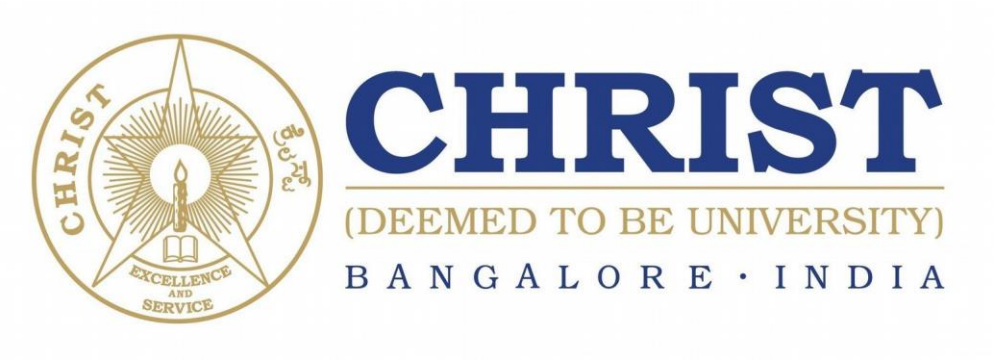


DEXTROCITY – TALENT MANAGEMENT SYSTEM

by

Jithin Thomas (2041110)

Pranav Prakash(2041124)

Under the guidance of

Dr. KN Saravanan & Dr. Vaidhehi V

A Project report submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Computer Applications of

CHRIST (Deemed to be University)

April - 2023

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled ***Dextrocity – Talent Management system*** *is a bona fide record of work done by* ***Jithin Thomas (2041110), Pranav Prakash (2041124)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of VI Semester BCA during the year 2022-23.*

Head of the Department **Faculty In charge**

Valued-by:

Name : Jithin Thomas
Pranav Prakash

Register Numbers : 2041110
2041124

Examination Centre : CHRIST (Deemed to be University)

Date : 24/04/2022

ACKNOWLEDGEMENTS

A large number of people contributed to the effective completion of the Dextrocity-Talent Management System project. First and foremost, we would like to extend our sincere gratitude to our Vice Chancellor **Dr. Fr. Joseph CC**, Head of the Department **Dr. Ashok Immanuel V**, Coordinator **Dr. Beaulah Soundarabai**, and the faculty for giving us this priceless opportunity to apply and advance our technical skills and knowledge through this project.

Perfection is unattainable, but hard effort and devotion may help you reach any objective. We did everything we could to meet the project's criteria. We are grateful to our guide teachers, **Dr. KN Saravanan** and **Dr. Vaidhehi V**, for allowing us to work under their supervision and for their unwavering support and encouragement. Our profound appreciation to our guide teachers for their enthusiastic interest and support throughout the period of our project.

We also wish to thank each and every one of our lecturers. Last but not least, we would like to thank everyone of our friends and family who, directly or indirectly, contributed to the success of the project. We are grateful for everyone's efforts.

ABSTRACT

"An artist cannot fail; it is a success to be one."- This project is dedicated to showcasing an artist's potential and providing a platform for them to demonstrate their hidden talents digitally. The idea is to come up with the concept of creating a Talent Management project where Artists may share their work of art, such as music, dancing, acting, public speaking, painting, and so on, with an audience wishing to hire an artist for a certain event.

Customers can utilize this platform to browse and compare a range of artists looking for opportunities to showcase their creativity. The project's objective is to develop an application software that will help artists increase the size of their potential audience and help customers choose the right performers for the event they are hosting.

The Talent Hunt Management System project's main goal is to manage information of artists, their projects, and their work while also giving customers an online platform to make bookings. Finding the perfect performers for the proper occasion can be solved using this internet method. The intention is to automate the tedious process of manually looking for artists in order to find a decent artist.

With the use of this website, the system can assist both customers and artists in establishing a communication link so that they may collaborate and perhaps book gigs. This website not only makes identifying good artists easier, but it also assists artists in finding new customers and gig opportunities.

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

1. INTRODUCTION

This document section will provide the overall understanding of the project idea, its uses, advantages and possible limitations. It is essential to understand the project idea before we head into the technical details.

1.1 BACKGROUND OF THE PROJECT

We aim to boost an artist's potential and provide a digital stage to showcase their skills. Our Talent Management project allows artists to display their works to an audience seeking to hire artists for events like music, dance, acting, public speaking, painting, etc. Customers can use this platform to choose from a wide range of artists for their events. The project aims to create an application that helps artists expand their audience and makes it easy for event organisers to pick the right performer. The Talent management project aims to manage artists' information, work and projects and provide customers with an online booking system. Finding the right artist for an event is now effortless through our digital solution. Our website helps customers and artists establish a communication channel for potential bookings and performances. This platform not only simplifies the search for talented artists and helps artists find new clients and performance opportunities.

1.2 OBJECTIVES

Our goal is to create a connection between artists seeking performance opportunities and customers looking for artists. Dextrocity offers a platform for less-known artists to showcase their abilities. Our platform includes filters to simplify the search for quality artists. Our online solution solves the challenge of finding the ideal performer for an event. We help both clients and artists overcome their challenges through our website. This service simplifies the search for talented artists and helps them discover new customers and performance opportunities.

1.3 PURPOSE, SCOPE AND APPLICABILITY

Purpose:

Dextrocity is addressing the high demand for entertainment jobs by providing a platform for talented entertainers and artists to showcase their skills to the world. Despite advancements in technology and social media, there need to be more systems to identify artists for events. Dextrocity aims to increase the visibility of undervalued artists and give

them a chance to earn a livelihood through their talents. The platform allows artists to share their works with a global audience and makes it easy for customers or event organisers to find the right performer for their events. Dextrocity facilitates communication between the two parties for potential bookings and shows.

Scope:

With the increasing talent and innovation among people, more artists are finding success. The challenge for young artists to earn a livelihood while creating quality content is significant. Dextrocity aims to address this issue by providing a platform for artists to showcase their work. Customers can easily compare and select from various talented performers without manual searching. This initiative also helps preserve the traditional folk artists in society.

Applicability:

The industry is open to well-known and lesser-known artists, including actors, dancers, musicians, magicians, folk artists, public speakers, and others. After their work has been verified, artists can register for an account on the website and post it there. Customers interested in a particular category of artists can register on the website. The website will act as the two parties' communication channels.

1.4 OVERVIEW OF THE REPORT

The report for this project will outline the website's requirements to provide a platform for artists and customers to interact and conduct events efficiently. The report will cover the functional and non-functional requirements for the website, including its purpose, scope, constraints, assumptions, and interfaces. It will also detail the user requirements, such as user authentication, event management, payment processing, and other vital features required to fulfil the website's purpose. The SRS report will serve as a crucial document providing a shared understanding between the development team, stakeholders, and customers about the software's functionalities, capabilities, and limitations.

2. SYSTEM ANALYSIS AND REQUIREMENTS

The analysis and requirements to design the talent management project have been specified inthe following section.

2.1 EXISTING SYSTEM

Customers cannot search for an artist on a digital platform under the current system. Finding the appropriate kind of artist that can deliver high-quality performances in a short amount of time is challenging. On the other hand, artists are limited to using social media platforms such as Facebook, Instagram, and others to distribute their work. Although social media gives content creators a forum to show off their work, there needs to be a comprehensive structure that allows customers to connect with different artists and compare them.

2.2 LIMITATIONS OF THE EXISTING SYSTEM

Artists with an immense potential struggle to build a name for themselves and are frequently obliged to abandon their talents to find work. New contacts and gig chances are difficult to come by for them. Customers looking for artists for an event frequently have to rely on manual searches to find a suitable artist. We hope that by developing this software, we will be able to address both of these concerns by serving as a communication link between them. This website will be a breeding ground for a diverse group of artists looking to display their work and earn gigpossibilities. This will benefit them both financially and in terms of expanding their reach. Digitizing the entire system makes it easy for clients/event organizers to find the right performer.

2.3 PROPOSED SYSTEM

By addressing the issues with the current applications, we decided to provide a new system for the entertainment industry where artists can display their talents and customers can organize theirevents effectively.

- Dextrocity will be a web application that allows artists to directly market and monetize their talents while allowing customers to book them for specific events.
- It is a time-saving tool with a very user-friendly UI that even a layperson can use.
- The customer can upload events for which they require artists, and the artists can register for these events.

2.4 BENEFITS OF THE PROPOSED SYSTEM

- Ease of use for artists and customers: The platform provides a single point of access forartists and customers to perform and conduct events.
- Increased efficiency: The website streamlines organizing and conducting events, reducingthe time and effort required.
- Improved customer experience: Customers can easily find and book artists for theirevents, increasing the chances of a successful outcome.
- Increased visibility for artists: The website provides a platform for artists to showcasetheir work and reach a wider audience.
- Increased revenue: By providing a platform for artists and customers to interact, thewebsite has the potential to generate revenue for both parties.
- Better collaboration: The platform facilitates communication between artists andcustomers, leading to better collaboration and improved event outcomes.
- Competitive advantage: The platform provides a unique solution in the market, giving thecompany a competitive advantage.

2.5 FEATURES OF THE PROPOSED SYSTEM

- The Artists can publish their content for better reach and grow their audience.
- Customers can find Artists of various kinds and book them for shows.
- Customers can publish their events to get more artist entries.
- Artists can look up events and register if they are interested.
- Users can avail of offers depending on their interaction on the website.
- Customers can like artists to save them for future events.
- A premium membership will provide additional features and better visibility on thewebsite.
- Customized email- The users can opt for email notifications if they want to be notifiedabout some new artists or the events organized.
- Real-time chatting and interaction between artists and customers.
- Realtime User support to handle any technical or non-technical issues.
- FAQs- Frequently asked questions will be answered.

- User authentication: The website will have a secure user authentication system to ensurethat only authorized users can access the platform.
- Event management: Customers can create and manage events on the platform, whileartists can search for and apply to perform at events.
- Payment processing: The website will have a secure payment processing system to enablecustomers to pay for events and artists to receive payment for their performances.
- Artist profile: Artists can create a profile on the platform to showcase their work,experience, and skills.
- Search and filtering: Customers can search for and filter artists based on specific criteria,such as location, genre, and experience.
- User reviews and ratings: Customers can leave reviews and ratings for artists after events,providing valuable feedback for both artists and customers.
- Customer support: The website will have a customer support system to handle customerinquiries and resolve any issues that may arise.
- Analytics: The platform will have a dashboard to provide insights into customer and artist behavior, allowing the company to make informed decisions.

2.6 SYSTEM REQUIREMENTS SPECIFICATION

As an outcome, the talent management system will allow customers to locate artists digitally rather than manually or through word-of-mouth advertising. The software allows artists topromote their work and earn money from concerts and performances. An explanation of the system requirements needed to implement the proposed features is given in the following section.

2.6.1 User Characteristics

- The Artists will use this platform to publish their original works, boost their visibility,and expand their opportunities to profit from their skills.
- Customers who sign up on this platform will have the option of comparing and selecting from various artists to meet their need for choosing an artist for any event they are organizing.

- The admins of this software will oversee the overall functioning of the website, handle artist and customer issues, check for discrepancies, provide technical support, etc.

2.6.2 Software And Hardware Requirements

Table 2.1: Software Requirements for Dextrocity

Component	Minimum	Recommended
Web Browser	Google Chrome 69.0 or Mozilla Firefox 71.0	Google Chrome 88.0 or Mozilla Firefox 84.0
HTML	HTML 5	HTML 5.2
CSS	CSS 3	CSS 3
JavaScript	ECMAScript 2016	ECMAScript 2018
PHP	PHP 6	PHP 8
MySQL	Server 5.6	Server 8.0
Server	LAMP 0.0.10 or WAMP 2.5	LAMP 0.0.12 or WAMP 3.0.3
Python	Python 3.8	Python 3.11

Table 2.2: Hardware Requirements for Dextrocity

Component	Minimum	Recommended
Processor	1.9 gigahertz (GHz) x86- or x64-bit dual-core processor	3.3 gigahertz (GHz) or faster 64-bit dual-core processor
Memory	2- GB RAM	4 - GB RAM
Storage	128 GB	512 GB

Display	Super VGA with a resolution of 1024 x 768	Super VGA with a resolution of 1024x 768 or more
Operating System	Windows 7 or Ubuntu 14.04 or OS X El Capitan 10.11	Windows 11 or Ubuntu 18.04 or Catalina 10.15.7
Python	Python 3.8	Python 3.11

2.6.3 Constraints

- The technology needed to develop this project can be complex, and the developmentteam may need to adapt quickly to new tools and technologies as they emerge.
- Integrating the platform with other systems, such as payment and social media, can be challenging and require significant testing and development effort.
- Protecting sensitive customer and artist data is a critical concern in projects like this. Thedevelopment team will need to take appropriate measures to ensure that data is secure andprotected from unauthorized access.
- Ensuring that artists and customers can effectively use the platform is a critical success factor for the project.

2.6.4 Functional Requirements

Table 2.3: Functional Requirements for Dextrocity

Requirement ID	Requirement	Description
FR_1	Access Mode	The Artists, Customers, Admins and Super Admin will be able to access the system with dedicated privilege sets. Each system usershould have their own set of functions anddedicated interface.

FR_2	Blog Posting	The Artists should be able to publish their blogs as and when required, and the Customers should be able to view them withoutconsiderable delay
FR_3	Events Publishing	The Customers should be able to publish events. Artists must be able to register for them.
FR_3	Chat Compatibility	The chatting between Artists and Customers should be kept secure. It should happen seamlessly.
FR_4	User Support	The admins should be able to provide real timesupport to any of the user issues and provide adequate solutions.
FR_5	Payments	The Users must be able to make payments through the payment gateway without any breaches.

2.6.5 Non-Functional Requirements

Table 2.4: Non- Functional Requirements

Requirement ID	Requirement	Description
NF_R1	Authenticity	The Artists and Clients registered with the website must be real people and not be using a false identity to engage in fraudulent activities.
NF_R2	Originality	Users must own any content published on the website; they cannot copy it from another source.

NF_R3	Usability	The released material shouldn't be deceptive and should be used to establish a line of communication between artists and customers.

2.7 BLOCK DIAGRAM

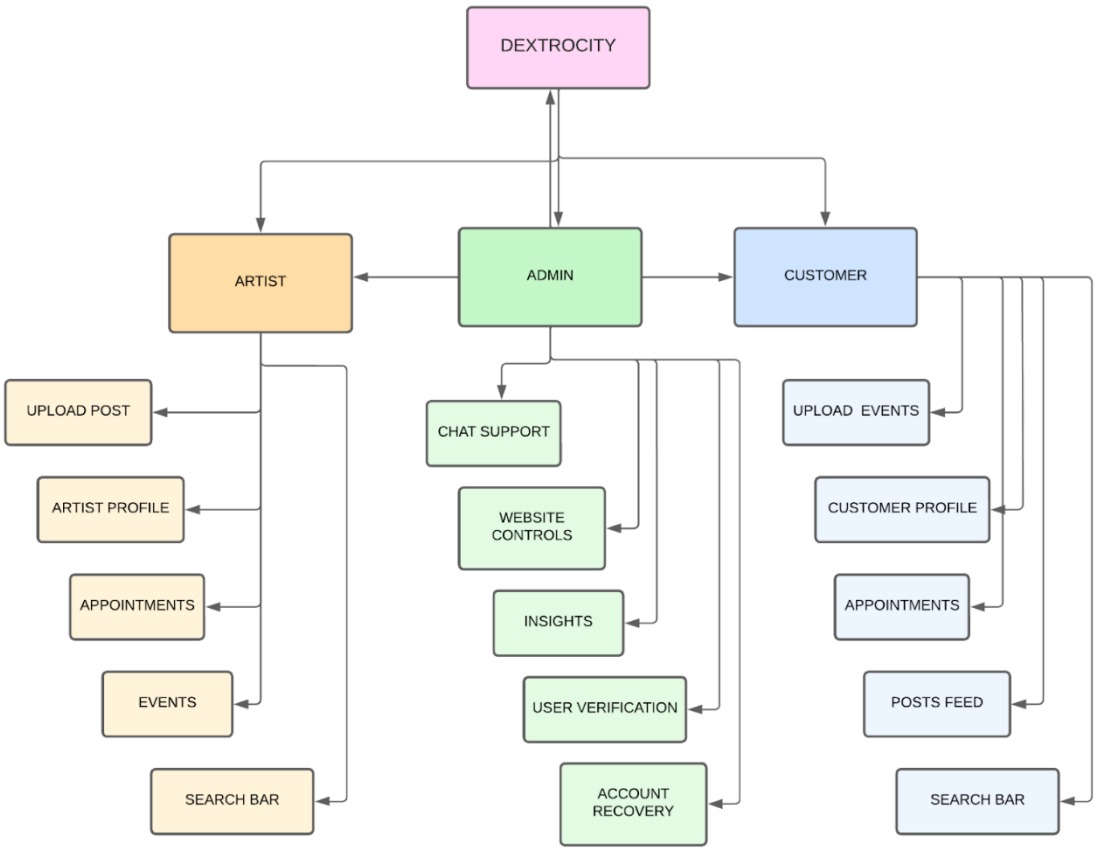


Fig 2.1: Block Diagram for Dextrocity

This block diagram represents the high-level control flow of the Dextrocity system. The homepage serves as the entry point for users, where they can then proceed to log in to their respective modules. The system is designed to provide various features to artists and customers, and the features for both users are interconnected. Changes made to the features of one user will impact the other user. The system is built to ensure that all updates and changes made by users are accurately recorded and reflected in the database, ensuring seamless information exchange and communication between artists and customers.

3. SYSTEM DESIGN

The design document for Dextrocity outlines the architecture and functionality of the platform, which aims to connect artists and customers in a seamless and efficient manner. This document provides an overview of the features, user interfaces, technologies, and workflows that will be incorporated into the system. It describes the various components of the system and their interactions, as well as the development, deployment, and maintenance of the platform.

3.1 SYSTEM ARCHITECTURE

The system architecture is a critical aspect of any software project, and the same holds for Dextrocity. The architecture serves as the foundation for the design, development, and deployment of the application. The system architecture will determine the overall performance, scalability, and security of the application, which are critical for the success of Dextrocity.

3.1.1 System Perspective

The system perspective in the context of the Dextrocity project involves defining the high-level architecture and components of the system, as well as the relationships and interactions between them. This includes the hardware and software infrastructure, the databases and data flow, the user interface and experience, and the various services and APIs that enable functionality. By understanding the system perspective, we can ensure that the project is built on a solid foundation that is scalable, reliable, and easy to maintain.

3.1.2 Architecture of Dextrocity

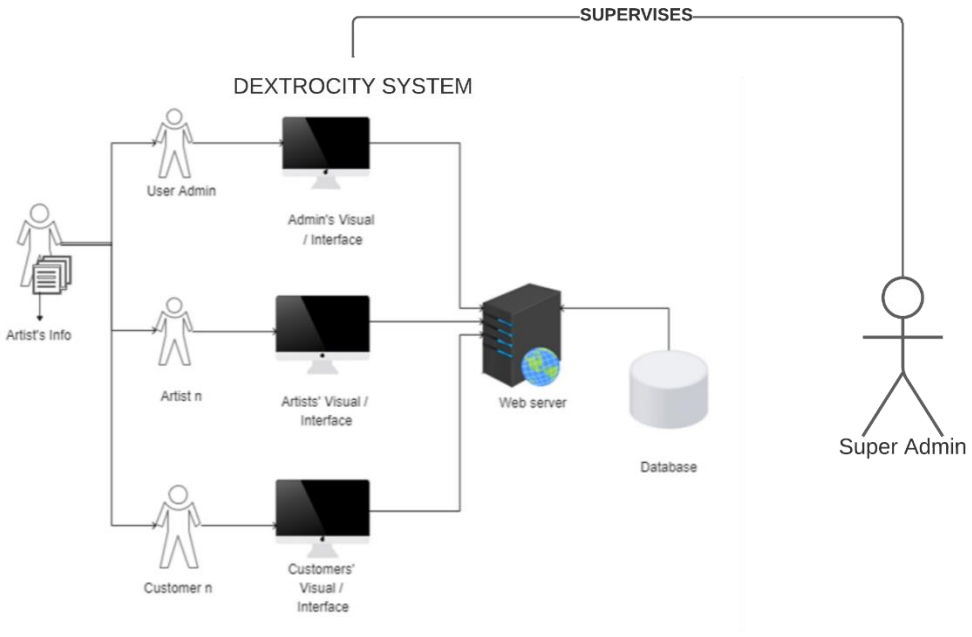


Fig 3.1: User Engagement System Architecture

The system perspective of the project Dextrocity revolves around the architecture depicted in the figure. The system primarily serves four types of users- Artists, Customers, Admins and Super Admin. The web server interacts with the database to fetch data and display it in a user-friendly manner according to the user's perspective. All the information presented to the users is fetched from the database. The users are also allowed to add new entries into the database through various actions within the system, such as posting and booking. This design ensures that the system is scalable and flexible enough to accommodate new features and user requirements in the future.

3.2 MODULE DESIGN

1. Admin Module: This paragraph describes the admin module of the system, which is managed by the system administrator. The admin module comprises four main categories: Login, Registration, Bookings, and Help Desk. The administrator is responsible for system administration work, ensuring the overall system functioning and security. They have the authority to control access to the website data, allowing only authorized personnel to access it and restricting unauthorized access.

2. Artist Module: The artist module is designed specifically for artists to showcase their talent and interact with customers. It includes features such as My Profile, Artist Registration, Client User feed, Search/filter bar, Events (Application). Artists can upload new content, apply for new events, and rate customers. This module is intended to enhance the artist's customer reach and provide them with the necessary tools to manage their online presence.

3. Customer Module: The customer module of the system is designed for the customers and includes several features such as the home page, client registration, search/filter bar, events (uploading), artist user feed, and bookings. Customers can interact with artists through this module, and can book artists for events, publish new events, and provide feedback on their previous experiences with an artist. This module is an essential component of the system, as it enables customers to discover and engage with artists, facilitating the growth of the platform.

4. Super Admin Module: The Super Admin module is the highest level of control in the Dextrocity system. It has access to all the features and functionalities of the website, and can manage all the data and information entered by the users. This module includes features like User Management, Artist Management, Event Management, and Content Management. The Super Admin module also has the ability to manage and control the access of other admins and users. This module is essential for maintaining the overall functioning and security of the Dextrocity system.

3.3 DATA FLOW DIAGRAMS

3.3.1 Level 0 DFD

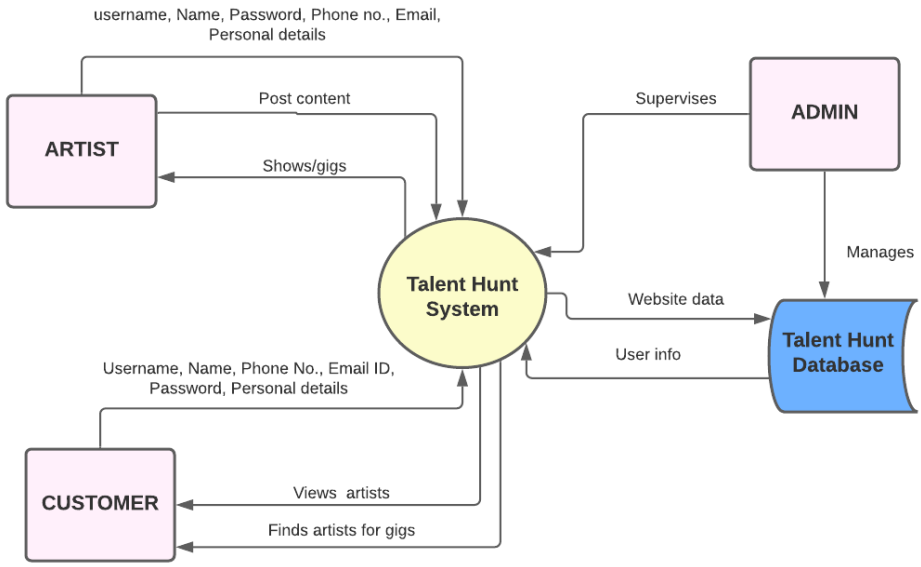


Fig 3.3: Level 0 DFD of Dextrocity

This level 0 DFD shows the overall flow of the system and how different entities interact with it. The system has two main entities, Artists and Customers, who use the Talent Management System to communicate with each other. The data related to Artists and Customers is stored in a shared database which acts as a central repository for all the data in the system. The system processes data inputs from the Artists and Customers to perform various functions like artist recommendation, bookings, and payment processing. The outputs of the system include various reports and updates to the Artists and Customers regarding their profiles and bookings. The level 0 DFD provides a high-level view of the system and is useful in understanding the overall functioning of the software.

3.3.2 Level 1 DFD

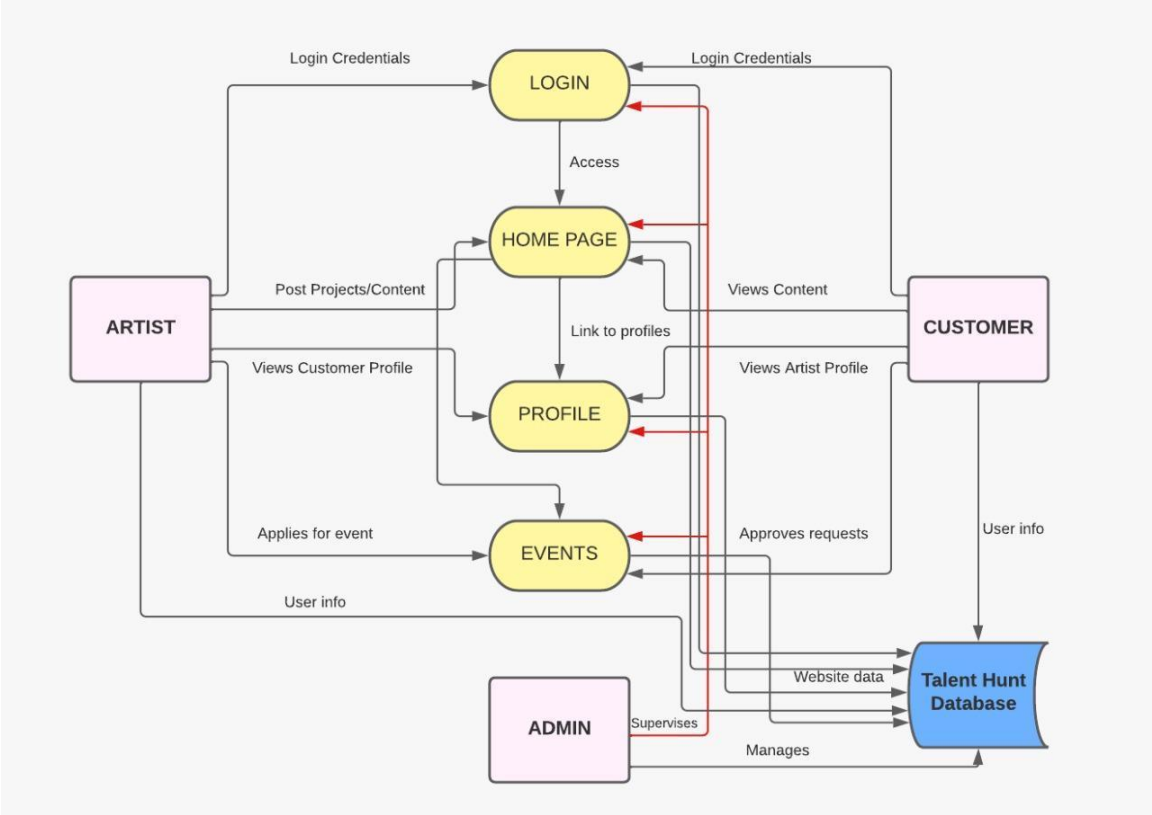


Fig 3.4: Level 1 DFD of Dextrocity

The Level 1 DFD provides a detailed view of the system's data flow between the modules and their features. The two main modules, Artists and Customers, are further divided into four categories - Login, Home Page, Profile, and Events. The Login feature allows users to log in to their respective modules, while the Home Page displays relevant information based on the user's role. The Profile section lets users view and edit their personal details, and the Events feature displays upcoming events related to the user's interests. The Level 1 DFD also shows that all data is stored in a shared database, enabling both modules to interact with each other seamlessly.

3.3.3 Level 2 DFD

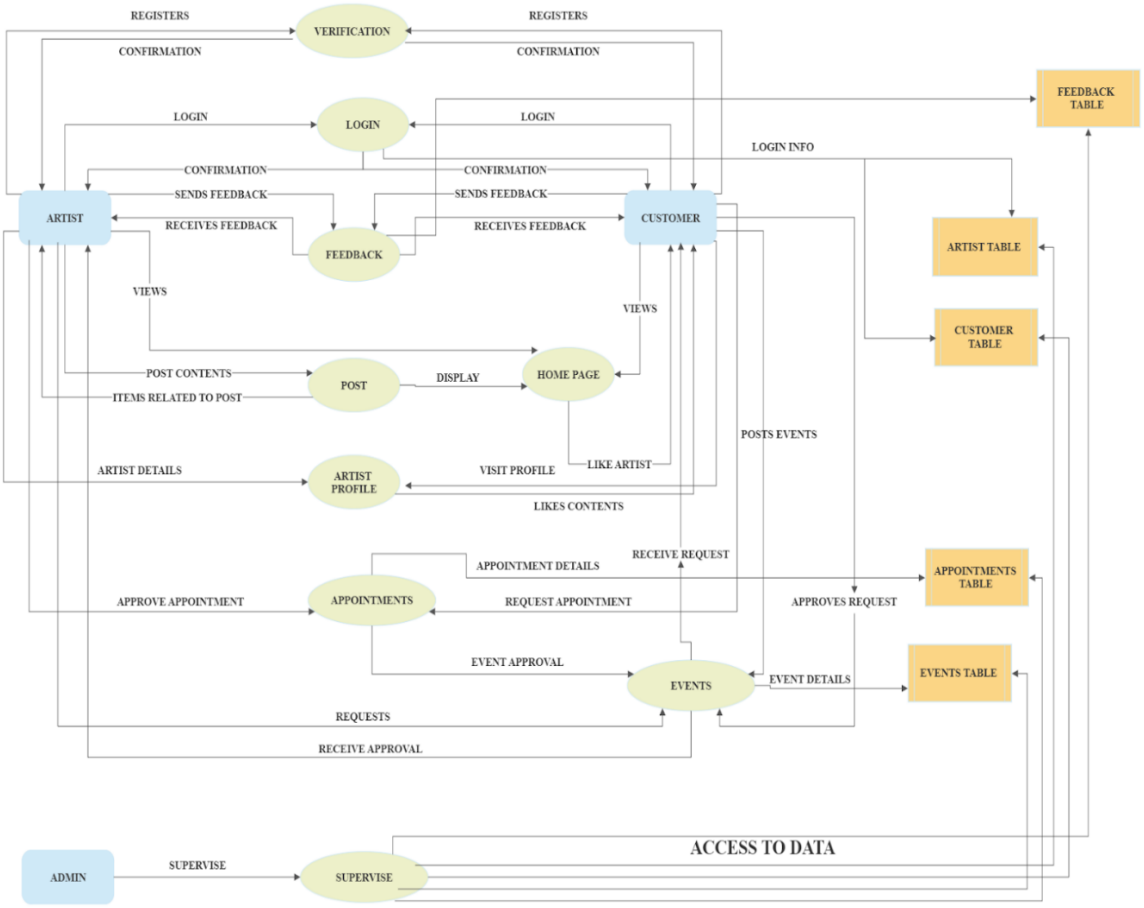


Fig 3.5: Level 2 DFD of Dextrocity

The level 2 DFD provides a more detailed view of the flow of data within the system. It shows how data is being utilized in different sections of the website and how it is processed and stored in a central database that is accessed by all the modules of the website. Each module is broken down into sub-processes, and the data flow between these sub-processes is clearly defined. This diagram helps to identify potential bottlenecks or areas of improvement in the system and is useful in ensuring that all the modules are working together seamlessly to provide a cohesive user experience.

3.4 ER DIAGRAM

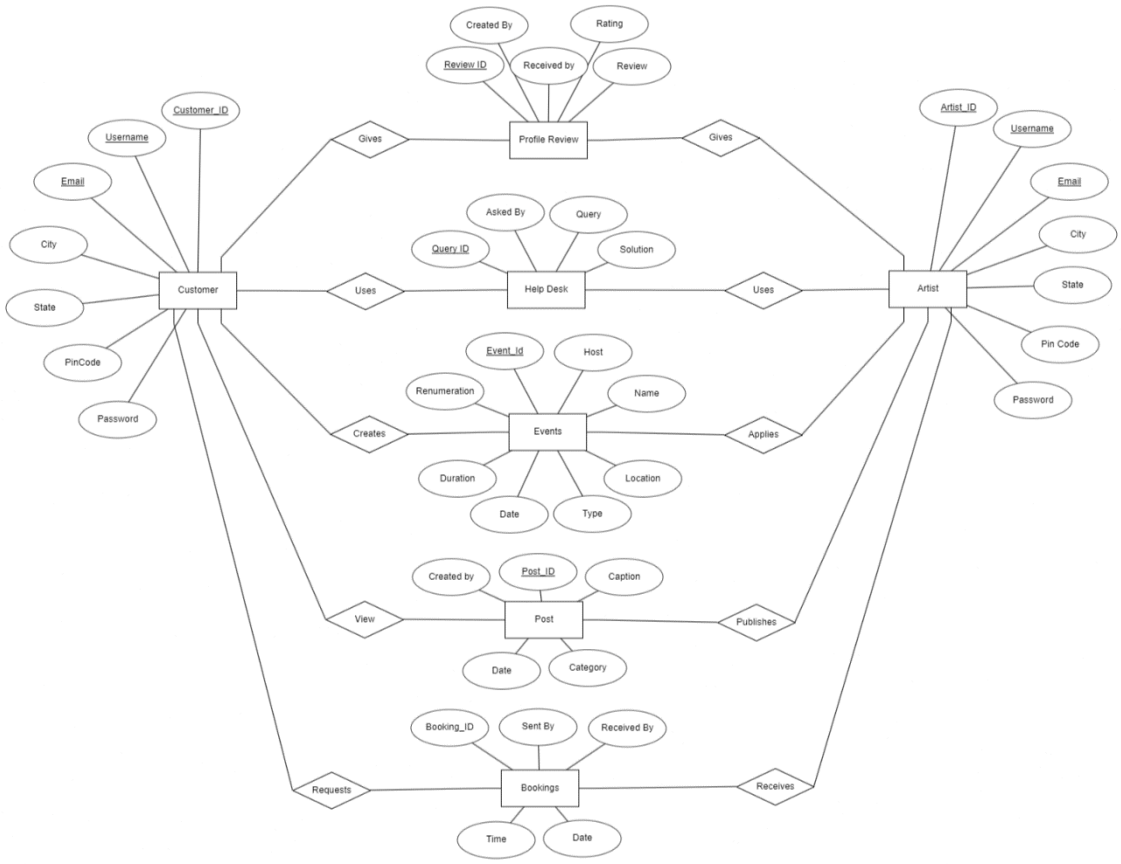


Fig 3.6: ER Diagram of Dextrocity

The ER diagram for this project showcases the relationships and entities involved in the Talent Management System. The entities include artists, customers, events, bookings, reviews, and help desk. The relationships between these entities are depicted in the diagram, such as the one-to-many relationship between customers and bookings, or the many-to-many relationship between artists and events. Overall, the ER diagram helps to visualize the structure of the project's database and the connections between its different components.

3.5 DATABASE DESIGN

Ensuring proper normalization of a database is critical for maintaining data consistency, avoiding redundancy, and upholding the ACID properties. With the Dextrocity project's database stored on AWS cloud servers, it can leverage the security and reliability of AWS infrastructure while also providing scalability as the project expands. Cloud storage also enables easy data access from anywhere, enabling efficient data management and analysis.

3.5.1 Proposed Database for Dextrocity

Table Name: New Artists

Field Name	**Data Type**	**Field Size for Display**	**Description**	**Constraint**
Artist_ID	integer	5	Unique ID of each artist	Primary Key
Name	varchar	40	Name of the artist	Not null
Username	varchar	40	Unique name of each artist	Not null
Phone_No	Long integer	10	Phone number of the artist	Unique key, not null
Email	varchar	30	E-mail of the artist	Not null
State	varchar	20	Location of the artist	Not null
Password	password	25	Password of the artist	Not null
Category	varchar	20	Artists' area of Expertise	Not Null

Table Name: New Customers

Field Name	**Data Type**	**Field Size for Display**	**Description**	**Constraint**
Customer_ID	integer	5	Unique id of each customer	Primary Key
Name	varchar	40	Name of the customer	Not null
Username	varchar	40	Unique name of each customer	Not null, Unique key
Phone_No	Long integer	10	Phone number of the customer	Unique key, not null
Email	varchar	30	E-mail of the customer	Not null
State	varchar	20	Location of the customer	Not null
Password	password	25	Password of the artist	Not null

Table Name: Blogs

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Blog_Id	integer	5	Unique id of each post	Primary key
Artist_username	varchar	40	Unique name of the artist	Foreign key from new artist
Media_URL	varchar	200	URL of the post	Not null
Caption	varchar	200	Description about the post	Not null

Location	varchar	100	Location of the artist/post	Not null

Table Name: Comments

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Blog_Id	integer	5	Unique id of each post	Foreign key from blog
Username	varchar	40	Unique name of each customer	Foreign key from new customer
Comment	varchar	100	Comment of each post	Not null

Table Name: Event Details

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Event_Id	integer	5	Unique id of each event	Primary key
Posted By	varchar	40	Username of the artist	Foreign key from the artist table
Event name	varchar	40	Name of the event	Not null
Location	varchar	30	Location of the event	Not null
Duration	integer	4	Duration of the event	Not null
Type	varchar	30	Type of the artist for the event	Not null
Date	date		Date of the event	Not null

Requirements	varchar	30	Requirements for conducting the event	Not null
Pay Range	float	6-8	Pay range for the event	Not null

Table Name: Applied Artists

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Artist_ID	integer	5	Unique id of each artist	Foreign key from artist table
Event_Id	Integer	5	Unique id of each event	Foreign key from event details
Role	varchar	30	Details of the event they are doing	Not null
Date	date		Date of the event	Not null
Status	Varchar2	20	Status of the artist for the event	Not null

Table name: Bookings

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Appointment_ID	integer	5	Unique id of each appointment	Primary key
Artist_ID	integer	5	Unique id of each artist	Foreign key from artist table

Customer_ID	integer	5	Unique id of each customer	Foreign key from customer table
Date	date		Date of the appointment	Not null
Time	Time		Time of the appointment	Not null
Mode_of_communication	varchar	30	Communication with the customer	Not null

Table Name: Admins

Field Name	**Data Type**	**Field Size of Display**	**Description**	**Constraint**
Admin_ID	integer	5	Unique id of each admin	Not Null
Username	varchar	25	Unique username of Admin	Primary Key
Role	varchar	50	Role of the admin	Not Null
Privileges	varchar	100	Privileges given to the admin	Not Null

3.5.2 Data Integrity and Constraints

Data integrity and constraints are crucial for the proper functioning of the Dextrocity project. To ensure data integrity, the database is designed to adhere to the 3-N normalization rules, which prevents data redundancy and ensures consistency across the database. Additionally, various constraints are put in place to ensure that the data entered by users is accurate and adheres to specific rules. For example, constraints such as data type, data format, and data range are implemented to ensure that the data entered is valid and meets the necessary requirements. Moreover, data security is also taken into

consideration to prevent unauthorized access and to ensure that user data is not compromised. Access controls and authentication mechanisms are put in place to restrict access to sensitive data and ensure that only authorized users can access the data. Overall, maintaining data integrity and constraints is vital for the success of the Dextrocity project and to provide a seamless and secure user experience.

3.6 INTERFACE DESIGN AND PROCEDURAL DESIGN

In the context of the Dextrocity project, Interface Design refers to the design of the user interface of the web application. It includes the layout, colour schemes, fonts, and other visual elements that make up the overall look and feel of the application. The interface design should be user-friendly, intuitive, and aesthetically pleasing to ensure that users enjoy using the application and can easily navigate through its different features. On the other hand, Procedural Design refers to the design of the software application's functionality and operations. It includes the design of the system architecture, the flow of data between different modules, and the development of algorithms and processes to perform various tasks within the application. The procedural design should be optimized for efficiency and accuracy, ensuring that the application can perform its functions effectively and without errors.

3.6.1 Proposed User Interface

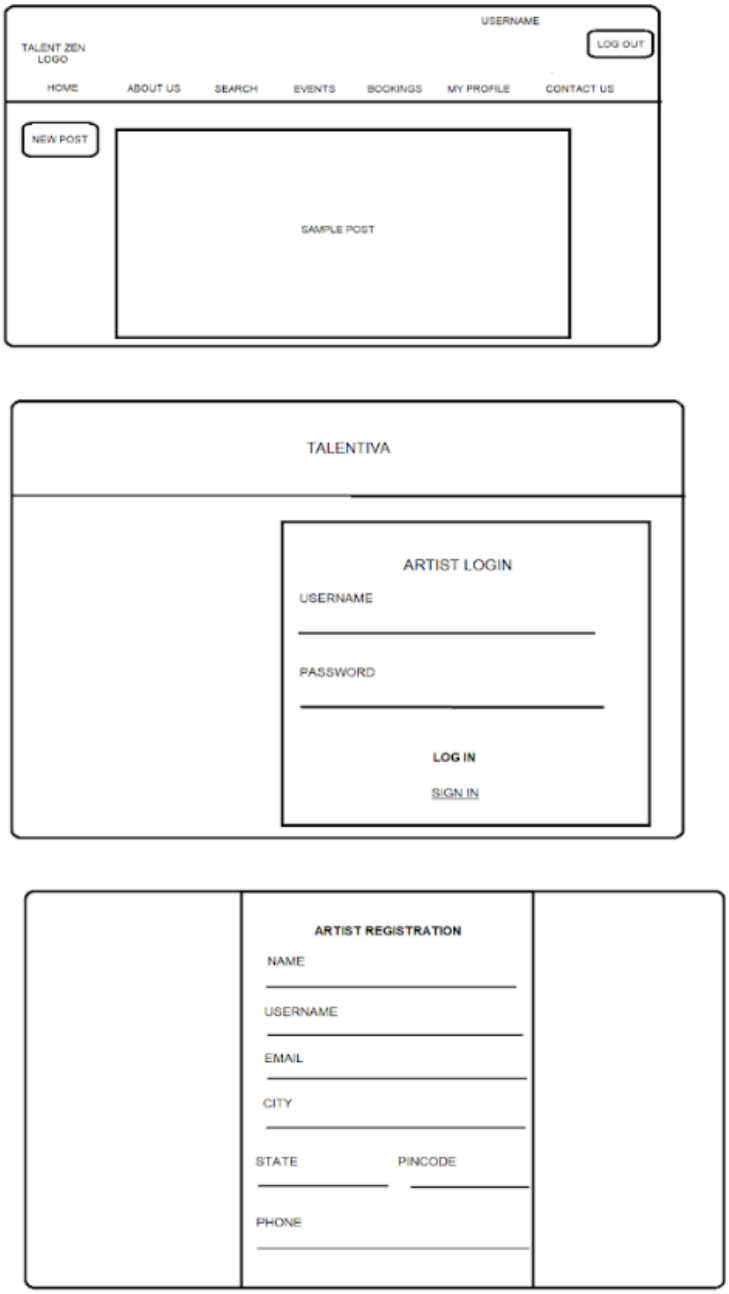


Fig 3.7: User Interface Design of Dextrocity

The interface design of the Dextrocity project will have a separate module for both the Customer and the Artist with slight differences in the layout of the pages. The registration and login pages will look almost the same for both types of users with some minor

differences. The home page for artists will allow them to post and manage their content, while the home page for customers will allow them to upload and manage events. The procedural design will focus on creating clear and concise steps for users to follow while navigating through the system. Proper validation checks will be implemented at each step to ensure the smooth functioning of the system

3.6.2 Proposed Process Logic of Modules

Dextrocity is a specialized platform that aims to connect customers with highly skilled and trained artists for their upcoming events. The website provides a user-friendly interface for artists and customers to interact and streamline the hiring process.

To start using the platform, artists must first register by providing their personal information, qualifications, and experience. They are also required to submit a sample project for verification. The administrator will then review the project's integrity and verify the artist's credentials. Once confirmed, the artist can upload their work to their profile and showcase their talent to potential clients.

Similarly, customers must register by providing basic information and selecting their preferred field of interest. The AI system then tailors the content in the feed to the customer's preferences. Customers can use the search box and various filters to sort through the artist data and find the perfect candidate for their event. Additionally, the AI system presents comparable results to the artists searched by the customer, assisting in finding the right match.

Customers can view an artist's profile, see their work, and schedule an appointment with them. Once a customer requests an appointment, the artist will receive a notification, and the appointment will be confirmed once approved by the artist. The platform also provides a tool for customers to post events in the events section, complete with all relevant information. Artists that are eligible and interested in participating in the event will have the opportunity to apply. The customer will have the final decision on whether or not to approve the request.

Feedback is an essential component of the platform. Both customers and artists can leave reviews on each other's profiles based on their interactions. This feedback is crucial for

maintaining the integrity of the platform, as it ensures that artists and customers receive the best possible experience.

In addition, there is an automatic help desk that customers and artists can access in the event of any website-related questions. It is mandatory for both parties to log in to the website each time to use the services provided by the platform. Admins are the super users who oversee the proper functioning of the website. They have the authority to restrict actions if found to be violating the website's policies, ensuring a safe and secure environment for everyone.

Overall, Dextrocity provides a one-stop-shop for customers to connect with talented artists and ensures a hassle-free and streamlined hiring process.

3.6.3 Actors and Use Cases

Actors:

1. Customer: a person who is looking to hire an artist for an upcoming event.
2. Artist: a person who has registered on the website and has a profile showcasing their work.
3. Admin: a superuser who oversees the functioning of the website and ensures that policies are being followed.
4. Super Admin: One who has supreme control over the website. He is responsible for managing admins and the overall working of the system.

Possible Use Cases:

1. Register as an Artist: an Artist can register on the website by providing personal information, qualifications, experience, and a sample project for verification.
2. Upload Content: once an Artist has been verified, they can upload their work to their profile page.
3. Search for Artists: a customer can search for Artists on the website using various filters to find the best match for their event.
4. Request Appointment: a customer can request an appointment with an Artist by selecting their profile and making a request.
5. Post an Event: a customer can post an event on the website, complete with all relevant information, and eligible Artists can apply to participate.

6. Leave Feedback: both Customers and Artists can leave feedback/reviews on their profiles about their interactions with the other party.
7. Admin Monitoring: the admin oversees the functioning of the website and ensures that policies are being followed. They have the authority to restrict actions if found to be violating the policies of the website.
8. View Artist Profile: A user can view an artist's profile, which contains information such as their portfolio, experience, and qualifications.
9. Search for Artist: A user can search for an artist based on various criteria such as art style, location, and availability.
10. Book Artist: A user can book an artist for their event by submitting a request and providing details such as event date, location, and required art services.
11. Apply for Event: An artist can apply to participate in an event posted by a user, providing details such as their availability and cost.
12. Confirm Booking: A user can confirm a booking request from an artist and proceed to finalize the contract and payment details.
13. Leave Review: A user or artist can leave a review on the other party's profile to share their experience and provide feedback.
14. Contact Support: A user or artist can contact customer support through the help desk for any website-related questions or issues.
15. Edit Profile: An artist or user can edit their profile information such as contact details, portfolio, and availability.
16. Admin Approvals: The website administrator can approve or reject artist registration requests, project submissions, and event postings to ensure the integrity of the platform.
17. AI Recommendations: The AI system can provide personalized recommendations to users based on their interests, search history, and previous interactions on the website.

3.7 REPORTS DESIGN

The reports generated by the Dextrocity website will provide valuable insights for both the artists and customers. For the artists, the reports will include details about the number of bookings, types of events, and customer feedback. This information will help them understand their audience better and improve their skills accordingly.

For the customers, the reports will provide an overview of their event bookings, including the artists they have booked, the date and time of the event, and any special requests they have made. This information will help them keep track of their events and ensure that everything runs smoothly.

The reports will be available in various formats, such as PDF and Excel, and can be customized to include specific details based on the user's preferences. The reports can be accessed via the website dashboard and can also be sent to the user's email address. Overall, the reports generated by the Dextrocity website will provide valuable insights and help users make informed decisions.

4. IMPLEMENTATION

Extensive research was conducted to implement this project, including a thorough analysis of search engine analytics, competitor activity, and current trends in the field. Despite this effort, identifying relevant analytics and utilizing coding techniques to analyze the available statistical data proved to be a challenging task that required exploring all possible resources.

4.1 CODING STANDARDS

To effectively implement the website, it was crucial to adhere to appropriate coding conventions. During the analysis of the available statistical data, several standards were followed.

- The statistical research and its modules were divided into multiple files, with each file containing code for one component of the statistical analysis.
- Files were grouped according to their type, separating code involving data from Search Engine Optimization results and code involving visualization and Organic Searches.
- To ensure clean and consistent code, a linting tool (eslint) was employed, which made code review easier.
- Additionally, the code was split into multiple smaller functions, each with a single responsibility to promote better understanding and simplification.
- Meaningful, simple and easy to understand comments were included where necessary.

4.2 CODING DETAILS

The codes below are the implementation of the various pages used in the website:

Index Page:

<html>

<head>

```
  <link rel="stylesheet" href="index.css" type="text/css">

  <script type="text/javascript">

  </script>

  <title>Dextrocity</title>
</head>

<body>

  <div class="vignette">
    <img id="toplogo" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/DC.png">
    <ul class="snip1189">
      <li class="current" onclick=""><a href="#">Home</a></li>
      <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
      <li><a href="loginpg.php">Login/Sign Up</a></li>
    </ul>
  </div>
  <h1 data-heading="DEXTROCITY">DEXTROCITY</h1>
  <div class="slider--container">

    <img class="slider--image"
src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/image1.jpg"
      alt="winter-01" />
    <img class="slider--image"
src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/image2.jpg"
      alt="winter-02" />
    <img class="slider--image"
src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/image3.jpg"
      alt="winter-03" />

  </div>


  <div class="cards">
    <button class="subcard" style="width: 280px; height:350px;">
      <img id="logo1" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/people.png">
      <div class="cardhead">Meet&nbspNew&nbspArtists</div>
      <div class="cardtext">A wide range of artists to choose from to help you make your
events a great success. </div>
```

```
    </button>

    <button class="subcard" style="width: 280px; height:350px;">
      <img id="logo1" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/event%20(1).png">
      <div class="cardhead">Events</div>
      <div class="cardtext">Numerous events with Artist requirements added on a daily
basis for Artists to choose from.
      </div>
    </button>

    <button class="subcard" style="width: 280px; height:350px;">
      <img id="logo1" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/busking.png">
      <div class="cardhead">Gig&nbspOpportunities</div>
      <div class="cardtext">Artists posting high quality content receive Gig opportunities
to showcase their talent on a
        regular basis.</div>
    </button>
  </div>
  <div class="Rectangle"></div>
  <img id="band" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/4301592.jpg">
  <div class="bandhead">Find&nbspArtists&nbspacross<br>the&nbspCountry</div>
  <div class="bandp">

We&nbsphere&nbspat&nbspTalentZen&nbspare&nbspdedicated&nbspto&nbspbringing
out&nbspbudding&nbsptalents&nbspacross<br>the&nbspcountry&nbspinto&nbspthe&n
bsplimelight&nbspand&nbspprovide&nbspwith&nbsppotential&nbspcustomers&nbspan
d<br>gig&nbspopportunities.&nbspFind&nbspyoung,&nbsptalented&nbspand&nbspext
ra&nbspordinary&nbspartists&nbsphere&nbspand<br>get&nbspthe&nbspright&nbspart
ists&nbspfor&nbspyour&nbspevents.&nbspWhat&nbspare&nbspyou&nbspwaiting&nbs
pfor?
  </div>
  <div class="maximize">Maximize&nbspYour&nbspPotential</div>
  <div class="maximizetext">

Show&nbspthe&nbspworld&nbspyour&nbspskill,&nbsplet&nbspthe&nbsppeople 
around&nbspyou&nbspknow&nbspwhat&nbspyou're&nbspcapable&nbspof,&nbspand&
nbspmake&nbspthe<br>most&nbspof&nbspwhat&nbspyou&nbsphave.&nbspWe 
at&nbspTalentZen&nbspwill&nbspmake&nbspevery&nbspeffort&nbspto&nbspprovide
&nbspyou&nbspwith&nbspthe&nbspgreatest<br>opportunity&nbspto&nbsppropel&nbs
pyou&nbspinto&nbspthe&nbspspotlight.
  </div>
  <img id="artimg" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/artimg.png">
```

```
  <div class="rectangle2"></div>
  <img id="logo2" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/DC.png">
  <!--Start of Tawk.to Script-->
  <script type="text/javascript">
    var Tawk_API = Tawk_API || {}, Tawk_LoadStart = new Date();
    (function () {
      var s1 = document.createElement("script"), s0 =
document.getElementsByTagName("script")[0];
      s1.async = true;
      s1.src = 'https://embed.tawk.to/63e09398c2f1ac1e20319ada/1goik13qm';
      s1.charset = 'UTF-8';
      s1.setAttribute('crossorigin', '*');
      s0.parentNode.insertBefore(s1, s0);
    })();
  </script>
  <!--End of Tawk.to Script-->

</body>

</html>
```

User Registration:

```
<?php
session_start();

include('Connect.php');

if(isset($_POST['register'])) {

    function validate($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $username = validate($_POST['username']);
```

```
$email = validate($_POST['email']);
$password = validate($_POST['password']);
$user_type = $_POST['user_type']; // added user_type variable

$secret_key = "af2b44bc63c81cc4662fb4c2f65e46a7";

// function to encrypt a password
function encrypt_password($password, $secret_key)
{
   $ciphering = "AES-128-CTR";
   $options = 0;
   $iv_length = openssl_cipher_iv_length($ciphering);
   $iv = openssl_random_pseudo_bytes($iv_length);
   $encryption = openssl_encrypt($password, $ciphering, $secret_key, $options, $iv);
   return base64_encode($encryption . "::" . $iv);
}

$encrypted_password = encrypt_password($password, $secret_key);

// check user type and set table name accordingly
if($user_type == "artist") {
   $table_name = "New_Artists";
} else if($user_type == "customer") {
   $table_name = "New_Customers";
} else {
   // handle error, invalid user type
   header("Location: loginpg.php?error=Invalid user type");
   exit();
}

$check_username = mysqli_query($con, "SELECT * FROM $table_name WHERE
username='$username'");
$check_email = mysqli_query($con, "SELECT * FROM $table_name WHERE
email='$email'");

if(mysqli_num_rows($check_username) > 0) {
   header("Location: loginpg.php?error=Username already exists");
   exit();
} else if(mysqli_num_rows($check_email) > 0) {
   header("Location: loginpg.php?error=Email already exists");
   exit();
} else {
   // Insert new user into database
   $insert_user = mysqli_query($con, "INSERT INTO $table_name (username, email,
PasswordHash) VALUES ('$username', '$email', '$encrypted_password')");
```

```
        if($insert_user) {
            $_SESSION['success'] = "Registration successful! You can now log in.";
            header("Location: loginpg.php");
            exit();
        } else {
            header("Location:ErrorMessage.html");
            exit();
        }
    }

} else {
    header("Location: loginpg.php");
    exit();
}
```

Login Page:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login/Sign up!</title>
    <link rel="stylesheet" href="login.css">
    <script>
            function showError(message) {
  var errorDiv = document.getElementById("error");
  errorDiv.innerHTML = message;
  errorDiv.style.display = "block";
  setTimeout(function() {
    errorDiv.style.display = "none";
  }, 5000); // Hide the error message after 5 seconds
}
</script>
</head>
<body>
    <div class="slider--container">

        <img class="slider--image" src="image1.jpg" alt="winter-01" />
        <img class="slider--image" src="image2.jpg" alt="winter-02" />
        <img class="slider--image" src="image3.jpg" alt="winter-03" />

      </div>
    <div class="cont" style= "height: 630px;">
        <div class="form sign-in">
```

```
        <h2>Sign In</h2><br>
        <form action="login.php" method="post">
          <?php if (isset($_GET['error1'])) { ?>

            <p style="margin-left: 230px; margin-top: 10px; color: red;"><?php echo
$_GET['error1']; ?></p>

        <?php } ?>
        <br>
        <label>
           <span>Username</span>
           <input type="text" name="username">
        </label>

        <label>
           <span>Password</span>
           <input type="password" name="password">
        </label>
          <a href="#">
           <button class="submit" type="submit" name="login">LOG IN</button>
          </a> </form>
        <p class="forgot-pass">Forgot Password ?</p>

        <div class="social-media">
          <ul>
            <li><img src="https://raw.githubusercontent.com/abo-elnoUr/public-
assets/master/facebook.png"></li>
            <li><img src="https://raw.githubusercontent.com/abo-elnoUr/public-
assets/master/twitter.png"></li>
            <li><img src="https://raw.githubusercontent.com/abo-elnoUr/public-
assets/master/linkedin.png"></li>
            <li><img src="https://raw.githubusercontent.com/abo-elnoUr/public-
assets/master/instagram.png"></li>
          </ul>
        </div>
      </div>

      <div class="sub-cont">
        <div class="img">
          <div class="img-text m-up">
            <h2>New here?</h2>
            <p>Sign up and discover great amount of new opportunities!</p>
          </div>
          <div class="img-text m-in">
            <h2>One of us?</h2>
            <p>If you already has an account, just sign in. We've missed you!</p>
```

```
          </div>
          <div class="img-btn">
            <span class="m-up">Sign Up</span>
            <span class="m-in">Sign In</span>
          </div>
        </div>
        <div class="form sign-up">
          <h2>Sign Up</h2>
        <form action="Artistregistration.php" method="post">
          <?php if (isset($_GET['error'])) { ?>

            <p style="margin-left: 200px; margin-top: 10px; color: red;"><?php echo
$_GET['error']; ?></p>

        <?php } ?>

          <label>
            <span>Username</span>
            <input type="text" name="username">
          </label>
          <label>
            <span>Email</span>
            <input type="email" name="email">
          </label>
          <label>
  <span style="display: block; margin-bottom: 5px;">Select User type</span>
  <select name="user_type" style="background-color:#white; color: #505f75;; padding:
14px; font-size: 16px; border: 2px #505f75;; cursor: pointer;">
  <option value="artist">Artist</option>
  <option value="customer">Customer</option>
</select>

</label>

          <label>
            <span>Password</span>
            <input type="password" name="password">
          </label>
          <label>
            <span>Confirm Password</span>
            <input type="password" name="confirmpassword">
          </label>
          <button type="submit" class="submit" name="register">Sign Up
Now</button></form>
        </div>
      </div>
```

```
    </div>
    <script src="./login.js"></script>
</body>
</html>
<?php
session_start();

include('Connect.php');
$secret_key = "af2b44bc63c81cc4662fb4c2f65e46a7";

// function to decrypt a password
function decrypt_password($encrypted_password, $secret_key) {
   $ciphering = "AES-128-CTR";
   $options = 0;
   list($encrypted_password, $iv) = explode("::", base64_decode($encrypted_password),
2);
   $decryption = openssl_decrypt($encrypted_password, $ciphering, $secret_key,
$options, $iv);
   return $decryption;
}

if (isset($_POST['username']) && isset($_POST['password'])) {

   function validate($data){
      $data = trim($data);
      $data = stripslashes($data);
      $data = htmlspecialchars($data);
      return $data;
   }

   $uname = validate($_POST['username']);
   $pass = validate($_POST['password']);

   if (empty($uname)) {
      header("Location: loginpg.php?error1=User Name is required");
      exit();
   } else if (empty($pass)) {
      header("Location: loginpg.php?error1=Password is required");
      exit();
   } else {
      $sql_artist = "SELECT * FROM New_Artists WHERE username = '$uname'";
      $result_artist = mysqli_query($con, $sql_artist);

      $is_artist = false;
      $table_name = "";
```

```
        if (mysqli_num_rows($result_artist)) {
            $is_artist = true;
            $table_name = "New_Artists";
        } else {
            $sql_customer = "SELECT * FROM New_Customers WHERE username =
'$uname'";
            $result_customer = mysqli_query($con, $sql_customer);

            if (mysqli_num_rows($result_customer)) {
                $table_name = "New_Customers";
            } else {
                header("Location: loginpg.php?error1=User does not exist");
                exit();
            }
        }

        $row = mysqli_fetch_assoc($is_artist ? $result_artist : $result_customer);

        $decrypted_password = decrypt_password($row['PasswordHash'], $secret_key);
        if ($decrypted_password === $pass) {
            echo "Logged in!";
            $_SESSION['Username'] = $row['username'];
            $_SESSION['Name'] = $row['name'];
            $_SESSION['Email'] = $row['email'];
            $_SESSION['DP'] = $row['DP'];
            $_SESSION['City'] = $row['city'];
            $_SESSION['AboutMe'] = $row['AboutMe'];
            $_SESSION['table_name'] = $table_name;

            // check if this is the first login and update the FirstLogin column accordingly
            $first_login = $row['FirstLogin'];
            if ($first_login == 1) {
                $sql_update_first_login = "UPDATE $table_name SET FirstLogin = 0
WHERE username = '$uname'";
                mysqli_query($con, $sql_update_first_login);
                header("Location: edit_profile.php");
            } else {
                // Redirect to the appropriate page based on the user type
                if ($table_name == "New_Artists") {
                    header("Location: Artist_blogfeed.php");
                } else {
                    header("Location: blogfeed.php");
                }
            }
            exit();
        } else {
```

```
            header("Location: loginpg.php?error1=Incorrect username or password");
            exit();
        }
    }
}
```

Blogs Page:

```
<?php

session_start();

if (isset($_SESSION['Name']) && isset($_SESSION['Username']))
{
    include('Connect.php');
  $user = $_SESSION['Username'];
  $sql3 = "SELECT * FROM {$_SESSION['table_name']} WHERE username =
'{$user}'";

  $res = $con->query($sql3);

    if ($res->num_rows > 0)
    {
      while($row = $res->fetch_assoc())
      {
 ?>
<html>
  <head>
  <link rel="stylesheet" href="blogfeed.css" type="text/css">
  <style>
    .vignette {
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        height: auto;
        box-shadow: 0 0 300px #1C0C5B inset;
        z-index: 1;
      }
      #toplogo
      {
        width: 100px;
        height: 100px;
        top:5px;
        left: 50px;
        position: fixed;
      }
      .btn-5 {
```

```
  position: fixed;
  width: 120px;
  height: 50px;
  line-height: 42px;
  top: 110px;
  left: 45px;
  padding: 0;
  border: none;
  background: #916BBF;
  background: linear-gradient(40deg, #916BBF 0%, #432e9b 100%);
}
.btn-5:hover {
  color: #916BBF;
  background: transparent;
   box-shadow:none;
}
.btn-5:before,
.btn-5:after{
  content:'';
  position:fixed;
  top:0;
  right:0;
  height:2px;
  width:0;
  background: #916BBF;
  box-shadow:
   -1px -1px 5px 0px #fff,
   7px 7px 20px 0px #0003,
   4px 4px 5px 0px #0002;
  transition:400ms ease all;
}
.btn-5:after{
  right:inherit;
  top:inherit;
  left: 45px;
  bottom:0;
}
.btn-5:hover:before,
.btn-5:hover:after{
  width:120px;
  transition:800ms ease all;
}


    </style>
  <script type = "text/javascript">
```

```
     document.querySelectorAll(".projcard-description").forEach(function(box) {
         $clamp(box, {clamp: 6});
});

    </script>

  </head>
  <body>
  <div class = "vignette">
    <img id="toplogo" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/DC.png" >
    <ul class="snip1189">
       <li class="current" onclick=""><a href="blogfeed.php">Blogs</a></li>
       <li><a href="events.php">Events</a></li>
       <li><a href="#">Bookings</a></li>
       <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
       <li><a href="u_profile.php">My Profile</a></li>
     </ul>

<div style="position: absolute; top: 0px;"><img src="<?php echo $row['DP']; ?>"
style="position: absolute; width: 35px;height: 35px;left: 1360px;top: 25px; border-radius:
50%;"></div>
<p style=" position: absolute; top: 13px; left: 1405px; font-family: PT Sans; font-size:
16px;  color: white;"><?php echo $row['name']; ?></p>
<a href="logout.php" style=" position: absolute; top: 50px; left: 1430px; font-family: PT
Sans; font-size: 15px;  color: white; text-decoration: none;">Log Out</a>
</div></div>
<div style="height: 40px;"></div>

<script type="text/javascript">
  var Tawk_API=Tawk_API||{}, Tawk_LoadStart=new Date();
  (function(){
  var
s1=document.createElement("script"),s0=document.getElementsByTagName("script")[0]
;
  s1.async=true;
  s1.src='https://embed.tawk.to/63e09398c2f1ac1e20319ada/1goik13qm';
  s1.charset='UTF-8';
  s1.setAttribute('crossorigin','*');
  s0.parentNode.insertBefore(s1,s0);
  })();
  </script>
  <a href="new_event.html"><button class="custom-btn btn-5" ><span>NEW
EVENT</span></button></a>
</body></html>
<?php
```

```
}
include('Connect.php');
$sql3 = "SELECT * from Blogs ORDER BY Blog_ID DESC";
$res = $con->query($sql3);

if ($res->num_rows > 0)
{

  while($row = $res->fetch_assoc())
 {
    ?>
<div class="projcard-container">

   <div class="projcard projcard-blue">
     <div class="projcard-innerbox">
     <img class="projcard-img" src="<?php echo  $row["thumbnail_url"] ?>" />
                              <div class="projcard-textbox">
                                       <div class="projcard-title"><?php echo  $row["caption"]
?></div>
                                       <div class="projcard-subtitle">Posted by: <?php echo
$row["username"] ?></div>
                                       <div class="projcard-bar"></div>
                                       <div class="projcard-description"><?php echo
$row["description"] ?></div>
       <button class="custom-btn btn-3">Read More</button>
       </div>
     </div>
   </div>
</div>

<?php
  }
}
else {
  echo "No Blogs have been made";
}

}
}
else
{

   header("Location: loginpg.php");
   exit();

}
```

```
 ?>
```

New Blog Post:

```
<?php

session_start();

if (isset($_SESSION['Name']) && isset($_SESSION['Username']))
{
  include('Connect.php');
  $user = $_SESSION['Username'];
  $sql3 = "SELECT * FROM {$_SESSION['table_name']} WHERE username =
'{$user}'";

  $res = $con->query($sql3);

    if ($res->num_rows > 0)
    {
      while($row = $res->fetch_assoc())
      {
 ?>
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"
rel="stylesheet">
<head>
    <link rel="stylesheet" href="./nbp.css">
</head>
<body>
  <div class="main-content">
    <!-- Top navbar -->

    <nav class="navbar navbar-top navbar-expand-md navbar-dark" id="navbar-main">
      <div class="container-fluid">

        <!-- Form --><ul class="snip1189">
            <li class="current" onclick=""><a href="#">Blogs</a></li>
            <li><a href="Artist_events.php">Events</a></li>
            <li><a href="#">Bookings</a></li>
            <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
            <li ><a href="Artist_profile.php">My Profile</a></li>
          </ul>


        <!-- User -->
        <ul class="navbar-nav align-items-center d-none d-md-flex">
          <li class="nav-item dropdown">
```

```
          <a class="nav-link pr-0" href="#" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <div class="media align-items-center">
              <span class="avatar avatar-sm rounded-circle">
                <img alt="Image placeholder" src= "<?php echo  $row["DP"] ?> ">
              </span>
              <div class="media-body ml-2 d-none d-lg-block">

                <span class="mb-0 text-sm  font-weight-bold"><?php echo  $row["name"]
?></span>

              </div>
            </div>
          </a>

        </li>
      </ul>
    </div>
    <a href="logout.php" style=" position: absolute; top: 55px; left: 1420px; font-family:
PT Sans; font-size: 15px;  color: white; text-decoration: none;">Log Out</a>
  </nav>
  <!-- Header -->
  <div class="header pb-8 pt-5 pt-lg-8 d-flex align-items-center" style="min-height:
600px; background-image:
url(https://raw.githubusercontent.com/creativetimofficial/argon-dashboard/gh-
pages/assets-old/img/theme/profile-cover.jpg); background-size: cover; background-
position: center top;">
    <!-- Mask -->
    <span class="mask bg-gradient-default opacity-8"></span>
    <!-- Header container -->
    <div class="container-fluid d-flex align-items-center">
      <div class="row">
        <div class="col-lg-7 col-md-10">
          <h1 class="display-2 text-white">Create New Blog</h1>
          <p>Share your skills to the world by making a new blog</p>

        </div>
      </div>
    </div>
  </div>
  <!-- Page content -->
  <div class="container-fluid mt--7">
    <div class="row">

      <div class="col-xl-8 order-xl-1">
        <div class="card bg-secondary shadow">
```

```
          <div class="card-header bg-white border-0">
            <div class="row align-items-center">
              <div class="col-8">
                <h3 class="mb-0">Please enter the details</h3>
              </div>
            </div>
          </div>
          <div class="card-body">
          <form action="BlogPost.php" method="POST" enctype="multipart/form-data">
 <div class="pl-lg-4">
   <div class="row">
     <div class="col-lg-6">
       <div class="form-group focused">
         <label class="form-control-label" for="title">Title</label>
         <input type="text" name="title" id="title" class="form-control form-control-
alternative" placeholder="">
       </div>
     </div>
   </div>

   <hr class="my-4">

   <div class="pl-lg-4">
     <div class="row">
       <div class="col-lg-4">
         <div class="form-group focused">
           <label class="form-control-label" for="thumbnail">Thumbnail (Image
Only)</label>
           <input type="file" name="thumbnail" id="thumbnail" class="form-control form-
control-alternative" accept="image/*" placeholder="">
         </div>
       </div>
     </div>

     <div class="row">
       <div class="col-lg-4">
         <div class="form-group focused">
           <label class="form-control-label" for="imagevideo">Image/Video (Image or
Video)</label>
           <input type="file" name="imagevideo" id="imagevideo" class="form-control
form-control-alternative" accept="image/*, video/*" placeholder="">
         </div>
       </div>
     </div>
   </div>
```

```
    <hr class="my-4">

    <div class="pl-lg-4">
      <div class="form-group focused">
        <label>Description</label>
        <textarea rows="4" class="form-control form-control-alternative"
name="description" placeholder="Write the body of your blog here"></textarea>
      </div>
    </div>

    <button class="btn btn-info" type="submit" >Post</button>
  </div>
</form>

          </div>
        </div>
      </div>
    </div>
  </div>
  <footer class="footer">
    <div class="row align-items-center justify-content-xl-between">
      <div class="col-xl-6 m-auto text-center">
        <div class="copyright">
          <p>Copyright by Dextrocity</p>
        </div>
      </div>
    </div>
  </footer>
</body>
<?php
     }
}
else{

    header("Location: loginpg.php");

    exit();

}
}
?>
```

Events Page:

```
<?php
```

```
session_start();

if (isset($_SESSION['Name']) && isset($_SESSION['Username']))
{
   include('Connect.php');
  $user = $_SESSION['Username'];
  $sql3 = "SELECT * FROM {$_SESSION['table_name']} WHERE username =
'{$user}'";

  $res = $con->query($sql3);

   if ($res->num_rows > 0)
   {
     while($row = $res->fetch_assoc())
     {
 ?>
<html>
  <head>
  <link rel="stylesheet" href="blogfeed.css" type="text/css">
  <style>
    .vignette {
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        height: auto;
        box-shadow: 0 0 300px #1C0C5B inset;
        z-index: 1;
      }
      #toplogo
      {
        width: 100px;
        height: 100px;
        top:5px;
        left: 50px;
        position: fixed;
      }
      .btn-5 {
    position: fixed;
    width: 120px;
    height: 50px;
    line-height: 42px;
    top: 110px;
    left: 45px;
    padding: 0;
    border: none;
```

```
    background: #916BBF;
    background: linear-gradient(40deg, #916BBF 0%, #432e9b 100%);
   }
   .btn-5:hover {
    color: #916BBF;
    background: transparent;
     box-shadow:none;
   }
   .btn-5:before,
   .btn-5:after{
    content:'';
    position:fixed;
    top:0;
    right:0;
    height:2px;
    width:0;
    background: #916BBF;
    box-shadow:
     -1px -1px 5px 0px #fff,
     7px 7px 20px 0px #0003,
     4px 4px 5px 0px #0002;
    transition:400ms ease all;
   }
   .btn-5:after{
    right:inherit;
    top:inherit;
    left: 45px;
    bottom:0;
   }
   .btn-5:hover:before,
   .btn-5:hover:after{
    width:120px;
    transition:800ms ease all;
   }


      </style>
    <script type = "text/javascript">
      document.querySelectorAll(".projcard-description").forEach(function(box) {
          $clamp(box, {clamp: 6});
});

     </script>

  </head>
  <body>
```

```
  <div class = "vignette">
    <img id="toplogo" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-
project/main/main/DC.png" >
    <ul class="snip1189">
       <li ><a href="blogfeed.php">Blogs</a></li>
       <li class="current" onclick=""><a href="events.php">Events</a></li>
       <li><a href="#">Bookings</a></li>
       <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
       <li><a href="u_profile.php">My Profile</a></li>
      </ul>

<div style="position: absolute; top: 0px;"><img src="<?php echo $row['DP']; ?>"
style="position: absolute; width: 35px;height: 35px;left: 1360px;top: 25px; border-radius:
50%;"></div>
<p style=" position: absolute; top: 13px; left: 1405px; font-family: PT Sans; font-size:
16px;  color: white;"><?php echo $row['name']; ?></p>
<a href="logout.php" style=" position: absolute; top: 50px; left: 1430px; font-family: PT
Sans; font-size: 15px;  color: white; text-decoration: none;">Log Out</a>
</div></div>
<div style="height: 40px;"></div>

<script type="text/javascript">
  var Tawk_API=Tawk_API||{}, Tawk_LoadStart=new Date();
  (function(){
  var
s1=document.createElement("script"),s0=document.getElementsByTagName("script")[0]
;
  s1.async=true;
  s1.src='https://embed.tawk.to/63e09398c2f1ac1e20319ada/1goik13qm';
  s1.charset='UTF-8';
  s1.setAttribute('crossorigin','*');
  s0.parentNode.insertBefore(s1,s0);
  })();
  </script>
  <a href="new_event.html"><button class="custom-btn btn-5" ><span>NEW
EVENT</span></button></a>
</body></html>
<?php
}
include('Connect.php');
$sql3 = "SELECT * FROM Blogs ORDER BY Blog_ID DESC";
$res = $con->query($sql3);

if ($res->num_rows > 0)
{
```

```
 while($row = $res->fetch_assoc())
{
   ?>
<div class="projcard-container">

   <div class="projcard projcard-blue">
    <div class="projcard-innerbox">
    <img class="projcard-img" src="<?php echo  $row["thumbnail_url"] ?>" />
                        <div class="projcard-textbox">
                                <div class="projcard-title"><?php echo  $row["caption"]
?></div>
                                <div class="projcard-subtitle">Posted by: <?php echo
$row["username"] ?></div>
                                <div class="projcard-bar"></div>
                                <div class="projcard-description"><?php echo
$row["description"] ?></div>
      <button class="custom-btn btn-3">Read More</button>
      </div>
    </div>
   </div>
</div>

<?php
 }
}
else {
 echo "No Blogs have been made";
}

}
}
else
{

   header("Location: loginpg.php");

   exit();

}
 ?>
```

New Event Post:

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"
rel="stylesheet">
<head>
   <link rel="stylesheet" href="./nep.css">
```

```
</head>
<body>
 <script>
  document.addEventListener('DOMContentLoaded', function() {
    const numberInputs = document.querySelectorAll('input[type="number"]');
    const dateInputs = document.querySelectorAll('input[type="date"]');

    numberInputs.forEach(input => {
      input.addEventListener('input', function() {
        if (this.value < 0) {
          this.value = 0;
        }
      });
    });

    const today = new Date().toISOString().split('T')[0];
    dateInputs.forEach(input => {
      input.setAttribute('min', today);
      input.addEventListener('change', function() {
        if (this.value < today) {
          this.value = today;
        }
      });
    });
  });
</script>



  <div class="main-content">
    <!-- Top navbar -->

    <nav class="navbar navbar-top navbar-expand-md navbar-dark" id="navbar-main">
      <div class="container-fluid">

        <!-- Form --><ul class="snip1189">
            <li><a href="blogfeed.php">Blogs</a></li>
            <li class="current" onclick=""><a href="events.php">Events</a></li>
            <li><a href="#">Bookings</a></li>
            <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
            <li ><a href="u_profile.php">My Profile</a></li>
          </ul>


        <!-- User -->
        <ul class="navbar-nav align-items-center d-none d-md-flex">
```

```
        <li class="nav-item dropdown">
          <a class="nav-link pr-0" href="#" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <div class="media align-items-center">
              <span class="avatar avatar-sm rounded-circle">
                <img alt="Image placeholder" src="https://demos.creative-tim.com/argon-
dashboard/assets-old/img/theme/team-4.jpg">
              </span>
              <div class="media-body ml-2 d-none d-lg-block">

                <span class="mb-0 text-sm  font-weight-bold">Jessica Jones</span>

              </div>
            </div>
          </a>

        </li>
      </ul>
    </div>
  </nav>
  <!-- Header -->
  <div class="header pb-8 pt-5 pt-lg-8 d-flex align-items-center" style="min-height:
600px; background-image:
url(https://raw.githubusercontent.com/creativetimofficial/argon-dashboard/gh-
pages/assets-old/img/theme/profile-cover.jpg); background-size: cover; background-
position: center top;">
    <!-- Mask -->
    <span class="mask bg-gradient-default opacity-8"></span>
    <!-- Header container -->
    <div class="container-fluid d-flex align-items-center">
      <div class="row">
        <div class="col-lg-7 col-md-10">
          <h1 class="display-2 text-white">New Event</h1>
          <p>To post an event that your hosting, please fill in the details below for the
artists to see.</p>
          <a href="#!" class="btn btn-info">Post</a>
        </div>
      </div>
    </div>
  </div>
  <!-- Page content -->
  <div class="container-fluid mt--7">
    <div class="row">

      <div class="col-xl-8 order-xl-1">
        <div class="card bg-secondary shadow">
```

```
        <div class="card-header bg-white border-0">
          <div class="row align-items-center">
            <div class="col-8">
              <h3 class="mb-0">Please enter the details</h3>
            </div>
          </div>
        </div>
        <div class="card-body">
          <form>
            <div class="pl-lg-4">
              <div class="row">
                <div class="col-lg-6">
                  <div class="form-group focused">
                    <label class="form-control-label" for="input-username">Title</label>
                    <input type="text" id="input-username" class="form-control form-
control-alternative" placeholder="">
                  </div>
                </div>

              </div>
              <div class="row">
                <div class="col-lg-6">
                  <div class="form-group focused">
                    <label class="form-control-label" for="input-first-
name">Location</label>
                    <input type="text" id="input-first-name" class="form-control form-
control-alternative" placeholder="Location of the Event">
                  </div>
                </div>

              </div>
            </div>
            <hr class="my-4">
            <div class="pl-lg-4">

              <div class="row">
                <div class="col-lg-4">
                  <div class="form-group focused">
                    <label class="form-control-label" for="input-city">Starting
Amount</label>
                    <input type="number" id="input-city" class="form-control form-control-
alternative" placeholder="Amount you want to pay">
                  </div>
                </div>
              </div>
```

```
            <div class="row">
              <div class="col-lg-4">
                <div class="form-group focused">
                  <label class="form-control-label" for="input-city">Image</label>
                  <input type="file" id="input-city" class="form-control form-control-
alternative" placeholder="">
                </div>
              </div>
            </div>

            <div class="row">
              <div class="col-lg-4">
                <div class="form-group focused">
                  <label class="form-control-label" for="input-
city">Duration(hour(s))</label>
                  <input type="number" id="input-city" class="form-control form-control-
alternative" placeholder="Event duration">
                </div>
              </div>
            </div>

            <div class="row">
              <div class="col-lg-4">
                <div class="form-group focused">
                  <label class="form-control-label" for="input-city">Date</label>
                  <input type="date" id="input-city" class="form-control form-control-
alternative" placeholder="">
                </div>
              </div>
            </div>

          </div>
          <hr class="my-4">
          <div class="pl-lg-4">
            <div class="form-group focused">
              <label>Description</label>
              <textarea rows="4" class="form-control form-control-alternative"
placeholder="Describe your event"></textarea>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
```

```
  </div>
  <footer class="footer">
    <div class="row align-items-center justify-content-xl-between">
      <div class="col-xl-6 m-auto text-center">
        <div class="copyright">
          <p>Copyright by Dextrocity</p>
        </div>
      </div>
    </div>
  </footer>
</body>
```

User Profile Page:

```
<?php
session_start();
if (isset($_SESSION['Name']) && isset($_SESSION['Username']) &&
isset($_SESSION['table_name'])) {
  include('Connect.php');
  $user = $_SESSION['Username'];
  $sql3 = "SELECT * FROM {$_SESSION['table_name']} WHERE username =
'{$user}'";

  $res = $con->query($sql3);

    if ($res->num_rows > 0)
    {
      while($row = $res->fetch_assoc())
      {
       ?>
        <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"
rel="stylesheet">
        <head>
        <link rel="stylesheet" href="./uprof.css">

        <script src="https://code.jquery.com/jquery-3.6.0.min.js">

        $(document).ready(function() {
        const maxChars = 50;
        $('figcaption h3').each(function() {
        var text = $(this).text();
```

```
      if (text.length > maxChars) {
        $(this).text(text.substr(0, maxChars) + "...");
      }
      });
      });

    </script>
    </head>
    <body>
      <div class="main-content">
  <!-- Top navbar -->

      <nav class="navbar navbar-top navbar-expand-md navbar-dark" id="navbar-main">
      <div class="container-fluid">

        <!-- Form --><ul class="snip1189">
            <li><a href="blogfeed.php">Blogs</a></li>
            <li><a href="events.php">Events</a></li>
            <li><a href="#">Bookings</a></li>
            <li><a href="https://dextrocityind.tawk.help/">FAQs</a></li>
            <li class="current" onclick=""><a href="u_profile.php">My Profile</a></li>
          </ul>

      <form class="navbar-search navbar-search-dark form-inline mr-3 d-none d-md-flex
ml-lg-auto">
        <div class="form-group mb-0">

        </div>
      </form>
      <!-- User -->
      <ul class="navbar-nav align-items-center d-none d-md-flex">
        <li class="nav-item dropdown">
          <a class="nav-link pr-0" href="#" role="button" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            <div class="media align-items-center">
              <span class="avatar avatar-sm rounded-circle">
                <img alt="Image placeholder" src="<?php echo  $row["DP"] ?>">
              </span>
              <div class="media-body ml-2 d-none d-lg-block">

                <span class="mb-0 text-sm  font-weight-bold"><?php echo
$row['name']?></span>
              </div>
            </div>
          </a>
          <div class="dropdown-menu dropdown-menu-arrow dropdown-menu-right">
```

```
            <div class=" dropdown-header noti-title">
              <h6 class="text-overflow m-0">Welcome!</h6>
            </div>
            <a href="../examples/profile.html" class="dropdown-item">
              <i class="ni ni-single-02"></i>
              <span>My profile</span>
            </a>
            <a href="../examples/profile.html" class="dropdown-item">
              <i class="ni ni-settings-gear-65"></i>
              <span>Settings</span>
            </a>
            <a href="../examples/profile.html" class="dropdown-item">
              <i class="ni ni-calendar-grid-58"></i>
              <span>Activity</span>
            </a>
            <a href="../examples/profile.html" class="dropdown-item">
              <i class="ni ni-support-16"></i>
              <span>Support</span>
            </a>
            <div class="dropdown-divider"></div>
            <a href="#!" class="dropdown-item">
              <i class="ni ni-user-run"></i>
              <span>Logout</span>
            </a>
          </div>
        </li>
      </ul>
    </div>
  </nav>
  <!-- Header -->
  <div class="header pb-8 pt-5 pt-lg-8 d-flex align-items-center" style="min-height:
600px; background-image:
url(https://raw.githubusercontent.com/creativetimofficial/argon-dashboard/gh-
pages/assets-old/img/theme/profile-cover.jpg); background-size: cover; background-
position: center top;">
    <!-- Mask -->
    <span class="mask bg-gradient-default opacity-8"></span>
    <!-- Header container -->
    <div class="container-fluid d-flex align-items-center">
      <div class="row">
        <div class="col-lg-7 col-md-10">
          <h1 class="display-2 text-white">Hello <?php echo  $row['name']?></h1>
          <p class="text-white mt-0 mb-5">This is your profile page. You can see the
progress you've made with your work and manage your blogs here</p>
          <a href="edit_profile.php" class="btn btn-info">Edit profile</a>
        </div>
```

```
    </div>
  </div>
</div>
<!-- Page content -->
<div class="container-fluid mt--7">
  <div class="row">
    <div class="col-xl-4 order-xl-2 mb-5 mb-xl-0">
      <div class="card-profile shadow">
        <div class="row justify-content-center">
          <div class="col-lg-3 order-lg-2">
            <div class="card-profile-image">
              <a href="#">
                <img src="<?php echo  $row['DP']?>" class="rounded-circle">
              </a>
            </div>
          </div>
        </div>
        <div class="card-header text-center border-0 pt-8 pt-md-4 pb-0 pb-md-4">
          <div class="d-flex justify-content-between">
            <a href="#" class="btn btn-sm btn-info mr-4">Connect</a>
            <a href="#" class="btn btn-sm btn-default float-right">Message</a>
          </div>
        </div>
        <div class="card-body pt-0 pt-md-4">
          <div class="row">
            <div class="col">
              <div class="card-profile-stats d-flex justify-content-center mt-md-5">
                <div>
                  <span class="heading">666</span>
                  <span class="description">Posts</span>
                </div>
                <div>
                  <span class="heading">4.2</span>
                  <span class="description">Ratings</span>
                </div>
                <div>
                  <span class="heading">23</span>
                  <span class="description">Bookings</span>
                </div>
              </div>
            </div>
          </div>
          <div class="text-center">
            <h3>
            <?php echo  $row['name']?>
            </h3>
```

```
            <div class="h5 font-weight-300">
              <i class="ni location_pin mr-2"></i><?php echo  $row['city']?>
            </div>
            <div class="h5 mt-4">
              <i class="ni business_briefcase-24 mr-2"></i><?php echo
$row['username']?>
            </div>
            <div>
              <i class="ni education_hat mr-2"></i>Dancer
            </div>
            <hr class="my-4">
            <p><?php echo  $row['AboutMe']?></p>
          </div>
        </div>
      </div>
      <div class="col-xl-8 order-xl-1">
        <div class="card bg-secondary shadow">
          <div class="card-header1 bg-white border-0">
            <div class="row align-items-center">
              <div class="col-8">
                <h3 class="mb-0">Posts</h3>
              </div>

            </div>
          </div>


   <?php
   }
    include('Connect.php');
    $user = $_SESSION['Username'];
    $sql = "SELECT * FROM {$_SESSION['table_name']} JOIN Blogs WHERE
Blogs.username = '{$user}' AND {$_SESSION['table_name']}.username =
Blogs.username";

    $result = $con->query($sql);

    if ($result->num_rows > 0)
    {

   while($rows = $result->fetch_assoc())
  { ?>
        <div class="projcard-container">

   <div class="projcard projcard-blue">
```

```
    <div class="projcard-innerbox">
    <img class="projcard-img" src="<?php echo  $rows["thumbnail_url"] ?>" />
                        <div class="projcard-textbox">
                                <div class="projcard-title"><?php echo  $rows["caption"]
?></div>
                                <div class="projcard-bar"></div>
      <button class="custom-btn btn-3">Read More</button>
      </div>
    </div>
   </div>
</div>
          <?php

  }
}
else {
  echo("No Posts Yet");
}
?>
         </div>
        </div>
       </div>
     </div>
   </div>
  </div>
  <footer class="footer">
   <div class="row align-items-center justify-content-xl-between">
    <div class="col-xl-6 m-auto text-center">
     <div class="copyright">
      <p>Copyright by Dextrocity</p>
     </div>
    </div>
   </div>
  </footer>
  <script type="text/javascript">
  var Tawk_API=Tawk_API||{}, Tawk_LoadStart=new Date();
  (function(){
  var
s1=document.createElement("script"),s0=document.getElementsByTagName("script")[0]
;
  s1.async=true;
  s1.src='https://embed.tawk.to/63e09398c2f1ac1e20319ada/1goik13qm';
  s1.charset='UTF-8';
  s1.setAttribute('crossorigin','*');
  s0.parentNode.insertBefore(s1,s0);
  })();
```

```
  </script>
</body>
<?php
 }
else
{

    header("Location: ErrorMessage.html");

    exit();

}
}
?>
```

Profile Edit Page:

```
<?php
session_start();
if (isset($_SESSION['Name']) && isset($_SESSION['Username']) &&
isset($_SESSION['table_name'])) {
  include('Connect.php');
  $user = $_SESSION['Username'];
  $sql3 = "SELECT * from {$_SESSION['table_name']} where username= '$user'";
  $res = $con->query($sql3);

  if ($res->num_rows > 0)
  {

    while($row = $res->fetch_assoc())
   {
      ?>

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700"
rel="stylesheet">
<head>
    <link rel="stylesheet" href="./edprof.css">
    <!-- jQuery library -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<!-- jQuery UI library -->
<link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-
ui.css">
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
<title>Profile Creation</title>
<script>
```

```
  $(document).ready(function() {
  // Define the list of cities
  var cities = [  "Mumbai",  "Delhi",  "Bangalore",  "Hyderabad",  "Ahmedabad",
"Chennai",  "Kolkata",  "Surat",  "Pune",  "Jaipur",  "Lucknow",  "Kanpur",  "Nagpur",
"Visakhapatnam",  "Bhopal",  "Patna",  "Ludhiana",  "Agra",  "Nashik",  "Vadodara",
"Gorakhpur",  "Rajkot",  "Meerut",  "Kalyan-Dombivali",  "Varanasi",  "Srinagar",
"Aurangabad",  "Dhanbad",  "Amritsar",  "Navi Mumbai",  "Allahabad",  "Ranchi",
"Howrah",  "Jabalpur",  "Gwalior",  "Vijayawada",  "Jodhpur",  "Madurai",  "Raipur",
"Kota",  "Guwahati",  "Chandigarh",  "Solapur",  "Hubballi-Dharwad",  "Bareilly",
"Moradabad",  "Mysore",  "Gurgaon",  "Aligarh",  "Jalandhar",  "Tiruchirappalli",
"Bhubaneswar",  "Salem",  "Mira-Bhayandar",  "Warangal",  "Thiruvananthapuram",
"Bhiwandi",  "Saharanpur",  "Gorakhpur",  "Guntur",  "Bikaner",  "Amravati",  "Noida",
"Jamshedpur",  "Bhilai",  "Cuttack",  "Firozabad",  "Kochi",  "Nellore",  "Bhavnagar",
"Dehradun",  "Durgapur",  "Asansol",  "Rourkela",  "Nanded",  "Kolhapur",  "Ajmer",
"Akola",  "Gulbarga",  "Jamnagar",  "Ujjain",  "Loni",  "Siliguri",  "Jhansi",
"Ulhasnagar",  "Jammu",  "Sangli-Miraj & Kupwad",  "Mangalore",  "Erode",
"Belgaum",  "Ambattur",  "Tirunelveli",  "Malegaon",  "Gaya",  "Jalgaon",  "Udaipur",
"Maheshtala",  "Davanagere",  "Kozhikode",  "Kurnool"]


  // Attach the autocomplete feature to the textbox
  $("#city").autocomplete({
    source: cities
  });
});

</script>
<style>
        #imagePreview {
                        width: 200px;
                        height: 200px;
                        background-position: center center;
                        background-size: cover;
                        margin: 0 auto;
                        border-radius: 50%;
                        overflow: hidden;
                }
                input[type="file"] {
                        display: none;
                }
                .label1 {
                        cursor: pointer;
                        display: block;
                        width: 200px;
                        height: 40px;
    margin-right: 300px;
```

```
    margin-top: 75px;
                        background-color: #3498db;
                        color: #fff;
                        text-align: center;
                        line-height: 40px;
                        border-radius: 4px;
                }
</style>
</head>
<body>
<img id="toplogo" src="https://raw.githubusercontent.com/Prawnsy/Dextrocity-project/main/main/DC.png" >
  <div class="main-content">
   <!-- Top navbar -->

   <nav class="navbar navbar-top navbar-expand-md navbar-dark" id="navbar-main">
    <div class="container-fluid">

      <!-- User -->
      <ul class="navbar-nav align-items-center d-none d-md-flex">
       <li class="nav-item dropdown">

       </li>
      </ul>
    </div>
   </nav>
   <!-- Header -->
   <div class="header pb-8 pt-5 pt-lg-8 d-flex align-items-center" style="min-height: 600px; background-image: url(https://raw.githubusercontent.com/creativetimofficial/argon-dashboard/gh-pages/assets-old/img/theme/profile-cover.jpg); background-size: cover; background-position: center top;">
    <!-- Mask -->
    <span class="mask bg-gradient-default opacity-8"></span>
    <!-- Header container -->
    <div class="container-fluid d-flex align-items-center">
     <div class="row">
      <div class="col-lg-7 col-md-10">
       <h1 class="display-2 text-white">Hello <?php echo  $_SESSION["Username"] ?></h1>
        <p class="text-white mt-0 mb-5">Welcome to Dextrocity! Please update your details before you begin to use our services. Thank you! </p>

      </div>
     </div>
    </div>
```

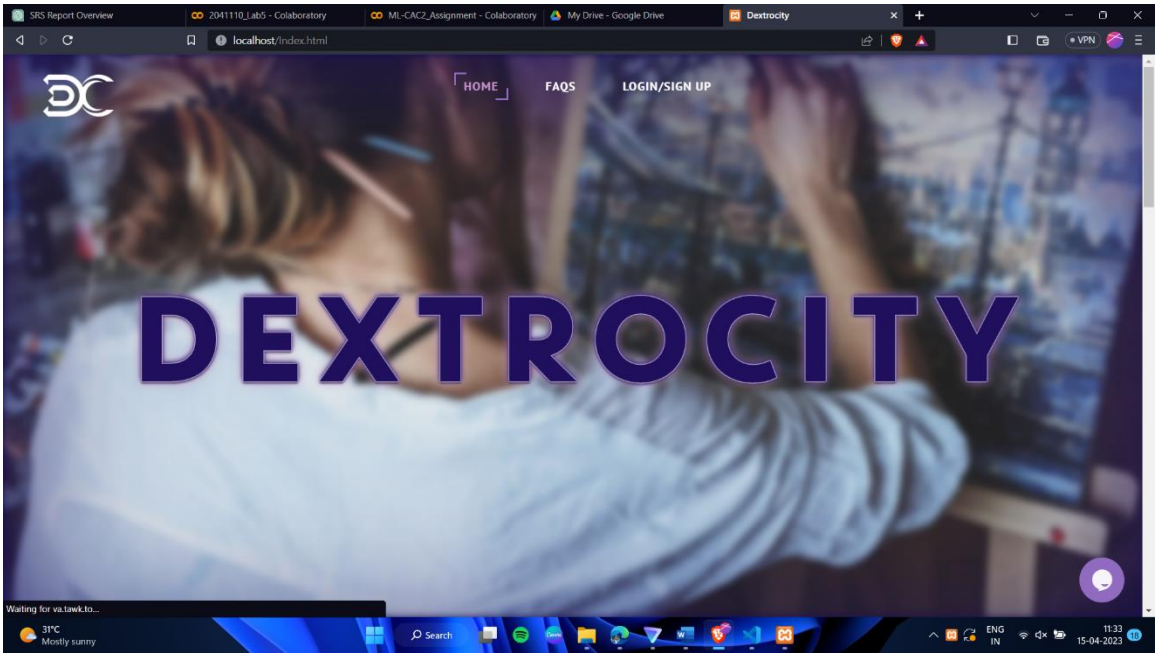
```
  </div>
  <!-- Page content -->
  <div class="container-fluid mt--7">
   <div class="row">

     <div class="col-xl-8 order-xl-1">
      <div class="card bg-secondary shadow">
       <div class="card-body">
       <form action="profileupdate.php" method="post" enctype="multipart/form-
data">
          <h6 class="heading-small text-muted mb-4">User information</h6>
          <div class="pl-lg-4">
          <div class="row">
          <div id="imagePreview" style="background-image: url('<?php echo
$row["DP"] ?>');"></div>
                <label class="label1"for="imageUpload">Upload Profile Picture</label>
                <input type="file" name= "imageUpload"id="imageUpload"
accept=".jpg, .jpeg, .png">

        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
        <script>
                function readURL(input) {
                        if (input.files && input.files[0]) {
                                var reader = new FileReader();
                                reader.onload = function(e) {
                                        var image = new Image();
                                        image.src = e.target.result;
                                        image.onload = function() {
                                                // Create a new HTML5 Canvas element to
store the cropped image
                                                var canvas =
document.createElement('canvas');
                                                var context = canvas.getContext('2d');

                                                // Calculate the square dimensions
                                                var squareSize = Math.min(this.width,
this.height);
                                                var offsetX = (this.width - squareSize) / 2;
                                                var offsetY = (this.height - squareSize) / 2;

                                                // Draw the cropped image to the canvas
                                                canvas.width = squareSize;
                                                canvas.height = squareSize;
                                                context.drawImage(this, offsetX, offsetY,
squareSize, squareSize, 0, 0, squareSize, squareSize);
```

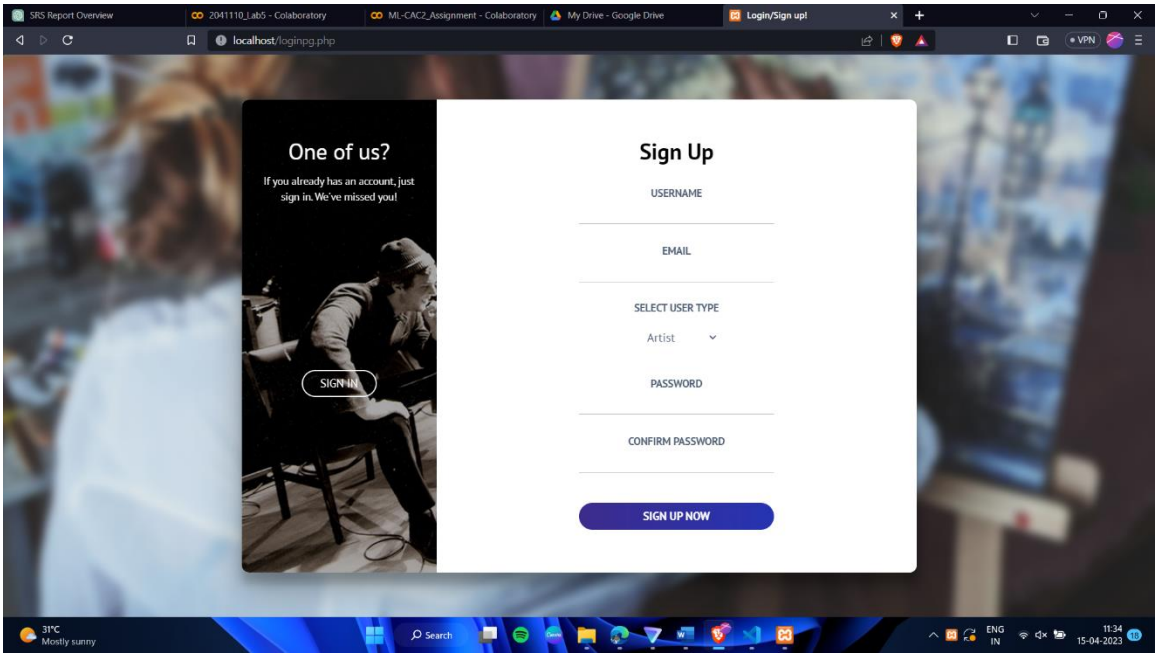
```
                                        // Update the image preview
                                        $('#imagePreview').css('background-image',
'url('+canvas.toDataURL()+')');
                                        $('#imagePreview').hide();
                                        $('#imagePreview').fadeIn(650);

                                        // Initialize Cropper.js with the cropped
image
                                        $('#imagePreview').cropper({
                                                viewMode: 1,
                                                autoCropArea: 1,
                                                aspectRatio: 1/1,
                                                strict: true,
                                                background: false,
                                                guides: false,
                                                highlight: false,
                                                dragCrop: false,
                                                cropBoxMovable: false,
                                                cropBoxResizable: false
                                        });
                                };
                        }
                        reader.readAsDataURL(input.files[0]);
                }
        }
        $("#imageUpload").change(function() {
                readURL(this);
        });
</script>
   </div>

    <div class="row">
      <div class="col-lg-6">
        <div class="form-group focused">
          <label class="form-control-label" for="name">Full name</label>
          <input type="text" name = "name"id="name" class="form-control form-
control-alternative" value="<?php echo  $row["name"] ?>"placeholder="<?php echo
$row["name"] ?>">
        </div>
      </div>

    </div>
    <div class="row">

     <div class="col-lg-6">
```

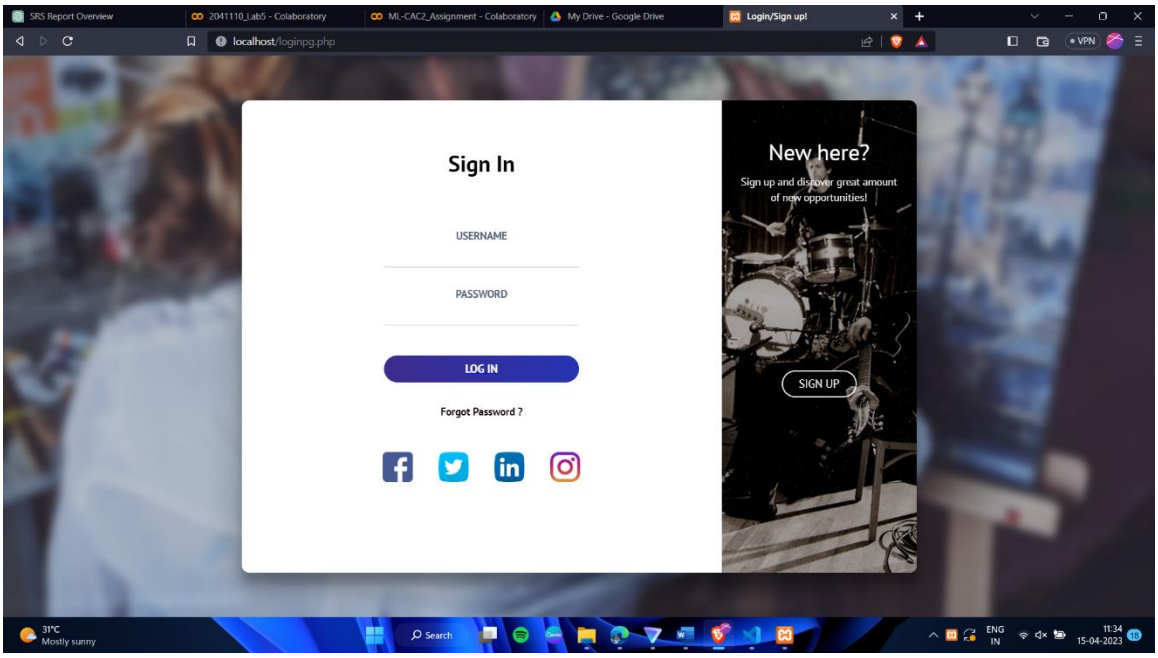
```
                <div class="form-group">
                  <label class="form-control-label" for="phone_no">Phone no.</label>
                  <input type="text" name="phone_no" id="phone_no" class="form-control
form-control-alternative" value="<?php echo  $row["phone_no"] ?>"placeholder="<?php
echo  $row["phone_no"] ?>">
                </div>
              </div>
            </div>
          </div>
          <hr class="my-4">
          <!-- Address -->
          <h6 class="heading-small text-muted mb-4">Contact information</h6>
          <div class="pl-lg-4">
            <div class="row">
              <div class="col-lg-6">
                <div class="form-group focused">
                  <label class="form-control-label" for="email">Email Address</label>
                  <input id="email" name="email" class="form-control form-control-
alternative" value="<?php echo  $row["email"] ?>" placeholder="<?php echo
$row["email"] ?>" type="text" DISABLED>
                </div>
              </div>
            </div>
            <div class="row">
              <div class="col-lg-6">
                <div class="form-group focused">
                  <label class="form-control-label" for="city">City</label>
                  <input type="text" name="city"id="city" class="form-control form-
control-alternative" value="<?php echo  $row["city"] ?>" placeholder="<?php echo
$row["city"] ?>">
                </div>
              </div>

            </div>
          </div>
          <hr class="my-4">
          <!-- Description -->
          <h6 class="heading-small text-muted mb-4">About me</h6>
          <div class="pl-lg-4">
            <div class="form-group focused">
              <label>About Me</label>
              <textarea name="AboutMe" id = "AboutMe" rows="4" class="form-control
form-control-alternative" placeholder="A few words about you ..."><?php echo
$row["AboutMe"] ?></textarea>
            </div>
            <button type="submit" name="update" class="btn btn-info">Save</button>
```

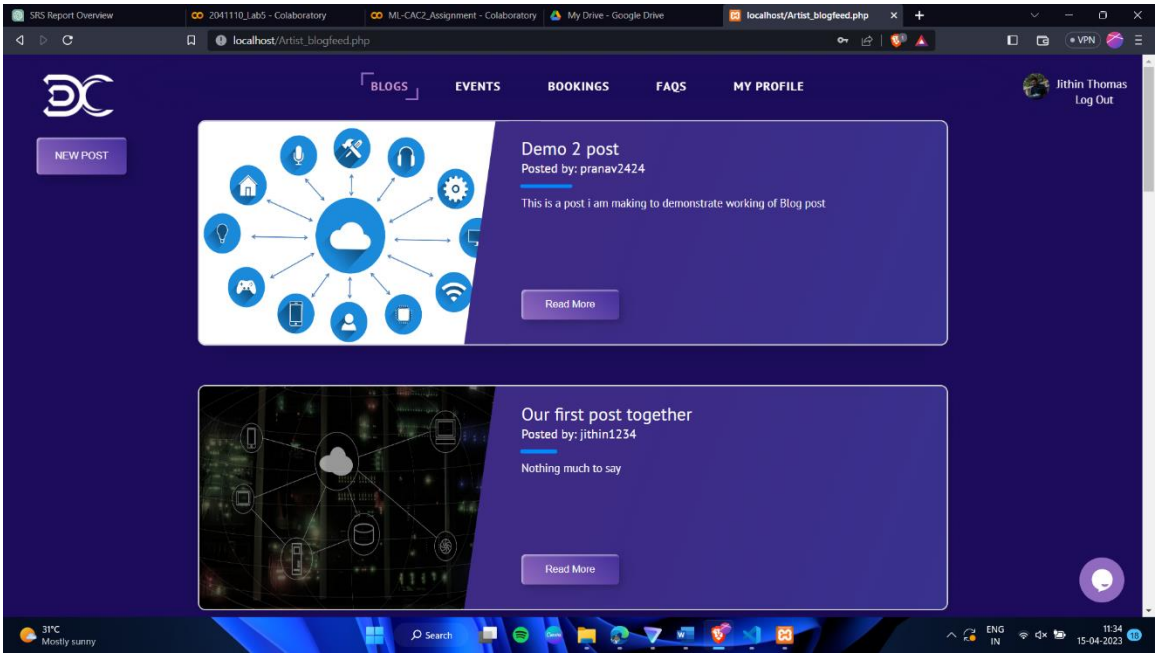
```
            </div>
           </form>
          </div>
         </div>
        </div>
       </div>
      </div>
     </div>
     <footer class="footer">
      <div class="row align-items-center justify-content-xl-between">
       <div class="col-xl-6 m-auto text-center">
        <div class="copyright">
         <p>Copyright by Dextrocity</p>
        </div>
       </div>
      </div>
     </footer>
    </body>
    <?php
     }
    }
    else {
     header("Location: ErrorMessage.html");
    }
     }
    else
    {

        header("Location: ErrorMessage.html");

        exit();

    }
    ?>
```

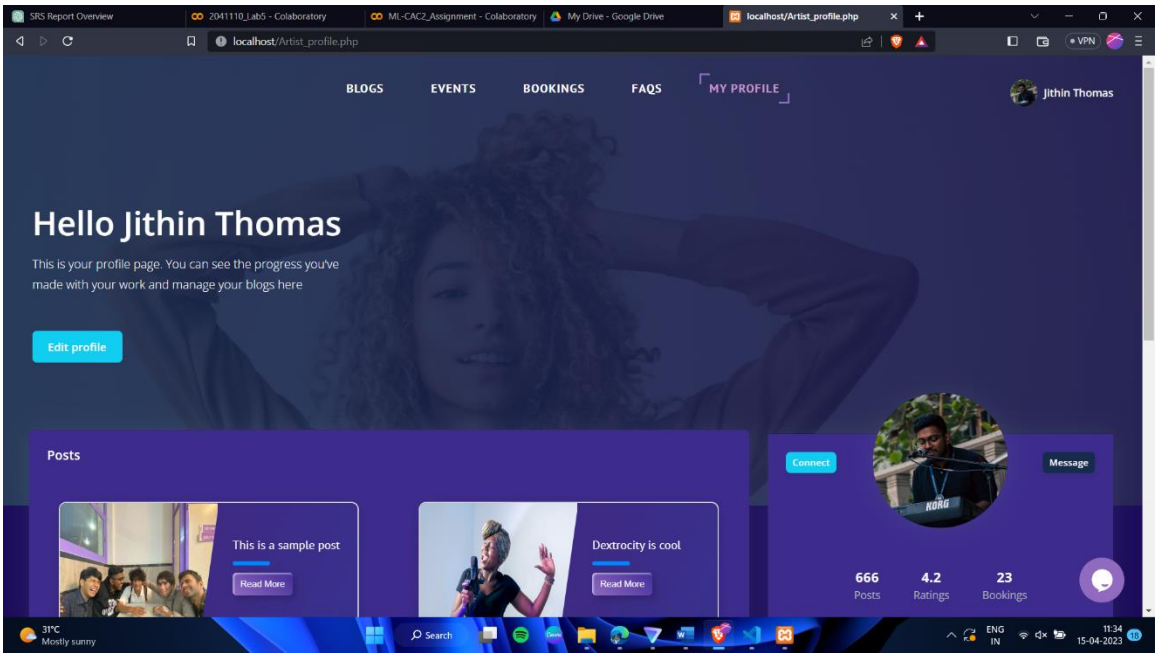
4.3 SCREENSHOTS

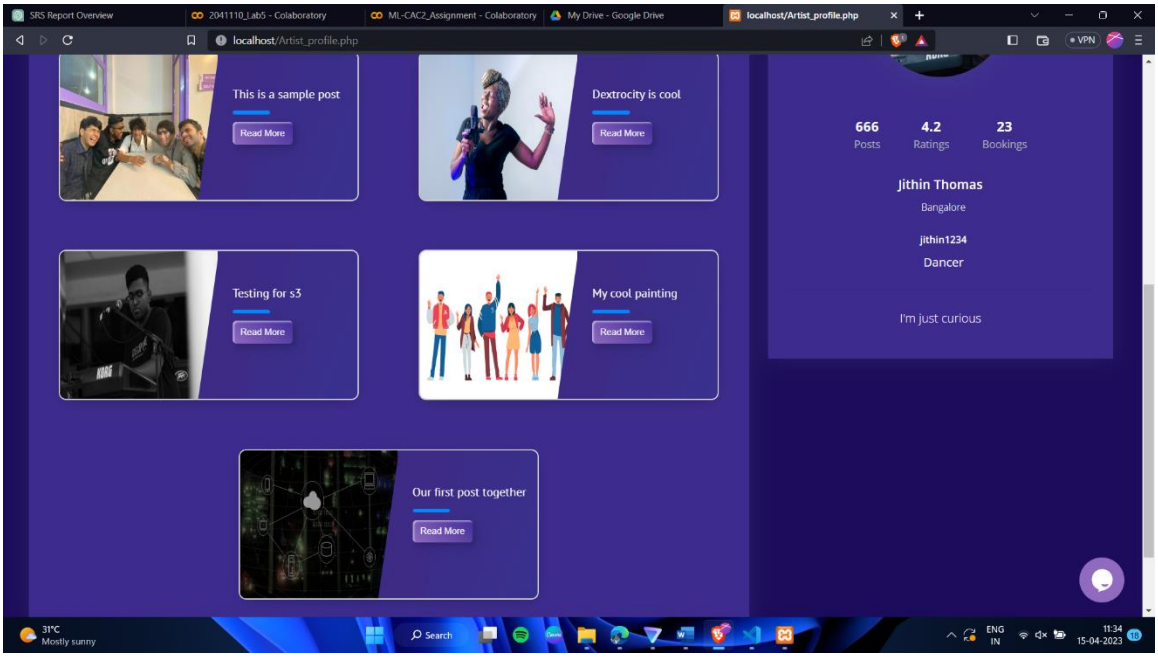


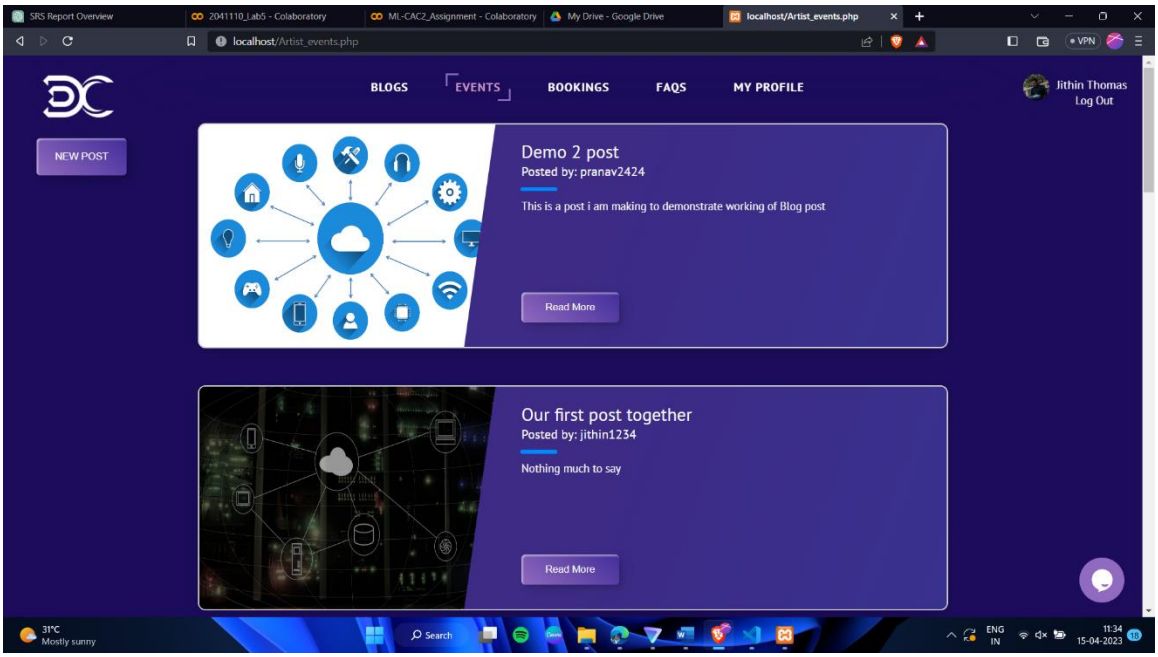


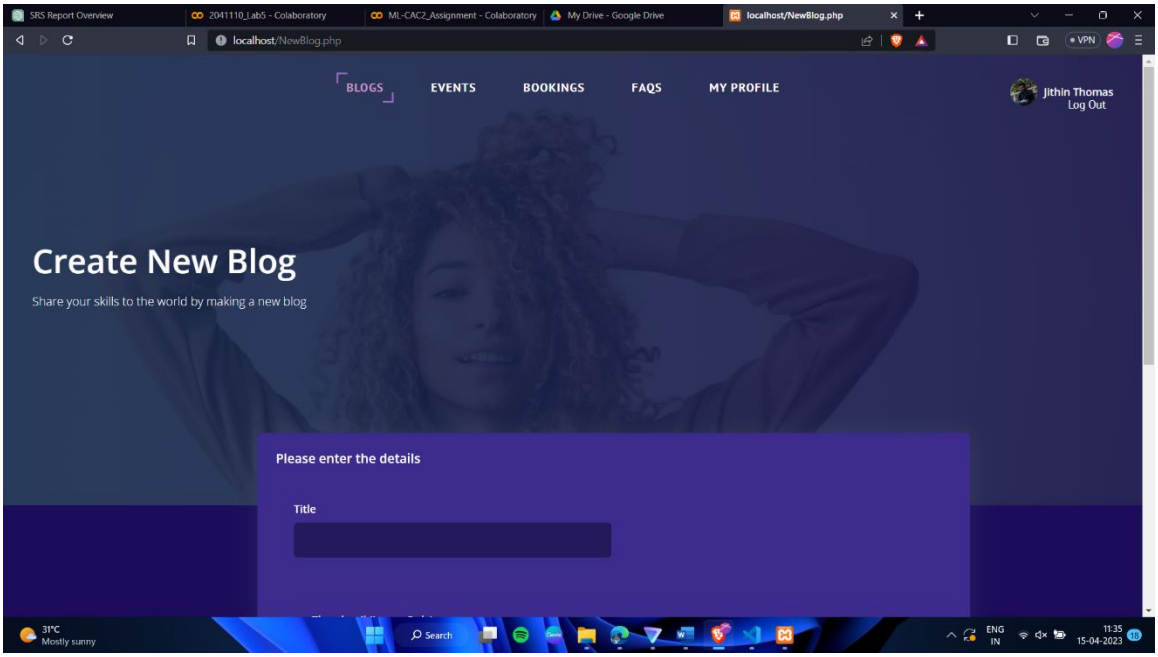

Sign In
USERNAME
PASSWORD
LOG IN
Forgot Password ?
New here?
Sign up and discover great amount of new opportunities!
SIGN UP

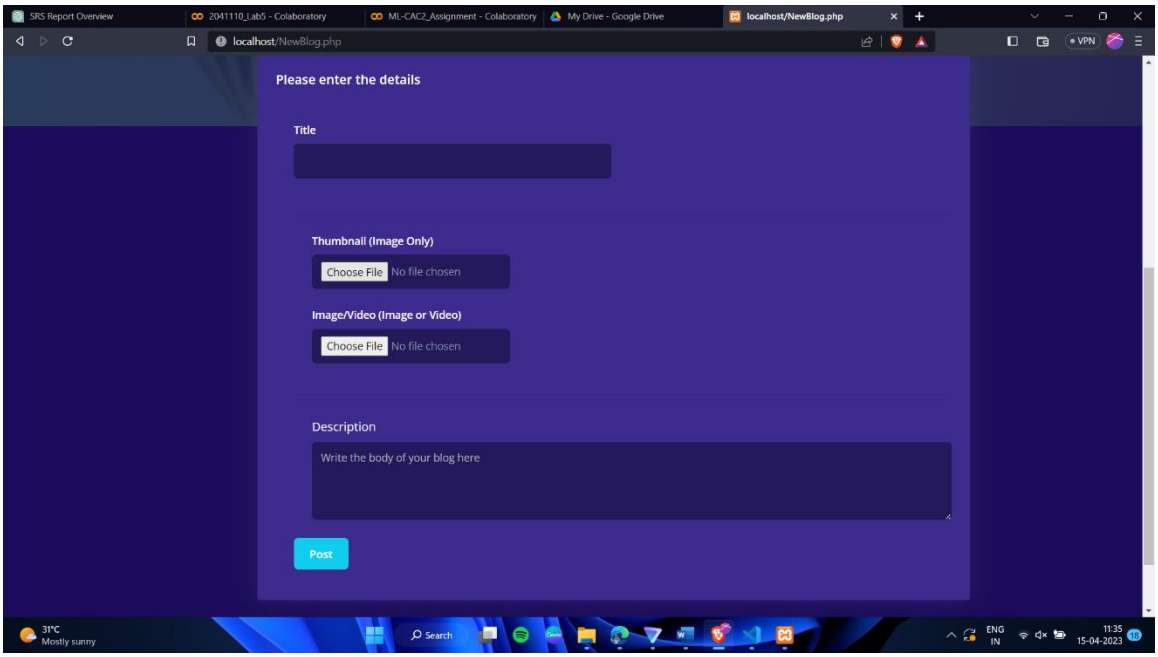

BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
NEW POST
Demo 2 post
Posted by: pranav2424
This is a post i am making to demonstrate working of Blog post
Read More
Our first post together
Posted by: jithin1234
Nothing much to say
Read More

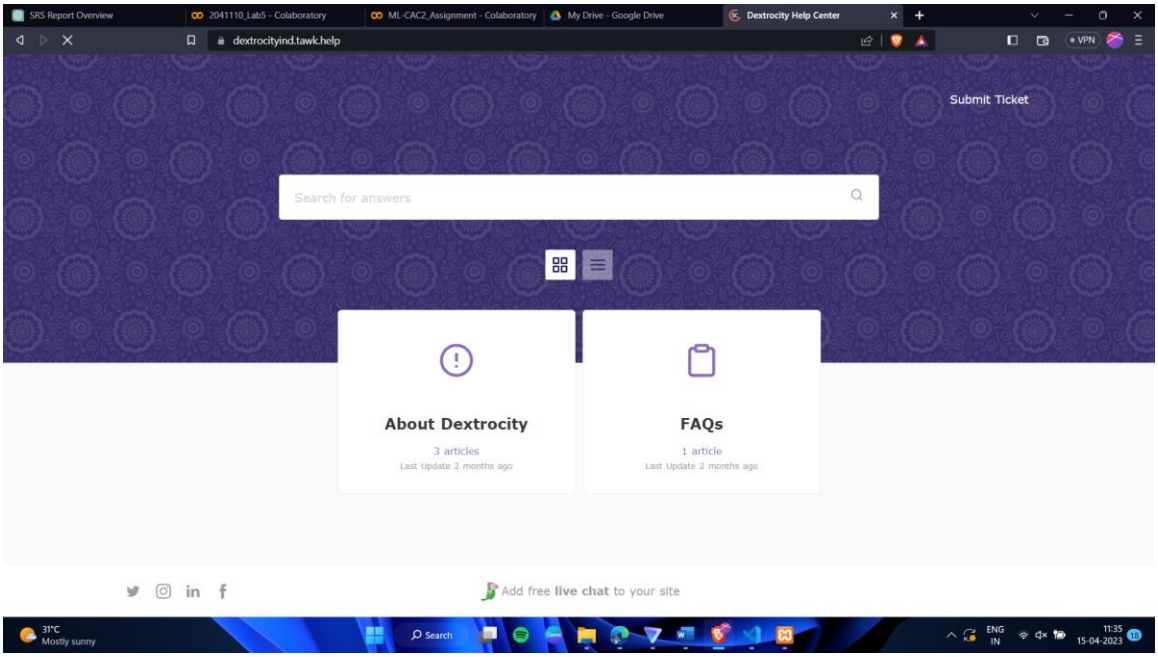

BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Hello Jithin Thomas
This is your profile page. You can see the progress you've made with your work and manage your blogs here
Edit profile
Posts
This is a sample post
Read More
Dextrocity is cool
Read More
Connect
Message
666
Posts
4.2
Ratings
23
Bookings

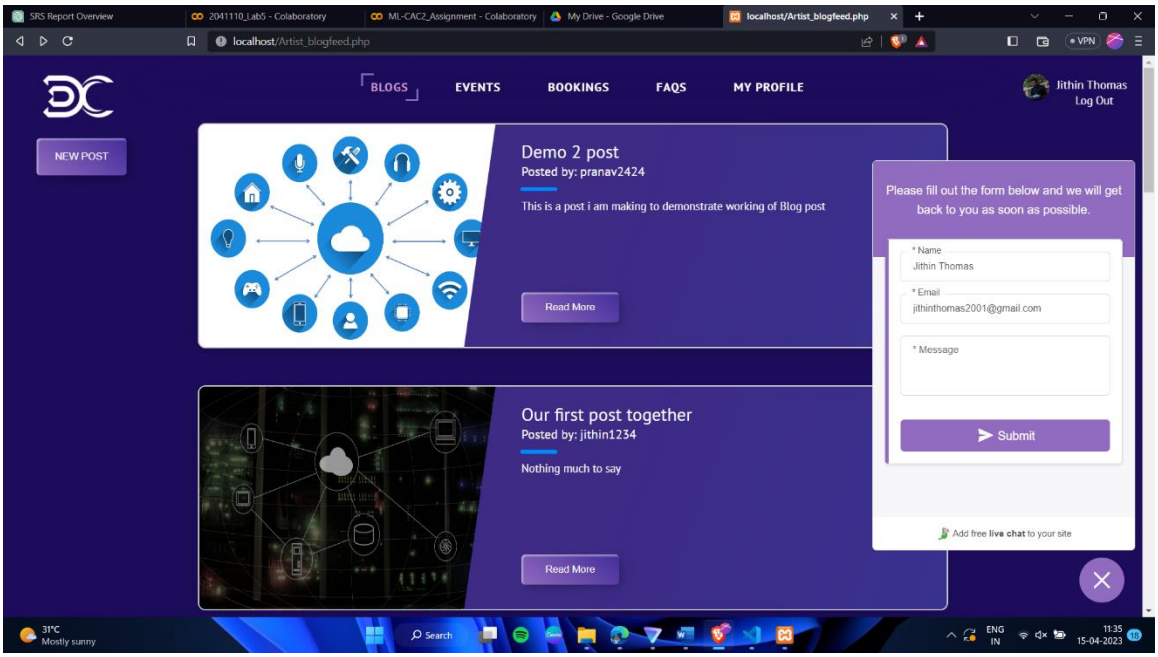

This is a sample post
Read More
Dextrocity is cool
Read More
666
Posts
4.2
Ratings
23
Bookings
Jithin Thomas
Bangalore
jithin1234
Dancer
I'm just curious
Testing for s3
Read More
My cool painting
Read More
Our first post together
Read More

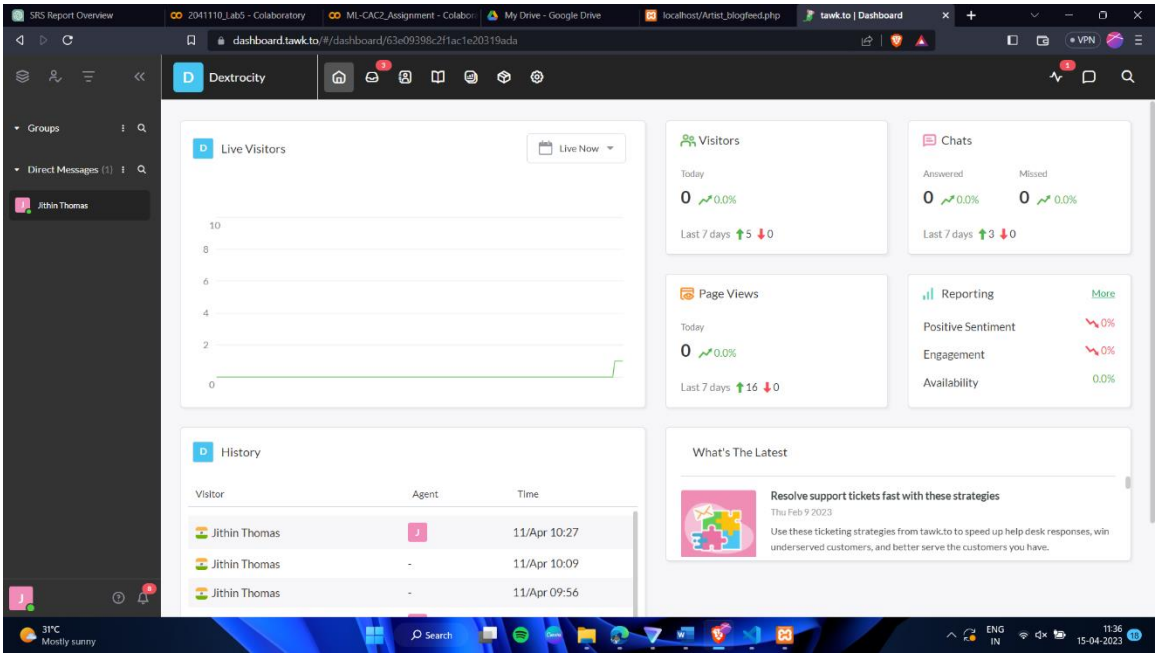

BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
NEW POST
Demo 2 post
Posted by: pranav2424
This is a post i am making to demonstrate working of Blog post
Read More
Our first post together
Posted by: jithin1234
Nothing much to say
Read More

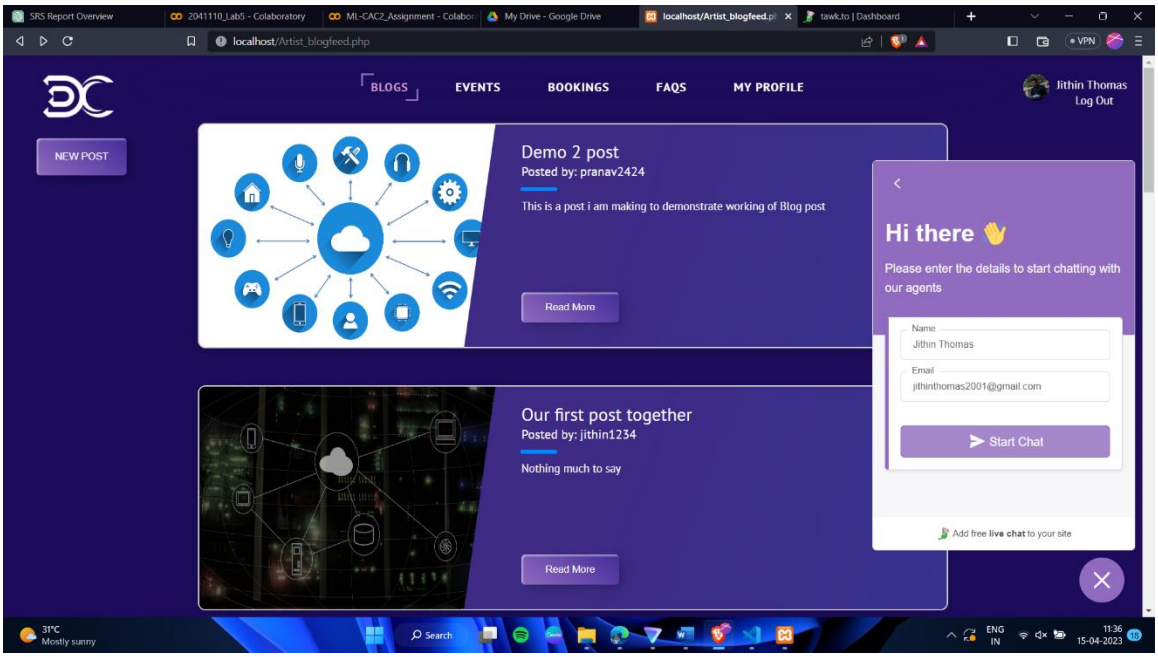

BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
Create New Blog
Share your skills to the world by making a new blog
Please enter the details
Title

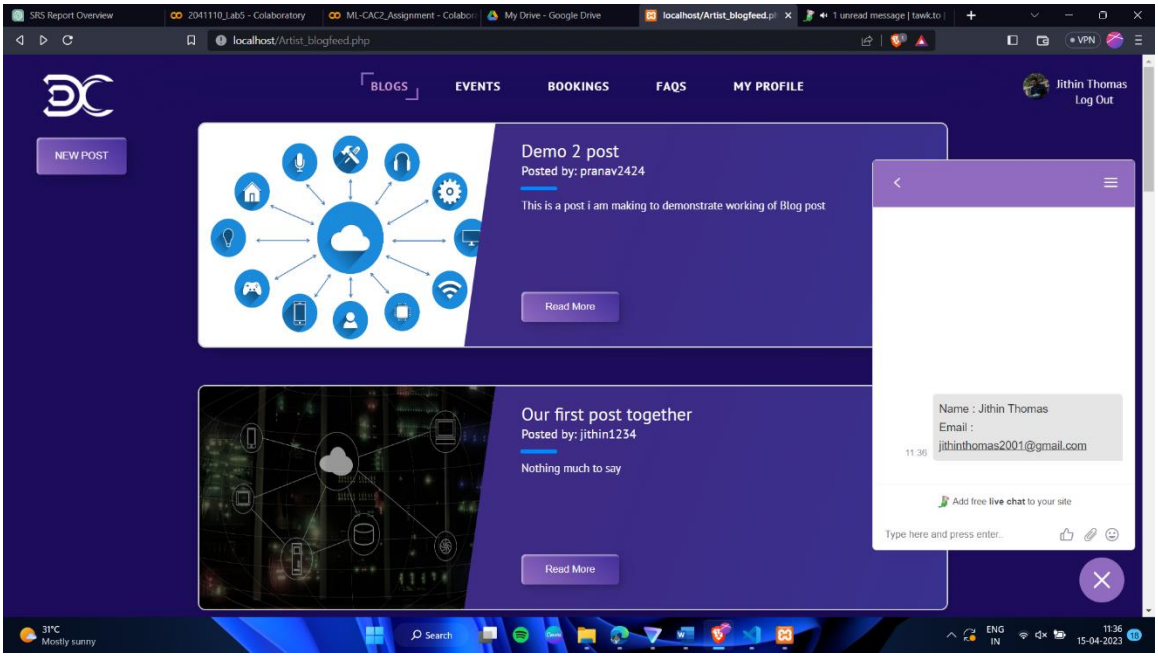

Please enter the details
Title
Thumbnail (Image Only)
Choose File No file chosen
Image/Video (Image or Video)
Choose File No file chosen
Description
Write the body of your blog here
Post

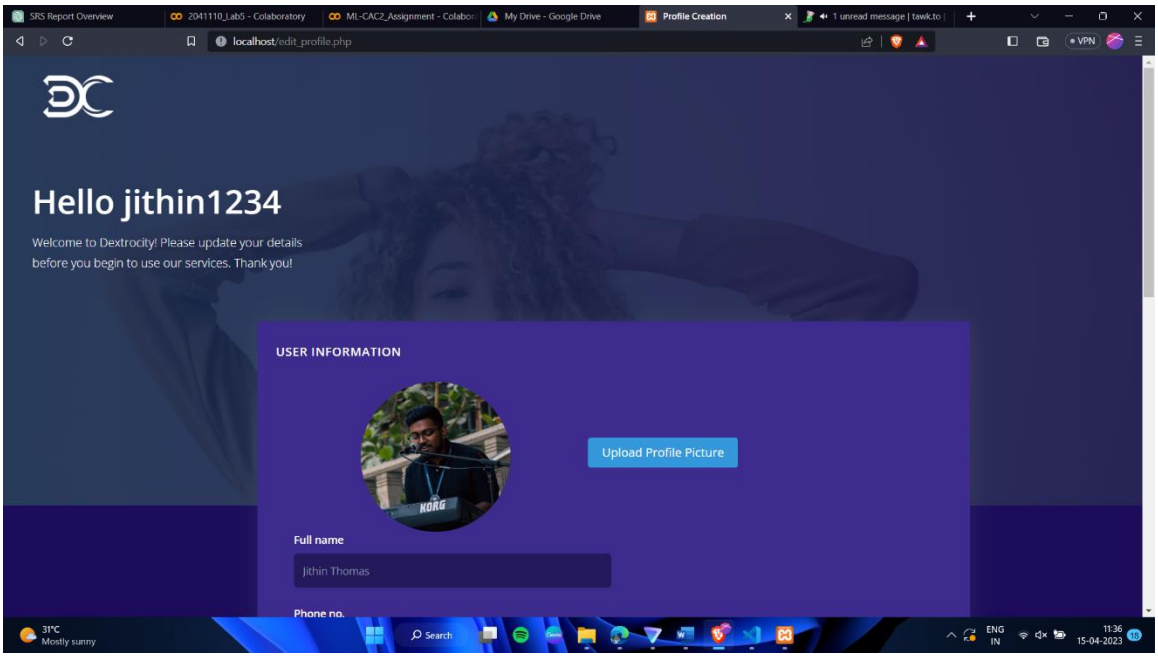

Submit Ticket
Search for answers
About Dextrocity
3 articles
Last Update 2 months ago
FAQs
1 article
Last Update 2 months ago
Add free live chat to your site

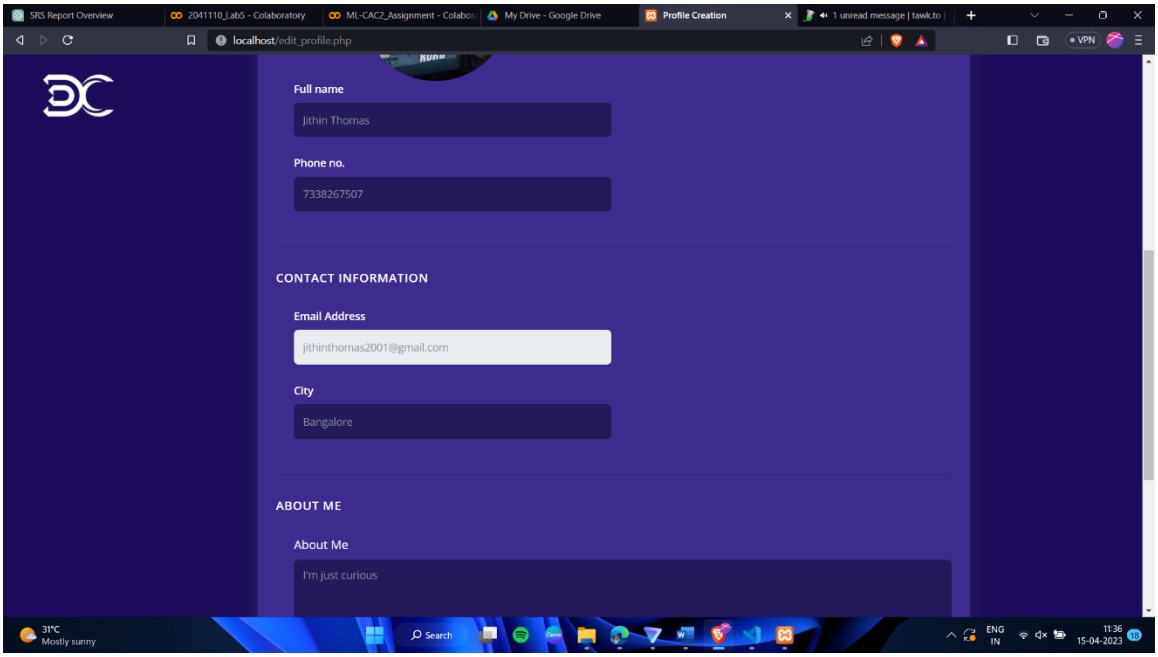

BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
NEW POST
Demo 2 post
Posted by: pranav2424
This is a post i am making to demonstrate working of Blog post
Read More
Please fill out the form below and we will get back to you as soon as possible.
* Name
Jithin Thomas
* Email
jithinthomas2001@gmail.com
* Message
Submit
Our first post together
Posted by: jithin1234
Nothing much to say
Read More

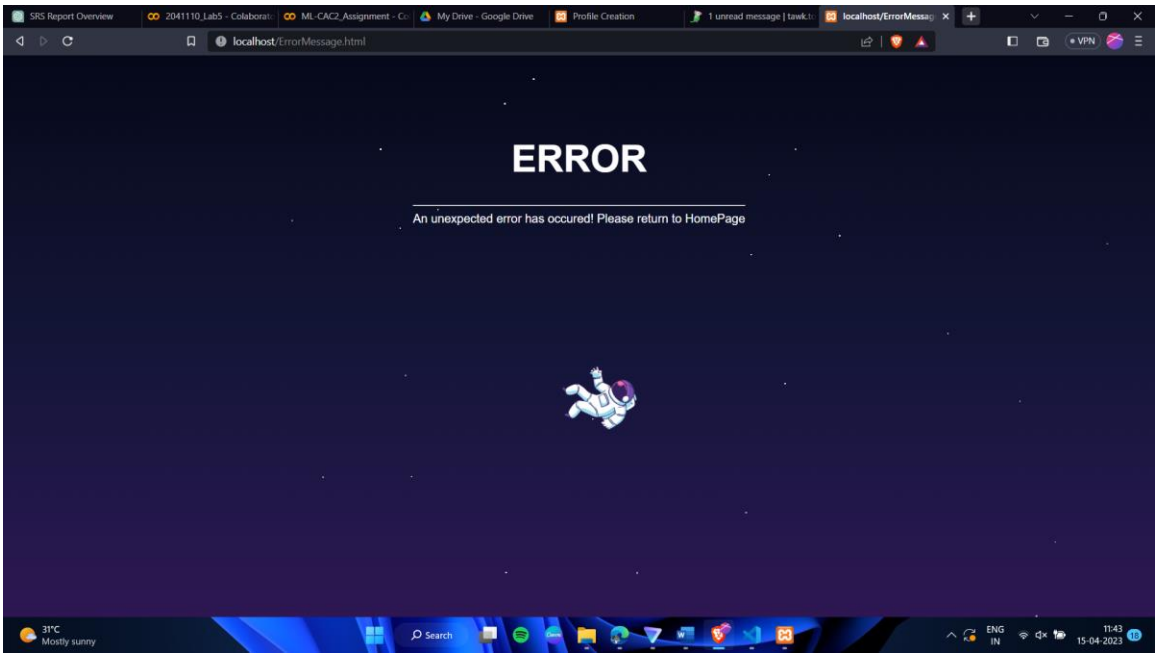

Dextrocity
Live Visitors
Visitors
Chats
Page Views
Reporting
History
What's The Latest
Resolve support tickets fast with these strategies


localhost/Artist_blogfeed.php
BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
NEW POST
Demo 2 post
Posted by: pranav2424
This is a post i am making to demonstrate working of Blog post
Read More
Our first post together
Posted by: jithin1234
Nothing much to say
Read More
Hi there
Please enter the details to start chatting with our agents
Name
Jithin Thomas
Email
jithinthomas2001@gmail.com
Start Chat
Add free live chat to your site


localhost/Artist_blogfeed.php
BLOGS
EVENTS
BOOKINGS
FAQS
MY PROFILE
Jithin Thomas
Log Out
NEW POST
Demo 2 post
Posted by: pranav2424
This is a post i am making to demonstrate working of Blog post
Read More
Our first post together
Posted by: jithin1234
Nothing much to say
Read More
Name : Jithin Thomas
Email :
jithinthomas2001@gmail.com
Add free live chat to your site
Type here and press enter..


Hello jithin1234
Welcome to Dextrocity! Please update your details
before you begin to use our services. Thank you!
USER INFORMATION
Upload Profile Picture
Full name
Jithin Thomas
Phone no.


Full name
Jithin Thomas
Phone no.
7338267507
CONTACT INFORMATION
Email Address
jithinthomas2001@gmail.com
City
Bangalore
ABOUT ME
About Me
I'm just curious


SRS Report Overview
2041110_Lab5 - Colaborat
ML-CAC2_Assignment - Co
My Drive - Google Drive
Profile Creation
1 unread message | tawk.to
localhost/ErrorMessag
localhost/ErrorMessage.html
VPN
ERROR
An unexpected error has occured! Please return to HomePage
31°C
Mostly sunny
Search
ENG
IN
11:43
15-04-2023

5. TESTING

5.1 TEST APPROACHES

- The website's need.
- Clearly stated and explicit testing objectives.
- Each user has their own profile.
- Develop a test plan that focuses on rapid-cycle testing.
- Robust software that is meant to test itself is created.
- Use effective formal reviews as a filter before testing.
- Conducted formal technical evaluations to assess the test strategy and test cases' nature, quality, and capability.
- Software testing towards the end

5.1.1 Testing Strategies

During the development and implementation of features or UI elements, the developer conducted unit testing to ensure their proper functioning. This process involved identifying all the units being developed and verifying that they behaved as expected when given different inputs. Given that this application includes modules that focus on statistical analysis of data related to Organic Searches, Paid Searches, and Search Engine Optimization results, it was crucial to ensure that any changes or additional data did not cause significant deviations from expected results. Appropriate actions were taken based on the test results to address any issues found during the testing process.

5.1.2 Functional Testing

Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application.

Functional Testing Process:

Functional testing involves the following steps:

- Identify function that is to be performed.
- Create input data based on the specifications of function.
- Determine the output based on the specifications of function.
- Execute the test case.
- Compare the actual and expected output.

The entire website was tested using the above steps and modifications were done based on the actual and expected output.

5.1.3 Integration Testing

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components is combined to produce output.

Integration testing is of four types:

- Top-down
- Bottom-up
- Sandwich
- Big-Bang

After unit testing each module of our website, it became possible to build an entire website with proper program structure based on the design that was planned for the website initially.

5.2 TEST CASES

Login Page:

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Artist Login with valid Data	1. Go to site 2. Enter Id 3. Enter Password 4. Click Submit	Userid = 100 Password = zxcf	User should Login into an application	As Expected	Pass
TU02	Check Artist Login with invalid Data	1. Go to site 2. Enter Id 3. Enter Password 4. Click Submit	Userid = 100 Password = asdf	User should not Login into an application	As Expected	Pass

Registration Page:

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU03	Check Artist registration with valid Data	1. Go to site 2. Enter the credentials 3. Click Submit	Valid Artist info	User will be registered	As Expected	Pass
TU04	Check Artist registration with invalid Data	1. Go to site 2. Enter the credentials 3. Click Submit	Invalid entry in certain fields	User will not be able to register	As Expected	Pass

Functional Testing:

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU06	Adding a New Post with valid data	1. Go to site 2. Enter the credentials 3. Click Add	Valid post will be registered	Post will be added	As Expected	Pass
TU07	Adding a New Post with invalid data	1. Go to site 2. Enter the credentials 3. Click Add	Invalid info/entry in certain fields	Post will not be added	As Expected	Pass

Integration Testing:

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU08	Adding new event with valid data	4. Go to site 5. Enter the credentials 6. Click Add	Valid event info	New event be inserted	As Expected	Pass
TU09	Adding new event with invalid data	4. Go to site 5. Enter the credentials 6. Click Add	Invalid info/entry in certain fields	Event will not be added	As Expected	Pass

6. CONCLUSION

6.1 DESIGN AND IMPLEMENTATION ISSUES

Debugging bugs, hosting the website, web security threats, performance, speed, and scalability are the main problems concerning the implementation of this website These problems have a significant impact on the website and make it challenging to build. In order for the user to have a good experience when using our website, it was necessary to make sure that a good UI/UX was developed. To find the ideal design, a lot of time and effort was invested. Since the data contains personal information about the user, proper data storage is crucial. A lot of study had to be completed before saving their data.

6.2 ADVANTAGES AND LIMITATIONS

Advantages

This website's main benefit is that it enables a variety of artists to showcase their talents while also earning an income doing so. They are directly exposed to organizations looking for different performers to perform at their events. As a result, both the artist and the organization profit from the website in terms of publicity.

Limitations

The limitations of the website are that it needs to have more security, hosting issues, and data collection. It is difficult to publicize the website on many platforms, and premium features must be implemented. The number of users that use the website will be less as the artists and customers need to know about the website.

6.3 FUTURE SCOPE OF THE PROJECT

The primary objective of Dextrocity is to link artists with clients so that they can perform at their events and earn money for their talent. The project's future scope includes hosting it on other platforms, connecting it with social media links to the website, API's, and a chat function where customers and artists can communicate with each other. The major feature that would be incorporated in the future is the premium feature, which is a subscription for artists and consumers to gain access to all content and read all facts about the artist/customer that non-premium members do not have access to.

REFERENCES

1. Automate the Boring Stuff with Python: Practical Programming for Total Beginners (1st Ed.), Al Sweigart, No Starch Press, 2015.

2. Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin Series), Robert C. Martin, PHI; First edition, 2017.

3. Soft Skills: The software developer's life manual, John Sonmez, Manning Publications; 1st edition, 2014.

4. Code Complete: A Practical Handbook of Software Construction, Steve McConnell, Microsoft Press; 2nd edition, 2004.

5. Programming Pearls, Jon Bentley, Addison-Wesley Professional; 2nd edition, 1999.

6. GEEKFORGEEKS – Code references: https://www.geeksforgeeks.org/

7. CODEPEN – UI/UX design reference https://codepen.io/

8 . LUCIDCHARTS- Diagram construction https://www.lucidchart.com/

9. FIGMA – UI/UX modelling/prototyping