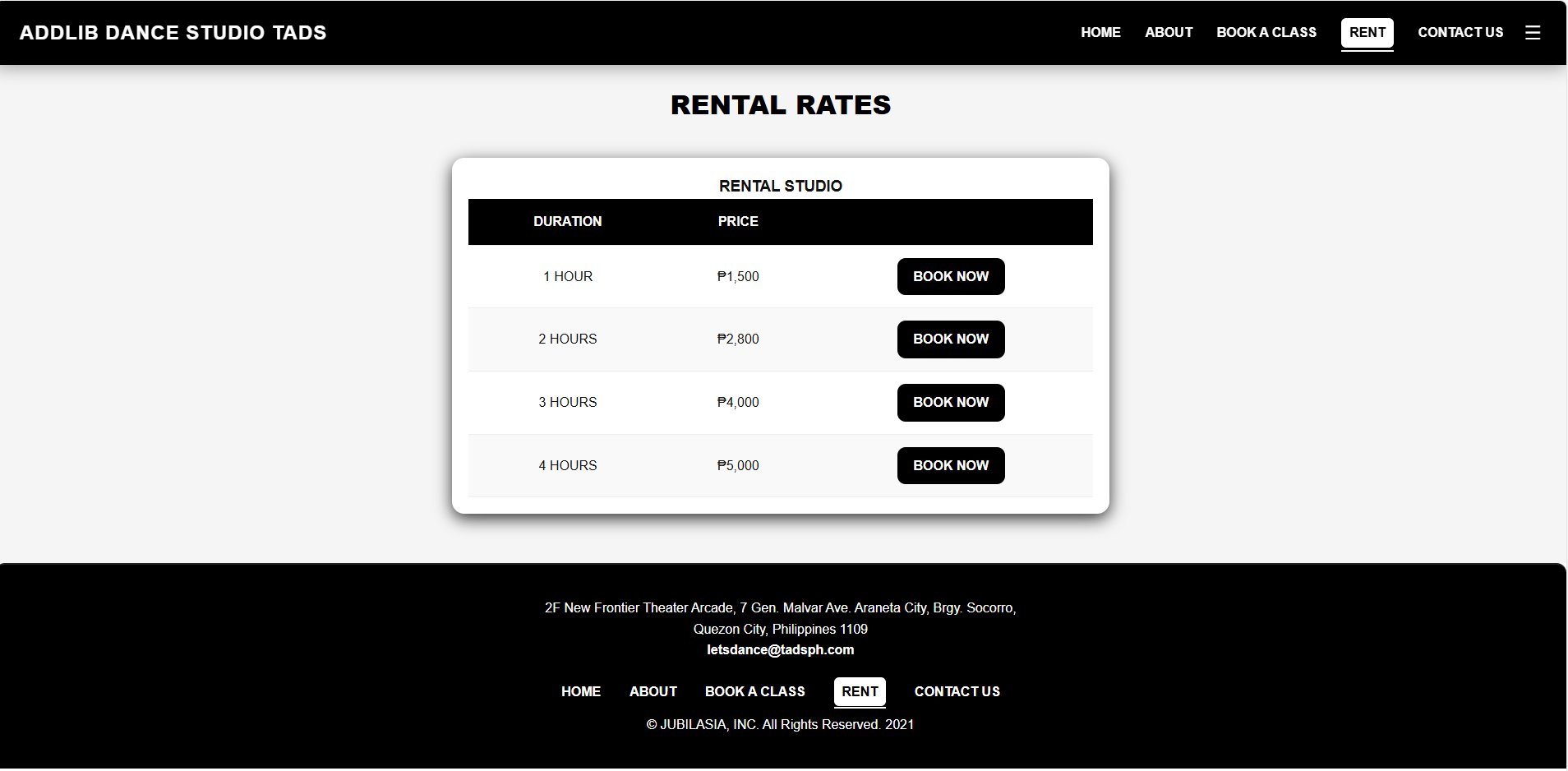


*Figure 3. “The About Page”*

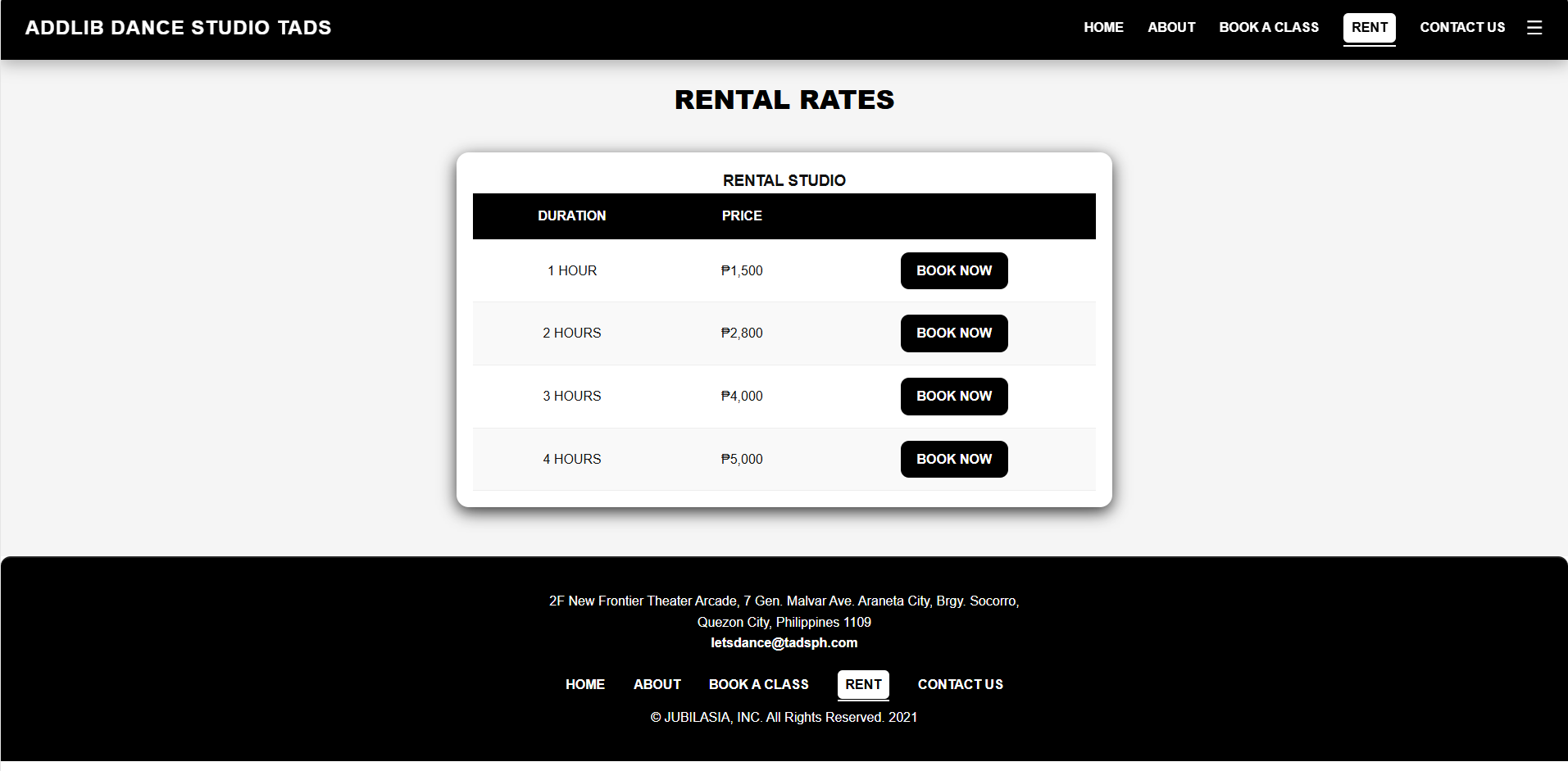


*Figure 4. “The Book a Class Page”*

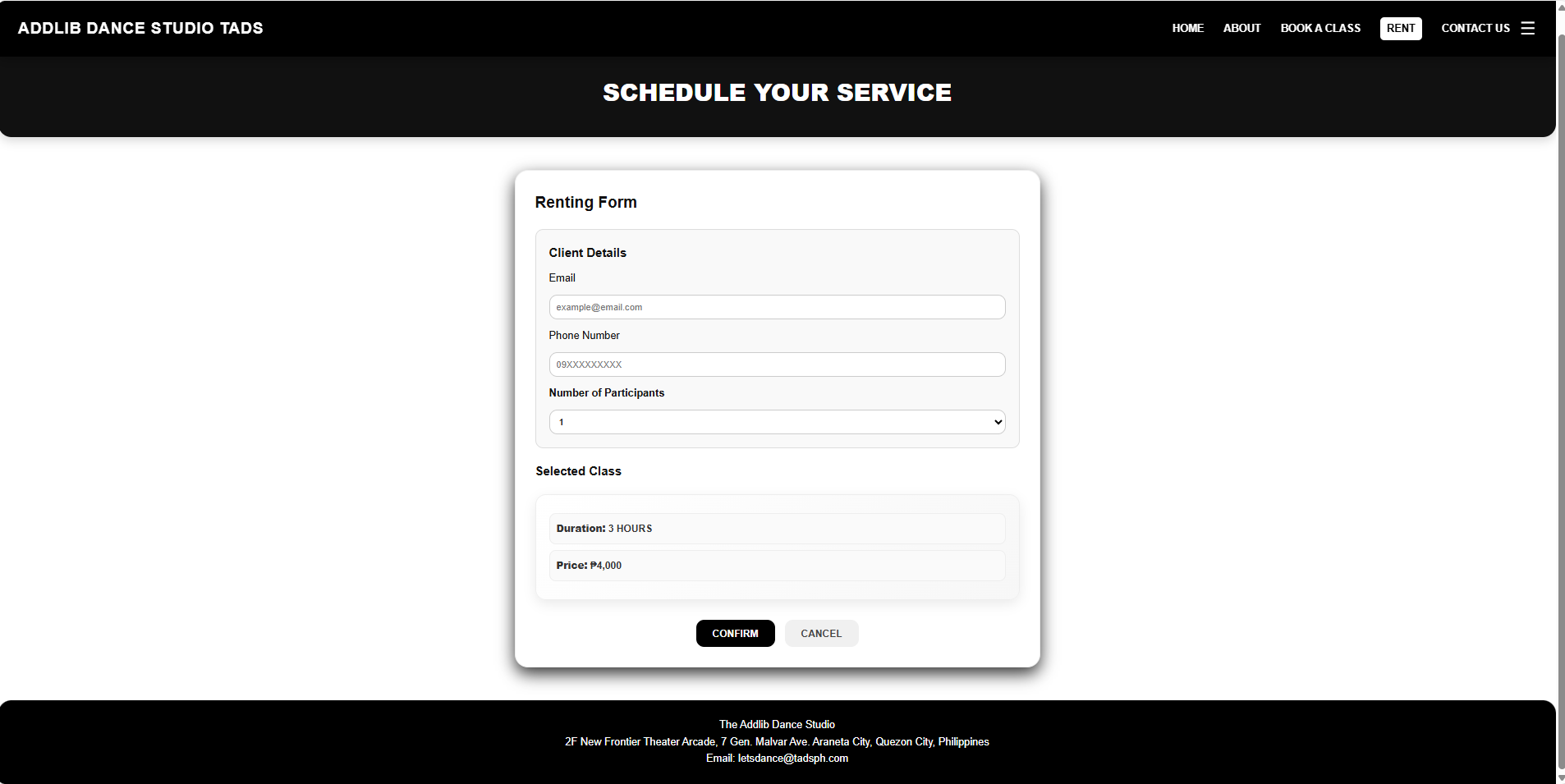




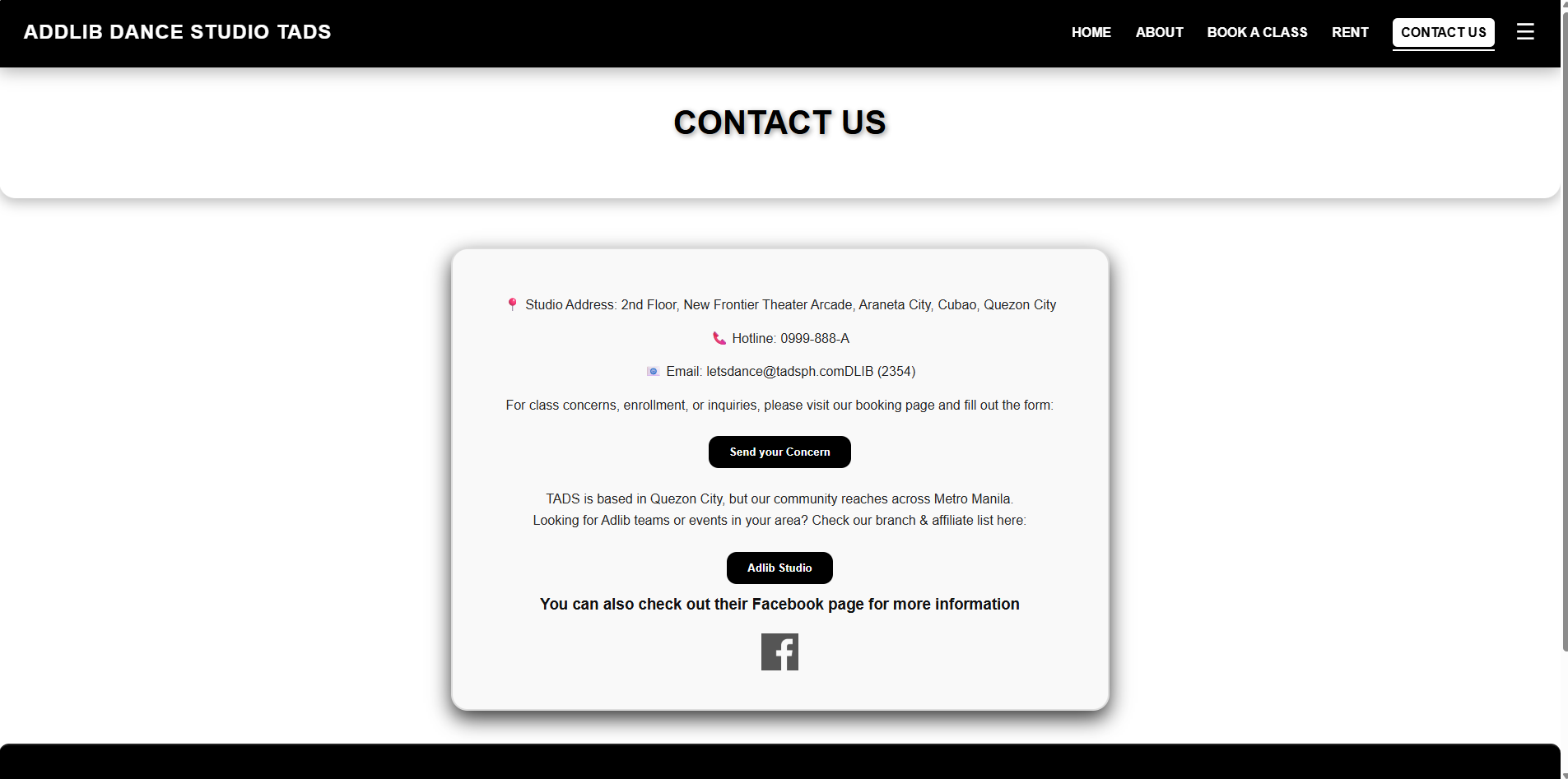
*Figure 5. “The Book a Class Fill Up Page”*



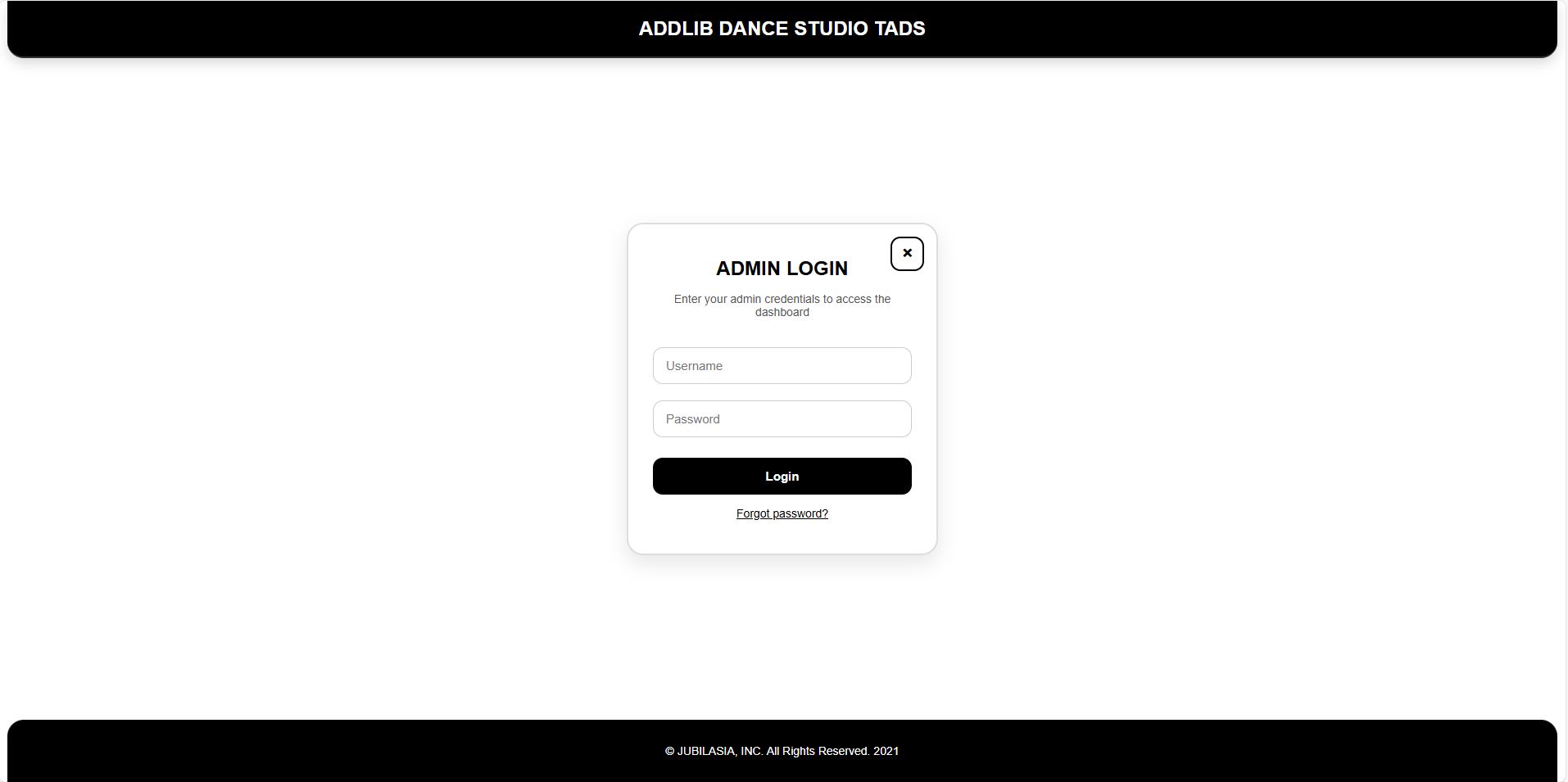
*Figure 6. “The Rental Rate Fill Up Page”*



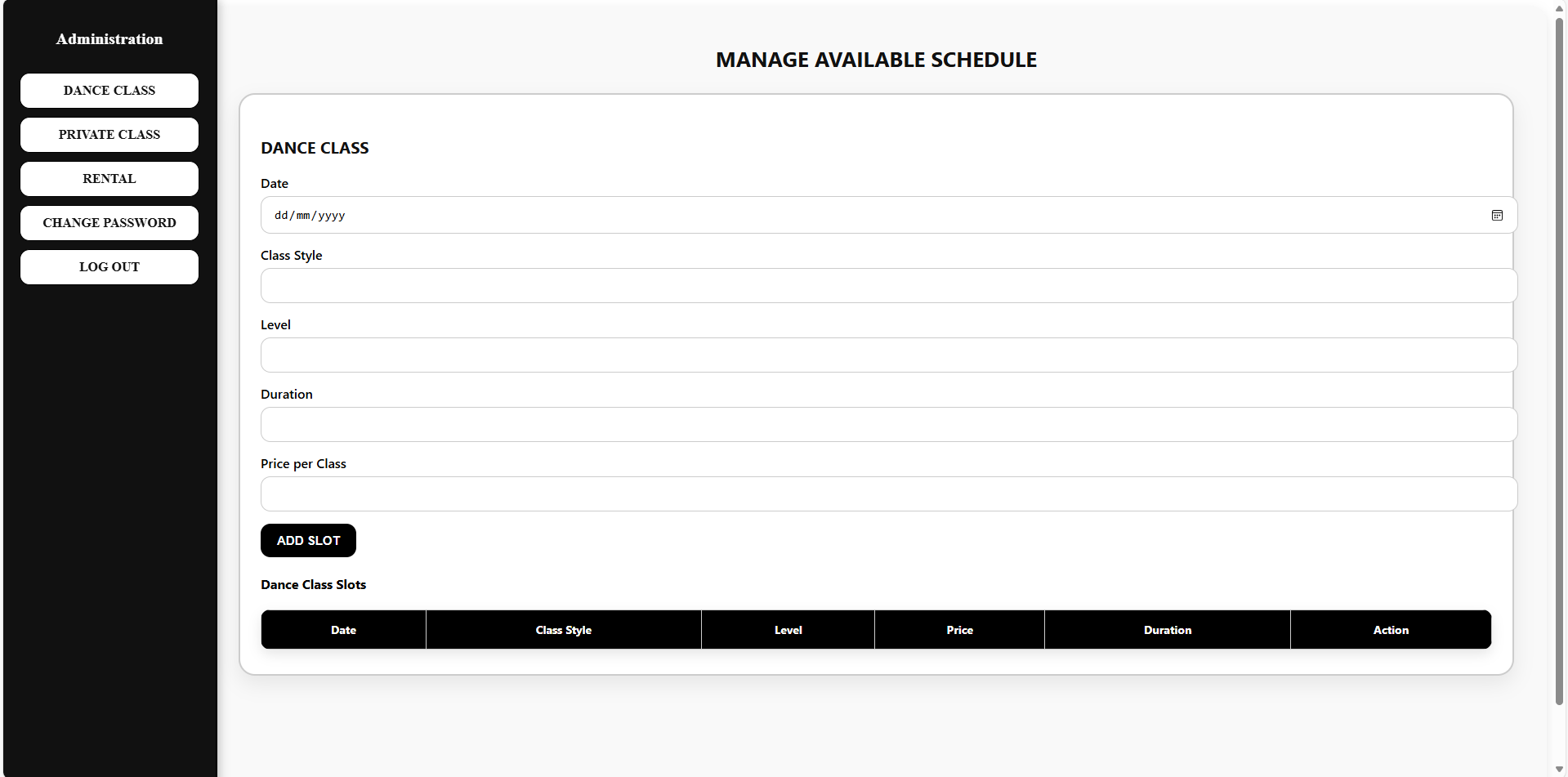
*Figure 7. “The Rent Fill Up Page”*



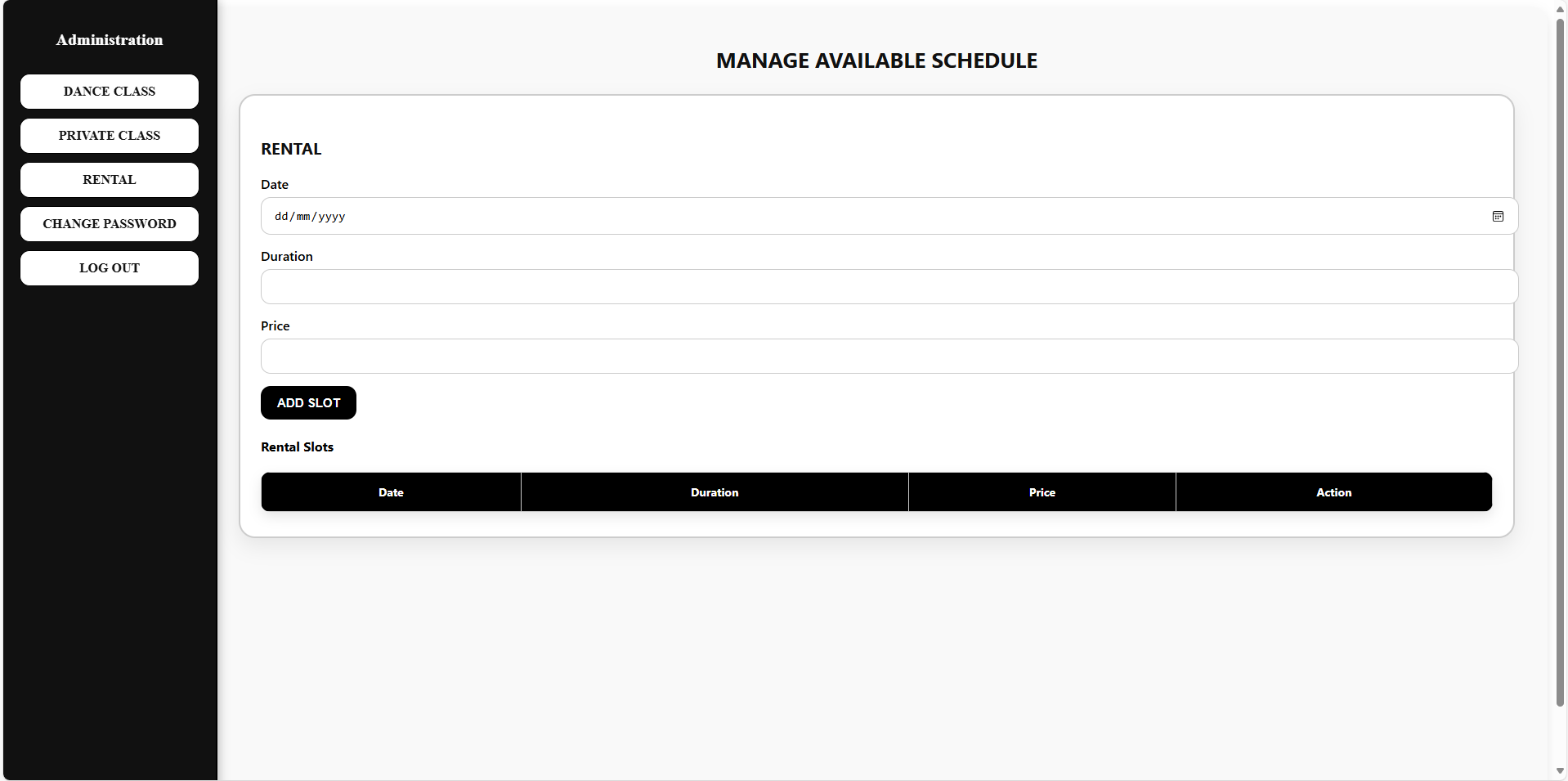
*Figure 8. “The Contact Us Page”*



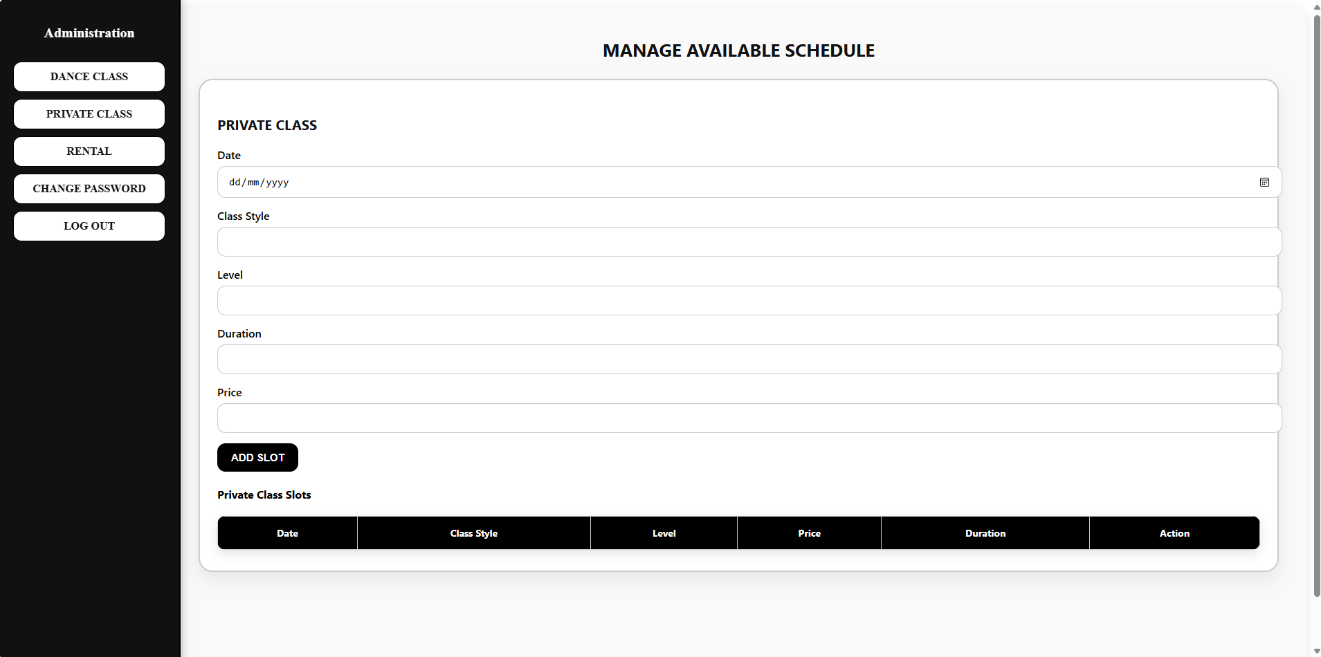
*Figure 9. “The Admin Log – in Page”*



*Figure 10. “The Manage Available Schedule for Dance Class Page”*



*Figure 11. “The Manage Available Schedule for Rental Page”*



*Figure 12. “The Manage Available Schedule for Private Class Page”*

**Design Principles and Usability Considerations**

The system will prioritize **clean and simple layouts and u**ser-friendly navigation. Ensure the website have minimal design but interactive to the clients. The admins staff must be able to practice or minimal training of the system. Error- prevention techniques will also ensure to prevent the errors.

**COMPONENTS DESIGN**

* **User Module -** Authentication and profile for student and instructors
* **Admin Module -** Manages the creation, updating and delete dance classes include the scheduling and instructors’ assignment.
* **Reservation Module -** Handles the reservation process, including creating, updating, and canceling reservations for classes.
* Payment Module - It ensures secure the transaction processing and updates payment records.
* **Notification Module -** Sends notifications to students and instructors regarding class schedules, reservation confirmations, and reminders.

**Dependencies**

The system manages these dependencies to ensure smooth system operation and maintain the security and data integrity

1. **User Management Module**

Modules depend on it for user authentication and authorization. The Reservation Management Module requires a valid user session to create or modify reservations.

1. **Class Scheduling Module**

Provides class data consumed by the Reservation Management Module to check availability and managing the schedule.

1. **Reservation Management Module**

Interacts closely with the Payment Processing Module to initiate and verify payments linked to bookings. It also triggers the Notification Module to send booking confirmations or cancellation alerts.

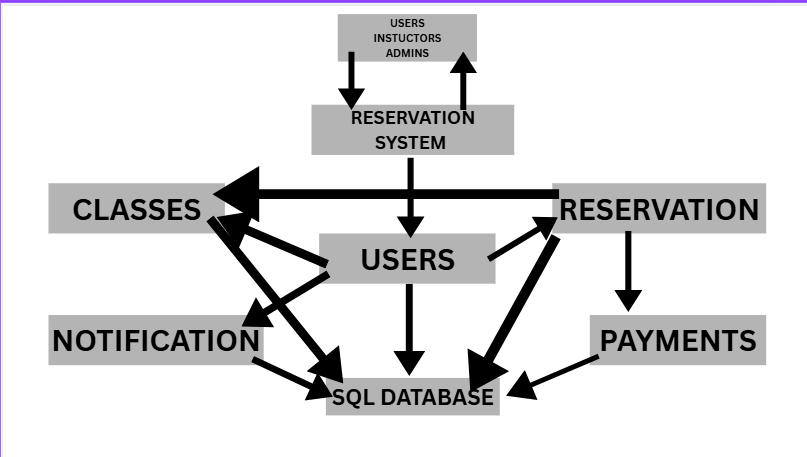
1. **Payment Processing Module**

It depends on external payment gateway APIs but exposes internal APIs for the Reservation Management Module to update payment status.

1. **Notification Module**

The interfaces with third-party messaging services and multiple modules to notify users of relevant events.

**DATA FLOW DIAGRAM**

****

*Figure 13. The Data flow of Diagram*

**SECURITY DESIGN**

The Reservation System must ensure the confidentiality and integrity of user data and system resources. It must protect the sensitive information such as personal user details, payment transaction and authentication credentials from unauthorized access. Authentication system uses token-based authentication with JSON users authenticate via email and password. Access control at the API level, ensuring users can only perform permitted actions. Input validation to prevent attacks. Database access is restricted by roles and network policies. Regular backups are encrypted and securely stored.

**PERFORMANCE DESIGN**

The system should respond to user request of all their transactions. Ensure it is available for users at all times, except during scheduled maintenance. It should also work smoothly on standard devices.Use server-side accessed data such as class schedules and user sessions. The backend services behind load balance to distribute traffic evenly across multiple instances, ensuring scalability and fault tolerance

**DEPLOYMENT PLAN**

**Preparation**

* Verify hosting environment and domain configuration
* Back up any previous data
* Ensure the code and dependencies are finalized

**Installation**

* Deploy the website to a server.
* Testing -Test links, forms, booking features, and responsiveness across devices.

**Deployment**

* Upload the website to the live server

**Maintenance**

* Monitor uptime and website analytic

**Hardware**

* Hosting server with at least 2 CPU cores 8gb ram, 100gb storage
* Stable internet connection
* Backup storage Software
* Database: MySQL
* Run time: node.js

**Version control**

* Use git for managing website code

**Configuration Management**

* Secure sensitive information
* Maintain environment specific configuration file

**MAINTENANCE AND SUPORT**

**Maintenance and Support**

Booking and Reservation System requires different maintenance and a clear support process to support operations to a long-term standard. This System maintenance and support section explains the guidelines, procedures, and escalation processes that help to sustain the system's efficacy, safety, and usability.

**System Maintenance and Support Guidelines**

* **Regular Backups**

To preserve the integrity of reservation data, daily incremental and full weekly database backup processes must be conducted. System Monitoring – Dedicated oversight of server performance, storage, and network usage is critical for uninterrupted operations.

* **Security Management**

Routine vulnerability assessments, implementation of protective measures, and user rights audits should be conducted to enforce organizational security policies.

* **Log Management**

System and error logs, user access logs, and other relevant information should be retained for auditing and troubleshooting to support log maintenance.

* **User Support**

Help desk, ticketing systems, chats, and other support channels are critical to assist with user and technical challenges.

**Software Update, Patch, And Bug Fix Procedures**

* **Updates**

It is a best practice to perform system updates during off-peak hours to avoid disrupting services.

* **Testing**

Update, patches, and bug fixes should be pr-deployed to a testing environment for compatibility and stability testing.

* **Deployment**

Validated changes are implemented to the live systems with proper version and rollback procedures.

* **Notification**

All users are informed in advance regarding scheduled maintenance, downtime, and other significant changes and updates.

* **Documentation**

All changes, patches, and fixes done should be documented for tracking and future reference.

**Issue Escalation and Resolution Procedures**

**Level 1 - Front-line Assistance**

User issues are captured at the front line using one of the available support channels: ticketing, email, or phone. Basic troubleshooting steps are performed. Connectivity issues, password resets, and login are among the basic troubleshooting steps.

**Level 2 - Technical Assistance**

If the initial troubleshooting steps taken at Level 1 do not resolve the issue, an advance troubleshooting step is performed. A system developer or technical personnel is contacted to check the system for software problems, database errors, or problems with integration.

**Level 3 - System Admin/Vendor Support**

System administrators or outside vendors are contacted for critical incidents, that are defined as prolonged downtime, database corruption, or significant security breaches. Feedback and Resolution: Users are informed of the solution and the issue is logged in the incidents tracking system.