# Technical Documentation

**{EMS (XYZ) System Application}**

{Rashed Hasan Qahah - 21026420}

1. **Introduction**
   1. **Problem Definition and Solution**

Therefore, the main problem that XYZ Events aims to solve by using the EMS event management system is the complexity and inefficiency that we currently face in planning and implementing events. Traditional approaches often involve disparate systems for communications, resource allocation, attendance management, and compliance, leading to confusion, errors, and wasted resources. Additionally, manual processes can be time-consuming and prone to human error, affecting the overall quality and success of events.

This may also lead to poor resource allocation, poor communication between team members, lack of insights based on data and information, and lack of attendance interaction. These shortcomings can lead to suboptimal event experiences, higher costs, and lower customer satisfaction. Existing manual or semi-automated processes may not provide the scalability and accuracy required to handle diverse and complex events, leading to errors such as double booking, mismanagement of attendance information, and insufficient compliance with regulations.

Therefore, to address these challenges, the solution is to develop a comprehensive and easy-to-use event management system (EMS) that takes advantage of advanced and effective technology in order to simplify and enhance the entire event management life cycle. XYZ Events requires an integrated and efficient solution that streamlines all aspects of event management into a cohesive, easy-to-use platform. This system needs to facilitate seamless communication between team members, improve resource allocation, provide detailed data analysis to make informed decisions, enhance attendee engagement through a seamless registration process, and ensure regulatory compliance. By addressing these issues, EMS aims to enhance efficiency, reduce errors, deliver exceptional value to customers, and set new standards in the event management industry.

The proposed solution is to develop an Event Management System (EMS) that incorporates advanced technologies and best practices to simplify the entire event management process. The EMS will feature an easy-to-use interface, comprehensive capabilities for event management, attendance recording and ticketing, and powerful data processing for compliance and analysis.

* 1. **System and user requirements**

**User Requirements:**

1. Event Organizers:

* Account Creation and Management: Ability to create and manage their accounts with personal details like name, email, and phone number, Password.
* Event Creation and Management: Tools to create, edit, and manage events with specifications such as event name, date, location, capacity, type, and description.
* Event Schedule View: Access to a comprehensive list of all their events to avoid scheduling conflicts.

1. Attendees:

* Registration: Simple registration process requiring personal details like name, email, and phone number.
* Event Selection: Ability to browse and select events from a list of available events.
* Ticketing: Option to select ticket types and prices.

**System Requirements:**

System requirements refer to the specific criteria that a system must meet in order to effectively deliver high-quality services to its users. System requirements for the XYZ Events System encompass:

Integration of databases: The integration of the XYZ Events System with the database should ensure that any newly added, altered, or removed events are securely and consistently saved. When a certain alteration occurs in the system, the database should accurately mirror that modification on the XYZ Events System by simply retrieving the data that already exists in the database.

**Physical components of a computer system, such as the processor, memory, and storage devices.**

To ensure optimal functionality and delivery of necessary services to system users and administrators, it is imperative to furnish the system with critical hardware components. Some of the components may comprise the following:

Server: To provide efficient and effective data processing, a server is necessary to handle the numerous requests from users on the online application. Furthermore, all crucial data pertaining to the events, and beyond, will be securely stored and preserved, and may be readily accessed at any time by authorized personnel. With a reliable server, consumers have the ability to utilize the system at their convenience for booking and viewing events. Additionally, administrators can efficiently manage events and perform other necessary tasks as needed.

An essential element for the system is a robust cooling system to provide regular maintenance and prevent the server from overheating, which could lead to system failure or catastrophic consequences. The purpose of the cooling systems is to maintain the server's temperature within an optimal range, neither too high nor too low. By doing this action, the server will be capable of operating efficiently and handling data and traffic in accordance with the necessary requirements.

Memory: Another essential hardware need for the system is a sufficient amount of RAM. This is due to the fact that having ample memory results in a substantial enhancement in the speed and efficiency of the system, enabling it to effectively handle numerous procedures and demands with greater speed and efficiency. By possessing a proficient memory, the system will have the capacity to promptly exhibit any modifications implemented within the system. For instance, when an administrator endeavors to modify some information regarding a preexisting event, the system should promptly exhibit the recently-updated data to the users.

Secondary storage is a crucial hardware component that is necessary for the efficient operation of the XYZ Events System. This feature is crucial for the system as it enables the storage of information pertaining to users, events, and administrators. Secondary storage has the capacity to efficiently store and retain large amounts of data over an extended period of time. This guarantees the safety, security, and preservation of data, even in the event of a system shutdown, in contrast to main memory or RAM.

**Software requirements** are an additional aspect of system requirements. These are computer programs or applications that the system can utilize to provide certain services or improve them. Software requirements encompass a variety of examples.

Firewall: One of the important things that the XYZ Events System can certainly take advantage of is using a firewall to ensure that no unauthorized access occurs. The firewall essentially acts as a layer of security or defense for the network, and will only allow certain individuals to access the system depending on how it is set up; as the firewall manages the ongoing and outgoing traffic for the network. In the context of the XYZ Events System, because the web application is a public service, there is no problem with pretty much any IP address or user accessing it. However, it should be ensured that no one other than the admin is able to access the administrator dashboard and use it. Thus, the firewall can be set up for the private network side of the company to ensure that only authorized people are allowed there. From this, we see that the firewall essentially prevents private data from being disclosed and used inappropriately.

Data Backup: Another thing the XYZ Events System can take advantage of regarding the software side of the requirements is to have a backup service for the system. This is because the data inside the system can easily be vulnerable, and if correct measures are not followed, then the data of the employees, users, and many more can be lost, resulting in even bigger issues for the company. By performing a backup on the data at least once a week, the company will make sure that in the event the data is lost, destroyed, or the company faces a cyber-attack of any sort, there is already a copy of the data that was not destroyed and was protected. The XYZ Events System can then operate on the data from the backup instead of suffering from the destruction of all data.

UML Design Tools: One of the important things for the XYZ Events System is for it to be designed well in terms of functionalities, database, and more. UML design tools are very useful as they give the developer a visual representation of how the XYZ Events System is going to look, along with the actions and methods it provides to the users prior to building the system itself. The design step is quite essential as it is also one of the SDLC phases for any system or software being built. After the design is completed, the developer can bring life to the design by applying it on the system using programming languages to develop the web application for the XYZ Events System.

Software Testing Tools: This is one of the essential things for the XYZ Events System as well; because it is also one of the SDLC phases of testing. Testing the software or system being built is very important to make sure that the validation and verification are being done properly. In the context of the XYZ Events System, it is absolutely important to apply testing to different areas within the system such as unit testing, performance testing, and more. With this, the company ensures that the software is performing as required, and the clients or customers will be pleased with using and taking advantage of the system.

1. **Development Environment**
   1. **development environment tools and methodologies**

The selection of tools and methodologies for an EMS system is closely interconnected, with the goal of creating a cohesive development environment that maximizes efficiency, effectiveness, collaboration, and adaptability. Let me explain and delve into the rationale behind each choice and identify the connections between them.

First, choosing Windows as an operating system (OS) aligns with the need for an easy-to-use interface, comprehensive support, and strong security features. Windows provides an intuitive platform for developers to work on, reducing the learning curve and enhancing productivity. Its compatibility with a wide range of third-party software and hardware ensures seamless integration of essential tools and components for EMS development. Furthermore, regular updates from Microsoft ensure that the operating system remains secure and up-to-date, protecting the EMS system from vulnerabilities. This choice is reinforced by the Agile methodology, where an easy-to-use Windows interface promotes collaboration and seamless transition between project phases, consistent with Agile's focus on customer collaboration and adaptability.

The choice of Windows as the operating system for EMS development is justified by its ease of use, comprehensive support, and security features. Windows provides a familiar and intuitive interface, which reduces the learning curve for developers and end users alike and provides many tools and programs that have always been available in Windows operating systems. This knowledge can facilitate smoother transitions and user acceptance of the EMS system. Furthermore, Windows' compatibility with a wide range of third-party software and hardware simplifies the integration process, which is critical for EMS development that may require communication with different devices and systems.

On the other hand, choosing Windows is consistent with the principles of the Agile methodology, especially its focus on collaboration and adaptability. The easy-to-use Windows interface enhances collaboration among team members, enabling them to work efficiently on different aspects of the project. Additionally, Windows' robust security features ensure the integrity and confidentiality of EMS data, addressing Agile's focus on risk management.

Secondly, in terms of integrated development environments (IDEs), Visual Studio Code (VS Code) is chosen due to its lightweight nature, extensive customization options, and strong community support. The versatility and efficiency of VS Code makes it an ideal choice for EMS development, enabling developers to design their own development environment to suit their specific needs and also providing many tools and extensions that help developers and programmers create effective and efficient projects. The lightweight nature of VS Code ensures smooth performance, which is essential for handling complex EMS projects. Extensive customization options and plug-in support improve productivity and collaboration, aligning with Agile's focus on agility and customer collaboration. Furthermore, VS Code's compatibility with different programming languages ensures compatibility with the diverse requirements of the EMS system, facilitating iterative development cycles.

Also choose Visual Studio Code (VS Code) as the IDE completes the choice of Windows operating system. VS Code's lightweight nature and extensive customization options make it well-suited for EMS development, ensuring smooth performance and enabling developers to tailor their development environment to suit their specific needs. Its compatibility with different programming languages meets the diverse requirements of EMS development, allowing developers to work seamlessly across different aspects of a project.

The choice of VS Code also aligns with Agile's focus on flexibility and customer collaboration. The versatility of VS Code allows for rapid iteration and adaptation to changing requirements, which is essential for an iterative Agile approach. Additionally, strong community support fosters collaboration and knowledge sharing between developers, reinforcing Agile principles of teamwork and communication.

Third, for diagramming software, Draw.io and Visual Paradigm were chosen for their ease of use, comprehensive features, and cloud integration capabilities. Draw.io provides an intuitive interface and a wide range of features for creating detailed diagrams, essential for visualizing and designing an EMS system architecture. Its cloud integration capabilities enhance collaboration, allowing team members to work together in real-time and ensuring designs are constantly updated and accessible. On the other hand, a visual model provides extensive diagramming capabilities and superior aesthetic quality, making it suitable for designing complex systems. Its integration with development tools and comprehensive language support aligns with EMS system requirements, facilitating seamless integration of charts into the overall project workflow.

So, choosing Draw.io and Visual Paradigm for your diagramming software complements the Agile development process by facilitating visual communication and collaboration. Draw. Io ease of use and comprehensive features allow developers to create detailed diagrams depicting the architecture and design of an EMS system. Its integration with cloud services enhances collaboration, allowing team members to work together in real-time and ensuring designs are constantly updated and accessible.

Visual Paradigm provides extensive diagramming capabilities and professional features that match the complexity of EMS development. Its integration with development tools simplifies workflow, enabling seamless integration of diagrams into the development process. The selection of both Draw.io and Visual Paradigm supports Agile methodology's focus on customer collaboration and continuous improvement, enabling stakeholders to provide feedback on EMS visual representations and facilitate iterative improvement.

Fourth, choosing an Agile methodology enhances the choice of tools by focusing on agility, customer collaboration, and risk management. Agile's iterative approach and regular feedback loops complement the features of Windows, VS Code, Draw.io, and Visual Paradigm, ensuring the development process remains dynamic and responsive to changing requirements. The iterative nature of Agile allows for continuous improvement and adaptation based on feedback and evolving stakeholder needs, aligned with the collaborative and iterative features of the identified tools.

Choosing an Agile methodology also links choices of operating systems, integrated development environments (IDEs), and planning software by providing a framework for effective iterative development, collaboration, and adaptability. Agile's iterative approach allows for continuous improvement and adaptation based on feedback and changing requirements, ensuring that the EMS system remains dynamic and responsive to stakeholder needs. Its focus on customer collaboration enhances communication and alignment between developers and end users, reinforcing the importance and ease of use of the EMS system.

* 1. **List of frameworks and libraries used in the development of the application**

**Frontend**: The frontend refers to the part of the software or application that users interact with directly. It encompasses the user interface (UI) components, layouts, and features that users see and interact with when using the system. In the context of XYZ Events' Event Management System (EMS), the frontend would include elements such as the user interface for event creation, attendee registration forms, event listings, navigation menus, and any other visual components that organizers and attendees interact with. The frontend is responsible for presenting data and enabling user interactions in a visually appealing and intuitive manner. Technologies commonly used in frontend development include HTML, CSS, JavaScript, and frontend library and frameworks like React, Angular, or Next.

**Frontend Technologies:**

**HTML**: HTML (Hypertext Markup Language) serves as the foundation of web pages, providing the structure and content of the EMS interface. By leveraging HTML, XYZ Events can ensure the creation of structured and semantically meaningful web pages. HTML facilitates the organization of content elements such as headers, paragraphs, forms, and lists, enabling a clear and intuitive user interface (UI). Additionally, HTML supports the integration of multimedia elements like images and videos, enhancing the visual appeal and interactivity of the EMS interface. With HTML, XYZ Events can design user-friendly pages that are accessible across different devices and platforms, ensuring a seamless experience for organizers and attendees alike.

**CSS**: CSS (Cascading Style Sheets) plays a crucial role in styling and formatting the visual presentation of HTML elements within the EMS interface. By implementing CSS, XYZ Events can customize the appearance of various UI components, including fonts, colors, layouts, and animations. CSS empowers XYZ Events to create visually appealing and cohesive designs that align with their brand identity and enhance user engagement. Moreover, CSS enables responsiveness, allowing the EMS interface to adapt dynamically to different screen sizes and resolutions, optimizing the user experience across desktops, tablets, and mobile devices. Through the strategic use of CSS, XYZ Events can elevate the aesthetics and usability of their EMS, leaving a lasting impression on organizers and attendees.

**The** **Bootstrap:** library plays a pivotal role in the development of the frontend for XYZ Events' Event Management System (EMS). Bootstrap, a widely-used CSS framework, provides a robust foundation for creating responsive and visually appealing web interfaces. By leveraging Bootstrap, XYZ Events ensures a consistent and modern design across the EMS, enhancing user experience for both event organizers and attendees. The library offers a rich collection of pre-designed components such as navigation bars, forms, buttons, and modals, which streamline the development process and maintain a cohesive aesthetic throughout the application. Furthermore, Bootstrap's responsive grid system allows the EMS interface to adapt seamlessly to various screen sizes and devices, ensuring accessibility and usability. Through the strategic use of Bootstrap, XYZ Events can deliver a polished and user-friendly interface that aligns with their brand and meets the dynamic needs of event management.

**JavaScript**: JavaScript serves as the backbone of dynamic and interactive functionalities within the EMS interface. By leveraging JavaScript, XYZ Events can enhance user engagement and interactivity by implementing features such as real-time updates, form validations, and event-driven interactions. JavaScript enables XYZ Events to create dynamic UI elements that respond to user actions, providing instant feedback and improving usability. Additionally, JavaScript frameworks like React can be utilized to streamline the development process and build complex UI components efficiently. With JavaScript, XYZ Events can enrich the EMS interface with seamless navigation, responsive design, and interactive elements, ensuring a delightful experience for users.

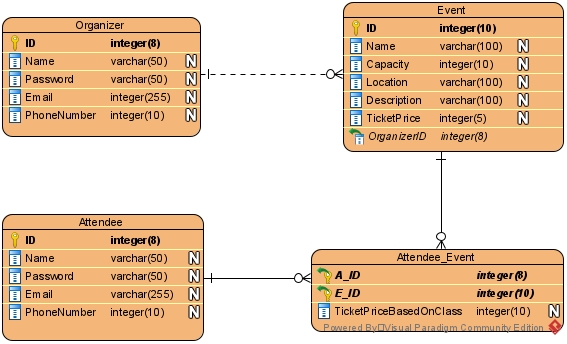
**Backend**: The backend, also known as the server-side, refers to the part of the software or application that runs on the server and is responsible for handling data storage, processing, and business logic. In the context of XYZ Events' EMS, the backend would include functionalities such as user authentication, event management, data storage and retrieval, and communication with external services or databases. The backend processes requests from the frontend, performs necessary operations, retrieves data from the database, and sends responses back to the frontend. Technologies commonly used in backend development include programming languages like PHP, Python, Ruby, or Node.js, as well as frameworks like Laravel, Django, Ruby on Rails, or Express.js. Additionally, databases such as SQL or NoSQL databases are often used to store and manage data in the backend.

**Backend Technologies:**

**SQL**: SQL (Structured Query Language) is a powerful tool for managing and querying relational databases, serving as the backbone of the EMS backend infrastructure. By utilizing SQL, XYZ Events can design and maintain a robust database schema to store and retrieve event-related data efficiently. SQL enables XYZ Events to perform complex database operations such as data insertion, retrieval, updating, and deletion, ensuring the integrity and consistency of attendee and event information. Moreover, SQL supports the implementation of advanced database features like transactions, constraints, and indexes, optimizing performance and reliability. With SQL, XYZ Events can build a scalable and secure backend system that meets the evolving needs of event management.

**PHP**: PHP (Hypertext Preprocessor) serves as the server-side scripting language for dynamic web application development, providing the logic and functionality of the EMS backend. By leveraging PHP, XYZ Events can handle user requests, process form submissions, and interact with the database to perform CRUD (Create, Read, Update, Delete) operations. PHP enables XYZ Events to implement business logic and server-side validation, ensuring data integrity and security. Additionally, PHP frameworks like Laravel can be utilized to expedite development and maintainability, offering features such as routing, authentication, and session management. With PHP, XYZ Events can build a scalable and performant backend system that powers the EMS with efficiency and reliability.

1. **Database Design:**
   1. **Entity-Relationship Diagram (ERD) depicting the database schema.**



**Entity-Relationship Diagram (ERD):**

The ERD provided outlines the structure of an event management database, featuring three primary entities: Organizer, Event, and Attendee. Additionally, it includes an associative entity, Attendee\_Booking, to facilitate the many-to-many relationship between Event and Attendee.

**Description of Database Tables:**

**Organizer Table:**

* ID: This is the primary key for the Organizer table, uniquely identifying each organizer with an integer value.
* Name: A varchar field that stores the name of the organizer, which is essential for identifying the organizer in a user-friendly manner.
* Password: A varchar field designated for the organizer's password, ensuring secure access to the organizer’s account.
* Email: A varchar field storing the email address, which is crucial for communication and account recovery.
* PhoneNumber: An integer field for storing the organizer's phone number, used for contact purposes.

**Event Table:**

* ID: This primary key uniquely identifies each event with an integer value.
* Name: A varchar field that holds the name of the event, facilitating easy identification.
* Type: A varchar field describing the type or category of the event (e.g., conference, workshop).
* Capacity: An integer indicating the maximum number of attendees allowed, which helps in managing event logistics.
* Location: A varchar field specifying the venue of the event.
* Description: A varchar field providing a brief overview of the event.
* TicketPrice: An integer indicating the price of attending the event.
* OrganizerID: A foreign key linking to the Organizer table, ensuring that each event is associated with a valid organizer.

**Attendee Table:**

* ID: The primary key for the Attendee table, uniquely identifying each attendee.
* Name: A varchar field for the name of the attendee, aiding in personal identification.
* Password: A varchar field for the attendee’s password, securing their account.
* Email: A varchar field storing the email address, important for communications and notifications.
* PhoneNumber: An integer field for the attendee’s phone number, used for contact purposes.

Attendee\_Booking Table:

* A\_ID: A foreign key referencing the Attendee table, indicating which attendee is booking the event.
* E\_ID: A foreign key referencing the Event table, indicating which event is being booked by the attendee.
* The composite primary key (A\_ID, E\_ID) ensures that each attendee can only book a specific event once, maintaining the integrity of the booking process.
  1. **Description of database tables, relationships, and constraints.**

**Relationships:**

**One-to-Many Relationship (Organizer - Event):**

This relationship indicates that one organizer can host multiple events. The OrganizerID in the Event table serves as a foreign key linking back to the Organizer table. This relationship ensures that all events are associated with a legitimate organizer, maintaining organizational structure and accountability. For instance, an organizer may be responsible for multiple workshops, conferences, or seminars, and this linkage helps in tracking the performance and involvement of each organizer across different events.

**Many-to-Many Relationship (Attendee - Event):**

The many-to-many relationship between Attendee and Event is managed by the Attendee\_Booking table. This table's composite primary key (A\_ID, E\_ID) ensures that each booking is unique, preventing duplicate bookings for the same event by the same attendee. This relationship is vital for capturing the dynamic nature of event participation where multiple attendees can join multiple events. This structure allows for comprehensive tracking of event attendance, which is crucial for logistical planning, resource allocation, and understanding attendee preferences and behaviors.

**Constraints:**

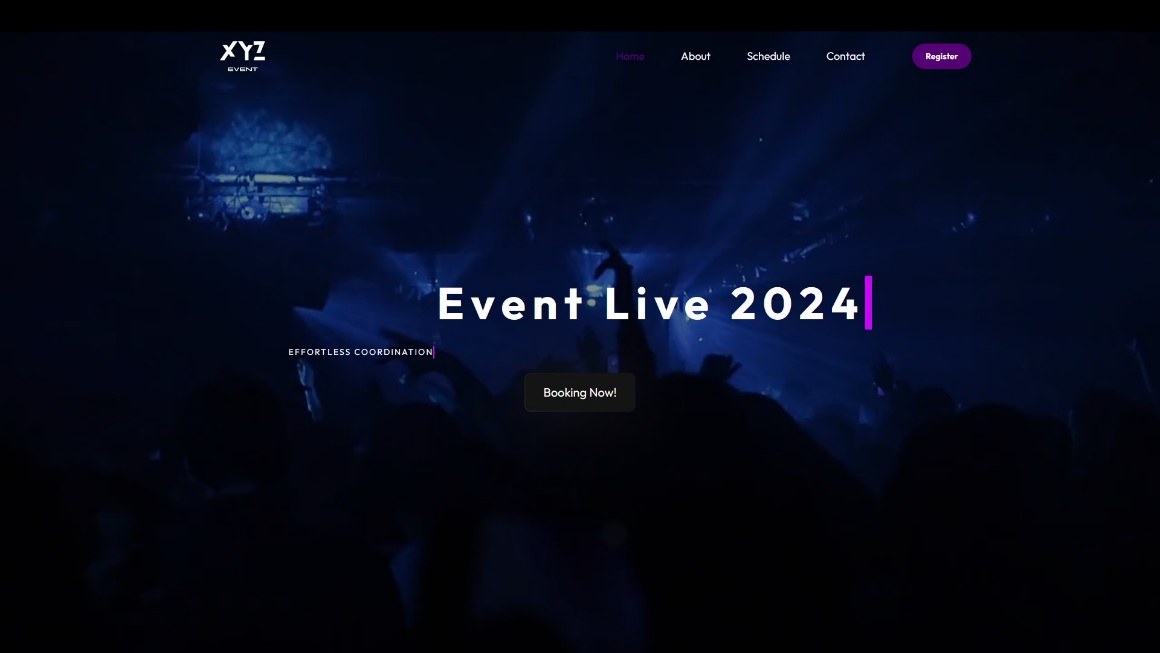
**Foreign Key Constraints:**

Foreign key constraints in the Event and Attendee\_Booking tables enforce referential integrity by ensuring that the entries in these tables are valid. For example, OrganizerID in the Event table must match an existing ID in the Organizer table, ensuring that an event cannot exist without a valid organizer. Similarly, A\_ID and E\_ID in the Attendee\_Booking table must correspond to valid IDs in the Attendee and Event tables, respectively. This integrity check is crucial for maintaining accurate and reliable data within the database.

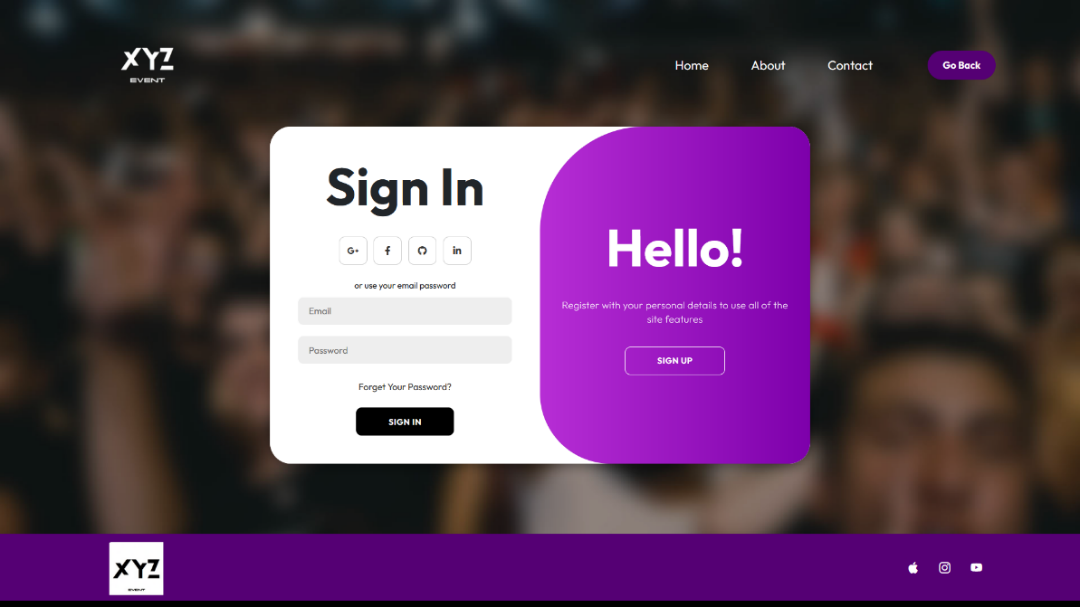
**Composite Primary Key:**

The composite primary key in the Attendee\_Booking table prevents an attendee from booking the same event multiple times, which is essential for preventing data redundancy and ensuring accurate booking records. This constraint helps maintain the quality of the booking data, ensuring that each entry is unique and meaningful. By enforcing this rule, the system can efficiently manage event capacities and avoid overbooking scenarios, thus enhancing the overall event management process.

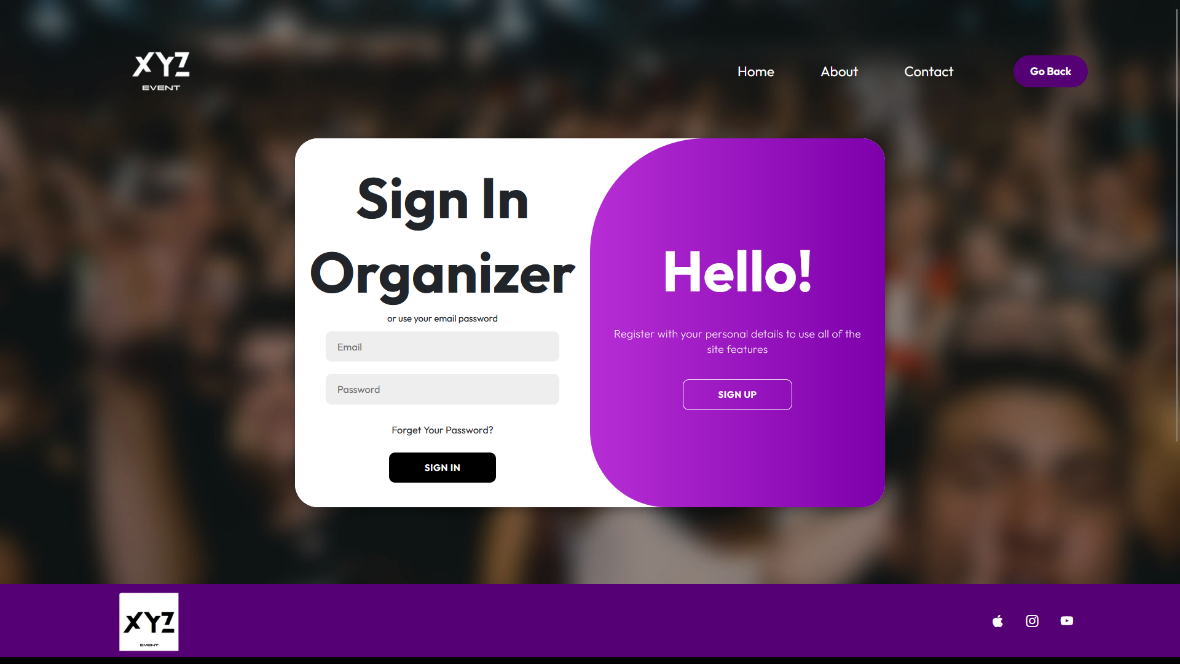
1. **User Manual**
   1. **Step-by-step instructions for using the application**



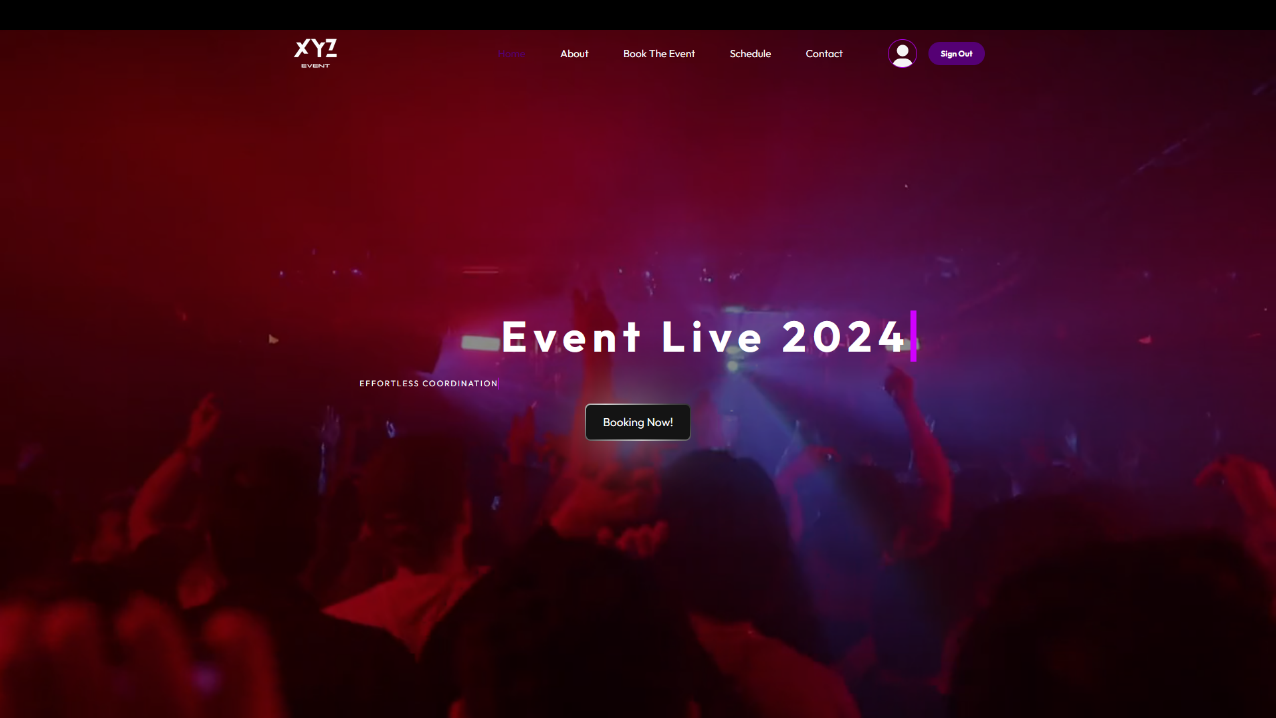
**First, the home page,** which contains a header and consists of a logo for the application and also navigation and contains elements found in the navigation, which are home, the schedule, about us, communication, and also contains a registration button for attendees and also for organizers.

The page also contains a landing section in which animated text appears in the middle of the page, as well as a background video, and also a button “Booking Now!”. The page also contains many sections that give beauty to the home page. These sections include the table, about us, and contact. The page also contains a footer that gives additional information about the site, including contact numbers and others.

**Secondly**, when you click on the registration button, there are two options through which registration for attendees and also registration for organizers appear. When you click on register for attendees, it takes you to the attendance registration page, through which you can create a new account and then you can log in smoothly and efficiently.

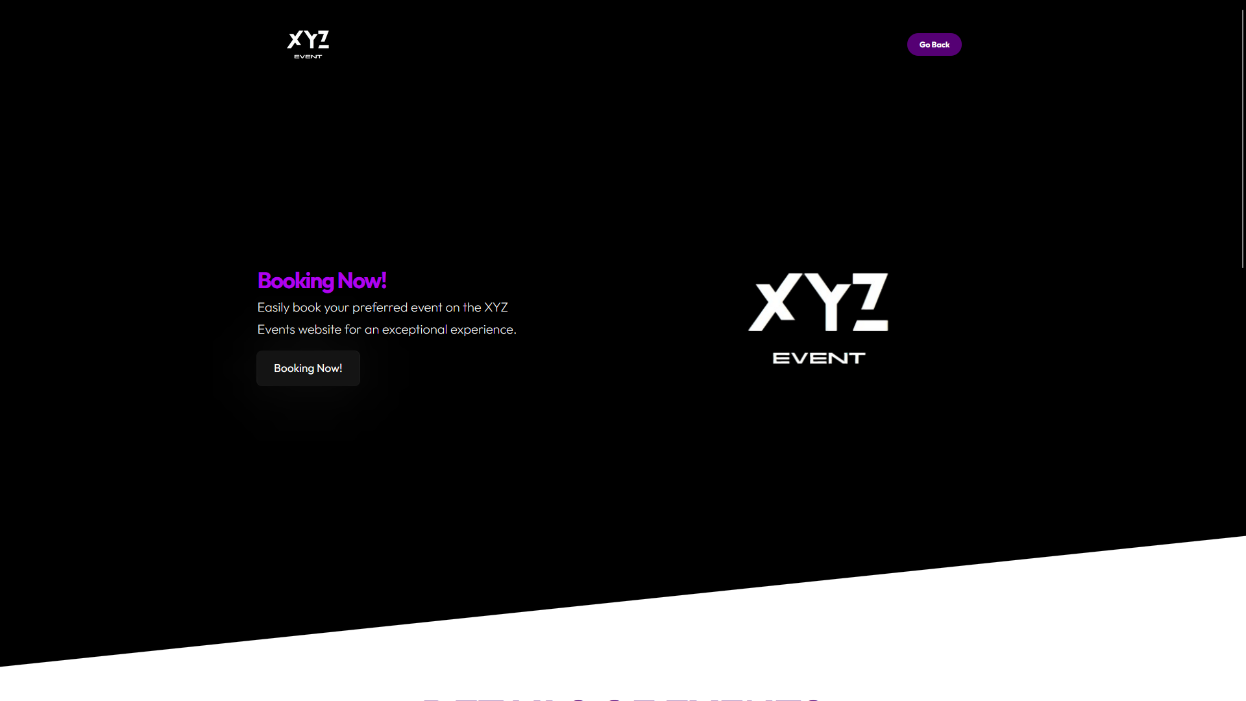


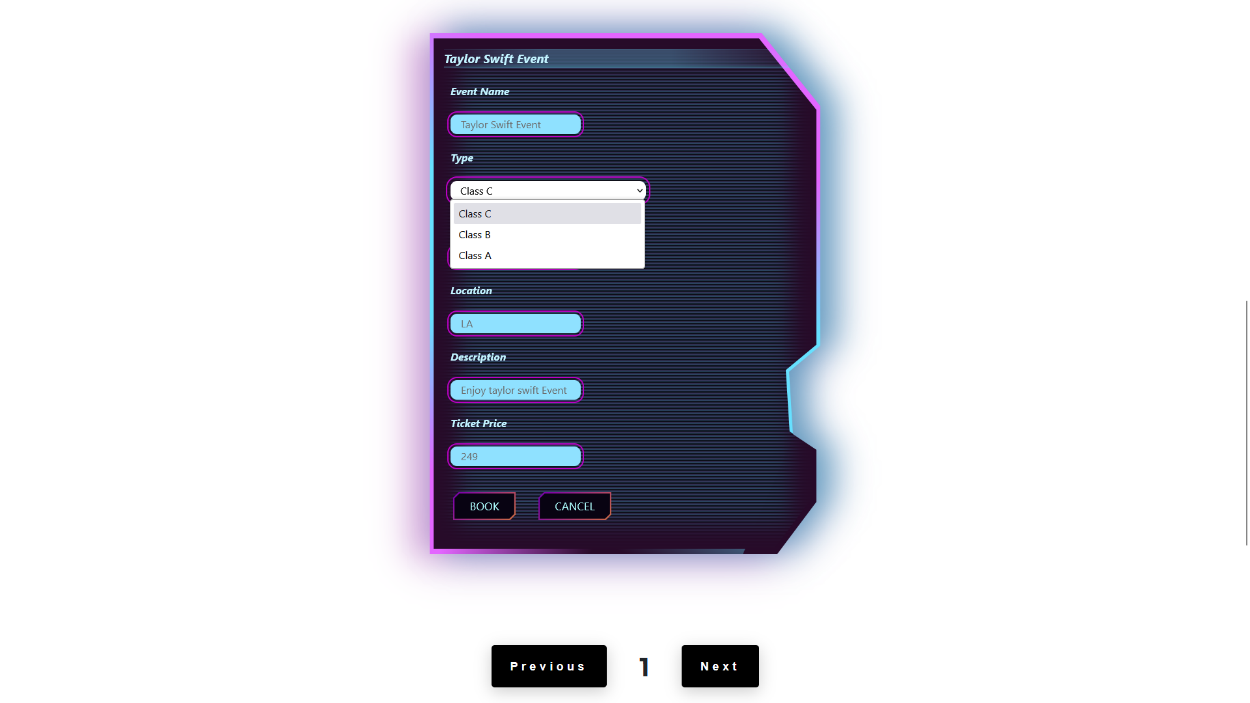
Thirdly, when you click on the registration button, there are two options through which registration for attendees and also registration for organizers appear. When you click on Register for Organizers, it takes you to the Registration for Organizers page, through which you can create a new account and then you can log in smoothly and efficiently.



Fourth, when you register as an attendee, it takes you to the attendees’ home page, which contains a header and consists of a logo for the application as well as navigation. It contains the elements in the navigation, which are the home page, Book the event, schedule, us, and communication. It also contains a sign-out button, and when you press the item in the header “Book the event” it takes you to the event booking page for attendees.

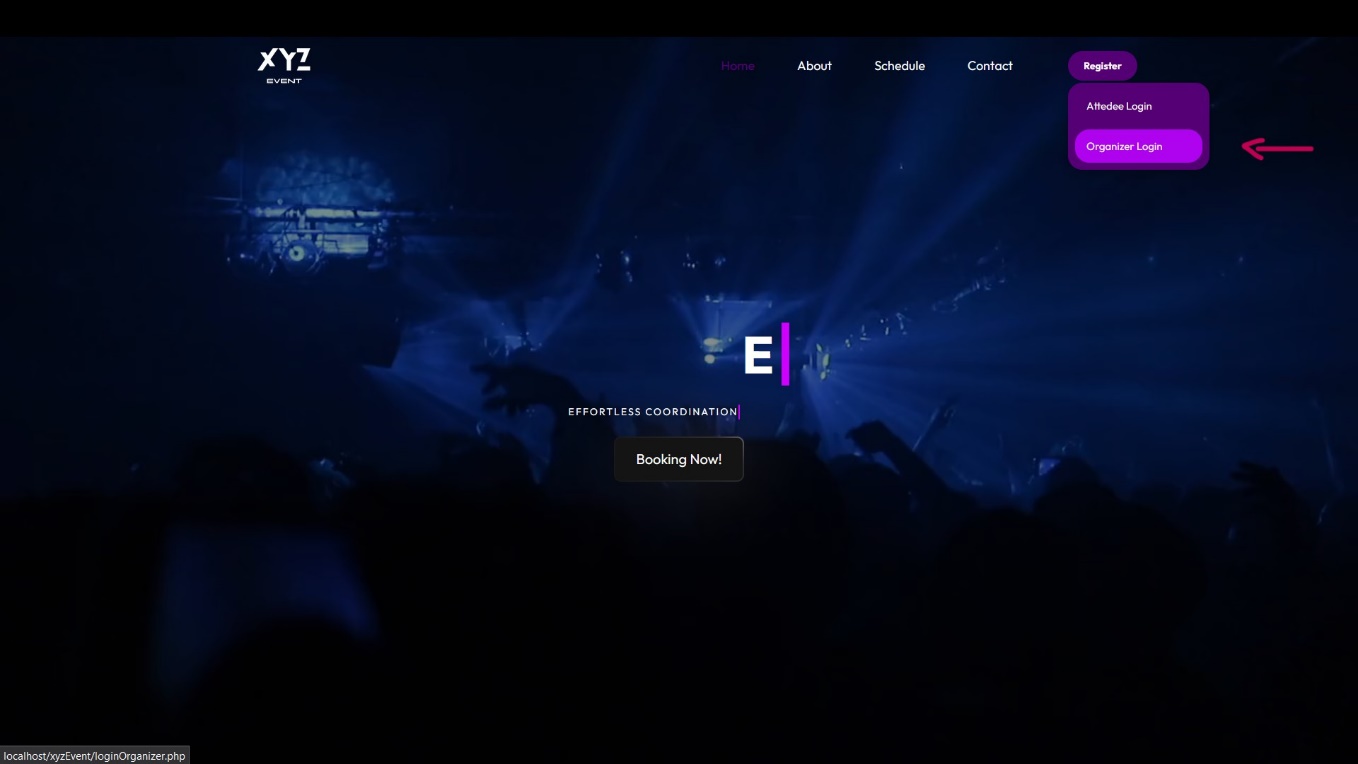
The page also has a landing section in which animated text appears in the middle of the page, a video in the background, and a “Booking Now!” button. Through it, it also takes you to the event booking page for attendees. The page also contains many sections that add beauty to the home page. These sections include Schedule, About Us, and Contacts. The page also contains a footer that gives additional information about the site, including contact numbers and more.





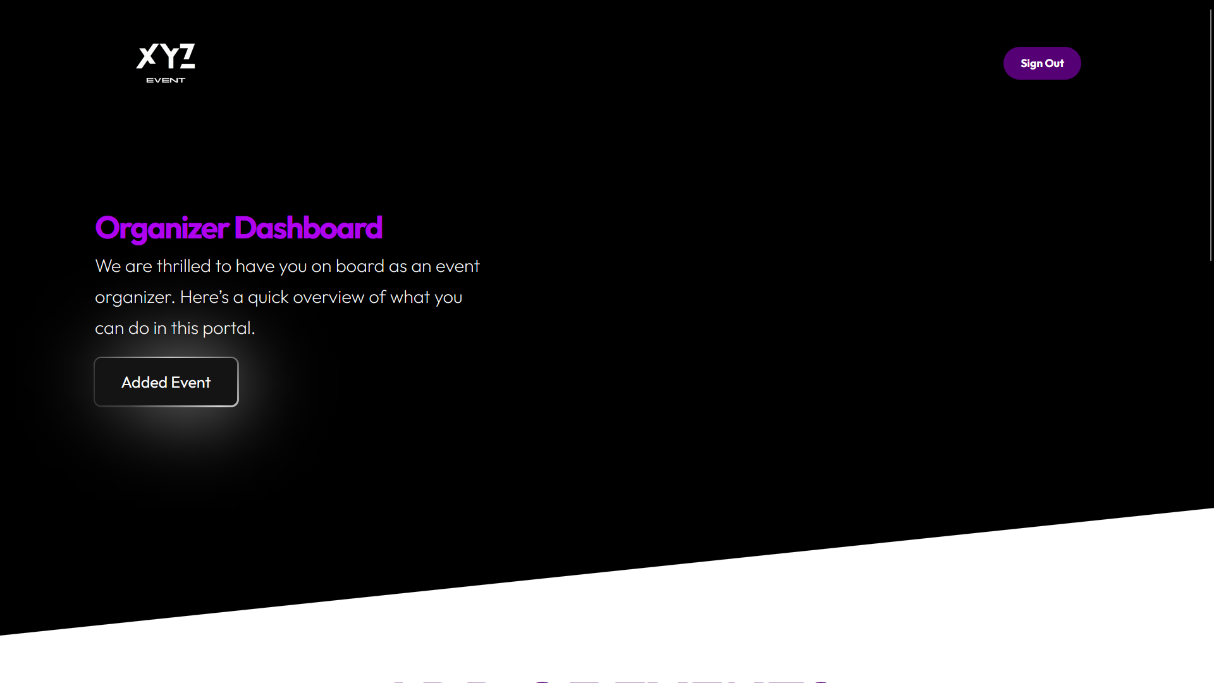
Fifthly, this is a special page for booking events for attendees, through which attendees can book events smoothly, easily and effectively. Therefore, it appears on the Landing page, which contains a paragraph and also a “Book Now” button, which when you click on it will take you to the bottom of the page. Immediately after that, Cards will appear for you. Through it, details of the events are shown, including the name of the event, the type of event, the type of ticket, the attendance capacity, the ticket price, a description of the event, and also the location of the event. There are two buttons through which you can book the event and the other button through which you can cancel the reservation.

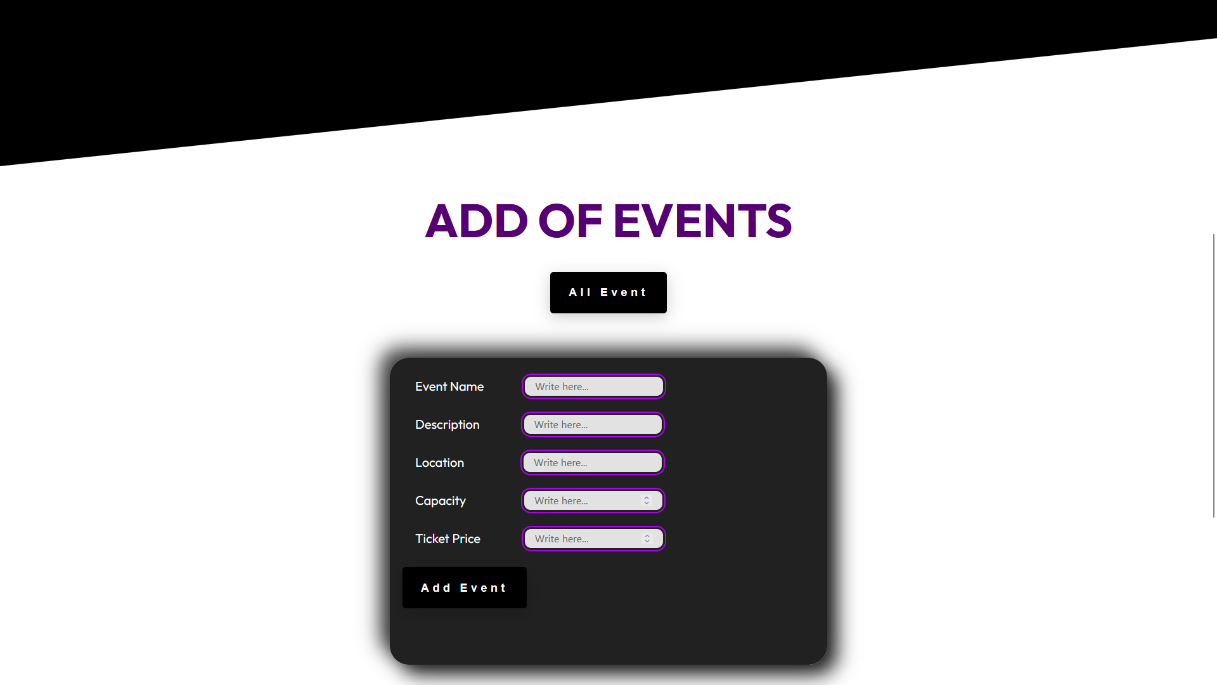
Below the card there is a feature through which you can, through the next and previous buttons, move between cards that contain event details, indicating the event details card number.



Sixth, when you register as a developer, it takes you to the general home page, which has a header and consists of a logo for the application as well as navigation. It contains the items in the navigation, which are Home, Schedule, About Us, and Contact. It also contains a registration button for both attendees and organizers, and when you click on the register button as an organizer, it takes you to a special registration page for organizers.

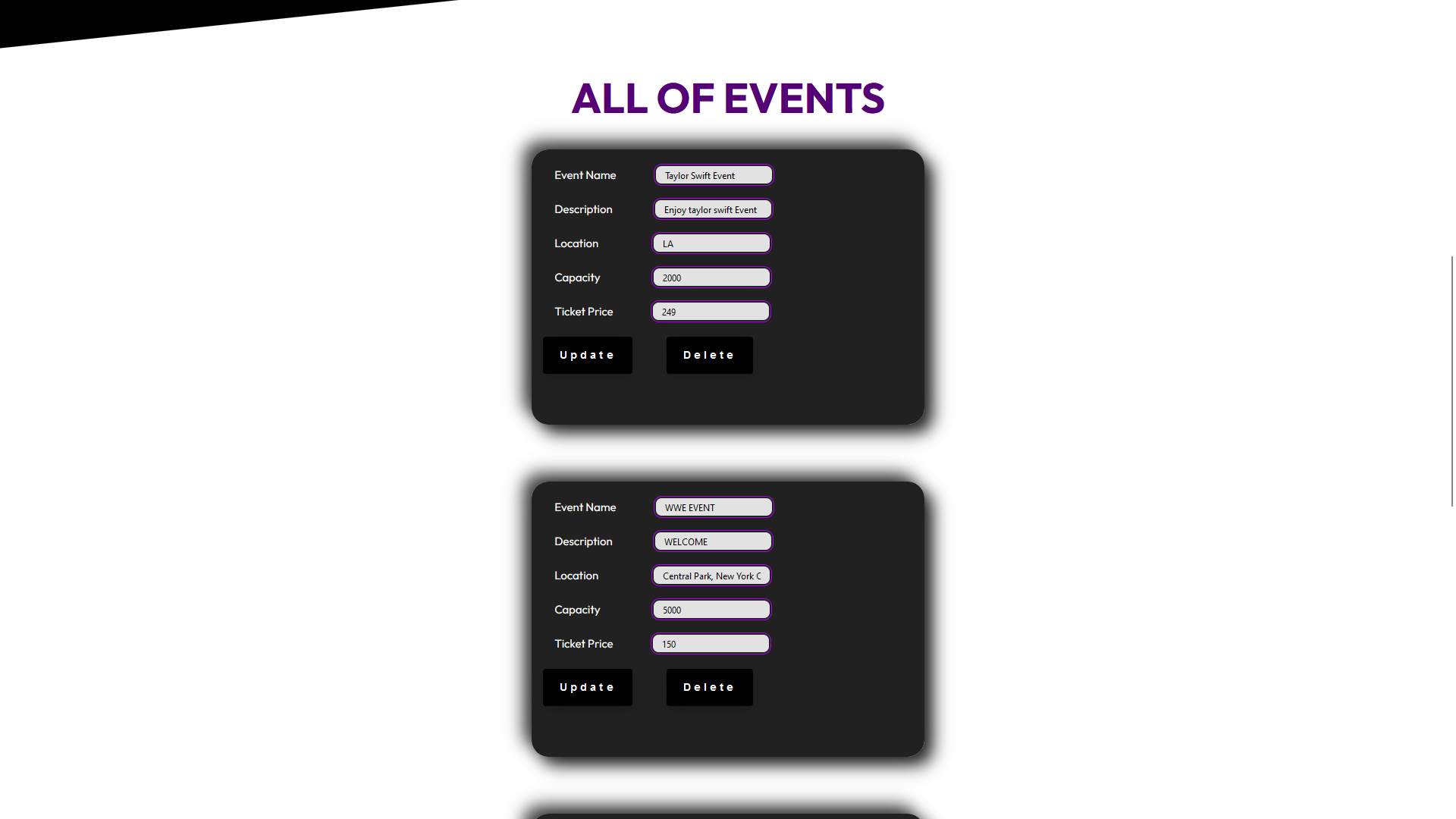
When you click on the registration button, there are two options through which the registration appears for attendees and also the registration for organizers. When you click on Register for Organizers, you will be taken to the Registration for Organizers page, through which you can create a new account and then you can log in smoothly and efficiently.





Seventh, when you log in as an organizer, it will take you to the organizer’s dashboard, which contains a Landing section and a button through which it will take you to add a new event by adding the event data to the card, as in the picture in front of you, and after adding the event data, there is an add event button inside the card, which will appear. All events are added by the organizer for attendees.

As you can see in the picture in front of you, there is a button above the card “All Events”, which when you click on it takes you to all the events that have been added in the form of cards, and through the two buttons on the cards.



The organizer can update the event information. The organizer can also delete the event, which is certainly any Interaction, through these processes, which will be modified and updated on the special event booking page for attendees.

1. **Testing Strategy**
   1. **Overview of the testing approach**

Testing Strategy: the testing strategy is considered to be the comprehensive plan that has to be made in order to see through the testing that will be performed for a software project. This strategy is essential because it outlines all the required testing in order to make sure that the application is functioning and performing as required for the functional, non-functional, user, and system requirements overall. For this strategy, there are three main kinds of testing that will be performed on the software project which are:

**Blackbox Testing:** The Blackbox testing will be performed on the XYZEvents system in order to see whether or not the main functionalities that are available to the user through the interface which is the web application are working correctly, and as required or not. This type of testing is important because the output or the result of a certain action would be tested in it. An example of that would be if we tried to test the login functionality in the application. If the user were to enter the correct user credentials, and the system was able to fetch their data from the database and be able to verify it, and then navigate the user to the homepage for example, that means that this type of Blackbox testing has succeeded for sure. In general, this type of testing will be performed on this system such that the output is the most important thing to be kept in mind.

**Whitebox Testing:** The Whitebox testing will be performed in the context of the XYZEvents system such that we would see whether or not there are errors or any kind of issues within the code. This is because the Whitebox testing only focuses on the internal issues of the application such as the code, and does not focus on the user side of the application. An example of that if in the form for the signup page, the **name** property for the password was “UserPassword”, and in the backend the developer tried to pass in the **$\_POST[“Password”]**, what will happen then is that the code will not work because the name was passed correctly to the backend, and therefore this test case will not be successful. In order for the Whitebox testing to not fail in this case, the name has to **Password** for both the form, and the backend when using it with the database. In general, this type of testing will be performed on this system such that the internal part of the application which is the code is the most important aspect for this testing.

User Acceptance Testing: The user acceptance testing is performed on the context of the XYZEvents system such that I would show the application or the software project to the end-user, and the end-user would provide me with their opinion, as well as some feedback which would make me determine whether or not the current software project is ready to be released or not, or if there are updates that has to be made for it too. Example for that is when I showed the application to my peers, and they provided me with feedback as well as their opinions on various things regarding the application which made think again on whether or not my application will be ready or not, and after tackling those points, I was able to achieve all the required points of the XYZEvents system.

* 1. **Test cases and scenarios**

Overview of the testing approach: The testing approach is considered to be an essential aspect or part when developing a new application, and therefore designing the Event Management System (EMS) seamlessly for XYZ Events is only possible with a carefully deliberated and comprehensive testing strategy. This goal is to achieve a flawless functioning of each part of the system from initial event registration up to the end event. My plan for achieving is as follows.

**Requirement Analysis and Test Planning**

Initially, I need to have a deep comprehension of what EMS should accomplish. In attempt to achieve this, I will evaluate every requirement and produce an elaborate test plan which describes our intentions, resources as well as timeframe. This stage prevents the omission of any critical aspects and lays ground for subsequent testing phases.

**Test Case Design**

Firstly, test cases are necessary so that every single feature as well as function in this application can be verified through some specific test cases. Another point is about coming up with test cases for user account creation, event management, attendee registration among others like creating event registration forms. While doing UI Testing one should pay much attention especially attending that it has been designed with user-friendly features including nav bars at page extremes which should be usable. This means that before validation takes place or during manual testing scenario-based approaches might also be used when necessary, during manual testing situations or they might even include more than that.

**Unit and Integration Testing**

When doing unit tests, I will try every component separately so that it does what is expected from it. This is meant to prevent problems before they start growing. Eventually, the integration ones are performed where different parts of the EMS are combined together seamlessly. They help me see whether there is any smooth interaction between database, business model and UI so that they work harmoniously.

**System and Usability Testing**

Making sure the whole EMS meets the specifications is what system testing is all about. Also, am going to test if aspects such as user accounts, event management or attendee registration are functioning all right. The purpose of usability testing is to ensure that this EMS is easy-to-use. Consequently, some users will be selected randomly so that their opinion can be sought regarding navigation bar, footer and what they think about it in general, then those areas will be worked on according to the input obtained.

**Performance and Security Testing**

Performance testing will evaluate how well the system works under different circumstances. I am going to carry out load testing to determine how it behaves during its period of highest use. Security testing will ascertain that EMS is protected from various forms of risks. This entails vulnerability scanning, penetration testing, and examining data encryption as well as access controls so that user data remains safe.

**Acceptance and Regression Testing**

We will involve the stakeholders in the UAT so that they can be sure that the EMS will meet all user needs while intending to be dispensed. I will check to see if this system can give exactly what is required by customers and if it can make them happy too. This exercise will be performed in order to ascertain that old functions are not lost just because there are new ones hence ruining the general performance of the system

**Deployment and Post-deployment Testing**

In the production context for the web application when it is released to the users, I will eventually test EMS. After deployment, smoke testing will be done to verify the system's critical functionality. This will make it easy to correct failures immediately when they occur through live and continuous monitoring; henceforth, having continuous and smooth system performance.

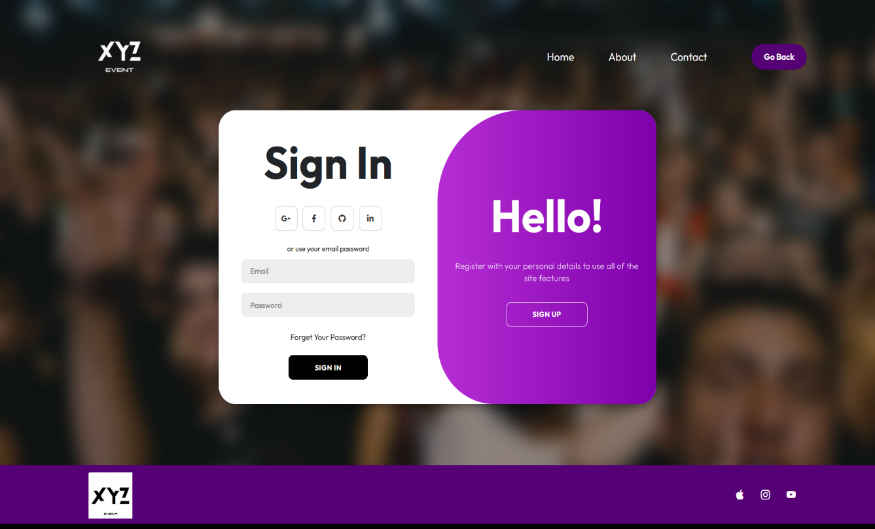
**Conclusion**

Through the use of such a testing methodology that is structured, I hope to give a very strong event management system for XYZ Events that is user-friendly and reliable. Every stage within our software test plan focuses on finding errors sooner rather than letting them build up among others so that we deliver great quality software. These testing processes are therefore aimed at assisting XYZ Events to have better event organization capabilities while increasing on productivity which epitomizes excellent customer service thereby leading to revolutionizing this industry in terms of coordination.

Test cases and scenarios: For this section, I will be testing different parts of the application and providing a description for it.

Testing Register Functionality for Attendees: I will be testing the register feature for the attendees when they want to create a new account in the system.

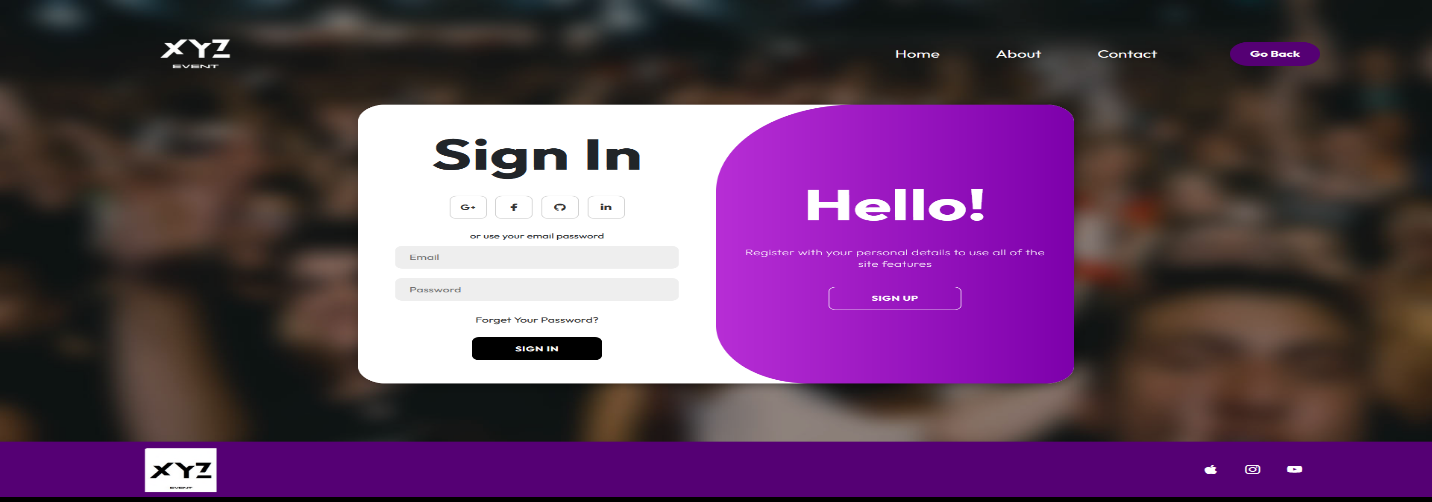
A new user would like to register as an attendee in the XYZEvents system, they navigate to the register page, and then they enter their details which are their name, email, password, and phone number and then they click on the register button which is supposed to send this data to the backend which the backend will execute and the result will show on the database based on the query written.

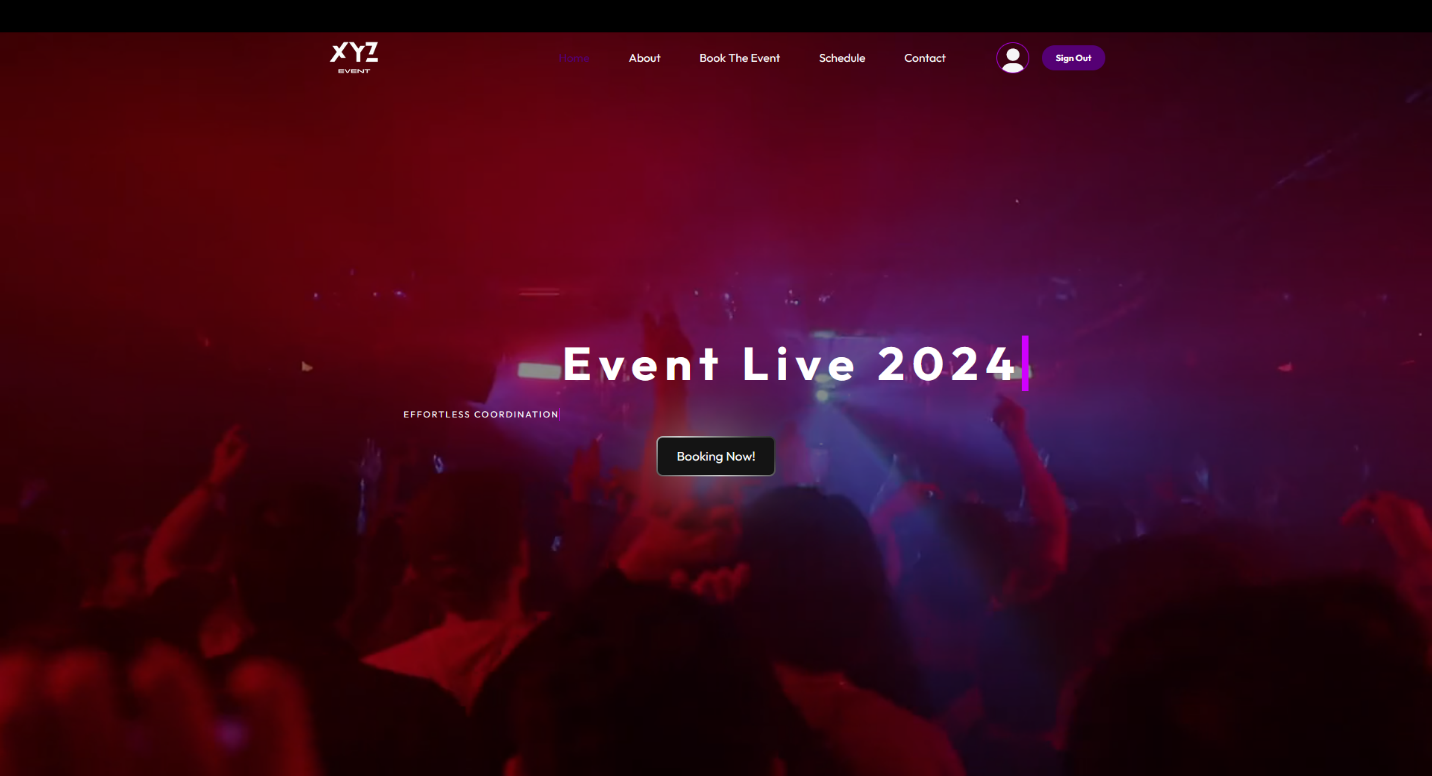


Testing Login Functionality for Both Attendees and Organizers: I will be testing the login functionality for both the organizer.

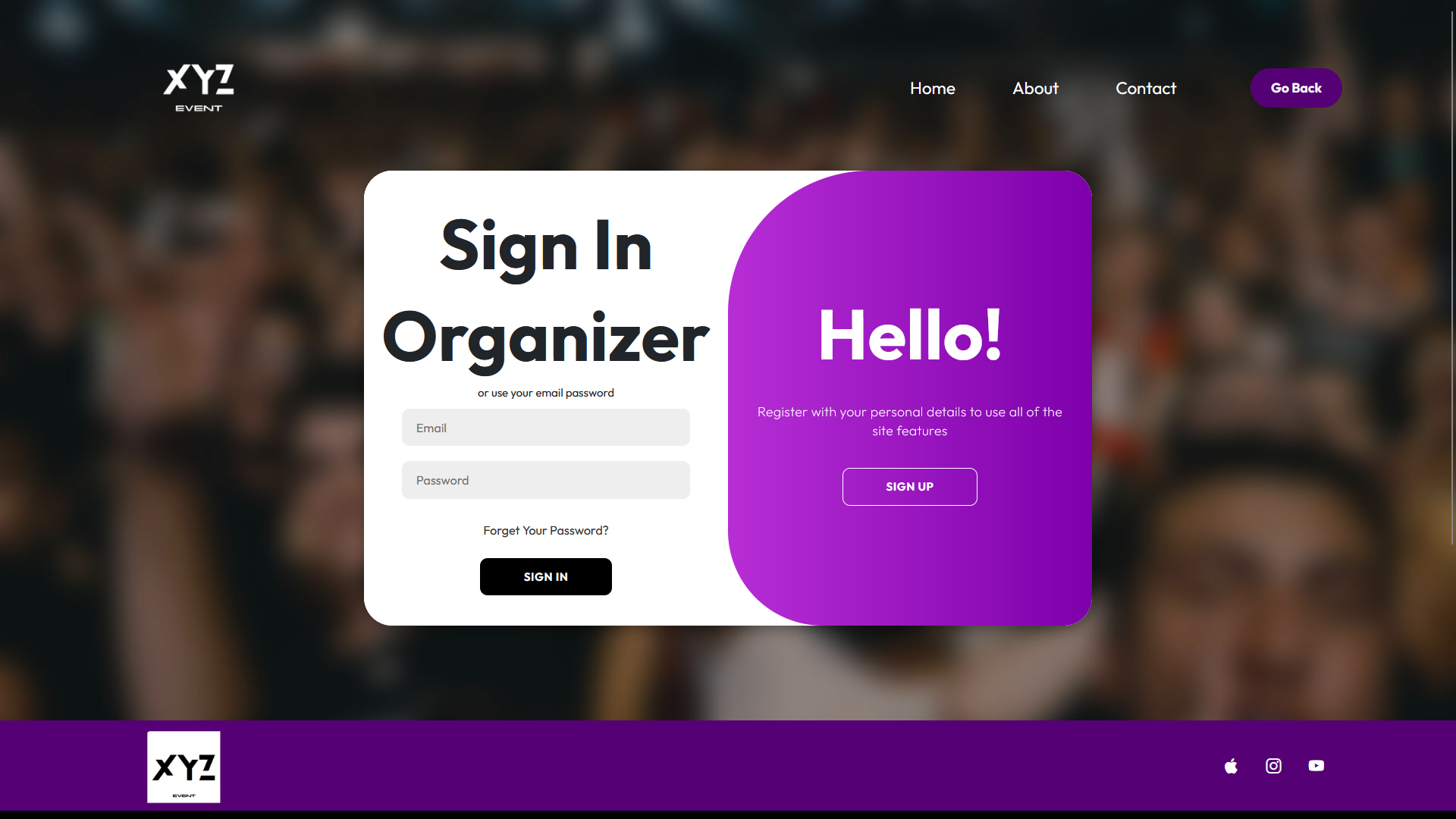
Suppose we have a user, either attendee or organizer that registered a new account, or they already have an account in the XYZEvents system, and they would like to login to the system and navigate to the homepage. The process would be simple, they would need to input their email as well as their password that are associated to them in the database, and then the system will check whether those login credentials are correct or not, if it is then the user is navigated to the homepage and into the system. Otherwise, the user will stay in the same page until they enter the correct data regarding them. It is important to mention that the organizer should go to the organizer login page, and the attendee would go to the attendee login page in order for the process to go successfully. So, the attendee would naturally go to the homepage for the attendees, and the organizer would go to his dashboard once they input the correct credentials.

**For Attendee:**



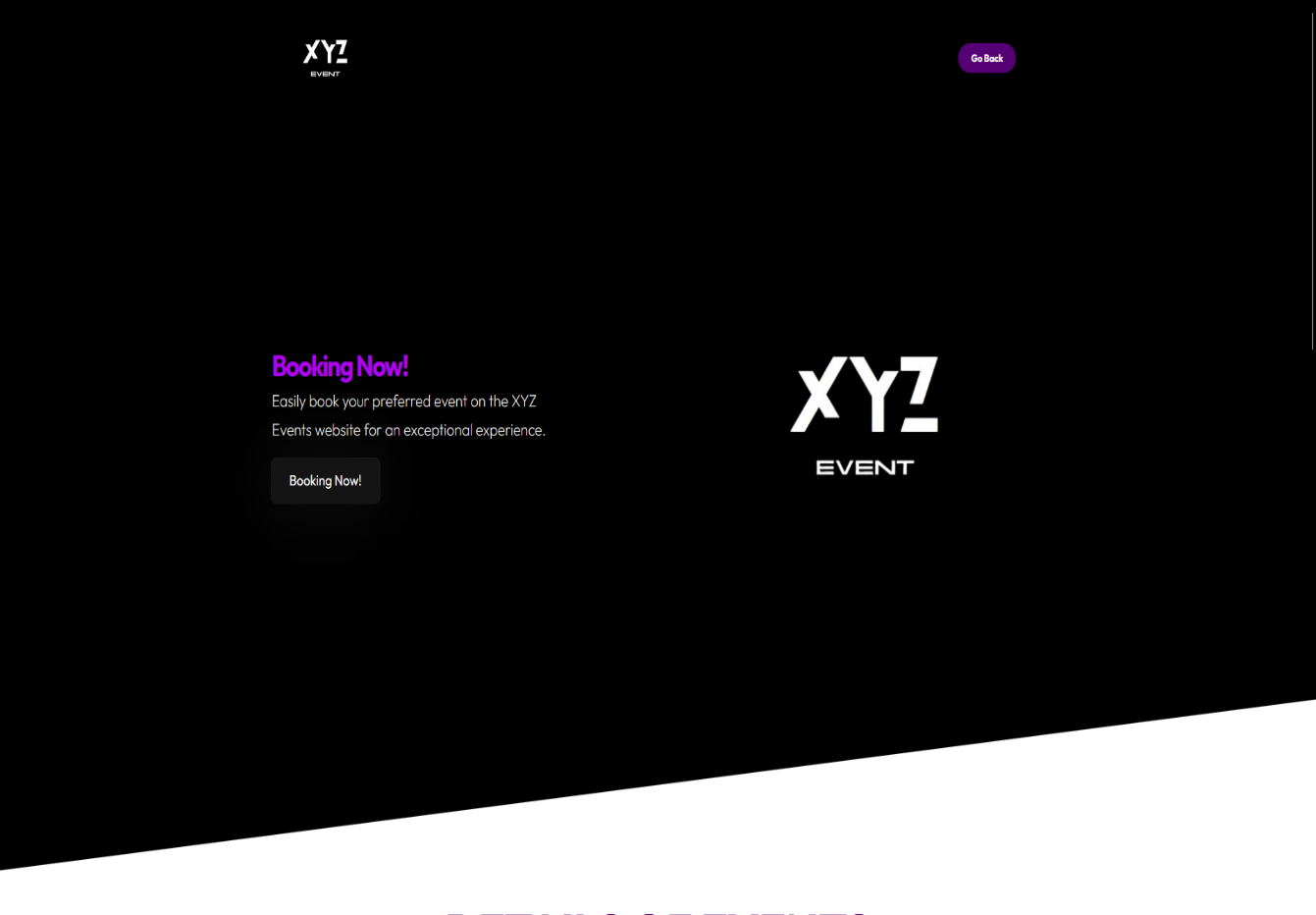


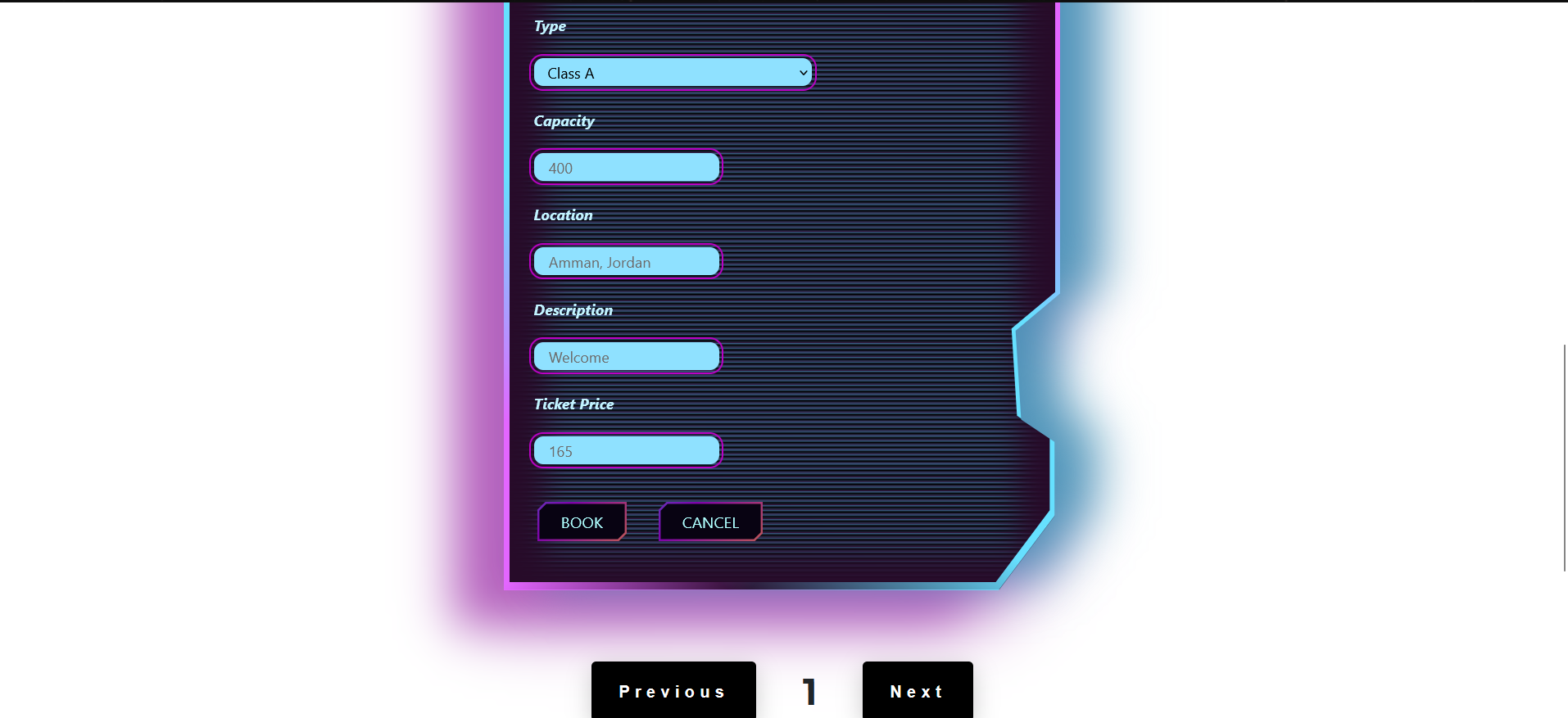
**For Organizer:**



**Testing the Booking feature for the Attendee:**

Suppose the attendee has entered their login credentials correctly, navigated to the homepage, and now wants to view all the available events and then book them. So, the attendee would click on the **booking** in the navigation bar and it would get them to the booking page where all the available events in the database are being read and displayed, and also, they are able to book each one of the events just by click on the **book** button for the event they are interested in booking. Once the book button is pressed, it will take the attendee ID from the session being held from the login, and the event ID for the event that was booked and add them both into the **Attendee\_Booking** table in the database. Then, the user will be registered for this event in the system.



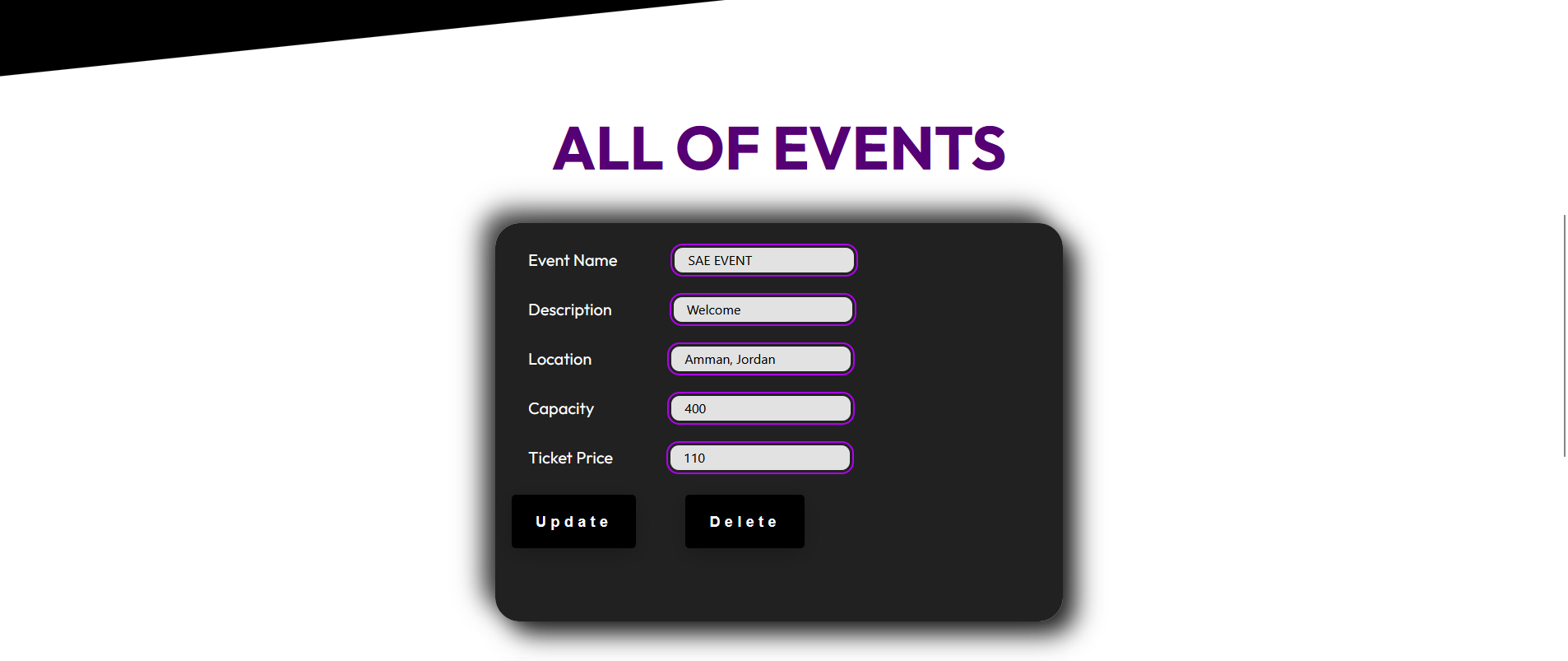


**Testing the Cancel feature for Attendees:**

Suppose that the attendee has already booked an event in the **booking** page by clicking on the **book** button for that event, and now they would like to cancel that event from the booking. In that scenario, the attendee would click on the **cancel** button which would delete the attendee ID and the event ID from the **Attendee\_Booking** table, and now the user has successfully cancelled the event.

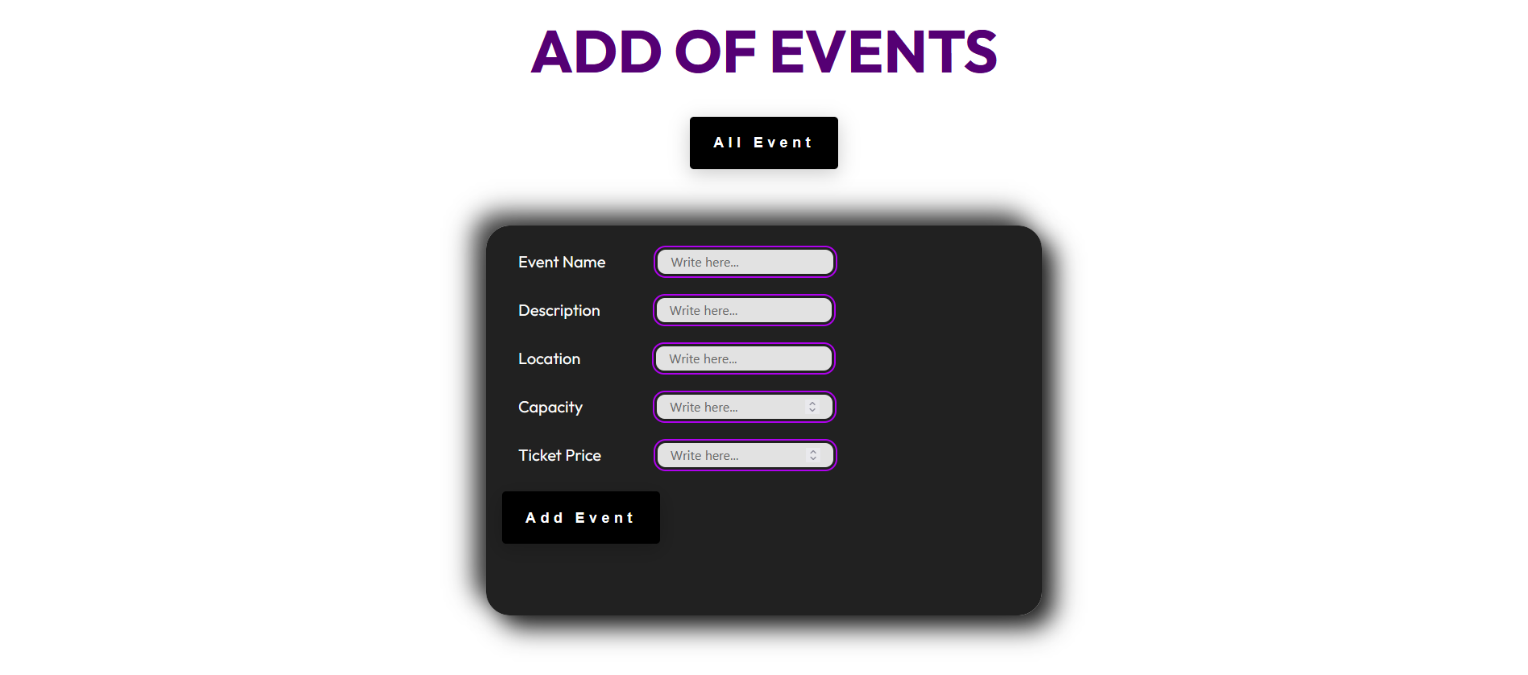
**Testing Viewing all events for Attendee & Organizers:**

Suppose either one of attendees or organizers navigated to the homepage and they would like to see the events page, as soon as they enter the events page, the XYZ Events system will read all the event data that exist in the database to both the attendee and organizer in the same manner. After viewing all the available events, the attendees will be able to book the events they see and also cancel them. And the organizers will be able to update and delete the events as required.



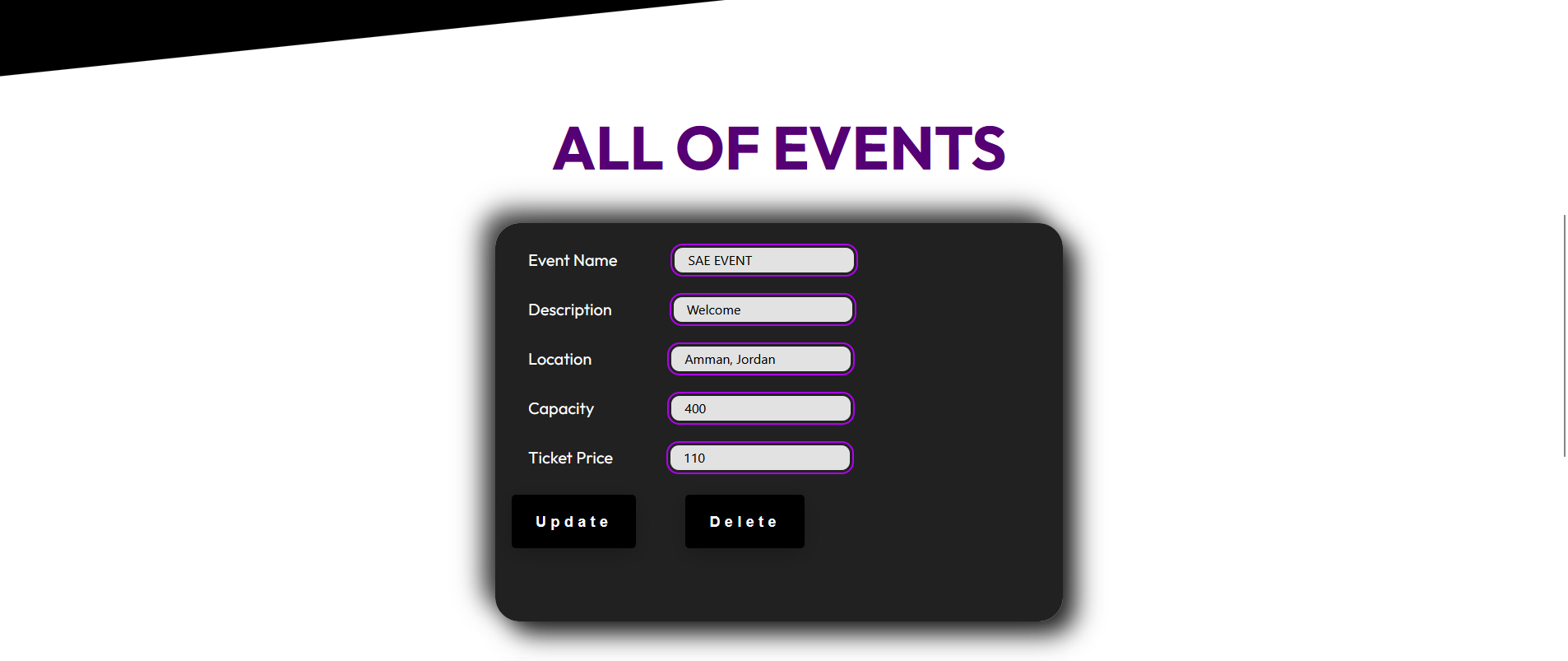
**Testing Adding a New Event for Organizers:**

Suppose the organizer has navigated to the add event page, and they would like to add a new event based on what is required for the system. So, the organizer is able to enter all necessary details for the event and then press on the add button which would insert the current event to the database and it would show a new event in the all events page where all the events are being displayed.



**Testing Updating an Event for Organizers:**

Suppose the organizer has navigated to the all events page, and they would like to update event based on what is required for the system. So, the organizer is able to enter details they would like to update for the event and then press on the update button which would update the current event to the database and it would show changes in the all events page where all the events are being displayed.



**Testing Deleting an Event for Organizers:**

Suppose the organizer has navigated to the all events page, and they would like to delete event based on what is required for the system. So, the organizer is able to delete event they would like to delete for the event and then press on the delete button which would remove the current event to the database and it would show all the events minus the one that was deleted in the whole events page where all the events are being displayed.