**DEPLOYMENT DOCUMENTATION 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 2025

**DEPLOYMENT DOCUMENTATION**

**INTRODUCTION**

The Event Venue Booking System is a web-based application developed for City of Dreams Manila. Its main goal is to make the event booking process faster and more convenient for clients. Through this system, users can browse available venues, view their details, and submit booking requests online. On the admin side, staff can review requests, check equipment availability, and send booking confirmations through email.

The purpose of this deployment is to set up the system in a working environment where it can be tested and used. It ensures that all parts of the system such as the front-end interface, back-end server, database, and API function correctly together. This process also helps identify and fix possible issues before launching the system for actual use.

For now, the system is only deployed on localhost for development and testing purposes. This setup allows the developer to make changes, fix issues, and test features directly before doing a full or live deployment later on.

**DEPLOYMENT PLAN**

The deployment of the Event Venue Booking System will be done in three main phases: Pre-Deployment, Deployment, and Post-Deployment. Since the system is currently in the development stage, the deployment is focused on a local environment (localhost) for testing and setup purposes. This approach allows me to check the functionality and fix any issues before moving to a live server.

**Deployment Schedule and Milestones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Description** | **Start Date** | **End Date** | **Status** |
| **Pre-Deployment** | Preparing the local environment, installing dependencies, and configuring database connections. | 10/25/2025 | 10/27/2025 | In Progress |
| **Deployment** | Setting up the system on localhost, running the Node.js server, and linking it to SSMS database. | 10/28/2025 | 11/02/2025 | In Progress |
| **Post-Deployment** | Conducting functionality testing, debugging, and system monitoring for performance. | 11/03/2025 | 11/06/2025 | Pending |

**Table 1:** Deployment Schedule and Milestones

**Overall Strategy**

The main strategy for deployment is to start small and local. The system is first deployed on a local machine using Visual Studio Code and Node.js to make sure that all features work correctly. Once all bugs are fixed and the system runs smoothly, it can later be deployed to an online web server for actual client use.

**DEPLOYMENT ENVIRONMENT**

The Event Venue Booking System is currently deployed on a local environment (localhost) for development and testing. The setup uses a personal computer as the main host, where both the server and database are installed. This allows easier testing, debugging, and modification of the system before moving it to a live server.

**Hardware Requirements**

|  |  |
| --- | --- |
| **Component** | **Specification** |
| **Processor** | Intel Core i3 or higher |
| **Memory (RAM)** | At least 8 GB |
| **Storage** | Minimum 256 GB HDD or SSD |
| **Network** | Stable internet connection (for testing email API and Google login) |
| **Client Devices** | Desktop or laptop with updated browser (Google Chrome, Edge, or Firefox) |

**Table 2:** Hardware Requirements

**Software Requirements**

|  |  |
| --- | --- |
| **Category** | **Details** |
| **Operating System** | Windows 10 or highe |
| **Development Tool** | Visual Studio Code |
| **Server Environment** | Node.js (Express Framework) |
| **Database** | Microsoft SQL Server (SSMS) |
| **Dependencies** | bcryptjs, body-parser, cors, dotenv, express, google-auth-library, mssql, nodemailer |

**Table 3:** Software Requirements

**Hosting Information**

For now, the system runs only on localhost for development and testing. The hosting is done locally through Node.js using the command node server.js. Once the system is fully tested and finalized, it can later be hosted on a cloud server or school web domain for public access.

**DEPLOYMENT PROCEDURES**

The system is deployed locally for testing using Node.js and Microsoft SQL Server.  
Below are the basic steps followed to deploy the system on localhost:

**Step 1:** Install the required software

* Install Node.js (latest LTS version).
* Install Microsoft SQL Server and SQL Server Management Studio (SSMS).
* Install Visual Studio Code for development.

**Step 2:** Set up the project folder

* Create a folder for the project Event Venue Booking System.
* Inside the folder, create subfolders such as public, routes, and config.
* Copy or save the main server file as server.js.

**Step 3:** Initialize Node.js and install dependencies

Open the terminal in VS Code and run:

* npm init -y
* npm install express mssql cors body-parser dotenv bcryptjs jsonwebtoken google-auth-library

**Step 4:** Configure the environment variables

* Create a .env file in the project root.
* Add the following environment settings:

DB\_USER=your user

DB\_PASSWORD=your password

DB\_SERVER=Your server name

DB\_NAME=EventVenueBooking Sytem

JWT\_SECRET=Your secret ke

GOOGLE\_CLIENT\_ID=your google client id

**Step 5:** Set up the database

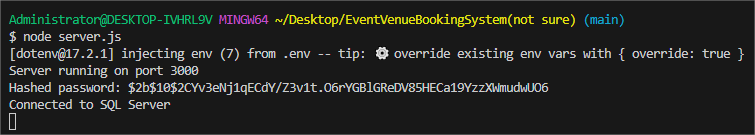
* Open SQL Server Management Studio (SSMS).
* Create a new database named EventVenueBookingSytem.
* Run your SQL script or manually create the following tables.

**Step 6:** Run the server

In VS Code terminal, type

* node server.js.

wait the message

**Image 1:** This is the message if run the server

**Step 7:** Access the system

* Open a browser and go to http://localhost:3000.
* You can now log in as admin or test booking as client using Google Sign-In.

**PRE-DEPLOYMENT STEPS**

**Step 1:** Backup existing data (if applicable)

* Since the system is currently on a local setup, there may not be existing data yet.
* But if testing data already exists in SQL Server, export it using SSMS by right-clicking the database
* Save the backup file in a safe folder for restoration later if needed.

**Step 2:** Set up the environment

* Install Node.js, Microsoft SQL Server, and SQL Server Management Studio (SSMS) on the machine.
* Install Visual Studio Code (VS Code) as the main development and deployment editor.
* Create a new project folder EventVenueBookingSy and include the source files (server.js, public folder, .env, etc.).

**Step 3:** Configure the database

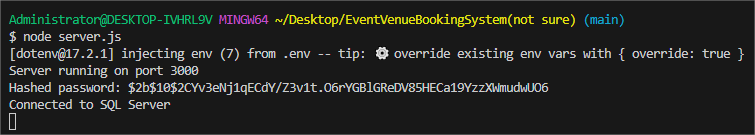
* Open SSMS, create a new database (e.g., eventease\_db).
* Run SQL scripts to create required tables.
* Test the connection between Node.js and SQL Server by running the server in VS Code node server.js.

**Step 4:** Check network connectivity and compatibility

* Ensure SQL Server is running and can accept local connections.
* Allow TCP/IP connections in SQL Server Configuration Manager.
* Make sure no other process is using port 3000 the default port in your app.

**Step 5:** Verify system readiness

* Confirm that dependencies in package.json are properly installed using



**Image 2:** This is the message if run the server

* Test the system on browser using http://localhost:3000.
* If the app runs without errors, proceed to deployment.

**DEPLOYMENT EXECUTION**

**Deploy application files or upload to server**

After completing the setup, follow these steps to deploy the Event Venue Booking System for City of Dreams Manila

**Configure system settings**

* Copy all project folders (HTML, CSS, JS files, and Node.js backend) to the working directory of your project.
* Open Microsoft SQL Server Management Studio (SSMS) and create a new database. Import the tables and data used by the system.
* Update the database connection in your Node.js project (usually found in .env or config.js) with the correct server name, database name, username, and password.

**Perform system initialization and check for errors.**

* Run npm install in the project directory to install all required dependencies.
* Start the Node.js server using node server.js or npm start.
* Open a browser and go to http://localhost:3000 (or the port you set) to access the system.
* Test the website by selecting a venue and logging in through Google.
* On the admin side, check if the booking details show correctly and confirm if the email notification is sent properly.
* After verifying everything works, the system can now be fully used by City of Dreams Manila.

**POST-DEPLOYMENT STEPS**

**Verify functionality with test cases**

After deploying the Event Venue Booking System, the next step is to make sure that everything is working properly. First, the system’s functionality should be verified using test cases. This includes checking if clients can successfully log in with their Google account, select venues, and submit bookings. On the admin side, the confirmation, cancellation, and email notification features must be tested as well.

**Monitor system performance and stability**

Once testing is complete, system performance and stability should be monitored to ensure that it runs smoothly without errors or crashes while on localhost. If any issues appear, they should be fixed before full use.

**Conduct user training**

Lastly, if the staff from City of Dreams Manila or other users will manage the system, a short user training can be conducted to help them understand how to navigate the system, handle bookings, and use the admin dashboard properly.

**USER TRAINING & SUPPORT**

**Training schedule for users**

For the City of Dreams Manila staff who will use the Event Venue Booking System, a short user training can be done after deployment. The training can be scheduled once the system is fully tested and working on localhost. During the session, users will be guided on how to log in using their Google account, select and book venues, and for the admin side, how to review, confirm, or cancel client requests.

**Documentation or manuals provided to users**

A simple user manual will also be provided, containing screenshots and step-by-step instructions on how to use each feature of the system. This will help users easily understand the process even without direct assistance.

**Support contact details for troubleshooting**

For technical support or troubleshooting, users can contact the system developer through email or chat in case they encounter errors or issues while using the system.

**RISKS & CONTINGENCY PLAN**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Impact** | **Mitigation Strategy** |
| **Server downtime** | High | Since the system is currently running on localhost, regular backups should be made to prevent data loss. When hosted online, ensure there’s a backup server and inform users ahead of maintenance. |
| **Database connection failure** | Medium | Test the database connection before deployment and double-check the configuration in the .env file to avoid errors. |
| **Email sending error** | Medium | Verify the SMTP settings in the Node.js server and ensure the internet connection is stable before sending confirmation emails. |
| **User login issues (Google)** | Low | Recheck Google API credentials and make sure OAuth permissions are properly configured |
| **User resistance** | Low | Conduct basic training for users and provide a simple guide on how to use the system effectively. |

**Table 4:** RISKS & CONTINGENCY PLAN

**DEPLOYMENT VERIFICATION & SIGN-OFF**

After the deployment of the Event Venue Booking System, several tests were conducted to ensure that all core features were functioning correctly. The client login using Google, venue selection, booking process, admin confirmation, and email notification were all tested successfully. The system performed well on localhost without any major issues or errors.

Below is the sign-off table to confirm that the deployment has been completed and verified by both the developer and the client.

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Role** | **Signature** | **Date** |
| Albania, John Maverick B. | Project Developer |  | 10-20-25 |
| City Of Dreams Manila | Client |  | 10-20-25 |