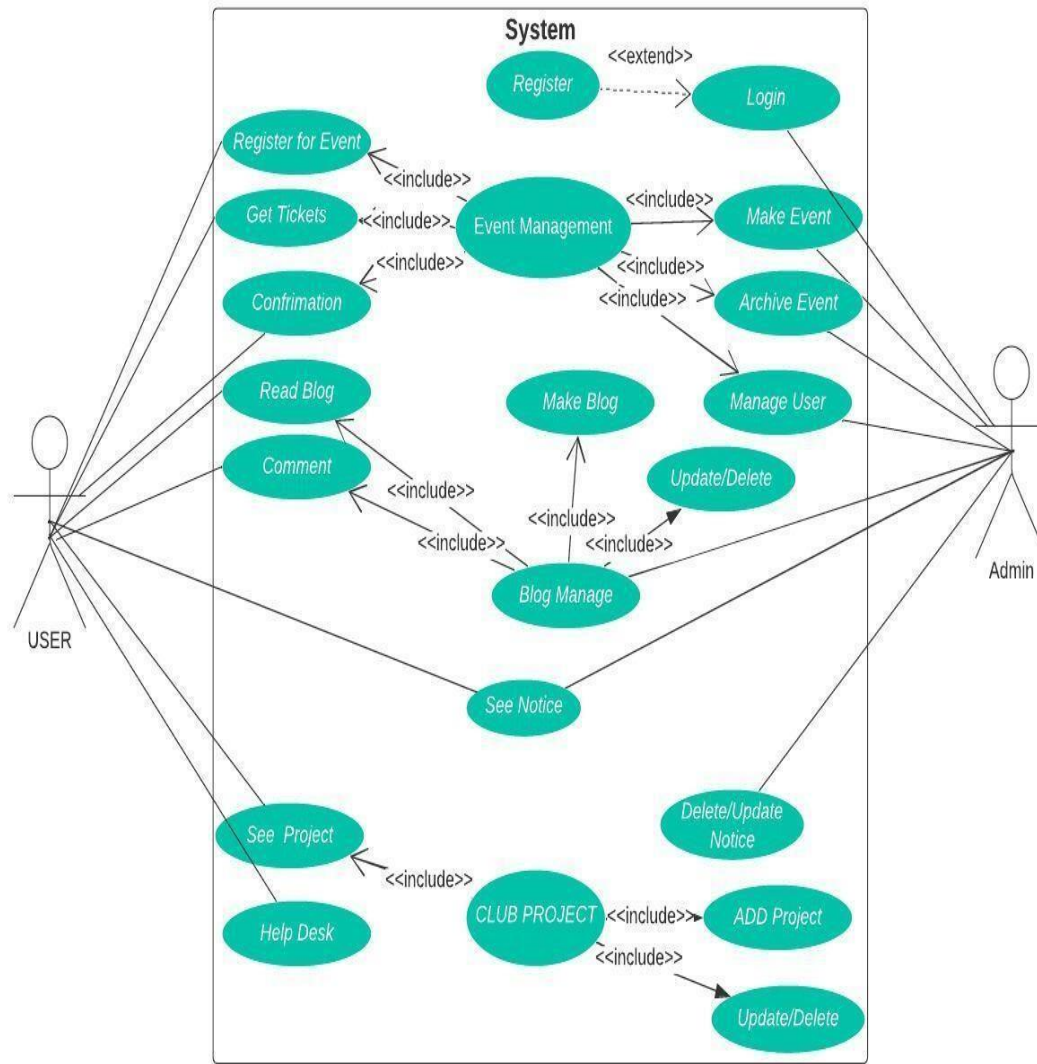


Task 1:

Use-case diagram:

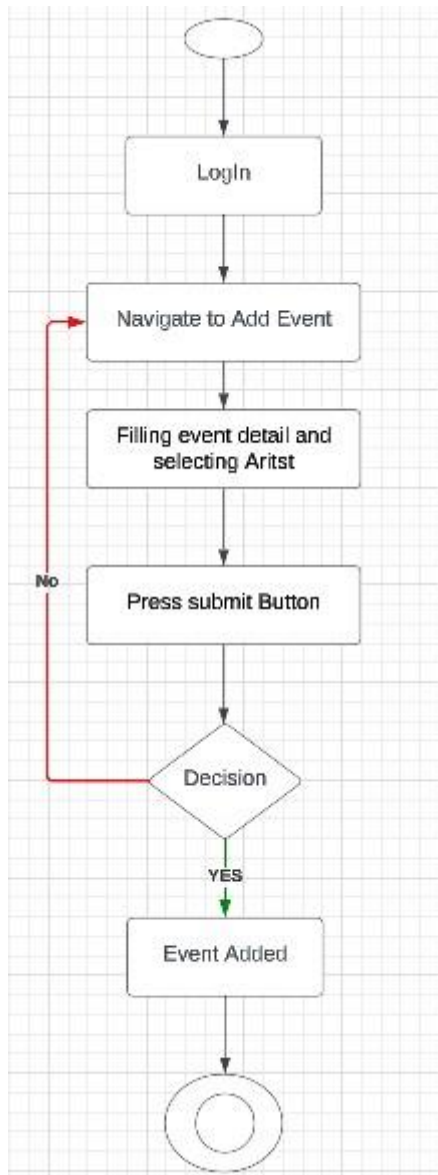


Use case diagram represent all the actors whose are related to the system and then show them how they can participate to the system. Here is the figure which will show the Use case diagram of the club management system with relations among tasks.

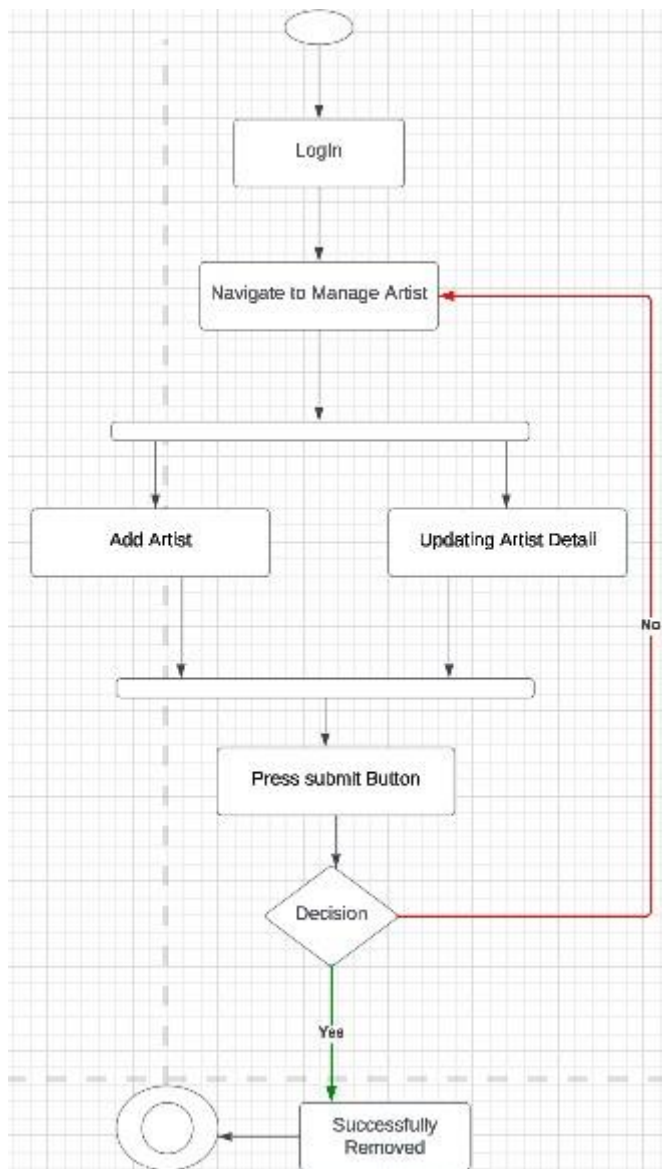
Activity diagrams:

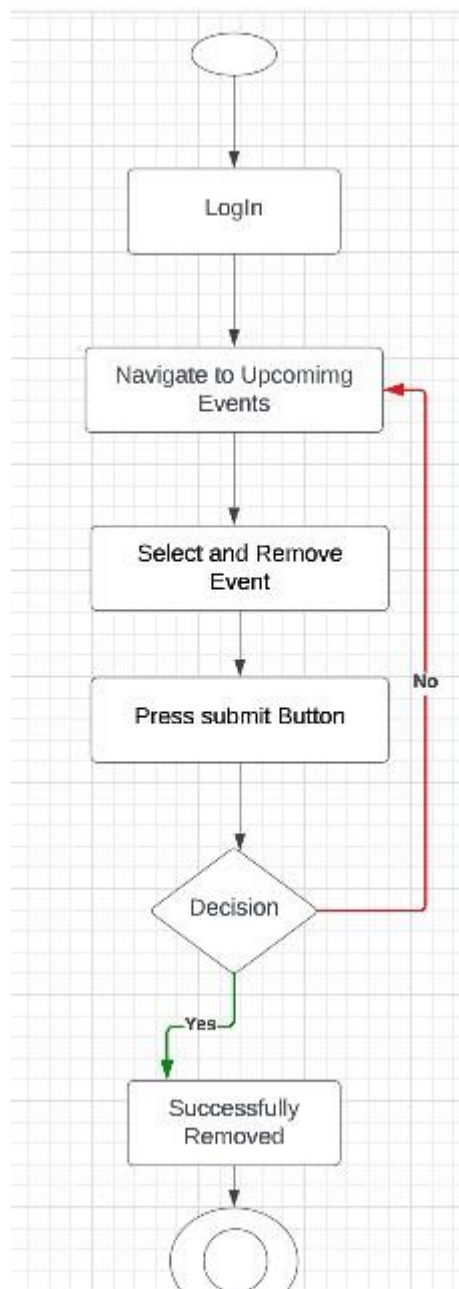
I can provide a breakdown of the steps in the activity diagram:

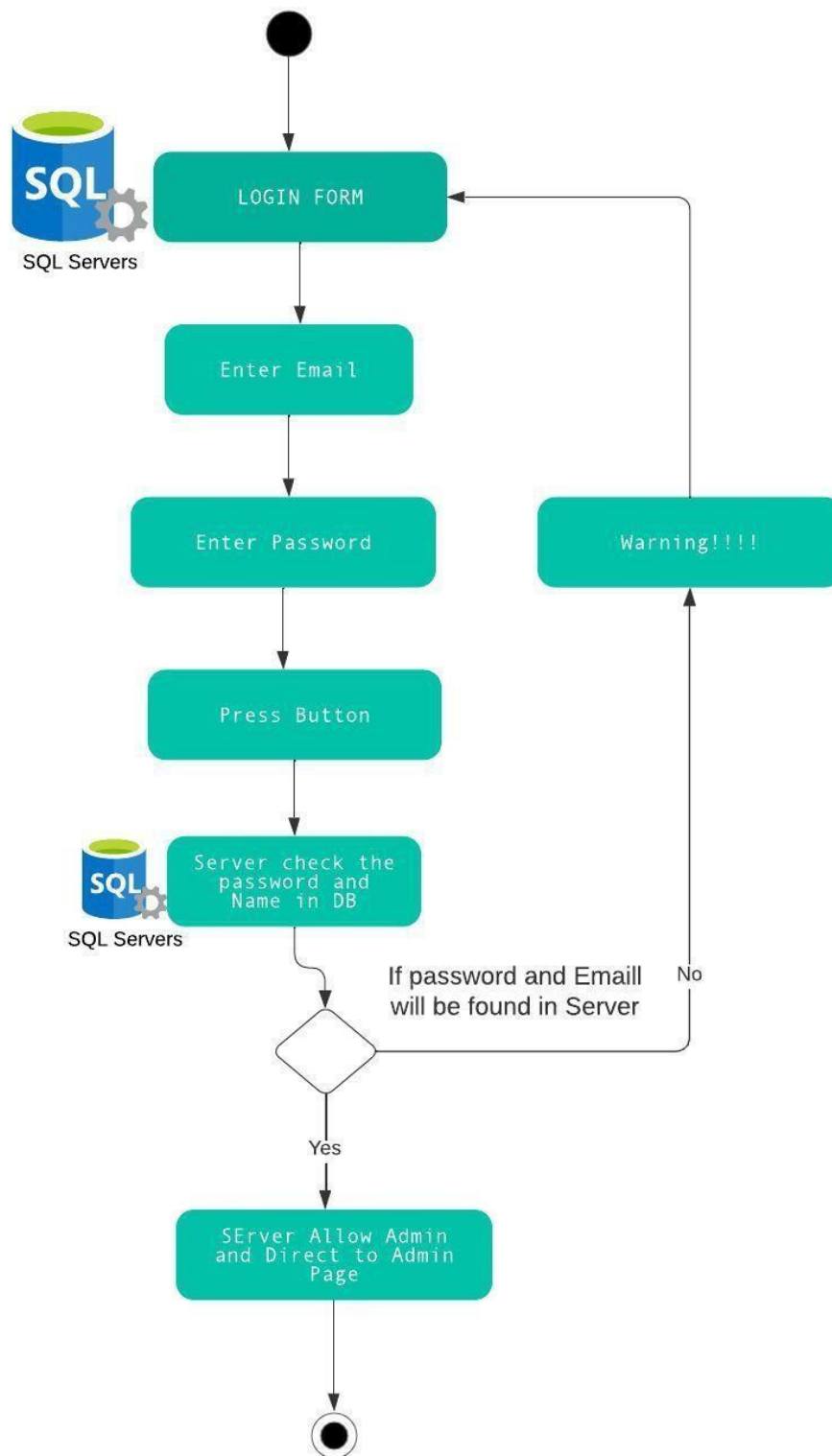
- Main Menu: The web app's primary navigation bar displays a list of choices, such as "Upcoming Events, recent Events, and Admin Pages, to the user.
- View Upcoming Events: The user may look at a schedule of events happening over the next six months and filter them based on date, category, or performer.
- View Past Events: The user can view a list of past events that have been held at the venue.
- Admin Pages: The staff member can access the admin pages by logging in with their credentials.
- Add or Remove Events: Date, start/end time, category, admission cost, attendance capacity, and performer information can all be added or removed from the database by the staff person.
- Add or Remove Artists: Employees have the ability to add and delete musicians from the database, which includes the artist's name, a brief biography, social network handles, and a promotional photograph.
- In general, the procedures entail using the system's interface to see and manage events and artists.



•







Sitemap:

I can provide a site map for the scenario you described. Here's a possible structure for the site map:

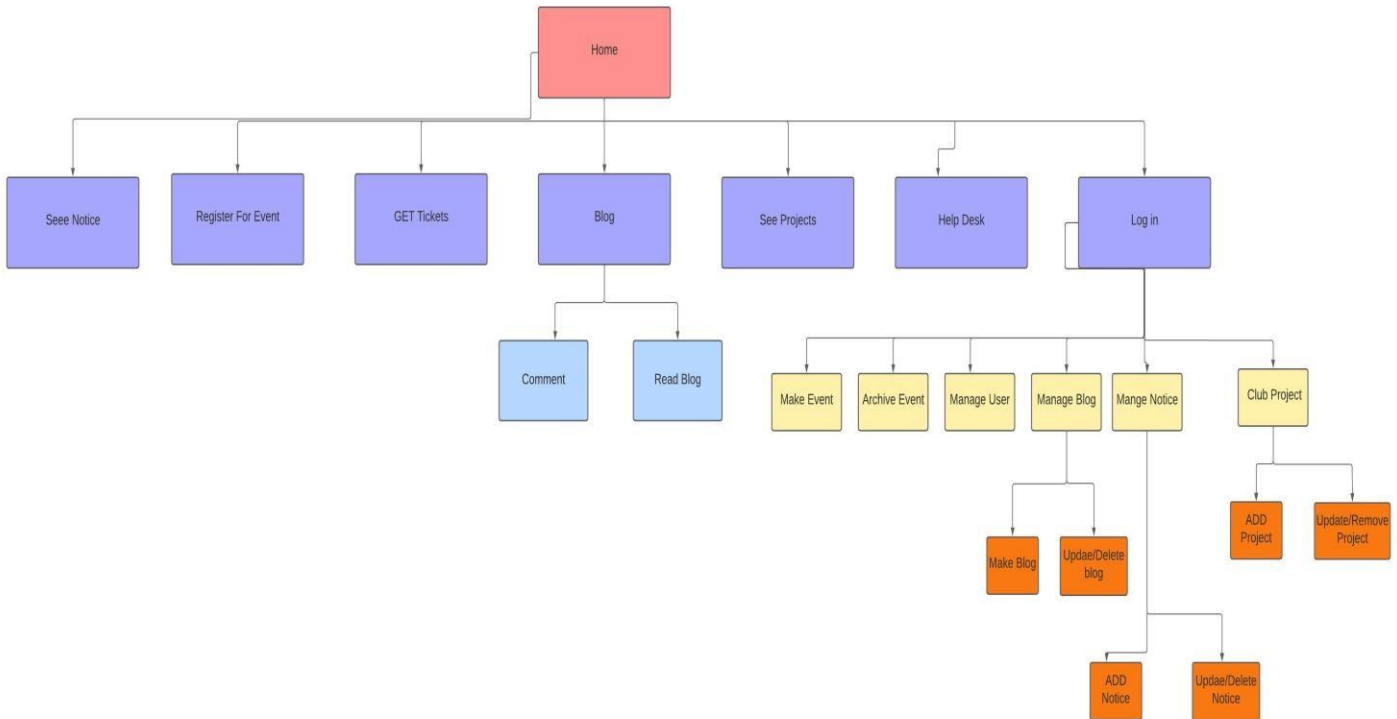
The Home page serves as the main landing page for the web app, providing access to the upcoming events, past events, and artist directory sections.

The Upcoming Events section is divided into three sub-sections for music, poetry, and comedy events. Each sub-section lists the relevant upcoming events, with links to view more details about each event.

The Past Events section provides access to a list of past events that have been held at the venue.

The Artist Directory section allows users to browse a list of all the artists that have performed at the venue, with links to view more information about each artist.

The Admin Pages section is only accessible to authorized staff members who have logged in. This section provides access to the login page, as well as the event and artist management pages. The Manage Events and Manage Artists pages allow staff members to add, edit, or delete events or artists from the database.

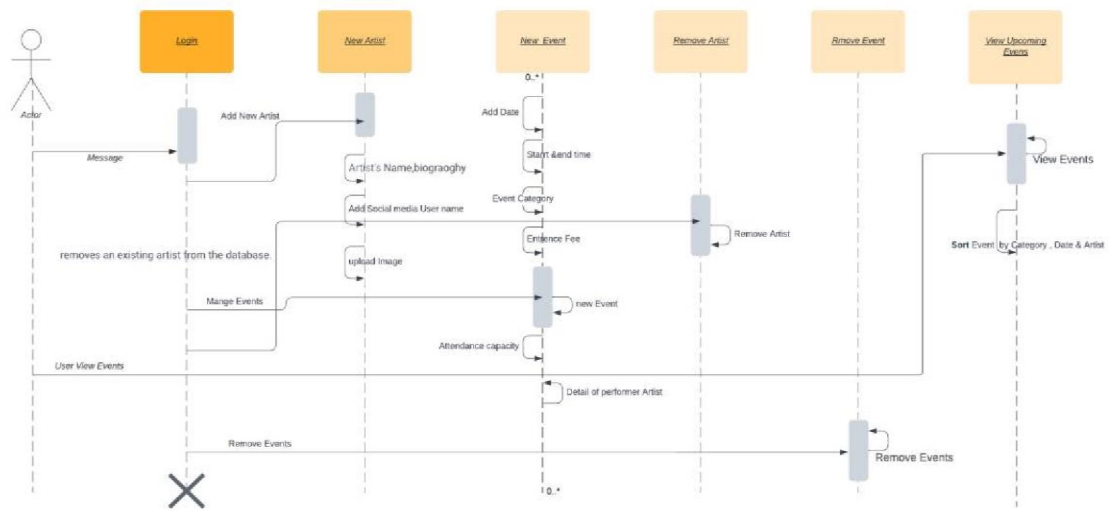


Sequence Diagram:

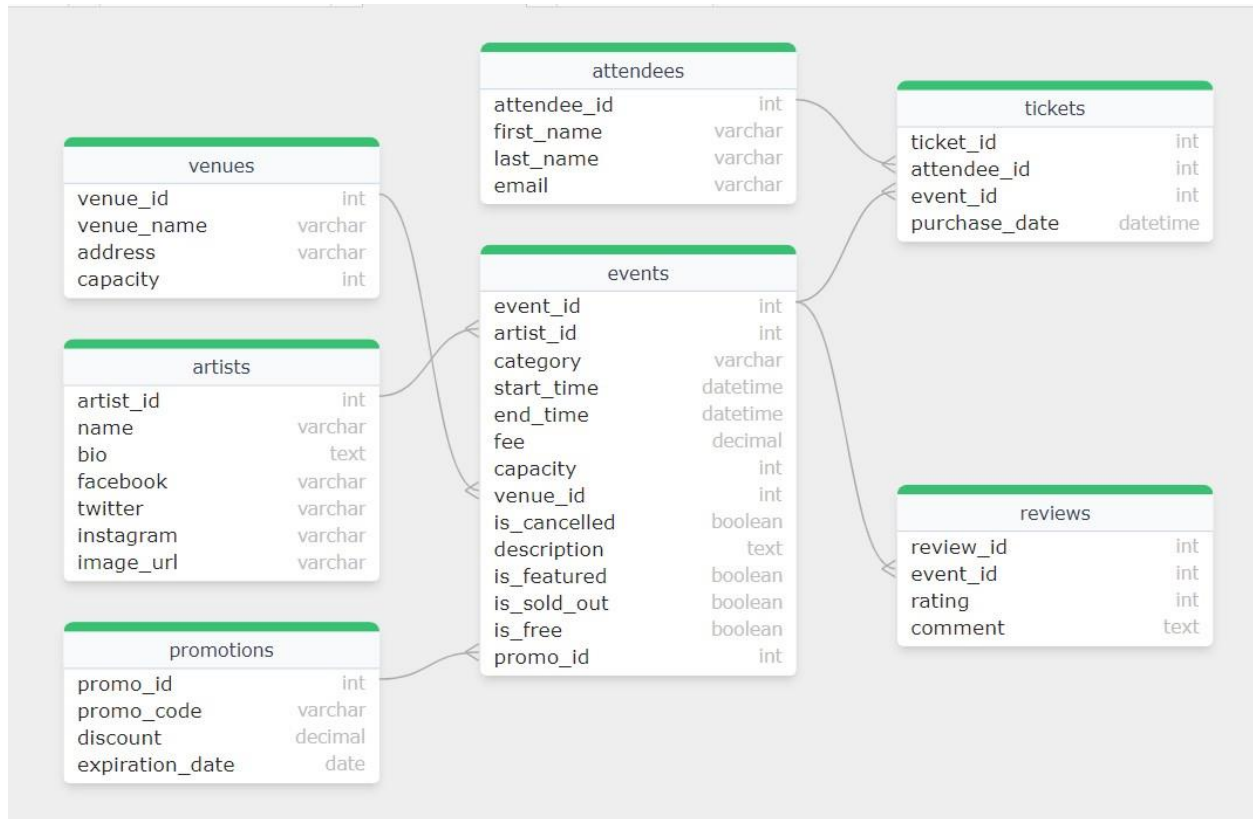
This sequence diagram depicts the fundamental interactions between a nightclub's event management system's user interface, database, and backend server. It shows how a user may sign in and do actions like adding, editing, removing and adding artists and removing events by simulating the exchange of messages between the various parts of the system. Creating a sequence diagram can help you see how different parts of the system interact with one another and where problems could arise.

Sequence diagram

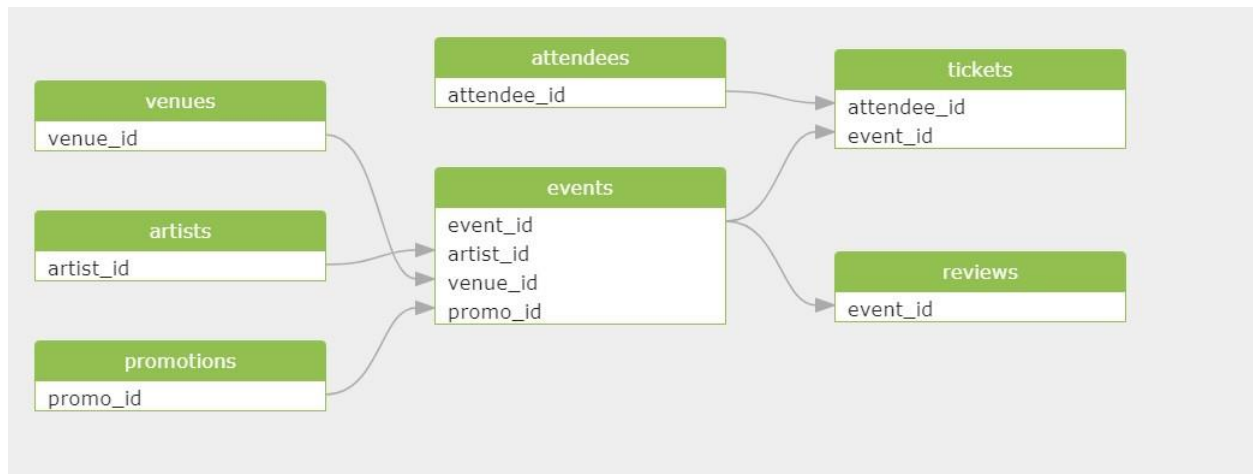
Date : March 16, 2022



Entity-Relationship Diagram In-Tool:



The SQL Flow:



Database:

```
CREATE TABLE artists ( artist_id INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255)  
NOT NULL,  
bio TEXT,  
facebook VARCHAR(255),  
twitter VARCHAR(255),  
instagram VARCHAR(255),  
image_url VARCHAR(255)  
);
```

```
CREATE TABLE events (event_id INT NOT NULL
AUTO_INCREMENT PRIMARY KEY, artist_id INT NOT
NULL, category VARCHAR(255) NOT NULL, start_time
DATETIME NOT NULL, end_time DATETIME NOT NULL,
fee DECIMAL(10,2) NOT NULL, capacity INT NOT NULL,
venue_id INT NOT NULL, is_cancelled BOOLEAN NOT
NULL DEFAULT FALSE,
description TEXT,
is_featured BOOLEAN NOT NULL DEFAULT FALSE,
is_sold_out BOOLEAN NOT NULL DEFAULT FALSE, is_free
BOOLEAN NOT NULL DEFAULT FALSE,
promo_id INT,
FOREIGN KEY (artist_id) REFERENCES artists (artist_id),
```

```
FOREIGN KEY (venue_id) REFERENCES venues (venue_id),
FOREIGN KEY (promo_id) REFERENCES promotions (promo_id)
);
```

```
CREATE TABLE venues (venue_id INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, venue_name  
VARCHAR(255) NOT NULL, address VARCHAR(255) NOT  
NULL, capacity INT NOT NULL  
);
```

```
CREATE TABLE attendees (attendee_id INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(255)
```

```
NOT NULL, last_name VARCHAR(255) NOT NULL, email  
VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE tickets (ticket_id INT NOT NULL
AUTO_INCREMENT PRIMARY KEY, attendee_id INT NOT
NULL, event_id INT NOT NULL, purchase date
DATETIME NOT NULL,
FOREIGN KEY (attendee_id) REFERENCES attendees (attendee_id), FOREIGN KEY (event_id)
REFERENCES events (event_id)
);
```

```
CREATE TABLE reviews (review INT NOT NULL
AUTO_INCREMENT PRIMARY KEY,
event_id INT NOT NULL,
rating INT NOT NULL,
comment TEXT,
FOREIGN KEY (event_id) REFERENCES events (event_id)
);
```

```
CREATE TABLE promotions ( promo_id INT NOT NULL  
AUTO_INCREMENT PRIMARY KEY, promo_code  
VARCHAR(255) NOT NULL, discount DECIMAL(10,2) NOT  
NULL, expiration_date DATE NOT NULL  
);
```

```
ALTER TABLE events ADD CONSTRAINT fk_events_artists FOREIGN KEY (artist_id) REFERENCES artists  
(artist_id);
```

```
ALTER TABLE events ADD CONSTRAINT fk_events_venues FOREIGN KEY (venue_id) REFERENCES venues  
(venue_id);
```

```
ALTER TABLE events ADD CONSTRAINT fk_events_promotions FOREIGN KEY (promo_id) REFERENCES  
promotions (promo_id);
```

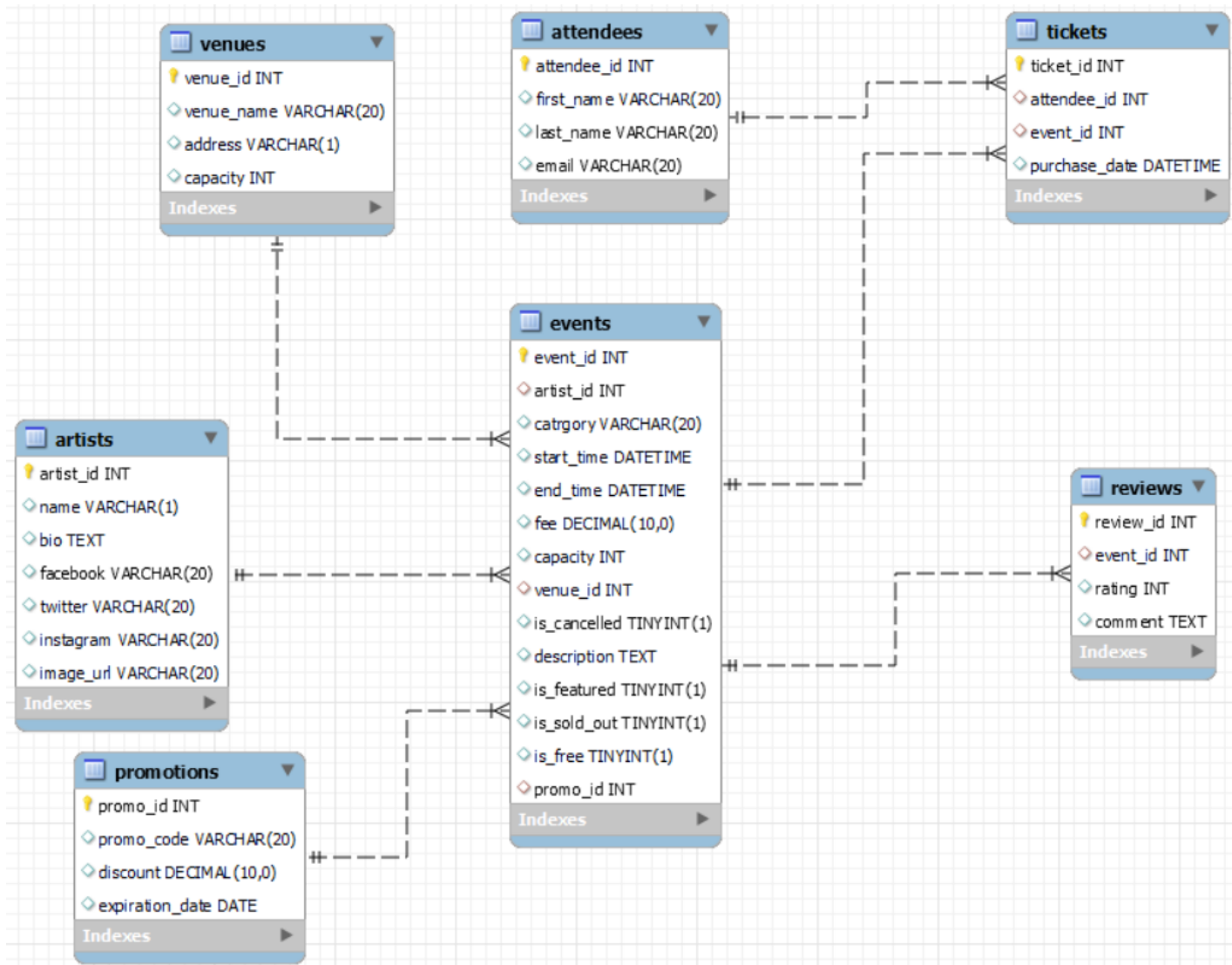
Created Tables(venues,artists,promotions,attendees,tickets,events and reviews) along with defined Keys in MYSQL-WorkBench for Erd-Fig also written manually in word with attached ss


```

4
5
6
7 ● create database softeng;
8 ● use softeng;
9 ● create table venues(venue_id int primary key, venue_name varchar(20), address varchar(1), capacity int);
10 ● create table artists(artist_id int primary key, name varchar(1), bio text, facebook varchar(20), twitter varchar(20), instagram varchar(20), image_url varchar(20));
11 ● create table promotions(promo_id int primary key, promo_code varchar(20), discount decimal, expiration_date date);
12 ● create table attendees(attendee_id int primary key, first_name varchar(20), last_name varchar(20), email varchar(20));
13 ● create table tickets(ticket_id int primary key, attendee_id int, event_id int, purchase_date datetime, foreign key (attendee_id) references attendees(attendee_id));
14 ● create table 'events'(event_id int primary key, artist_id int, catrgory varchar(20), start_time datetime, end_time datetime, fee decimal, capacity int, venue_id int
15   , is_cancelled boolean, 'description' text, is_featured boolean, is_sold_out boolean, is_free boolean, promo_id int, FOREIGN KEY (Artist_id) references artists(artist_id)
16   , FOREIGN KEY (venue_id) references venues(venue_id), foreign key (promo_id) references promotions(promo_id));
17 ● create table reviews(review_id int primary key, event_id int, rating int, 'comment' text, foreign key (event_id) references 'events'(event_id))

```

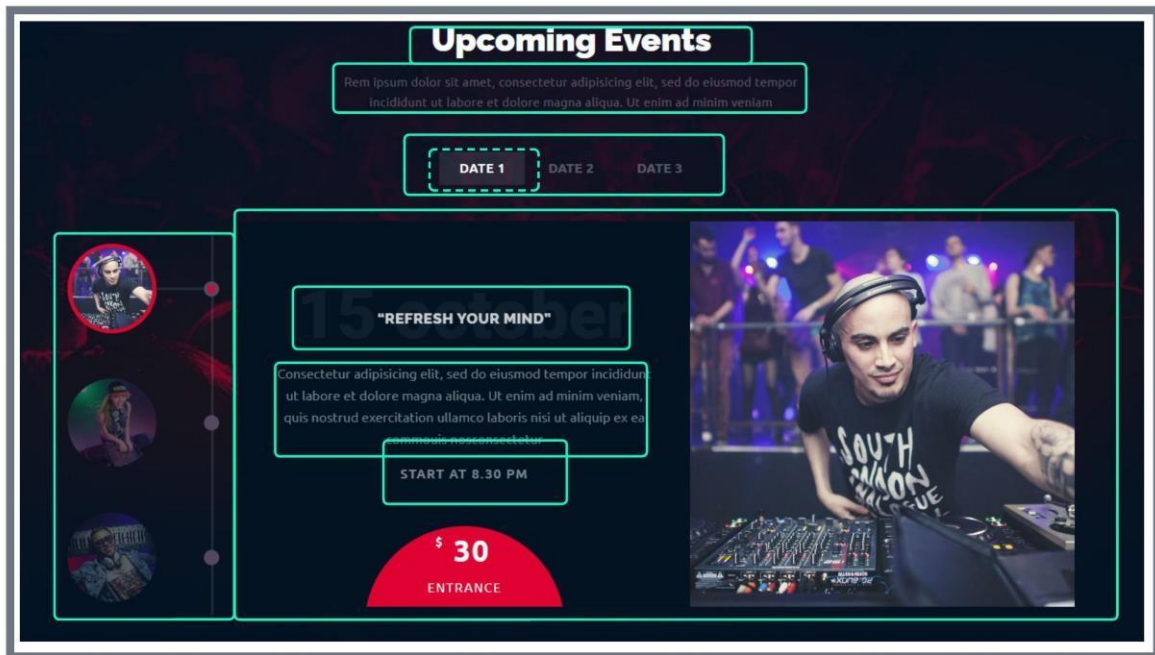
ERD-FIG in MYSQL WORK-BENCH:



by asking my module leader I created high, low and medium Fidelity Designs but more focused on med and low Fidelity as asked in docx

DESIGN:

Desktop High Fidelity Design:



Our Services

Rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam



CORPORATE EVENTS

Lor sit amet, ctetur adipisicing elit, sed deiusmod tempor incididunt ut labore et



BRANDED EVENTS

Dolor sit amet, tetur adipisicing elit, sed do eiu smod temporin cididunt ut labore



COMMERCIAL SHOOTS

Dolor sit amet, tetur adipisicing elit, sed do eiu smod temporin cididunt ut labore



HAN PARTIES

Psum dolor sit amet, ctetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et



BIRTHDAY PARTIES

Olor sit tetur adicing elit, sed do eiusmod tempor incididunt ut labore et



VIP SERVICE

Isum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut

WE CAN DANCE FOREVER

Rem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod idunt ut
labore et dolore magna aliqua. Ut enim ad minim inim veni

MAKE AN ORDER



Add Event

Rem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

Event

Date

Start Time

End Time

Entrance Fee

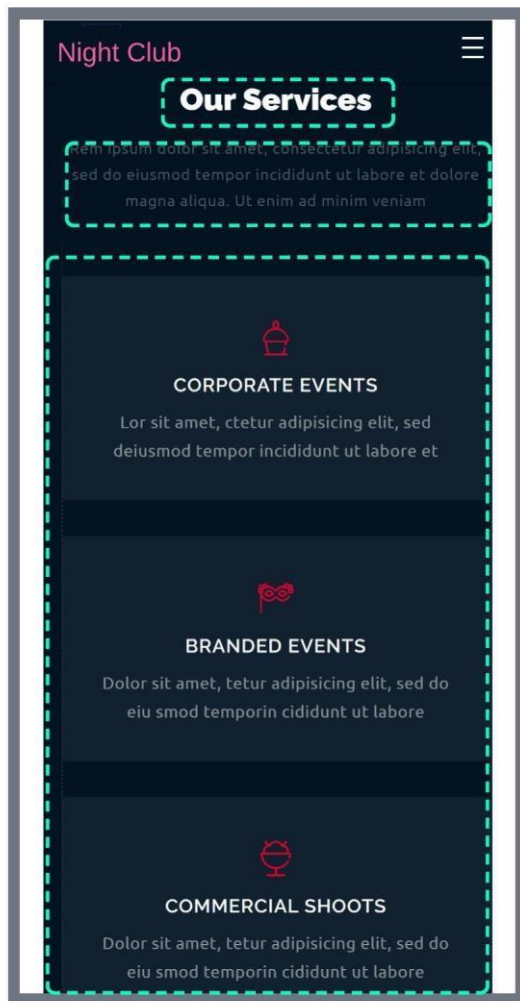
Capacity

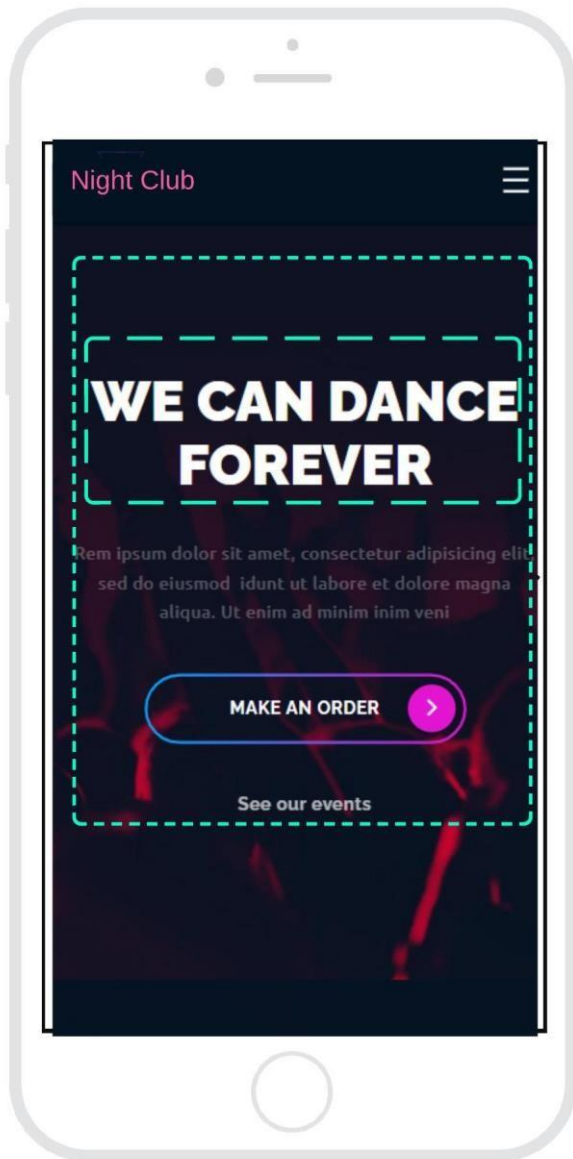
About Artist

Submit



Mobile High Fidelity Design:







Upcoming Events

Rem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

DATE 1

DATE 2

DATE 3

15 october

"REFRESH YOUR MIND"

Consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

START AT 8.30 PM

\$ 30

ENTRANCE

Night Club



Add Event

Rem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

Event

Date

Start Time

End Time

Entrance Fee

Capacity

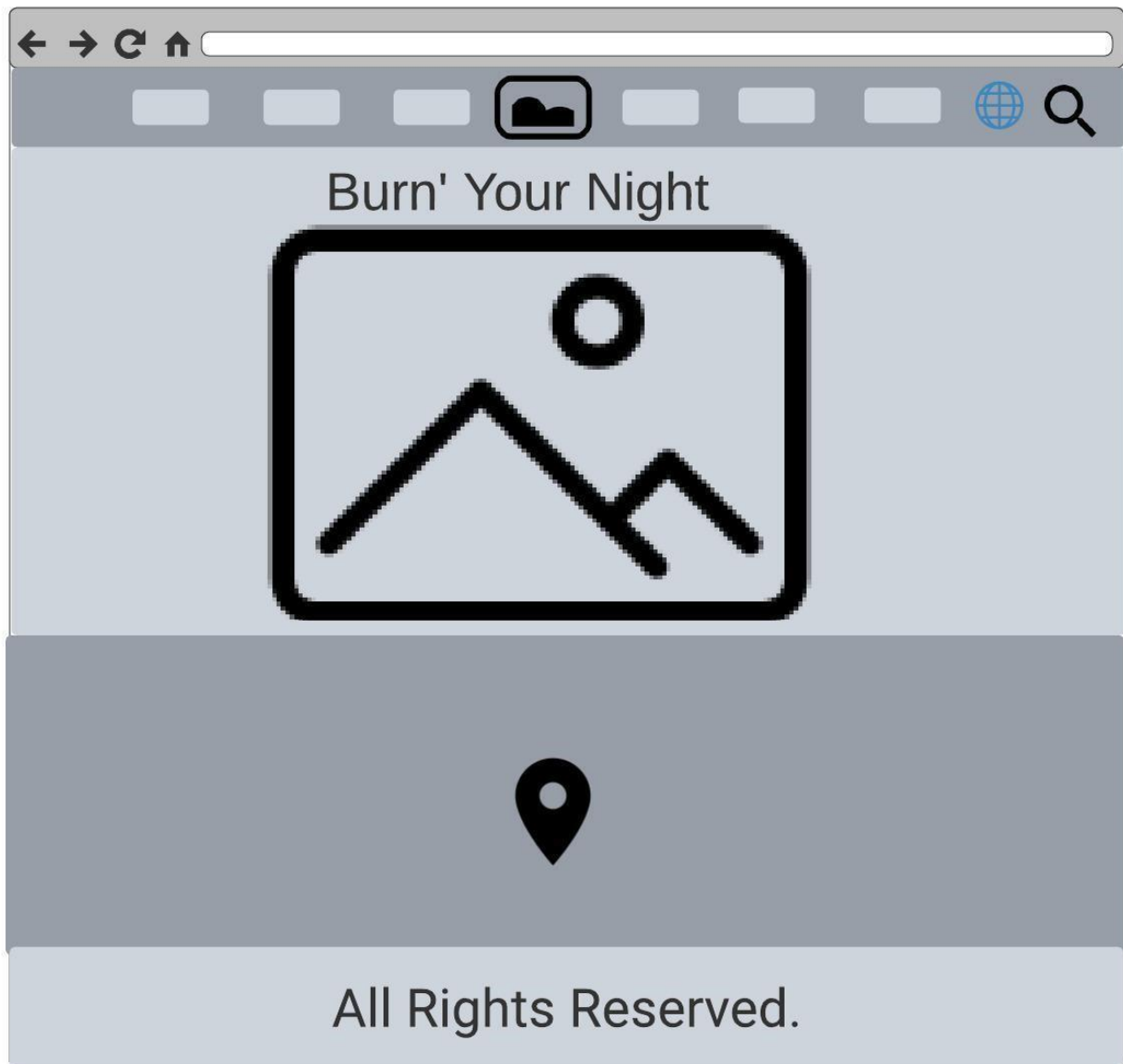
About Artist

Submit



Desktop Low Fidelity Design:

- **MASTER PAGE**



- LOGIN PAGE

← → ↻ ⬆

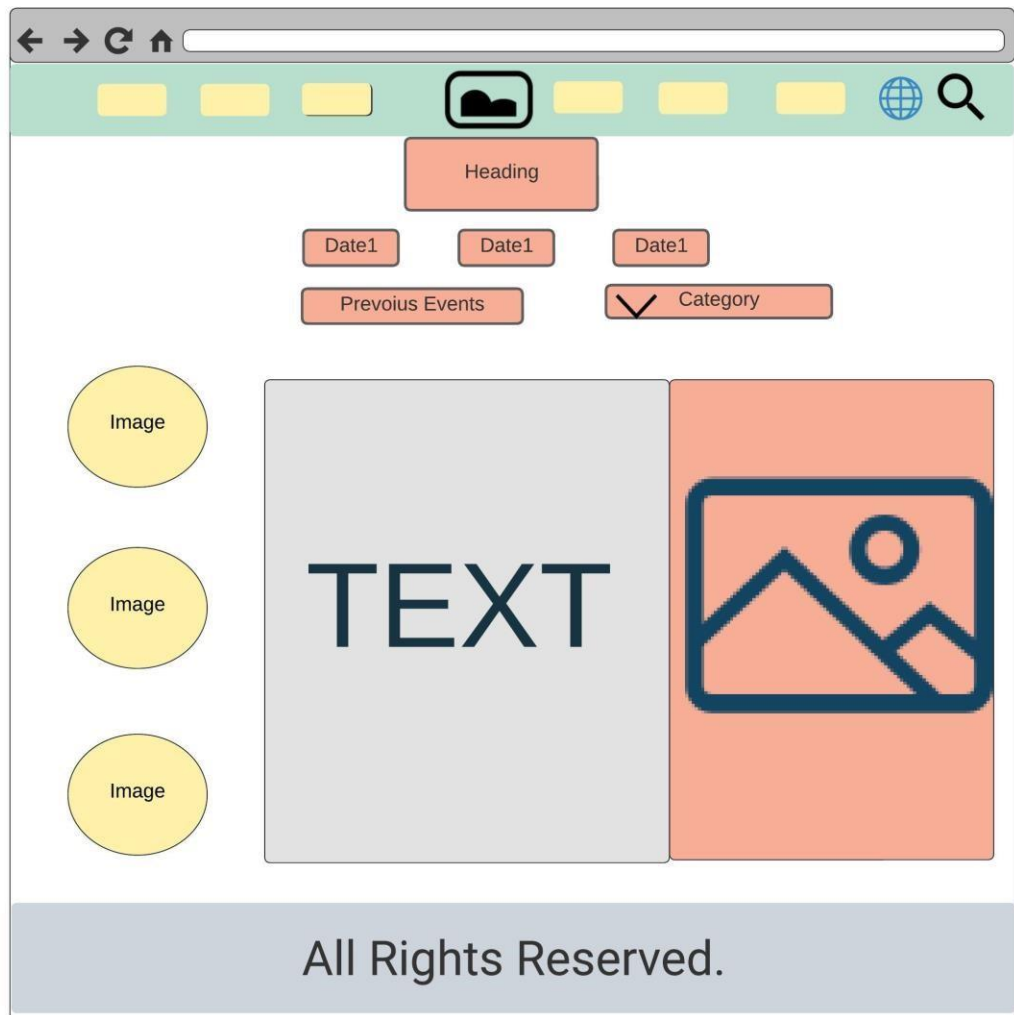
LOGIN

Name

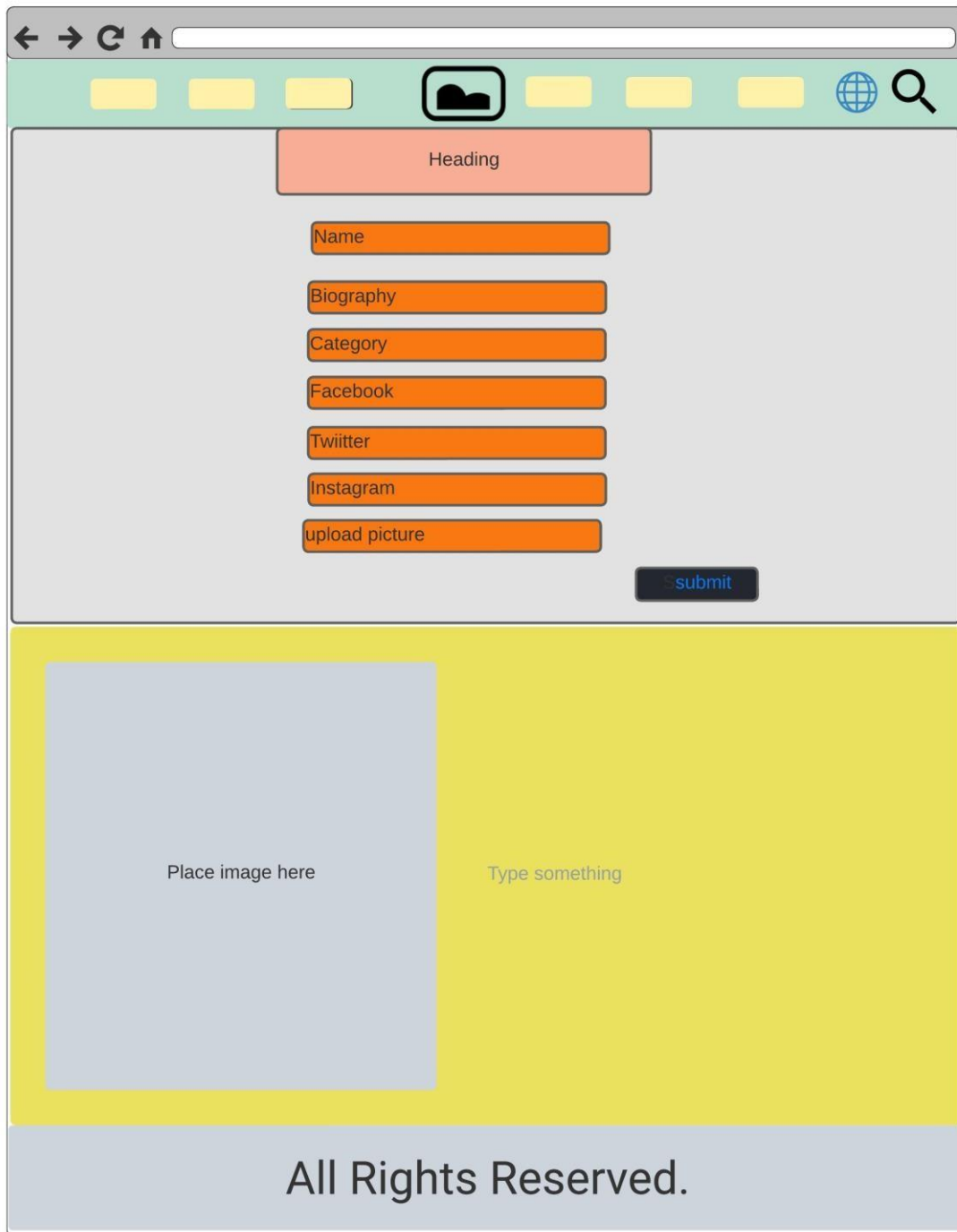
Password

All Rights Reserved.

- UPCOMING & PREVIOUS EVENTS



- **ADD ARTIST PAGE**



← → ↻ ⬆

[Yellow Box] [Yellow Box] [Yellow Box] [Image Icon] [Yellow Box] [Yellow Box] [Yellow Box] [Globe Icon] [Search Icon]

Heading

Name

Biography

Category

Facebook

Twitter

Instagram

upload picture

submit

Place image here

Type something

All Rights Reserved.

- **ADD Event PAGE**

← → ↻ ⬆

Yellow square icons, Cloud icon, Search icon

Heading

Description

TEXT

TEXT

TEXT

TEXT

TEXT

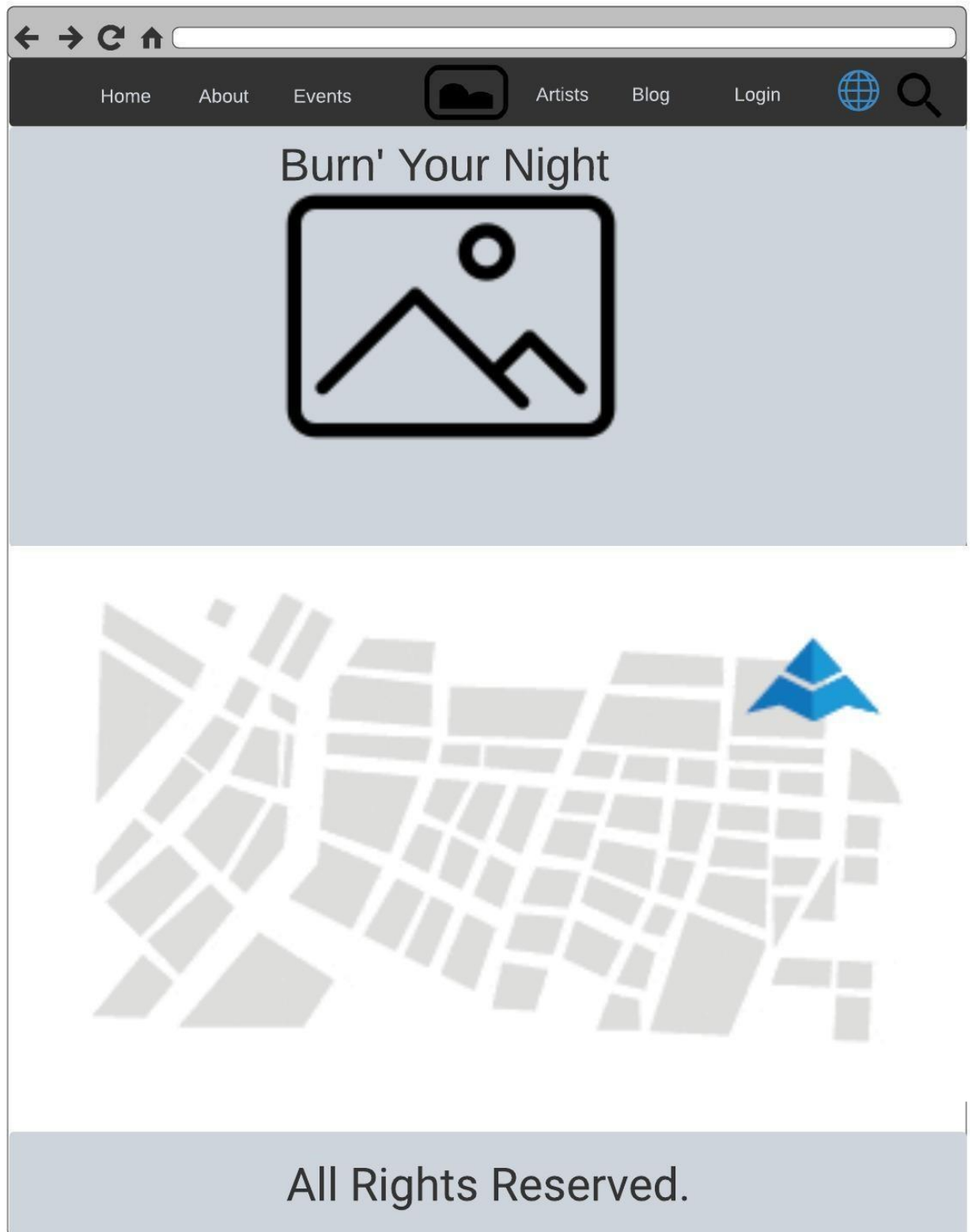
TEXT

Submit

back

-
- *Desktop Medium fidelity Design:*

- MASTER / HOME PAGE



- LOGIN PAGE

← → ↻ ⬆

LOGIN

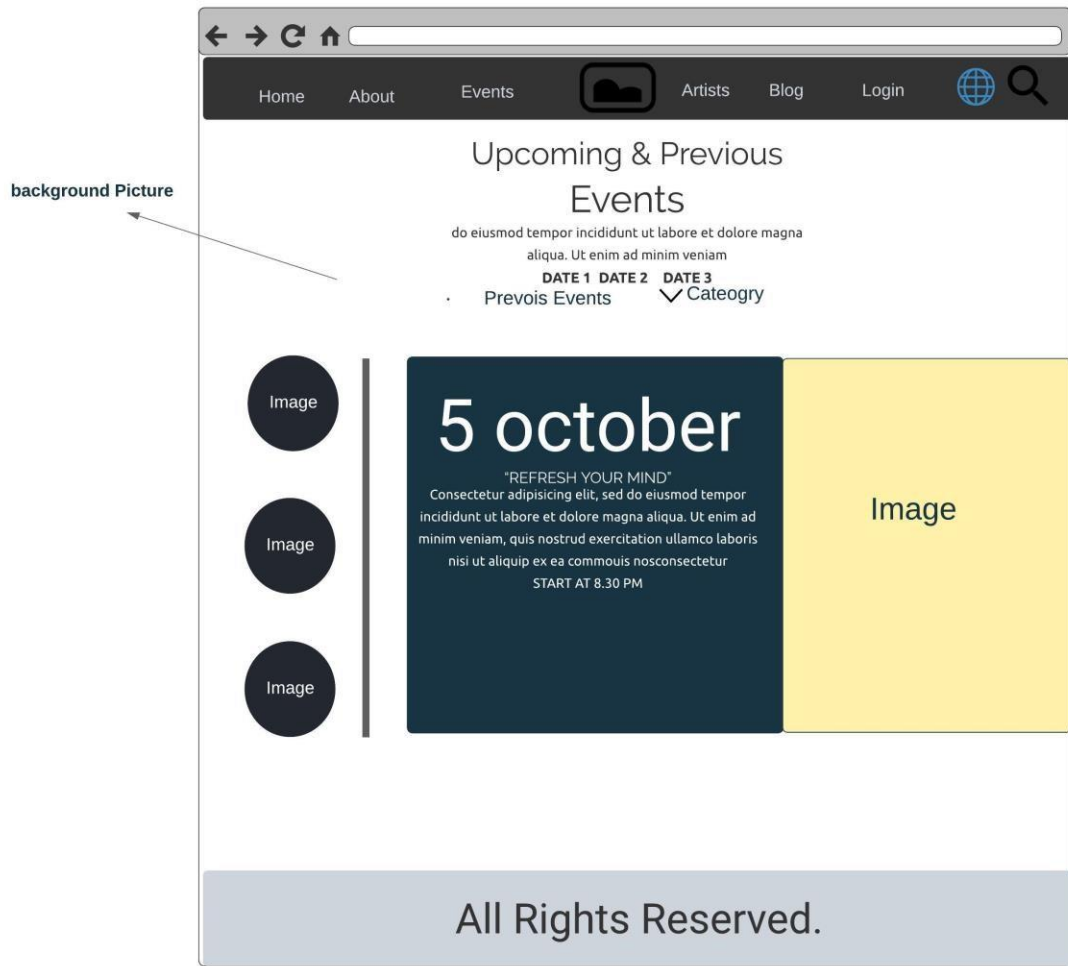
Name

Password

Login

[Forget password?](#)

- UPCOMING/ Previous EVENTS



- **ADD Event PAGE**

← → ↻ 🏠

Home About Events Artists Blog Login 🔍 🌐

ADD Events

Rem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

background Picture

Event Date Start Time

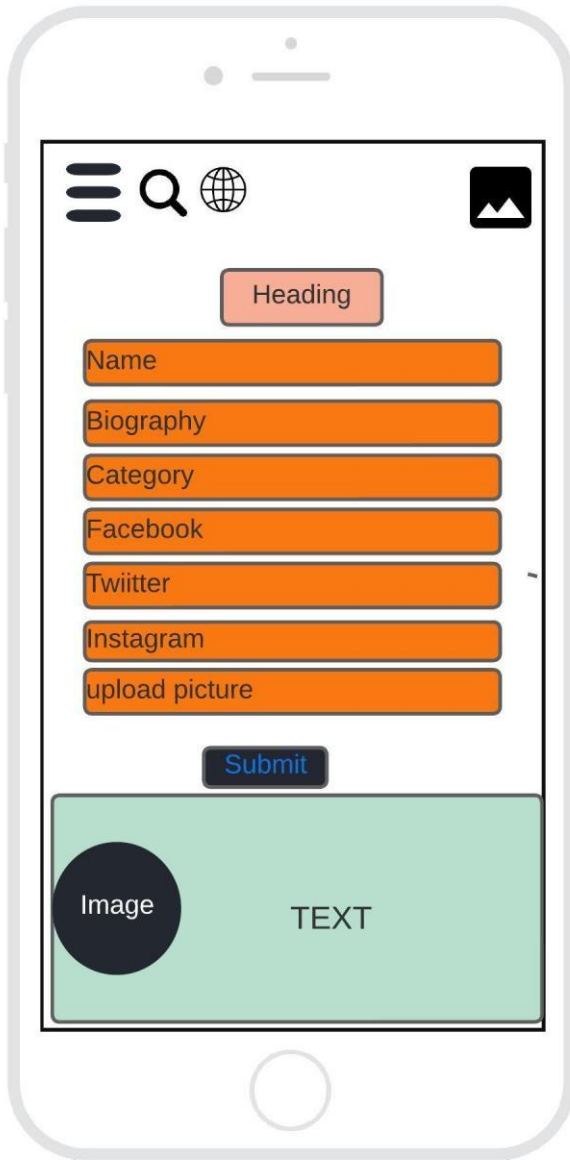
End Time Entrance Fee Capacity

About Artist

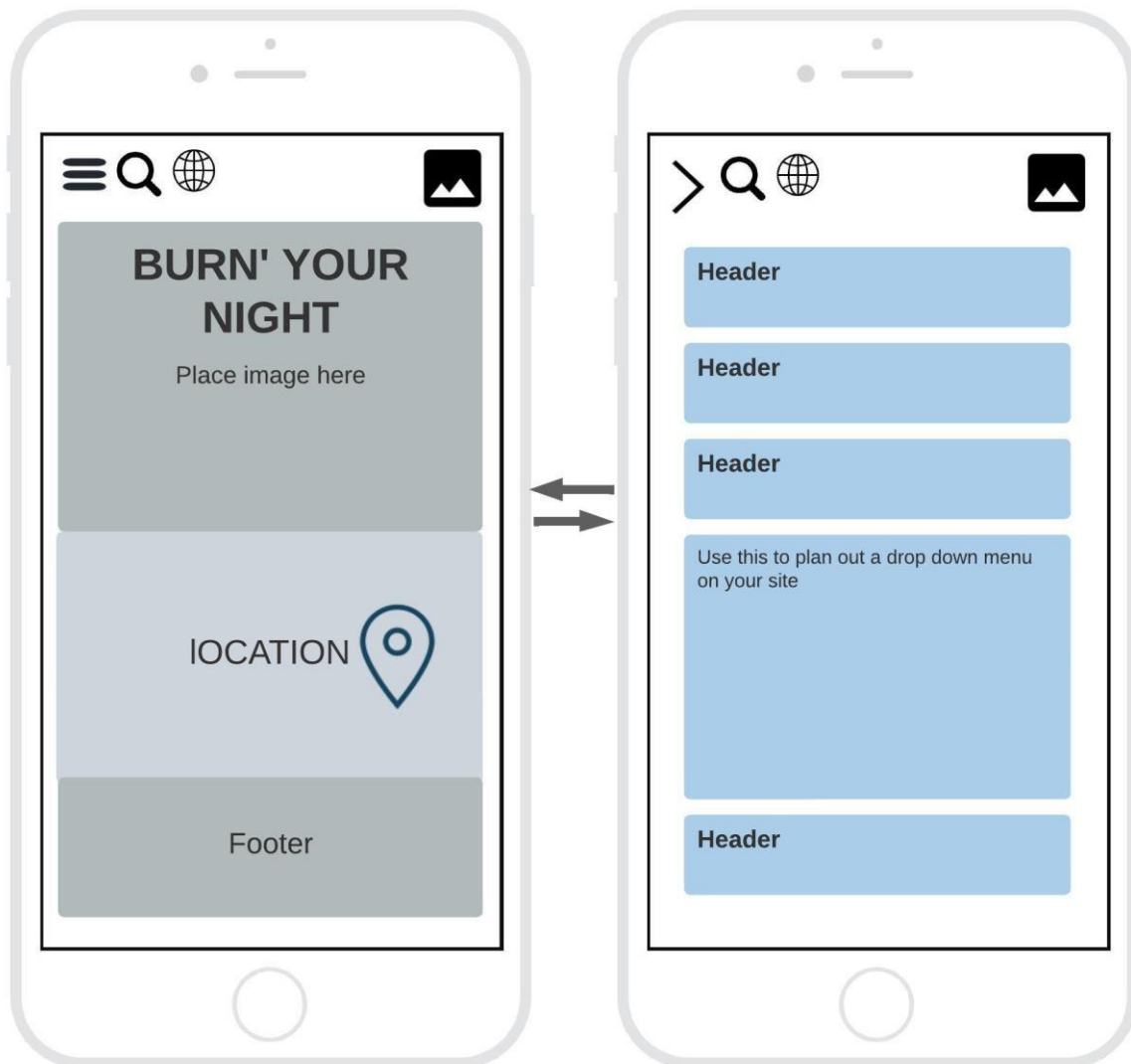
Submit

Low fidelity Mobile:

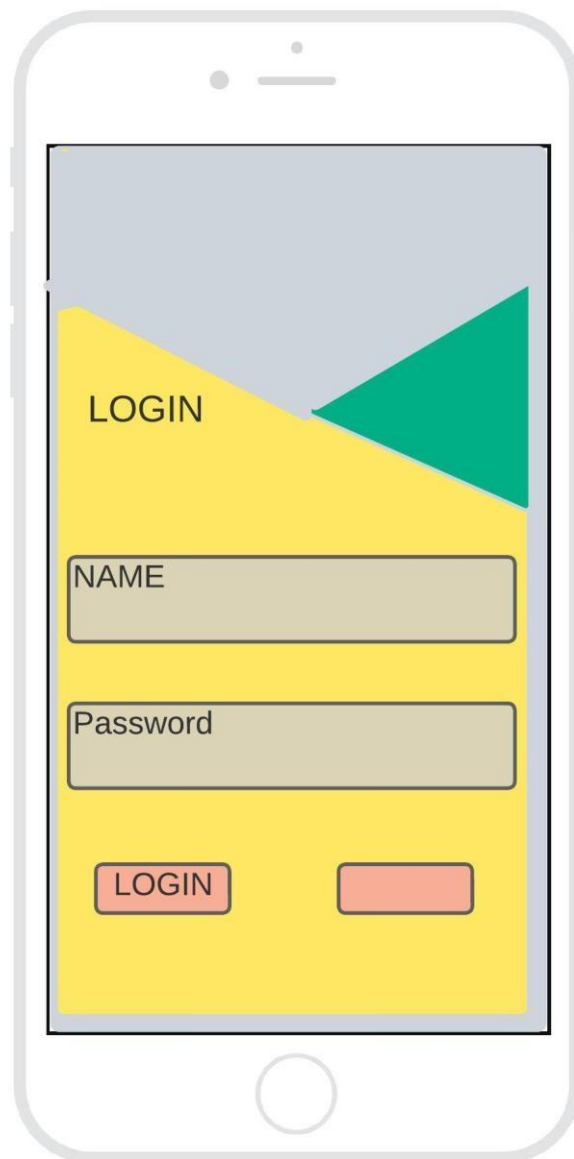
- **ADD ARTIST PAGE**



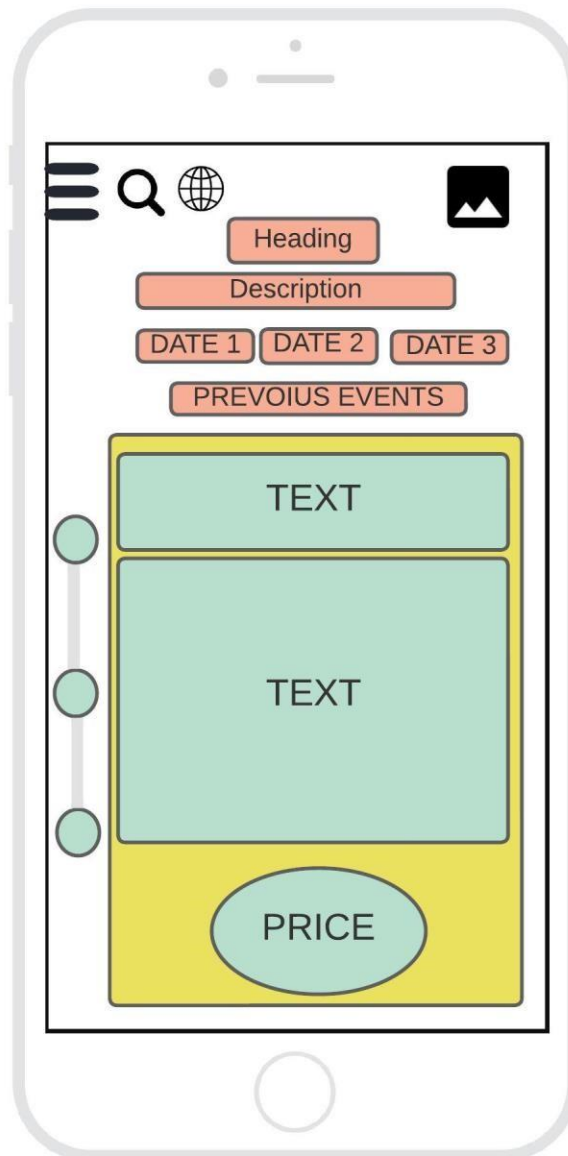
- MASTER PAGE



- LOGIN PAGE



- Upcoming & Previous PAGE



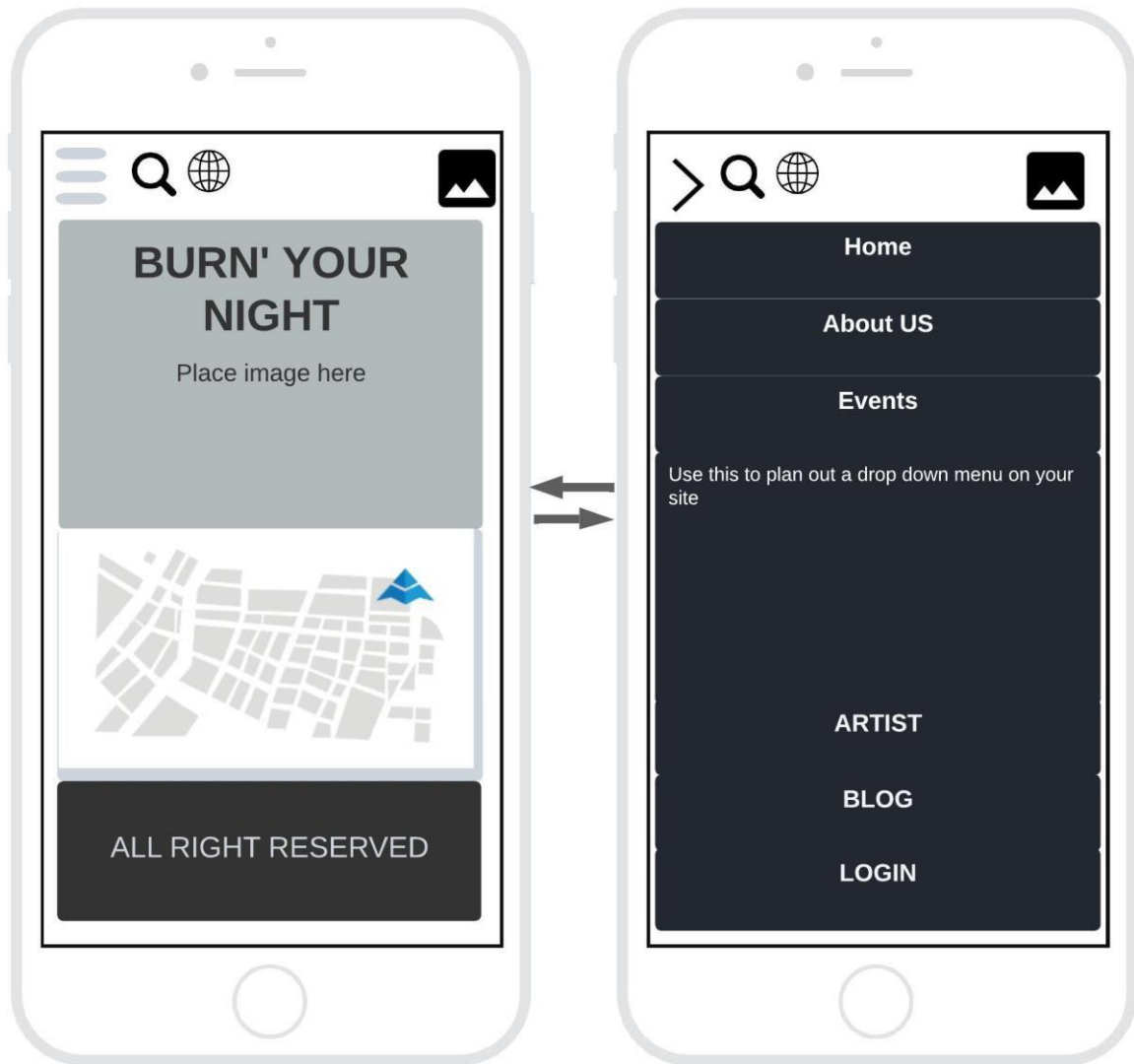
E

- **ADD Event Page**

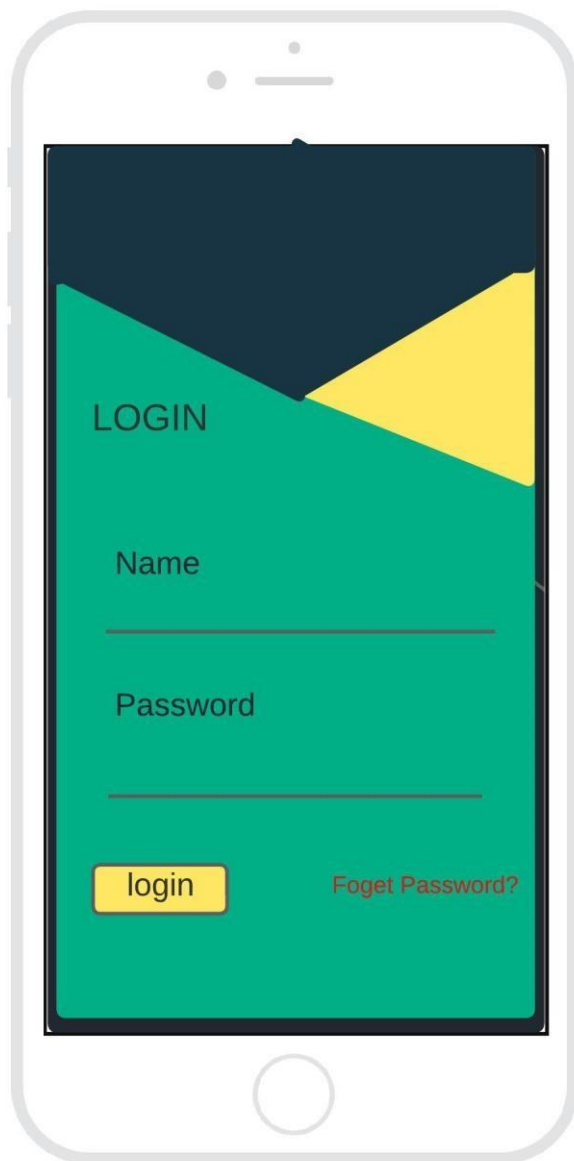
A mobile application interface for adding an event page. The screen is framed by a light gray border representing a smartphone. At the top, there is a header bar with four icons: a hamburger menu, a magnifying glass, a globe, and a square icon with a mountain silhouette. Below the header, the form consists of the following elements: a pink rounded rectangle labeled "Heading"; six yellow rounded rectangles, each labeled "TEXT", stacked vertically; a larger yellow rounded rectangle labeled "TEXT" at the bottom of the text input section; and a red rounded rectangle labeled "Submit" at the very bottom of the form area. The entire form is contained within a black rectangular border.

Medium fidelity Mobile:

- **MASTER /HOME PAGE**



- LOGIN PAGE



- ADD ARTIST PAGE

Add Artist

Name

Biography

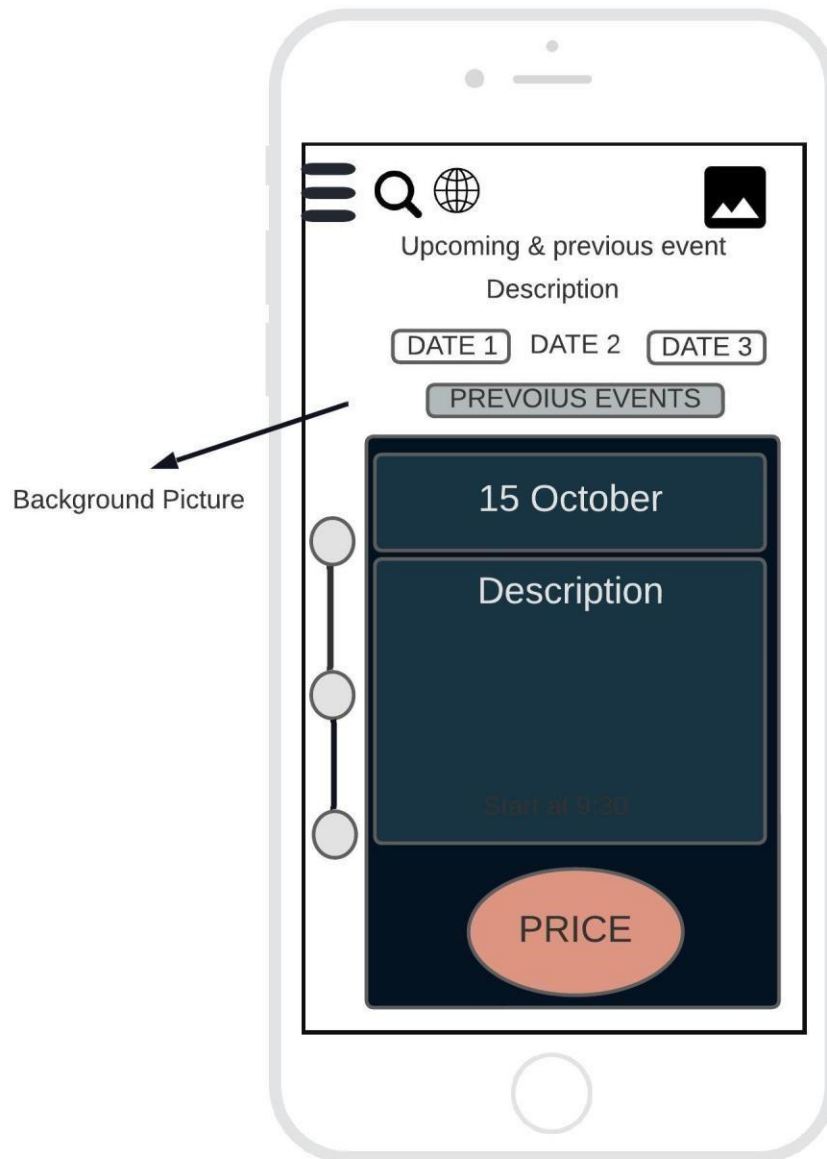
Category

Submit

Image

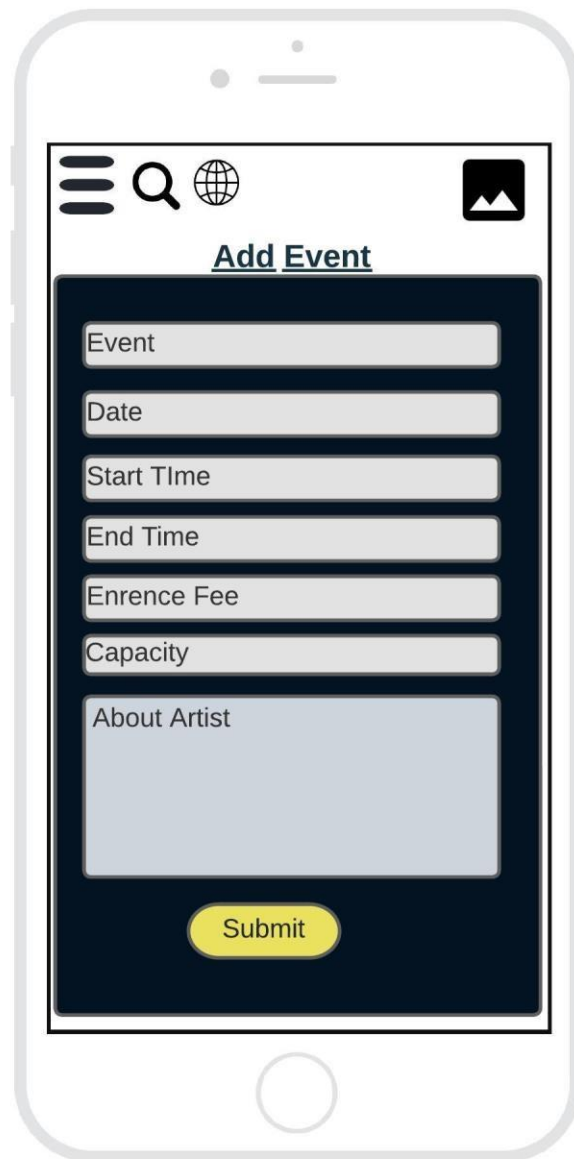
Describe about previous Artist

- Upcoming & Previous Event



- **ADD EVENT**

and adapt to varied screen sizes.



A mobile app mockup showing a form titled "Add Event". The form is displayed on a smartphone screen with a dark blue background. The form fields are white and stacked vertically. The fields are: "Event", "Date", "Start Time", "End Time", "Entrance Fee", "Capacity", and "About Artist" (a larger text area). A yellow "Submit" button is at the bottom of the form. The top of the screen shows a navigation bar with icons for a menu, search, globe, and a profile picture.

Event

Date

Start Time

End Time

Entrance Fee

Capacity

About Artist

Submit

were asked to include discussion of any design choices we made for views and any innovative GUI elements (input or output) that would be used

The nightclub event management web app can benefit from these cutting-edge GUI components by becoming more interesting, interactive, and valuable for users. This will improve the app's standing in the market.

These cutting-edge UI features are perfect for a web app that organises events at nightclubs:

An Event Countdown timer displays the number of seconds, minutes, or hours until an event begins, creating an air of anticipation and excitement among the audience.

The Event Suggestions system helps users find new events that may pique their interest by making suggestions based on their previous searches and preferences.

"Dynamic Event Maps" are maps that indicate the location and status of various events and are updated in real-time to aid users in making more efficient plans for the evening.

Login View: The login view should be straightforward with clear instructions. The desktop version should have a larger email and password input field, while the mobile version should stack input fields vertically. A "Remember Me" checkbox might keep users logged in after closing the programme.

Signup View: The signup view should capture all necessary information with minimal user interaction. The desktop version should feature a larger input field for the user's name and email address, while the mobile version should stack input fields vertically. A "Password Strength" metre that rates the user's password is an interesting GUI addition.

Event Creation View: The event creation view should walk users through the event creation process with clear instructions. The desktop version should have a larger event name and description input field, while the mobile version should employ a vertical style with input fields stacked on top. "Time Selection" GUI elements let users quickly set event start and finish times.

Event Details View: The event details view should include the event name, date, time, location, and attendees to help users organise their events. The mobile version should stack content

vertically, whereas the desktop version should use a table or list structure. A "Map View" that displays the event's location is an interesting GUI element.

Event Search View: The event search view should let users search by location, date, and category. The desktop version should have a wider search bar with filter choices on the side, while the mobile version should utilise a vertical layout with search and filter options stacked. A "Quick Search" option that lets users search events by keywords or categories is an innovative GUI piece.

The night club event management web software should be easy to use with unique GUI elements. While maintaining functionality, desktop and mobile versions should be responsive

AJAX and JSON:

By allowing for asynchronous content loading and data interchange between the client and server, AJAX and JSON may be employed in the interface design of the web application to improve the user experience.

Some potential applications of AJAX and JSON in the aforementioned scenario are as follows:

a) Login screen: AJAX may be used to submit the user's login credentials to the server upon form submission, avoiding the need to reload the page. After a successful login, the server can send back a JSON object indicating the user's status.

When a user accepts the form to add a new artist, the data is sent to the server through AJAX without the need to reload the page.

b) Add new artist view: Using this information, the server may send back a JSON object indicating whether or not the artist was successfully added.

AJAX and JSON may be used to submit the form data and get a response from the server indicating whether the event was added successfully or not, just like they can in the add new artist view.

c) Add new event view: Use AJAX and JSON to submit the form data and get a response from the server indicating whether the event was added successfully or not, just like the add new artist view.

d) See a list of past and future events without having to refresh the page completely with AJAX. Using JavaScript, the server may send back the event data as a JSON object, which can then be displayed dynamically on the website. In addition, AJAX might be utilised to provide filtering and sorting of the event list without requiring the website to be reloaded.

B: Accessibility

Accessibility is a vital aspect of any web application, since it ensures that all users, regardless of their skills or limitations, may properly access and utilise the application. In this section, I will describe how the web application created for Task A) is accessible to a wide range of users, including those with visual, auditory, motor, or cognitive disabilities.

To ensure accessibility, the online application will be developed in accordance with the Web Content Accessibility Guidelines (WCAG). The World Wide Web Consortium's Web Content Accessibility Guidelines (WCAG) are a set of internationally recognised principles for developing accessible web content. They encompass a wide variety of accessibility requirements that cater to the needs of various user groups. The WCAG rules have three levels of conformity: A, AA, and AAA, with AAA being the greatest level of adherence.

Listed below are specific design considerations made to ensure accessibility for various user groups:

- Visual impairments: People with visual impairments can view web material via screen readers or magnifiers. To facilitate this, the online application will have a clear and uniform layout, with appropriate headings, labels, and alt text for images. The colour palette will also be chosen to offer individuals with low vision with adequate contrast and legibility. ARIA (Accessible Rich Internet Applications) features will be utilised to offer screen readers and other assistive technologies with more information.
 - Auditory impairments: To access audio information, users with hearing problems may rely on captions or transcripts. All audio content on the online application will be accompanied by captions or transcripts, and users will have the choice to enable or disable subtitles as necessary.
 - Motor impairments: It may be challenging for users with motor impairments to utilise a mouse or keyboard to navigate the web application. To meet this, the

online application will be created with keyboard navigation in mind and all functionality will be accessible through keyboard. Attention states and other visual cues will be used to facilitate keyboard navigation inside the application. ARIA characteristics will also be used to provide assistive devices with additional information.

- Cognitive impairments: Users with cognitive impairments may have difficulty understanding complex or abstract concepts, or may have difficulty with memory or attention. To accommodate this, the web application will be designed with clear and simple language, with minimal jargon or technical terms. The interface will also be designed to minimize distractions and provide clear feedback to the user.

To ensure accessibility for the **night club web application**, the following design decisions were made:

- Clear and consistent layout: The web application will have a clear and uniform interface to facilitate site navigation for people with visual impairments. This contains appropriate headings, labels, and image alternative text. ARIA characteristics will also be used to provide assistive devices with additional information.
- Color scheme: The color scheme of the web application will be chosen to provide good contrast and readability for users with low vision. This ensures that users with visual impairments can easily read and navigate the site.
- Keyboard navigation: The online application will be created with keyboard navigation in mind, allowing users with motor impairments to easily explore the site using the keyboard alone. Attention states and other visual cues will be used to facilitate keyboard navigation inside the application.

Simple language: The language used in the online application will be straightforward and clear, with no use of jargon or technical phrases. By doing this, it is made sure that people with cognitive disabilities can readily understand the site's information.

Accessible forms: The web application should include forms that are accessible to all users, including those with disabilities. The forms should have clear and concise labels, and provide helpful instructions and error messages. HTML5 provides several form input types and attributes that can be used to create accessible forms.

Login mechanism: The administration pages of the web application should be protected by a login system so that only authorised club personnel can access them. All users, including those with disabilities, should have access to the login method. Providing clear and concise labels and instructions and utilising ARIA roles and attributes to provide additional information for screen reader users are methods for ensuring accessibility.

Assistive technology: The web application must be compatible with assistive technologies, including screen readers, magnifiers, and voice recognition software. Providing alternative language for images, utilising suitable heading structures, and ensuring that interactive features are keyboard accessible are a few strategies that can be utilised to ensure compatibility with assistive technology.

To implement these design decisions, the web application can be built using a web framework like **React(JS,JQUERY)** or **Angular**, which provides built-in support for accessibility features like keyboard navigation and ARIA attributes. The use of a design system like Material Design or Bootstrap can also help ensure accessibility, as these systems provide pre-built components that are designed with accessibility in mind.

In addition, the online application will adhere to the Web Content Accessibility Guidelines (WCAG) so that it complies with worldwide accessibility requirements. Specifically, the WCAG guidelines include recommendations for increasing the accessibility of web content for disabled people. The criteria are structured into three conformance levels: A, AA, and AAA. We will aim for Level AA conformity for this web application, which contains requirements to ensure that content is perceivable, operable, comprehensible, and resilient.

Overall, guaranteeing accessibility is crucial for the success of the nightclub's web application, as it ensures that all users, regardless of their skills or disabilities, can access and utilise the site. The web application can deliver a positive user experience for all users by adhering to the WCAG criteria and adopting design decisions that fit the demands of various user groups.

C. Validation:

In the web application for the night club scenario, there are several places where user input is required.

These include adding and editing events, adding and editing artists, and logging in as admin. Validation checks are necessary to ensure that the data entered by users is correct and does not cause errors or security vulnerabilities in the system.

The validation of user input can occur on both the client-side and server-side of the web application. Client-side validation involves checking the user's input in real-time as they fill out the form or input data into the application. Server-side validation involves checking the data on the server-side after it has been submitted by the user.

Client-side and server-side validation are two essential methods for ensuring that user input is accurate and secure. In the context of a nightclub event management system, it is crucial to validate user input for login, signup, and search features in order to prevent illegal access and assure the accuracy of data provided into the system.

Client-side validation can be used to validate the format of the email address and password entered by the user, whilst server-side validation can be used to validate the email address and password against the database.

Client-side validation:

The email input should conform to the standard email format (for example, example@email.com).

The password should meet the minimum length and complexity requirements (e.g. at least 8 characters, containing both letters and numbers).

Server-side validation:

The email address and password should be checked against the database to ensure that they match the user's information.

If the login attempt fails, the user should receive an error message explaining that the login credentials are invalid.

Signup validation:

Client-side validation can be used to confirm that the user has entered all needed fields and that the input format is proper, whilst server-side validation can be used to ensure that the email address has not previously been registered with the system.

Client-side validation:

All mandatory elements (such as name, email address, and password) must be completed.

The email input must adhere to the proper email format.

The password should meet the minimum length and complexity requirements.

Server-side validation:

The user's email address should be verified against the database to ensure that it has not previously been registered.

If the email address is already registered, an error message noting that the email address is already in use should be displayed to the user.

Client-side validation can be used to ensure that the search input is in the correct format and meets any required criteria, whilst server-side validation can be used to validate that the search query delivers valid results from the database.

Client-side validation:

The format of the search input should be proper (e.g. a string of characters).

The search input must satisfy all applicable requirements (e.g. at least 3 characters in length).

Server-side validation:

It is necessary to conduct the search query against the database to check that it delivers legitimate results.

If the search query yields no results, the user should be presented with an error message indicating that no events fit the search parameters.

In a nightclub event management system, both client-side and server-side validation are essential for ensuring that user input is accurate and safe. Client-side validation can be used to offer the user with quick feedback and enhance the user experience, whilst server-side validation can be used to ensure that the data submitted into the system is accurate and safe. By combining these methods of validation, we may build a more robust and trustworthy system for controlling nightclub events.

For example, in the form for adding and editing events, the following validation checks could be carried out:

Client-side validation:

- Confirm that the date format given is valid (e.g. YYYY-MM-DD).
- Confirm that the start and end times are formatted correctly (e.g. HH:MM).
- Confirm that the admission charge is a genuine number.
- Confirm that the maximum occupancy is a valid number.

Server-side validation:

- Confirm that the entered date is not in the past.
- Confirm sure the start time is earlier than the finish time.
- Confirm that the entry fee is positive.
- Verify that the maximum capacity for attendance is not negative.
- Confirm that the selected event category is valid (e.g. Music, Poetry or Comedy).
- Confirm that the selected artist is valid (i.e. exists in the database).
- Confirm that the event does not conflict with any other scheduled events.

For the form for adding and editing artists, the following validation checks could be carried out:

Client-side validation:

- Verify that the artist field is not blank.
- Verify that the biography is not excessively long (e.g. limited to 500 characters).
- Verify that the social media usernames are formatted properly (e.g. @username).

Server-side validation:

- Verify that the name of the artist is not already in the database.
 - Verify the authenticity of the social media usernames (e.g. exist on the respective platforms).
 - Verify that the artist's publicity photo is not excessively large (e.g. limited to 5MB).
- For the login form for admins, the following validation checks could be carried out:

Client-side validation:

- Check that the username and password are not empty.

Server-side validation:

- Check that the username and password are correct and match a valid admin account in the database.

Use secure hashing techniques to store and compare database passwords.

In accordance with the WCAG criteria for accessibility, the web application should provide users with visual or cognitive impairments with clear and succinct error messages when validation tests fail. In addition, the online application should employ ARIA (Accessible Rich Internet Applications) features to aid screen readers and other assistive devices in comprehending form inputs and any errors.

For the user input shown in, the following validation checks would be carried out:

1. Event Date and Time: The date and time for an event must be valid and cannot conflict with other events that have already been scheduled. This validation check will be performed on the server, as it needs a database query for existent events. Here is an example pseudocode for performing this validation check:

```
if (eventDateTime is not a valid date and time) { display
error message "Invalid date and time entered." } else if
(eventDateTime conflicts with existing events) { display
error message "Event date and time conflicts with another
scheduled event."
} else {
    save event to database
}
```

2. Entrance Fee: The admission charge cannot be negative and must be a genuine currency amount (e.g., \$10.50). This validation check can be performed on the client or server using JavaScript. Here is an example of pseudocode that uses SQL to do this validation check on the server:

```
if (entranceFee is not a valid currency amount or entranceFee < 0)
{
    display error message "Invalid entrance fee entered."
} else {
```

```
        save event to database
    }
```

3. Maximum Attendance Capacity: Maximum capacity must be an integer greater than zero. This validation check can be performed on the client or server using JavaScript. Here is an example of pseudocode that uses SQL to do this validation check on the server:

```
if (attendanceCapacity is not a valid number or attendanceCapacity <= 0)
{
    display error message "Invalid attendance capacity entered."
} else {
    save event to database
}
```

4. Artist Details: The artist details cannot be left blank (i.e., name, biography, social media usernames, and PR image). This validation check can be performed on the client or server using JavaScript. Here is an example of pseudocode for utilising SQL to do this validation check on the server:

```
If (artistName or artistBio or socialMedia or artistImage is blank) {
    display error message "Artist details cannot be left blank."
} else {
    save event to database
}
```

It is excellent practise to execute validation checks on both the client and server sides to guarantee that every user input is checked prior to being processed or saved to the database. Client-side validation checks can offer the user with quick feedback, while server-side validation checks provide an additional layer of security and assist prevent malicious attacks or errors.

Hacking:

Protecting a website from hacking involves implementing various security measures at different levels. Here are some steps that can be taken to improve the security of a PHP-based website:

- Keep PHP up-to-date: While new security patches are frequently published to address known vulnerabilities, it is essential to maintain PHP updated to the newest version.
- Employ parameterized queries To prevent SQL injection attacks, parameterized queries must be implemented. This involves substituting placeholders in SQL statements with user supplied data. Here's one eg:

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE username  
= ?");  
$stmt->execute([$username]);
```

- Sanitize user input: To prevent malicious code insertion, every user input should be sanitised. Using the filter input () function or regular expressions, it is possible to ensure that only acceptable data is accepted.
- Use HTTPS: HTTPS facilitates safe Internet communication by encrypting data exchanged between the server and client. Obtaining an SSL/TLS certificate and configuring the server to use HTTPS can do this.
- Implement authentication and permissions: Authentication verifies that users are who they say they are, whereas authorization restricts access to resources based on a user's role or rights. This is possible using session management and access control lists.

Use a firewall: By limiting incoming traffic from questionable IP addresses and established attack vectors, a firewall can prevent unwanted access to the server.

- Frequently back up data: The website and database should be backed up on a regular basis to guarantee data can be restored in the event of a security breach or system failure. The following PHP code demonstrates the implementation of basic authentication and authorization:

```
// Start a new session session_start();
```

```
// Check if the user is already logged in if
(asset($_SESSION['user'])) {
    // User is already logged in, redirect to dashboard
    header("Location: dashboard");

    exit;
}

// Check if the login form was submitted if
($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the username and password from the form
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Validate the username and password if ($username
    == 'admin' && $password == 'password') {
        // Authentication successful, set session variables
        $_SESSION['user'] = 'admin';
        $_SESSION['role'] = 'admin';

        // Redirect to dashboard
        header("Location: dashboard");    exit;
    } else {
        // Authentication failed, show error message
        $error = 'Invalid username or password';
    }
}
```

}

If the user is not already logged in, this code verifies whether the login form was submitted. It then verifies the username and password and, if authentication is successful, sets session variables. The session variables can then be utilised throughout the website to implement authorisation checks.

D. One additional innovative item of functionality or feature:

A suggestion engine is a unique feature that might be introduced to the club event promotion web application. This feature would recommend events to users based on their past preferences and behaviour.

The recommendation engine could be developed using a collaborative filtering algorithm that analyses user behaviour data to recognise patterns and generate recommendations. This algorithm might be incorporated into the user's account in order to track their historical event attendance, event category preferences, and artist preferences.

To implement the recommendation engine, the web application would have to collect and store user behaviour data in a database. The programme would then utilise this information to provide user-specific event recommendations. These recommendations could be displayed on the dashboard or in a distinct section of the online application.

To make the recommendation engine accessible and user-friendly, it might be linked with the existing event browsing and filtering capabilities. While creating an account, users could opt-in to the recommendation engine by stating their preferences, and they could modify these preferences at any time. In addition, the system might be fine-tuned as more data is collected and examined.

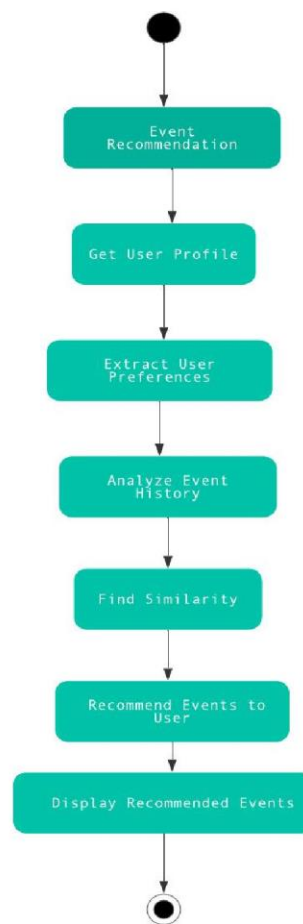
Included in low-fidelity mock-ups for the recommendation engine might be wireframes of the user dashboard and event pages, with a space for personalised event recommendations. Midfidelity designs may contain more comprehensive prototypes of the recommendation

engine's interface, demonstrating how the algorithm provides individualised event recommendations and how users can modify their preferences.

Overall, the recommendation engine would be a valuable and unique addition to the web app for club event advertising, helping to tailor the user experience and setting the web app apart from the competitors.

The following activity diagram illustrates the new feature's process:

Activity diagram



The activity diagram begins with collecting the user's profile data, such as name, age, and interests. This information is used to determine the types of events in which they are interested in participating.

The next stage is to examine the user's event history by examining the events they've attended in the past and utilising this information to predict the kind of events they may be interested in in the future.

After analysing the user's preferences and event history, the algorithm looks for similarities between the user's preferences and planned club events. This could involve evaluating genres of music or sorts of performers to determine which events best suit the user.

Last but not least, the system proposes events to the user based on their preferences and event history, and shows these events on the website. The user can then peruse the suggested events and select those they wish to attend.

This new feature would distinguish the web app from rivals by giving individualised event recommendations to users, so boosting their experience and increasing their likelihood of attending club activities.

E. Graphs:

Chart.js:

Chart.js is a JavaScript library for the creation of responsive, interactive, and modifiable charts and graphs. It supports numerous chart kinds, including line, bar, pie, donut, radar, polar area, and bubble charts. Chart.js

also allows you to animate your charts and offers numerous customization choices, including colours, fonts, labels, tooltips, and legends.

Advantages:

- a) Chart.js is a compact, lightweight library, making it simple to load and use.
- b) It is extremely user-friendly and simple to use, with extensive online documentation, examples, and tutorials.
- c) Chart.js supports a range of chart types and offers a large number of customisation options, making it a flexible library that can be utilised for a variety of applications.
- d) It is open-source, free to use, and requires no commercial licence.

Disadvantages:

- a) Chart.js may not be as powerful or feature-rich as some other charting libraries, and may not be suitable for very complex or large-scale projects.
- b) It may be less flexible than some other libraries in terms of customization options, although this is not a major issue for most use cases.

D3.js:

Data-driven documents (D3.js) is a JavaScript data visualisation toolkit. It offers a variety of tools and methods for building bespoke data visualisations, such as charts, graphs, maps, and animations. D3.js is highly flexible and can be used to generate extremely sophisticated and interactive graphics, although its learning curve is steeper than that of competing tools.

Advantages:

- a) D3.js is an extremely strong and versatile library, capable of producing complex and interactive visuals that other libraries may not be able to handle.
- b) It is open-source, free to use, and requires no commercial licence.
- c) D3.js includes a multitude of tools and functions for working with data, such as parsing, formatting, filtering, and altering data, which can be quite handy when developing bespoke visualisations.

Disadvantages:

- a) D3.js has a higher learning curve than comparable libraries, and mastering it may need more time and effort.
- b) It may be harder to use and modify than other libraries, particularly for less-experienced developers.
- c) D3.js may not be suited for uncomplicated or simple visualisations, as it is built for more complex and configurable use cases.

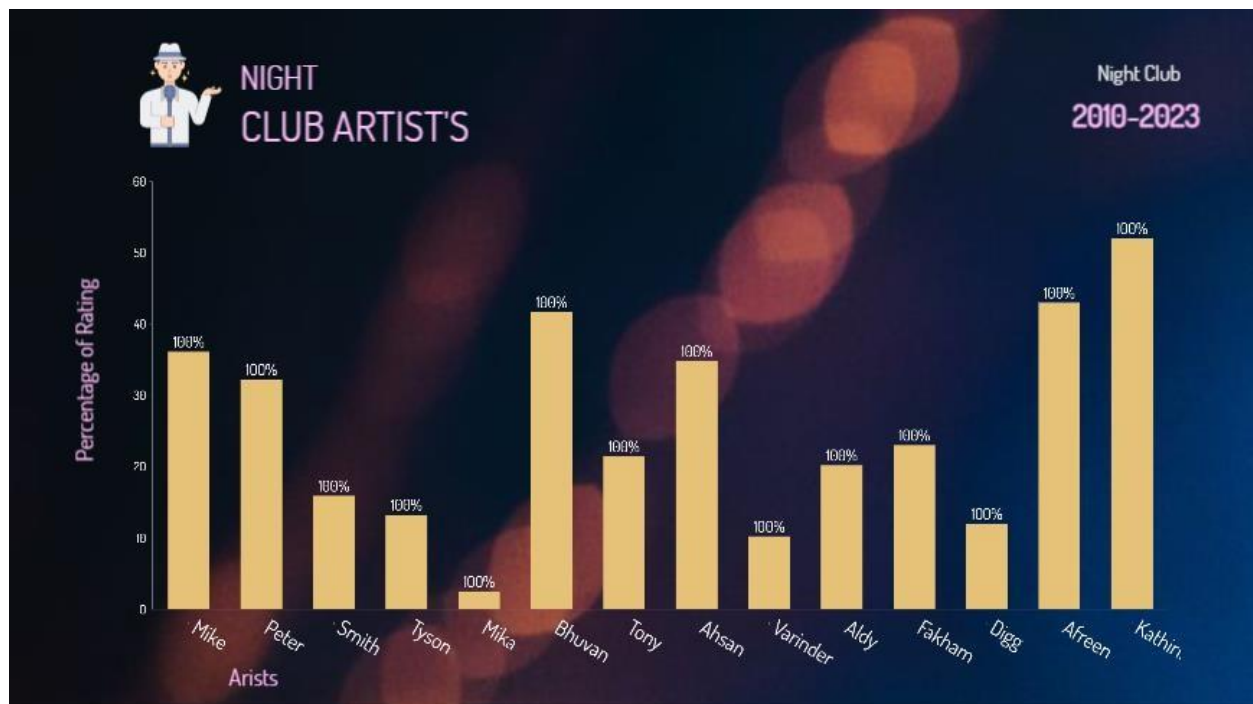
Both Chart.js and D3.js can be utilised to generate a range of relevant visuals for club managers' report generation. For instance, they can be used to produce bar charts displaying the number of events by category, line charts displaying the number of events each month, and pie charts displaying the distribution of events by artist or venue, among other visualisations. These visualisations can assist managers in comprehending data trends and patterns quickly and efficiently, allowing them to make informed decisions about future events.

Ultimately, the decision between Chart.js and D3.js will depend on the project's particular demands and requirements. D3.js is more suited to elaborate and customisable visualisations that require more data processing and modification than Chart.js is.

- **GRAPH OF EVENT ADUIENCE**



GRAPH OF ARTIST'S RATING



- GRAPH OF ARTIST'S BY CATEGORY

