

Project Report

On

### EventaList :Event Management System

**Submitted to D Y Patil International University, Akurdi, Pune in partial fulfilment of full-time degree**

Bachelor of Computer Applications

**Submitted By:** Name: Shekhar Patil PRN: 20210801123

Name: Manish Parihar PRN: 20210801124

Name: Rohit Patil PRN: 20210801128

Under the Guidance of

### Dr. Sarika Jadhav

School of Computer Science, Engineering and Applications

### D Y Patil International University, Akurdi,Pune, INDIA, 411044

[Session 2023-24]



# CERTIFICATE

This report on EventaList Event Management System Using Php and Mysql with Docker is submitted for the partial fulfillment of project,which is part of Bachelor of Computer Applications curriculum, under my supervision and guidance.

#### Ms. Hetal Thaker

(DYPIU Guide)

#### Dr. Swapnil Waghmare Dr. Maheshwari Biradar

(Project Coordinator) (HOD BCA & MCA)

#### Dr. Bahubali Shiragapur Director

School of Computer Science Engineering & Applications D Y Patil International University, Akurdi

Pune, 411044, Maharashtra, INDIA

## DECLARATION

I, hereby declare that the following Project which is being presented in the Project entitled as EventaList Event Management System Using Php and Mysql with Docker is an authentic documentation of my own original work to the best of my knowledge. The following Project and its report in part or whole, has not been presented or submitted by me for any purpose in any other institute or organization. Any contribution made to my work, with whom i have worked at D Y Patil International University, Akurdi, Pune, is explicitly acknowledged in the report.

Name: Chaitanya Mule

PRN No: 20210801057 Signature :

Name: Aditya Gaonkar

PRN No:20210801058 Signature :

Name: Bushra Tamboli

PRN No:20210801033 Signature :

## ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide Ms. Hetal Thaker, for her valuable help and guidance. We are thankful for the encouragement that she has given us in completing this Project successfully.

It is imperative for us to mention the fact that the report of project could not have been accomplished without the periodic suggestions and advice of our project supervisor Dr. Swapnil Waghmare.

We are also grateful to our respected, Dr. Bahubali Shiragapur(Director), Dr. Maheshwari Biradar (HOD, BCA & MCA) and Hon’ble Vice Chancellor, DYPIU, Akurdi, Prof. Prabhat Ranjan for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing support and encouragement.

Name: Chaitanya Mule PRN No: 20210801057

Name: Aditya Gaonkar PRN No: 20210801058

Name: Bushra Tamboli PRN No: 20210801033

### Abstract

1. Requirement Analysis:

This phase involves gathering and analyzing requirements from stakeholders. It includes identifying user needs, functional requirements, and system constraints.

1. Database Design:

Designing the database schema to store user information and event data efficiently. This includes defining tables, relationships, and constraints.

1. User Interface Design:

Designing a user-friendly interface with intuitive controls and layouts. This involves creating wireframes or prototypes to visualize the user experience.

1. Authentication Implementation:

Developing the authentication mechanism to allow only registered users to access the system. This may involve implementing login/signup functionality, password hashing, and session management.

1. User Management Module:

Developing modules to manage user information, including CRUD (Create, Read, Update, Delete) operations for user profiles. This includes storing personal details such as name, address, and contact information.

1. Event Management Module:

Developing modules to manage events, including features for creating, editing, and deleting events. This involves storing event details such as title, description, date, time, location, etc.

1. Search Functionality:

Implementing search capabilities to allow users to search for events or user details based on various criteria. This may involve implementing filters, sorting options, and keyword search.

**keywords:**

* PHP:

PHP (Hypertext Preprocessor) is a popular server-side scripting language used for web development.

* MySQL:

MySQL is a relational database management system (RDBMS) used for storing and managing structured data.

* Docker:

Docker is a platform for developing, shipping, and running applications using containerization.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** |  |
| **[DECLARATION](#_bookmark0)** | **i** |
| **[ACKNOWLEDGEMENT](#_bookmark0)** | **ii** |
| **[ABSTRACT](#_bookmark0)** | **iii** |
| **[LIST OF FIGURES](#_bookmark0)** | **v** |
| **[1 INTRODUCTION](#_bookmark1)** | **1** |
| [1.1 Background](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 1 |
| [1.2 Objectives](#_bookmark3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 1 |
| [1.3 Purpose](#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 1 |
| [1.4 Scope](#_bookmark5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| [1.5 Applicability](#_bookmark6) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| **[2 PROJECT PLAN](#_bookmark7)** | **3** |
| [2.1 Problem Statement](#_bookmark8) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| [2.2 Requirement Specification](#_bookmark9) . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| [2.3 Time Line chart](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| **[3 PROPOSED SYSTEM AND METHODOLOGY](#_bookmark12)** | **6** |
| [3.1 System Architecture](#_bookmark13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| [3.2 Implementation](#_bookmark16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| [3.2.1 Data Flow Diagrams](#_bookmark17) . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| [3.2.2 UML Diagrams](#_bookmark21) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
| [3.2.3 Decision Tree](#_bookmark26) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 16 |
| [3.2.4 Data Dictionary](#_bookmark28) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 17 |
| [3.2.5 Decision Table](#_bookmark29) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 19 |
| **[4 RESULTS AND EXPLANATION](#_bookmark30)** | **20** |
| [4.1 Implementation Approaches](#_bookmark31) . . . . . . . . . . . . . . . . . . . . . . . . . . . | 20 |
| [4.2 Pseudo Code](#_bookmark32) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 21 |
| [4.3 Testing](#_bookmark33) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 23 |
| **[5 CONCLUSION](#_bookmark34)** | **24** |
| **[6 REFERENCES](#_bookmark35)** | **25** |

# List of Figures

1. [Time Line Chart](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
2. [ERD Diagram](#_bookmark14) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
3. [Flow Chart](#_bookmark15) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
4. [Level 0 DFD](#_bookmark18) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
5. [Level 1 DFD](#_bookmark19) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
6. [Level 2 DFD](#_bookmark20) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
7. [Use Case Diagram](#_bookmark22) 11
8. [Sequence Diagram](#_bookmark23) 13
9. [Activity Diagram](#_bookmark24) 14
10. [Class Diagram](#_bookmark25) 15
11. [Decision Tree](#_bookmark27) 16
12. [Decision Tree](#_bookmark27) 16

## INTRODUCTION

#### Background

This existing system is not providing secure registration and profile management of all the users properly. This system is not providing on-line Help. This system doesn’t provide tracking of users activities and their progress. This manual system gives us very less security for saving data and some data may be lost due to mismanagement. This system is not providing event management through internet. This system is not providing proper events information. The system is giving manual information through the event management executer.

#### Objectives

The objective of this application is to develop a system that effectively manages all the data related to the various events that take place in an organization. The purpose is to maintain a centralized database of all event related information. The goal is to support various functions and processes necessary to manage the data efficiently.

#### Purpose

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. This system maintains user’s personal, a and contact details. This system will provide on line help and search capabilities. User friendliness is provided in the application with various controls provided by system rich user interface. Authentication is provided for this application only registered users can access. Event information files can be stored in centralized database which can be maintained by the system. This system provides the users to manage the events systematically.

#### Scope

The scope of an event management system is broad, encompassing a range of functionalities from administrative event management to user-focused booking processes.

The application can cater to various needs, from organizing small personal events to managing large-scale conferences and public gatherings.

To enhance its usability and reach, the system can integrate with popular calendar applications and ensuring seamless cross-platform functionality

#### Applicability

An event management system showcases its broad applicability across various domains and industries, serving as a versatile tool for efficient event planning, organization, and participation. From individuals organizing personal gatherings to businesses managing large conferences and public events, this application proves its utility. It is particularly valuable in sectors requiring precise coordination and attendee management, such as corporate events, entertainment, and education. The system enhances workflow in industries like healthcare for medical conferences, the legal sector for seminars, and government agencies for public events. Freelancers, entrepreneurs, and small businesses benefit from its streamlined booking and promotion capabilities. Whether in tourism, hospitality, real estate, or non-profit sectors, the adaptability of this application supports effective event handling, compliance with regulations, and improved operational efficiency, making it an indispensable tool in today’s interconnected world.

## PROJECT PLAN

#### Problem Statement

**NUMBER OF MODULES**

The system after careful analysis has been identified to be presented with the following

modules:

Event Management System Module

In EMS project we use PHP and MySQL database. It has two modules.

1. Admin Module

2. User Module

**Admin Module**

1. Dashboard: In this section, admin can see all detail in brief like listed categories,

Sponsors, Total Events, Total Registered Users, Total Booking, Total New Booking,

Total Confirmed Booking and Total Cancelled Booking.

2. Category: In this section, admin manage event category (add and update).

3. Manage Sponsors: In this section, admin can add sponsors and manage sponsors details

(Add/Update/Delete).

4. Events: In this section, admin manage events (add and update).

5. Manage Users: In this section, admin can update details of registered users and also

block them.

6. Manage Booking: In this section, admin can manage booking by cancel and confirm it.

7. News: In this section admin manage news (add and delete).

8. Website Setting: In this section, admin can update about us and another general website

setting.

Admin can also update his profile, change password and recover password.

**User Module**

In this module there is two types of user guest user and registered user.

Guest User: In this guest user can see only general information like about us, event details,

Contact details and new about events.

Registered users can do following activity

1. Books the events.

2. Update his/her own profile.

3. Change Password.

4. Users can also cancel booking which is not confirmed.

5. Registered user can also recover his/her own password.

#### Requirement Specification

**HARDWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| Processor | : | Intel P-IV based system |
| Processor Speed | : | 2.0. GHz |
| RAM | : | 1GB |
| Hard Disk | : | 40GB to 80GB |

**SOFTWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| Database | : | MySQL |
| Server | : | Apache |
| Frontend | : | HTML |
| Scripting language | : | Java Script |
| IDE | : | Sublime |
| Technology | : | PHP |

.

#### Time Line chart

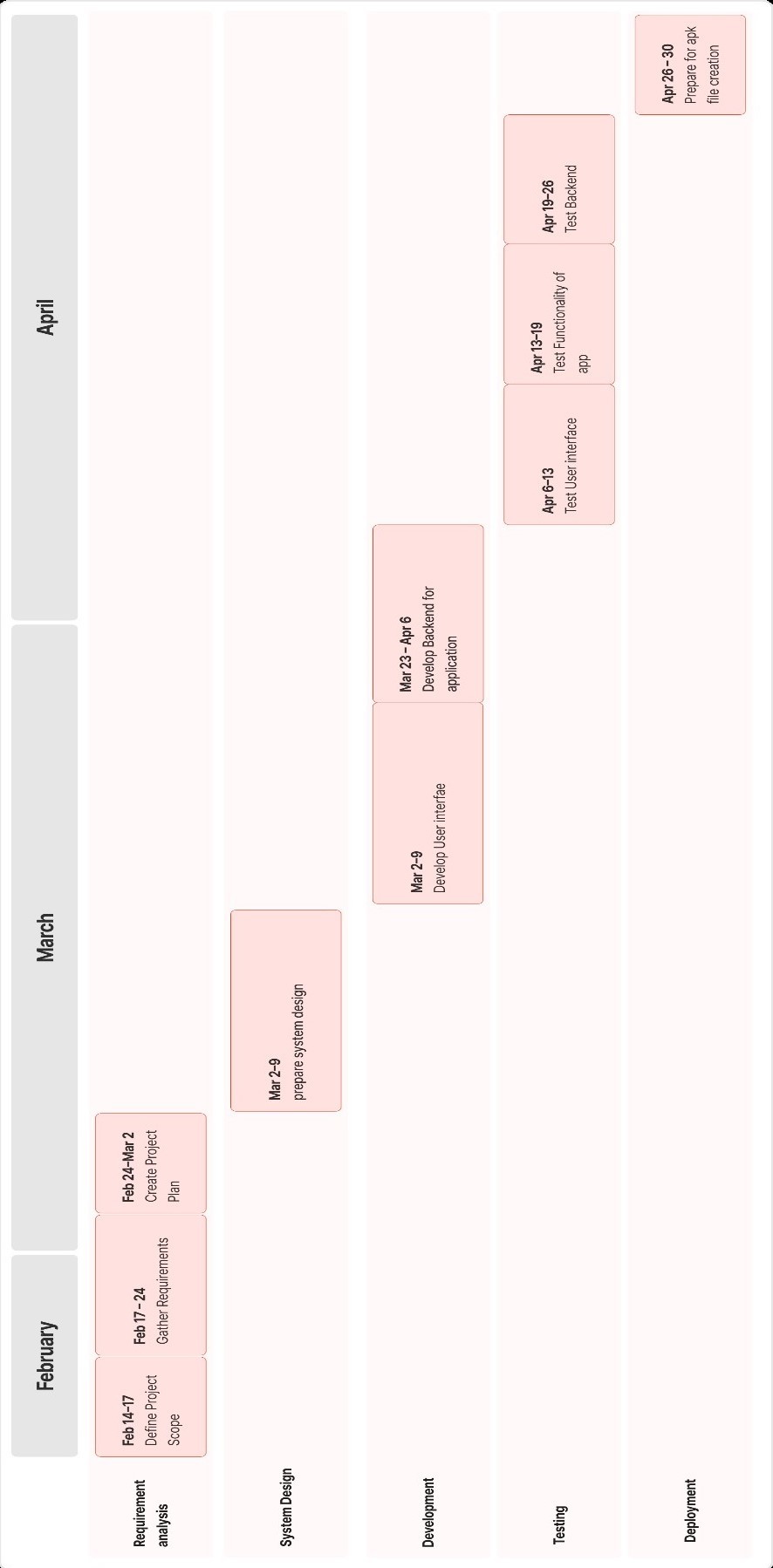


Fig. 1: Time Line Chart

## PROPOSED SYSTEM AND METHODOLOGY

#### System Architecture

The system architecture refers to the overall design and structure of the software system, including how its components interact with each other and with external systems.

#### Flow Chart

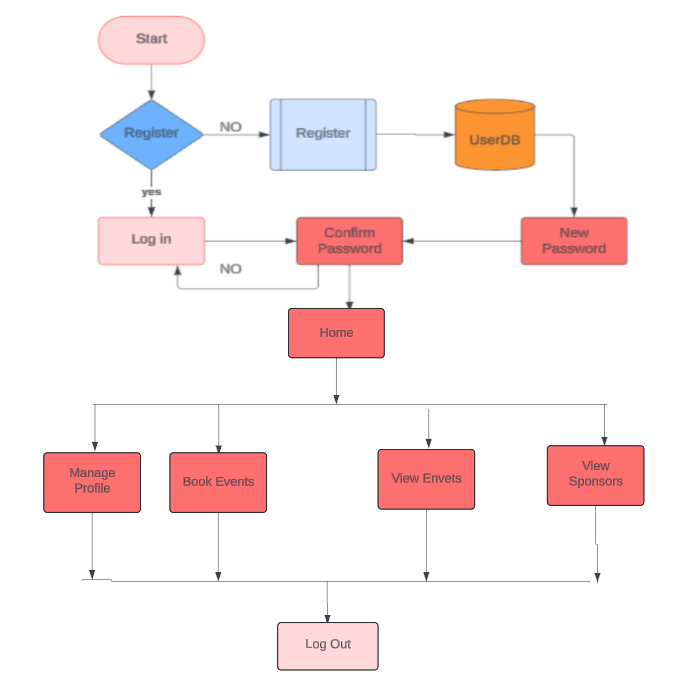
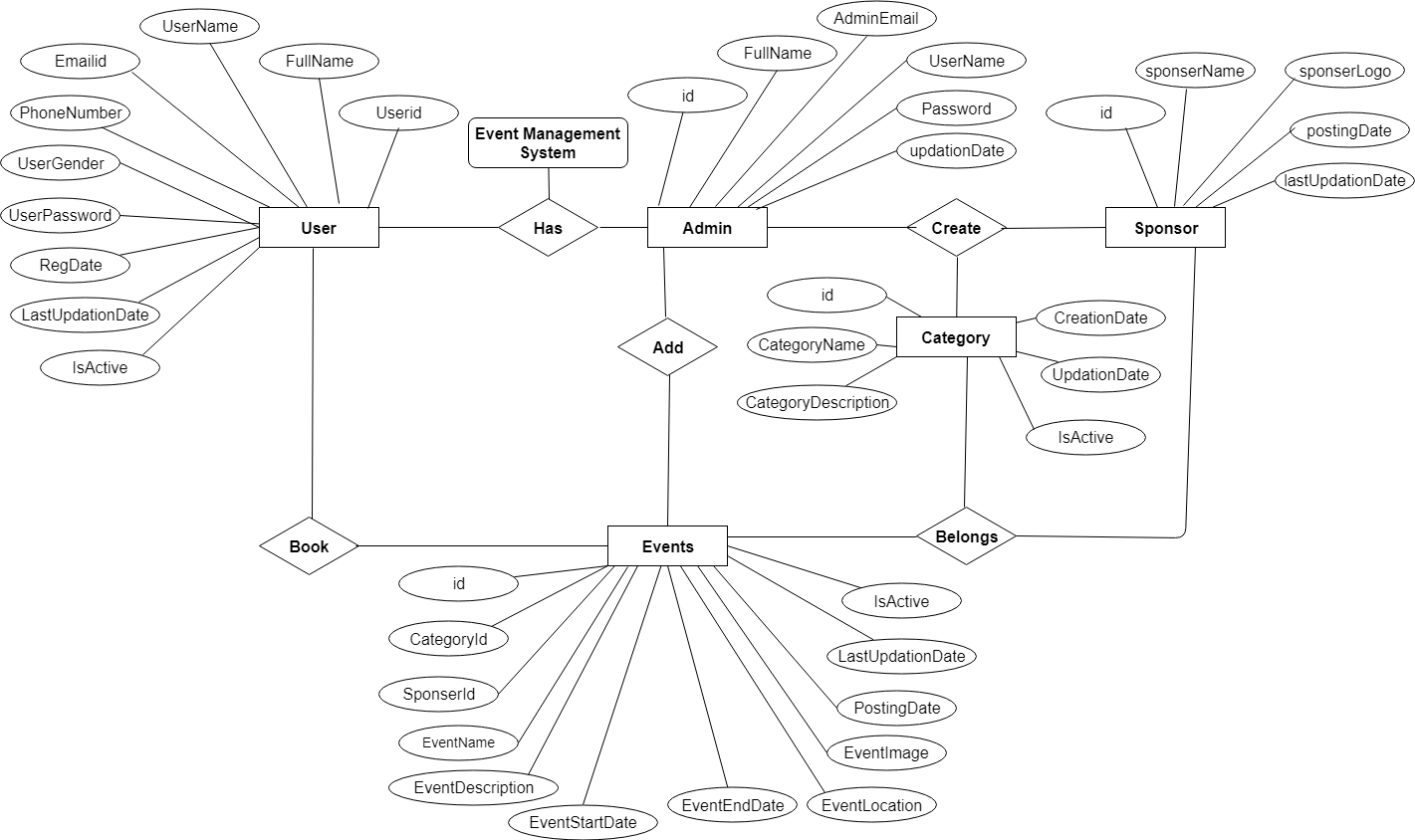


Fig. 3: Flow Chart

1. **ERD Diagram**

This is an ER model of a Eventalist -event management website. The entities are represented in rectangular boxes and relationships between different entities are represented by a diamond shaped box.

Fig. 2: ERD Diagram

* 1. **Implementation**
     1. **Data Flow Diagrams**

The Zero Level DFD is an abstract view of overall system. This is also called as Context Diagram in which entire system is represented as single process with

its relationship with external entities such as admin, user etc.

* + - 1. **Level 0 DFD**

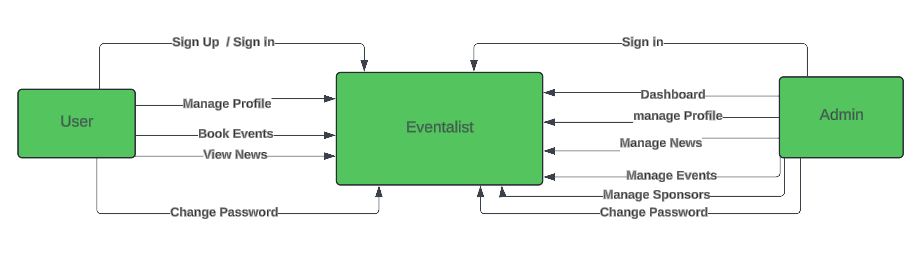


Fig. 4: Level 0 DFD

#### Level 1 DFD

The first DFD of event management web shows more details of processing. Level 1 DFD list all the major sub processes that makes the entire system.

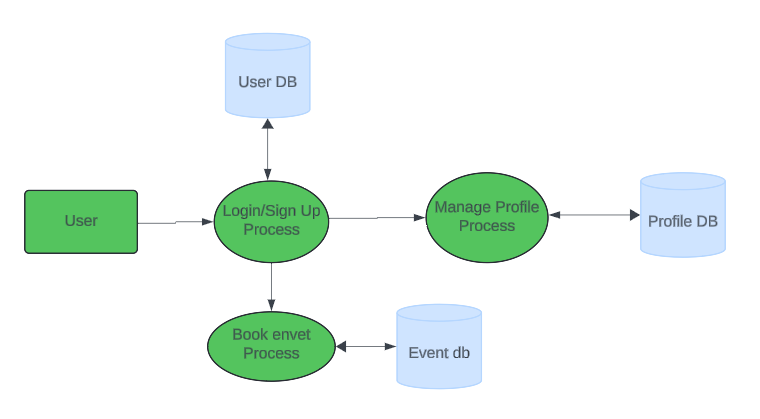


Fig. 5: Level 1 DFD

#### Level 2 DFD

Level 2 DFD for event management web dives even deeper into the concept of Level 1 DFD.

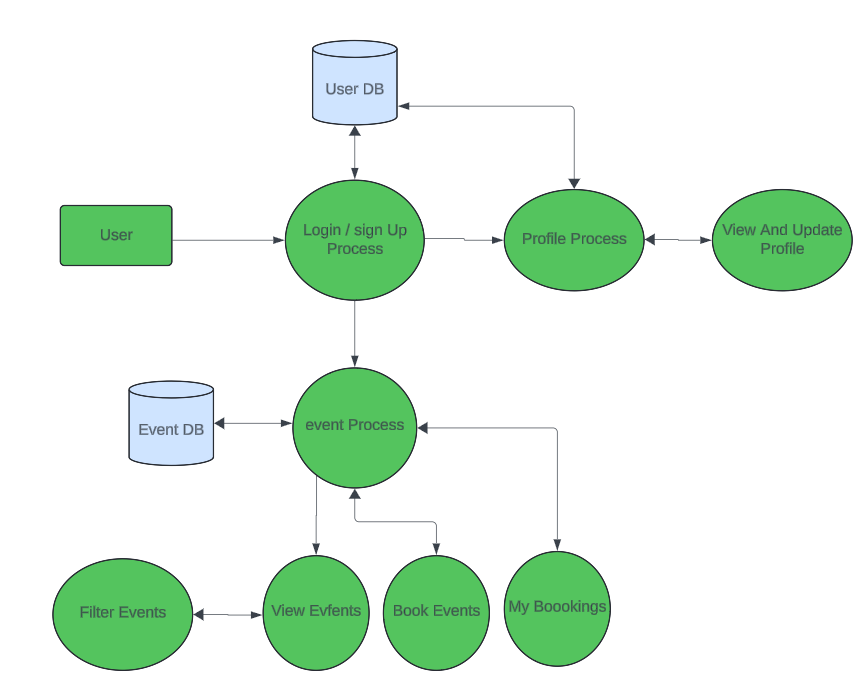
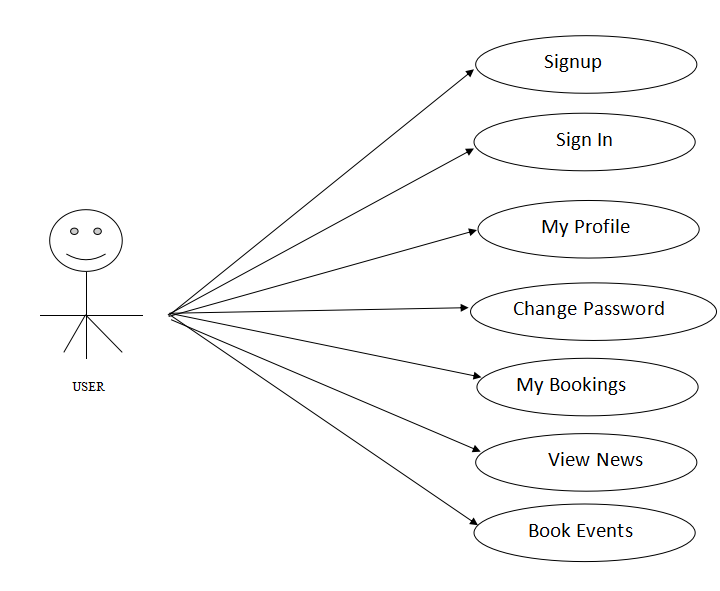


Fig. 6: Level 2 DFD

#### UML Diagrams

* + - 1. **Use Case Diagram**

A use case diagram is a type of Unified Modeling Language (UML) diagram that is used to model the interactions between a system and its actors.



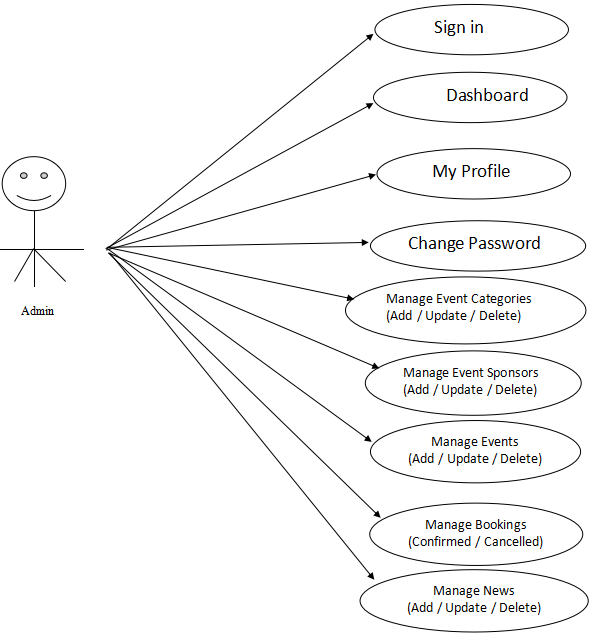


Fig. 7: Use Case Diagram

#### Sequence Diagram

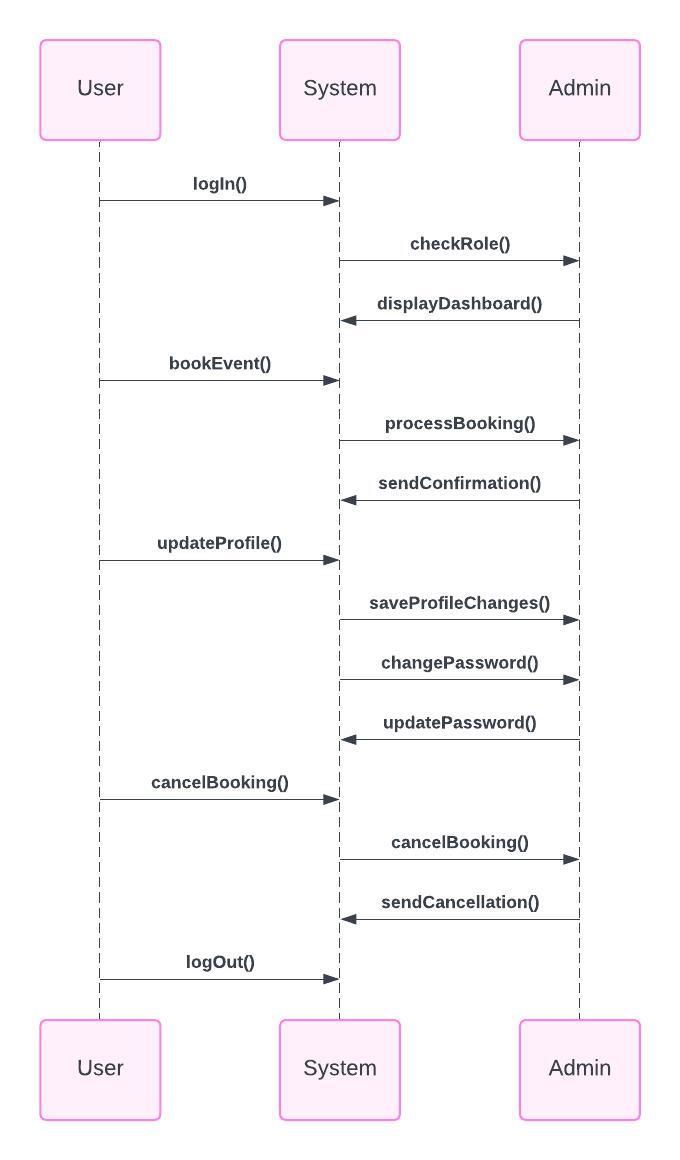


Fig. 8: Sequence Diagram

#### Activity Diagram

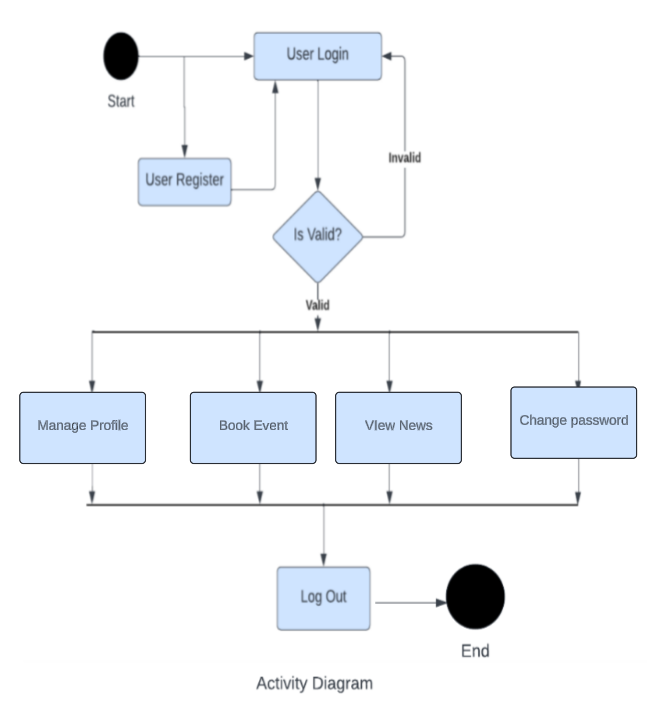
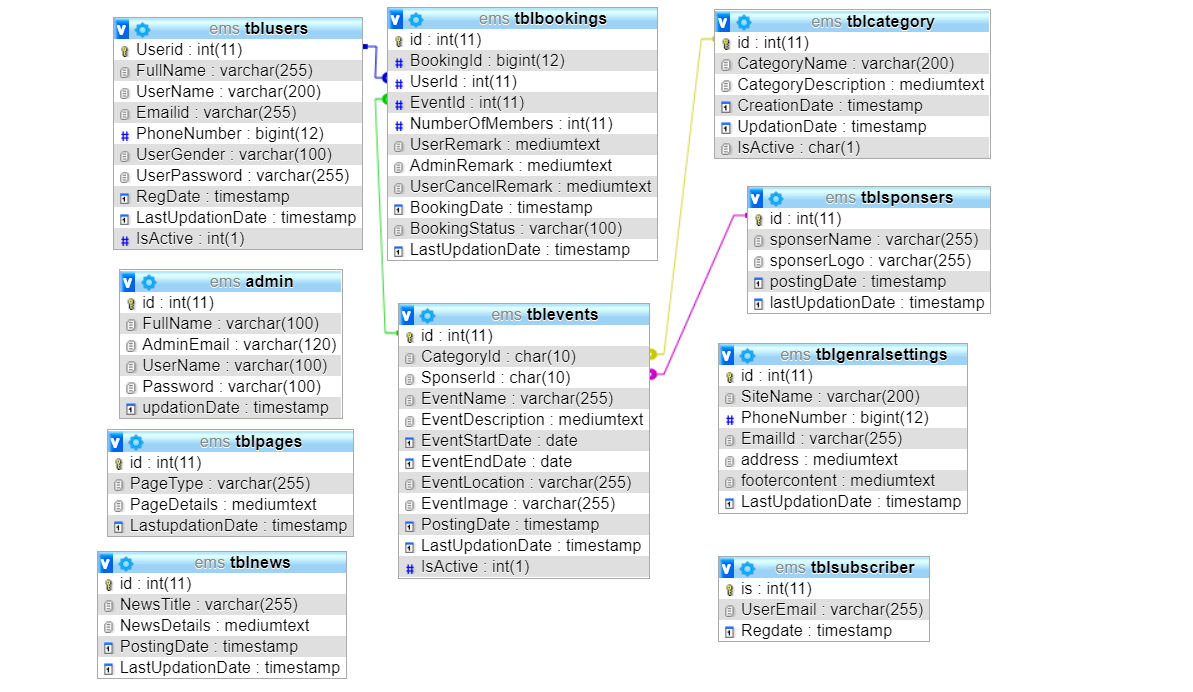


Fig. 9: Activity Diagram

#### Class Diagram

Fig. 10: Class Diagram

#### Decision Tree

A decision tree diagram is a graphical representation of a decision tree. It visually shows the flow of decisions and their possible outcomes in a hierarchical structure, making it easy to understand and interpret.

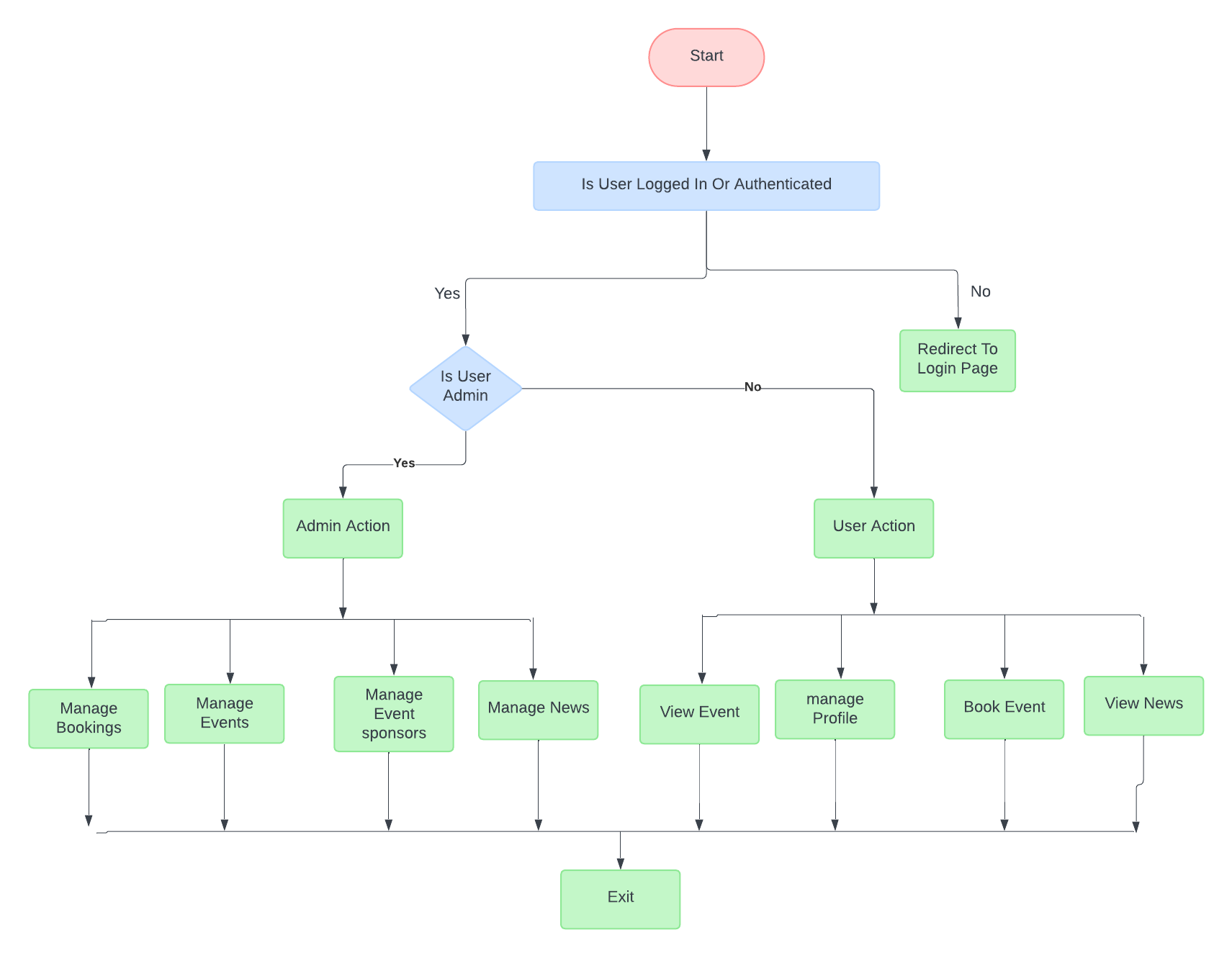
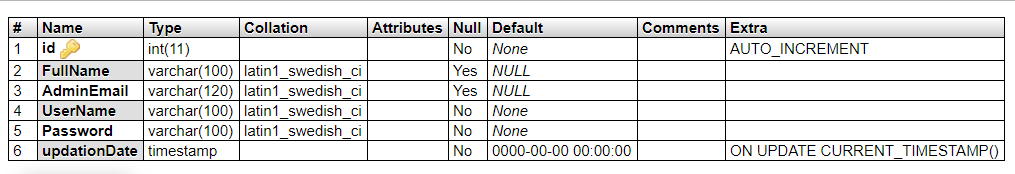
****

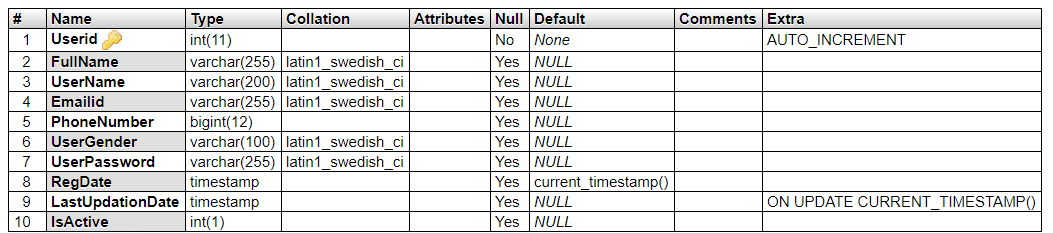
Fig. 11: Decision Tree

* + 1. **Data Dictionary**

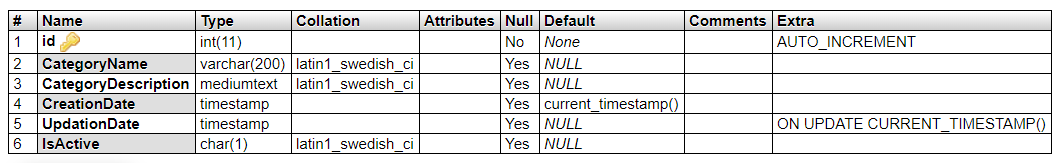
**tbladmin :** This table store the admin login detail



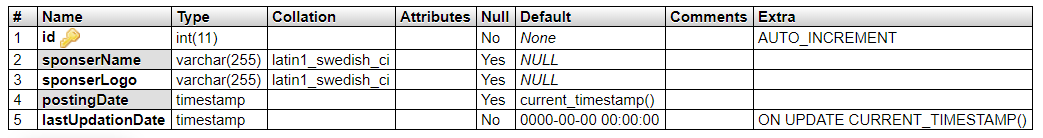
**tblusers:** This table store the user personal and login details.



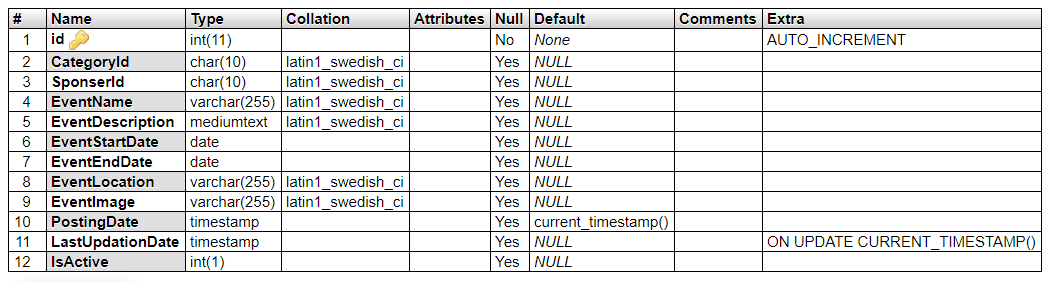
**tblcategory:** This table store the event category details.



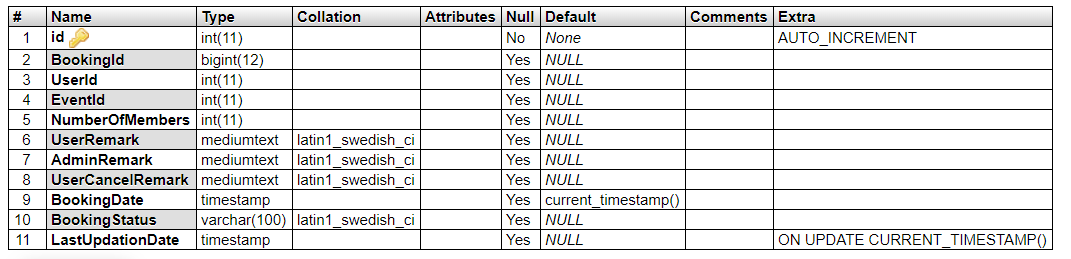
**tblsponsers:** This table store the event sponsor details.



**tblevents:** This table store the event full details



**tblbookings:** This table store the event booking details



* + 1. **Decision Table**

| **Action** | **Condition 1** | **Condition 2** | **Condition 3** | **Action 1** | **Action 2** | **Action 3** |
| --- | --- | --- | --- | --- | --- | --- |
| Admin | Logged In | Role: Admin | - | Dashboard | Manage Category | Manage Sponsors |
|  |  |  |  |  | Manage Events | Manage Users |
|  |  |  |  |  | Manage Booking | Manage News |
|  |  |  |  |  | Website Settings | Update Profile |
|  |  |  |  |  | Change Password | Recover Password |
| User | Logged In | Role: User | - | Book Event | Update Profile | Change Password |
|  |  |  |  |  | Cancel Booking | Recover Password |
| GuestUser | Any | Any | Any | View General Information | - | - |

## RESULTS AND EXPLANATION

#### Implementation Approaches

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

**TECHNOLOGIES USED**

**Programming Language**

**PHP**

* PHP stands for PHP: Hypertext Preprocessor
* PHP is a server-side scripting language, like ASP
* PHP scripts are executed on the server
* PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.)
* PHP is an open source software
* PHP is free to download and use

**MYSQL**

* MYSQL is a database server
* MYSQL is ideal for both small and large applications
* MYSQL supports standard SQL
* MYSQL compiles on a number of platforms
* MYSQL is free to download and use

**CSS**

* Cascading Style Sheets (CSS)
* Simple mechanism

**HTML**

* We have an HTML file that defines the structure of the webpage.

#### Pseudo Code

// User Actions

PROCEDURE ViewEvents

events = GET EVENTS FROM DATABASE ()

FOR EACH event IN events

DISPLAY EVENT DETAILS (event)

DISPLAY BUTTON "Book Now"

END FOR

WHILE user is interacting:

IF user clicked on "Book Now" button THEN

BOOK EVENT (event)

END IF

END WHILE

END PROCEDURE

PROCEDURE BookEvent(event)

PROMPT USER TO CONFIRM BOOKING ()

SAVE BOOKING TO DATABASE ()

DISPLAY "Booking successful!"

END PROCEDURE

PROCEDURE ViewNews

newsArticles = GET NEWS FROM DATABASE ()

FOR EACH article IN newsArticles

DISPLAY NEWS ARTICLE (article)

END FOR

END PROCEDURE

PROCEDURE SignUp

PROMPT USER TO ENTER USERNAME, PASSWORD, EMAIL

IF username is not already taken THEN

CREATE NEW USER (username, password, email)

DISPLAY "Account created successfully. Please log in."

ELSE

DISPLAY "Username is already taken. Please choose another one."

END IF

END PROCEDURE

PROCEDURE LogIn

PROMPT USER TO ENTER USERNAME AND PASSWORD

IF username and password are valid THEN

AUTHENTICATE USER (username, password)

DISPLAY "Login successful. Welcome, {username}!"

ELSE

DISPLAY "Invalid username or password. Please try again."

END IF

END PROCEDURE

// Admin Actions

PROCEDURE ManageEvents

DISPLAY EVENTS MANAGEMENT INTERFACE ()

// Options for adding, editing, or deleting events

END PROCEDURE

PROCEDURE ManageNews

DISPLAY NEWS MANAGEMENT INTERFACE ()

// Options for adding, editing, or deleting news articles

END PROCEDURE

PROCEDURE ManageSponsors

DISPLAY SPONSORS MANAGEMENT INTERFACE ()

// Options for adding, editing, or deleting sponsors

END PROCEDURE

PROCEDURE ViewBookings

DISPLAY BOOKINGS MANAGEMENT INTERFACE ()

// Display list of bookings with options to manage or view details

END PROCEDURE

PROCEDURE AdminLogIn

PROMPT ADMIN TO ENTER USERNAME AND PASSWORD

IF username and password are valid THEN

AUTHENTICATE ADMIN (username, password)

DISPLAY "Admin login successful. Welcome, {username}!"

ELSE

DISPLAY "Invalid username or password. Please try again."

END IF

END PROCEDURE

#### Testing

1. For New User Registration:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case**  **No.** | **Test case title** | **Test Data** | **Expected result** | **Actual result** | **Outcome** | **Action Required** |
| Tc-1 | Enter Username | Abc | It should accept user  name | User name accepted | Pass | No action required |
| Tc-2 | Enter valid email-id | [abc@gmail.com](mailto:abc@gmail.com) | It should accept valid  Email-id | Email  accepted | Pass | No action required |
| Tc-3 | Enter invalid email-id | [#abc@gmail.com](mailto:%23abc@gmail.com) | It should not be accepted | Not accepted | Fail | Gives alert window of Invalid Email ID |
| Tc-4 | Enter invalid password | 1234 | It should not be accepted | Not Accepted | Fail | Gives a Popup Window of Password Must be >8 |
| Tc-5 | Enter a  Valid Password | 12345678 | It should accept valid  password | Password Accepted | Pass | No Action Required |

1. For User login:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case numb**  **er** | **Test case title** | **Test data** | **Expected result** | **Actual result** | **Outcome** | **Action required** |
| Tc-1 | Enter  Invalid Username | Abc | It should not be accept | User name Not accepted | Pass | Gives alert window of  Invalid details |
| Tc-2 | Enter valid Username | Shruti123 | It should accept user  name | User name accepted | Pass | No Action Required |
| Tc-3 | Enter invalid password | 1234 | It should not be accepted | Not Accepted | Fail | Gives a Popup Window of Password Must be >8 |
| Tc-4 | Enter a  Valid Password | 12345678 | It should accept valid  password | Password Accepted | Pass | No Action Required |

## Future Scope

1. **Reduced entry work:** By automating processes and providing user-friendly interfaces, the system has reduced the manual effort required to enter and manage event information.
2. **Easy retrieval of information:** Users can easily search for and retrieve event information, leading to improved efficiency in accessing relevant data.
3. **Reduced errors due to human intervention:** Automation and standardized data entry processes have helped minimize errors that can occur with manual data entry, resulting in more accurate information.
4. **User-friendly screens to enter the data**: The system provides intuitive and user-friendly interfaces for entering event details, making it easier for users to input information accurately.
5. **Portable and flexible for further enhancement:** The system's architecture allows for easy scalability and future enhancements, ensuring that it can adapt to changing needs and requirements.
6. **Web-enabled**: Being web-enabled means that the system can be accessed from anywhere with an internet connection, providing flexibility and accessibility to users.
7. **Fast finding of information:** The system allows for quick and efficient retrieval of information, enabling users to find the event details they need promptly.

## CONCLUSION

The **“Event Management System”** was successfully designed and is tested for accuracy

and quality. During this project we have accomplished all the objectives and this project

Meets the needs of the organization. The developed will be used in searching, retrieving

And generating information for the concerned requests.

**GOALS ACHIVIED**

Reduced entry work

Easy retrieval of information

Reduced errors due to human intervention

User friendly screens to enter the data

Portable and flexible for further enhancement

Web enabled.

Fast finding of information request

## REFERENCES

1. www.w3schools.com
2. ***php****.net*
3. *en.wikipedia.org/wiki/****PHP***
4. www.hotscripts.com/category/**php**/
5. www.**apache**.org
6. www.**mysql**.com/click.php?e=35050