**MAINTENANCE DOCUMENTATION FOR EVENT VENUE BOOKING SYTEM FOR CITY OF DREAMS MANILA**

A Project Proposal 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:

Albania, John Maverick B.

October 202

**MAINTENANCE DOCUMENTATION**

**INTRODUCTION**

The Event Venue Booking System was developed for City of Dreams Manila to make the process of booking event venues faster and more convenient. It allows clients to view available venues, select their preferred one, and submit bookings online using their Google account. On the admin side, it helps staff manage client requests, confirm bookings, and send email notifications efficiently.

Regular maintenance is important to keep the system running smoothly and to avoid errors that may affect users. Maintenance ensures that the system stays updated, secure, and compatible with any new software versions used in the environment.

The scope of maintenance includes checking and fixing bugs that may appear during use, updating software dependencies in Node.js, applying security patches, and ensuring the connection between the system and the database remains stable.

**MAINTENANCE PLAN**

The maintenance plan for the Event Venue Booking System focuses on keeping the system stable, secure, and efficient after deployment. The main strategy is to regularly check the system’s performance, apply necessary updates, and quickly address any issues reported by users from City of Dreams Manila.

he types of maintenance included are

**Corrective Maintenance** - Fixing bugs or errors that may occur during the use of the system, such as login issues, failed bookings, or email errors.

**Adaptive Maintenance** - Making adjustments when there are changes in software versions or technologies used, like updates in Node.js or SSMS that affect compatibility.

**Perfective Maintenance** - Improving certain features or adding small enhancements to make the system more user-friendly and efficient based on client feedback.

**Preventive Maintenance** - Performing regular updates on system dependencies, checking for potential security issues, and running tests to prevent major problems in the future.

**MAINTENANCE SCHEDULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Description** | **Frequency** | **Responsible Person** | **Status** |
| Security Updates | Apply patches and update dependencies in Node.js and other tools. | Monthly | Developer | Scheduled |
| Bug Fixes | Fix any errors or issues found during system use or testing. | As needed | Developer | Pending |
| System Performance Check | Monitor and test the system’s speed and performance on localhost. | Quarterly | Developer | |  | | --- | |  |  |  | | --- | | Started | |
| Feature Enhancements | Add or improve minor system functions | As needed | Developer | Planned |

**Table 1:** Maintenance Shedule

**ISSUE TRACKING & BUG REPORTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Issue ID** | **Description** | **Severity** | **Reported By** | **Date Reported** | **Status** |
| BUG001 | Google login sometimes fails to authenticate properly. | High | Project Developer | 9/10/2025 | Fixed |
| BUG002 | Email notification not sending after admin confirmation. | Medium | Project Developer | 10/12/2025 | In Progress |
| BUG003 | Venue images take time to load on slower networks. | Low | Project Developer | 10/14/2025 | Pending |
| BUG004 | Booking details not displaying correctly on admin dashboard. | High | Project Developer | 8/4/2025 | Fixed |
| BUG005 | Minor layout issue on mobile view of venue page. | Low | Project Developer | 10/17/2025 | Pending |

**Table 2:** ISSUE TRACKING & BUG REPORTS

**BACKUP & RECOVERY PLAN**

**Describe the backup strategy**

The backup strategy for the Event Venue Booking System is focused on preventing data loss and ensuring that all booking and client information is safely stored. Since the system is currently running on localhost, database backups are created manually using Microsoft SQL Server Management Studio (SSMS). Full backups are done weekly, and copies of the backup files are saved in an external drive or secure cloud storage like Google Drive for added safety.

**Provide recovery steps in case of data loss or system failure**

In case of data loss or system failure, the recovery process involves restoring the most recent database backup file in SSMS. The developer will also re-upload all project files and verify that the Node.js environment and dependencies are properly reinstalled. After restoration, the system will be tested to ensure that all data and functions, such as login, booking, and admin confirmation, are working correctly again.

**BACKUP PROCEDURES**

**Frequency of backups**

Backups for the Event Venue Booking System are performed weekly to keep all client booking information and system data safe. In some cases, especially after major updates or maintenance activities, additional backups are done manually to ensure that no recent data is lost. This helps maintain the integrity of both the system files and the database.

**Storage locations**

The storage locations for these backups include the local computer where the system runs on localhost, for offline safety, and a cloud storage platform such as Google Drive for secure remote access. Keeping multiple copies of the backup ensures quick recovery in case of accidental deletion, hardware failure, or system malfunction. Each backup file is properly labeled with the date and time it was created to make restoration easier when needed.

In addition, the source code of the system is regularly pushed to GitHub using Gitversion control. This provides an extra layer of protection for the project files and allows easy tracking of code changes. It also makes it convenient to restore or update the system anytime by simply cloning the repository from GitHub.

**RECOVERY STEPS**

**Steps to restore a backup in case of failure**

In case of system failure or data loss, the recovery process for the Event VenueBooking System follows a structured procedure to ensure that normal operations are restored efficiently and safely. First, the most recent backup file is identified and verified to ensure that it is complete and not corrupted. Once verified, the system administrator restores the backup into the SQL Server Management Studio (SSMS) database by importing restoring the database directly through the “Restore Database” function. After the database has been successfully restored, the Node.js server is reconnected to the database, and the system is tested to confirm that all data and functionalities are working as expected.

**Contact information for technical support**

If the issue involves frontend or backend file corruption, the Git or GitHub repository is used to retrieve the latest stable version of the system source code. This makes the recovery process faster and ensures that no significant code updates are lost.

For technical assistance, users or administrators can contact the system developer. Support is available via email and phone for urgent concerns related to data recovery or server failure.

**PERFORMANCE MONITORING**

**Define key performance indicators (KPIs) to monitor system health**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Description** | **Threshold** | **Monitoring Tool** |
| **Response Time** | Time taken for pages and booking forms to load | 2 seconds | Google Lighthouse |
| **Error Rate** | Percentage of failed or unsuccessful user requests | 20% | Log Analyzer |
| **Database Performance** | Measures query speed and database response efficiency | 500 ms per query | SQL Profiler |
| **System Resource Usage** | Tracks CPU and memory utilization during peak operations | CPU 70%,  Memory 80% | Task Manager / Performance Monitor |

**Table 3:** Define key performance indicators (KPIs) to monitor system health

**SECURITY MEASURES**

**Outline security policies and measures**

The Event Venue Booking System implements several security measures to ensure the safety and integrity of both client and admin interactions. Access to the admin dashboard is strictly limited to authorized personnel, preventing unauthorized manipulation of bookings or sensitive information. Clients are required to log in using their Google account before they can submit a booking, which ensures proper authentication and minimizes the risk of fraudulent submissions. All input fields in the booking form are validated to protect the system from malicious data or injection attacks.

**Include access control rules, authentication mechanisms, and encryption methods**

Even though the system is currently on localhost, it is designed to handle sensitive information, such as client emails, securely within the database. Access control rules are applied to prevent clients from performing actions reserved for administrators, such as confirming or deleting bookings. Admin actions, including confirmations, cancellations, and deletions, are logged to maintain an audit trail. Finally, confirmation emails are sent to clients only after the admin approves their requests, ensuring a controlled and secure communication process**.**

**DOCUMENTATION UPDATES**

Any changes made to the system documentation are recorded to ensure that the information reflects the latest updates and improvements. This includes revisions to the system flow, descriptions of new features, updates to security measures, and modifications to user instructions or guidelines. Keeping the documentation current helps both developers and users understand the system accurately and ensures that future maintenance or enhancements can be performed efficiently.

**CONCLUSION & RECOMMENDATIONS**

The maintenance tasks completed for the Event Venue Booking System include updating the system flow, implementing security measures, validating booking inputs, and ensuring that admin actions are properly logged. These tasks have helped improve the overall reliability, security, and functionality of the system, making it easier for both clients and administrators to use.

For future improvements, it is recommended to deploy the system online with HTTPS for secure client-server communication, implement additional access controls for more granular admin permissions, and consider integrating automated notifications for clients regarding booking status updates. Regular updates to the system documentation should also be maintained to reflect any new features or changes.