AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FALL 2023-24

ADVANCE DATABASE MANAGEMENT SYSTEM [A]

PROJECT REPORT

A PHOTOGRAPHY CLUB MANAGEMENT SYSTEM

Group Information

NAME	ID	CONTRIBUTION	TOPIC
SUMAYEA KHATUN	20-43837-2	25%	Introduction, Scenario Description, ER Diagram, Normalization
MD. RAIYAN AL SULTAN	19-41639-3	25%	User Interface, Query Writing, Relational Algebra
NAZIM UDDIN	19-41112-2	25%	Proposal, Class Diagram, Conclusion, Future Work
MD TUIHN REZA	18-39074-3	25%	Content, Use Case Diagram, Activity Diagram, Schema Diagram, Table Creation, Create users, Assign roles and Grant privileges, Synonym, Report writing

Table OF Content

Introduction

The Photography Club Management System is here to elevate experience within our photography community. In this user-friendly platform, we've streamlined the management process, allowing you to focus more on capturing moments and less on administrative hassles. Key features include easy member registration, event coordination, and a hassle-free hiring system for photographers. Navigate through our intuitive interface to discover upcoming events, connect with fellow members, and explore the diverse talents within our community. Whether club member can be a photographer. If someone looking to hire a skilled lens artist, our system provides a seamless and accessible space for everyone. With the Photography Club Management System, we aim to create a space where creativity flourishes, connections thrive, and the joy of photography takes center stage.

Project Proposal

Background

In the dynamic realm of photography, the need for efficient club management systems has become evident. Photography clubs bring together enthusiasts, professionals, and hobbyists to share knowledge, organize events, and foster a creative community. The "Photography Club Management System" aims to address the challenges associated with coordinating club activities, membership management, and event organization in a seamless and organized manner.

Problem Domain and Root Cause

- Membership Management: Tracking and managing member details, renewals, and participation.
- Event Coordination: Efficiently organizing and promoting photography events, workshops, and exhibitions.
- Resource Management: Handling equipment rentals, studio bookings, and inventory tracking.
- Communication: Ensuring effective communication among club members and administrators.
- Documentation: Managing and organizing photography competition entries, judging criteria, and results.

Objectives

- Efficient Membership Management: Streamline the process of onboarding, renewals, and member communication.
- Seamless Event Coordination: Facilitate the planning, promotion, and execution of photography events.

- Resource Optimization: Manage equipment bookings, studio reservations, and inventory efficiently.
- Enhanced Communication: Provide a platform for effective communication among club members and administrators.
- Streamlined Documentation: Organize photography competition entries, judging criteria, and results systematically.

Solutions

The Photography Club Management System will be developed using a combination of technologies, including JavaScript for interactive web interfaces and Python for backend processing. Key features of the system include:

- Member Management: Track and manage member details, renewals, and communication through a centralized database.
- Event Planning and Promotion: Create, promote, and manage photography events with features for registration and attendance tracking.
- Resource Allocation: Implement a system for equipment bookings, studio reservations, and inventory management.
- Communication Platform: Provide a user-friendly interface for members and administrators to communicate effectively.
- Document Repository: Create a centralized repository for photography competition entries, judging criteria, and results.

Target User and Benefits

The Photography Club Management System is designed for photography clubs and organizations, benefiting:

- Program coordinator Streamlined membership management, event coordination, and resource optimization.
- Members: Easy access to event details, communication with other members, and simplified attendance tracking.
- Program Coordinator: Centralized access to photography competition entries, judging criteria, and results.
- Event Coordinator: Efficient planning, promotion, and execution of photography events.
- Marketing Team: Simplified equipment bookings, studio reservations, and inventory management.

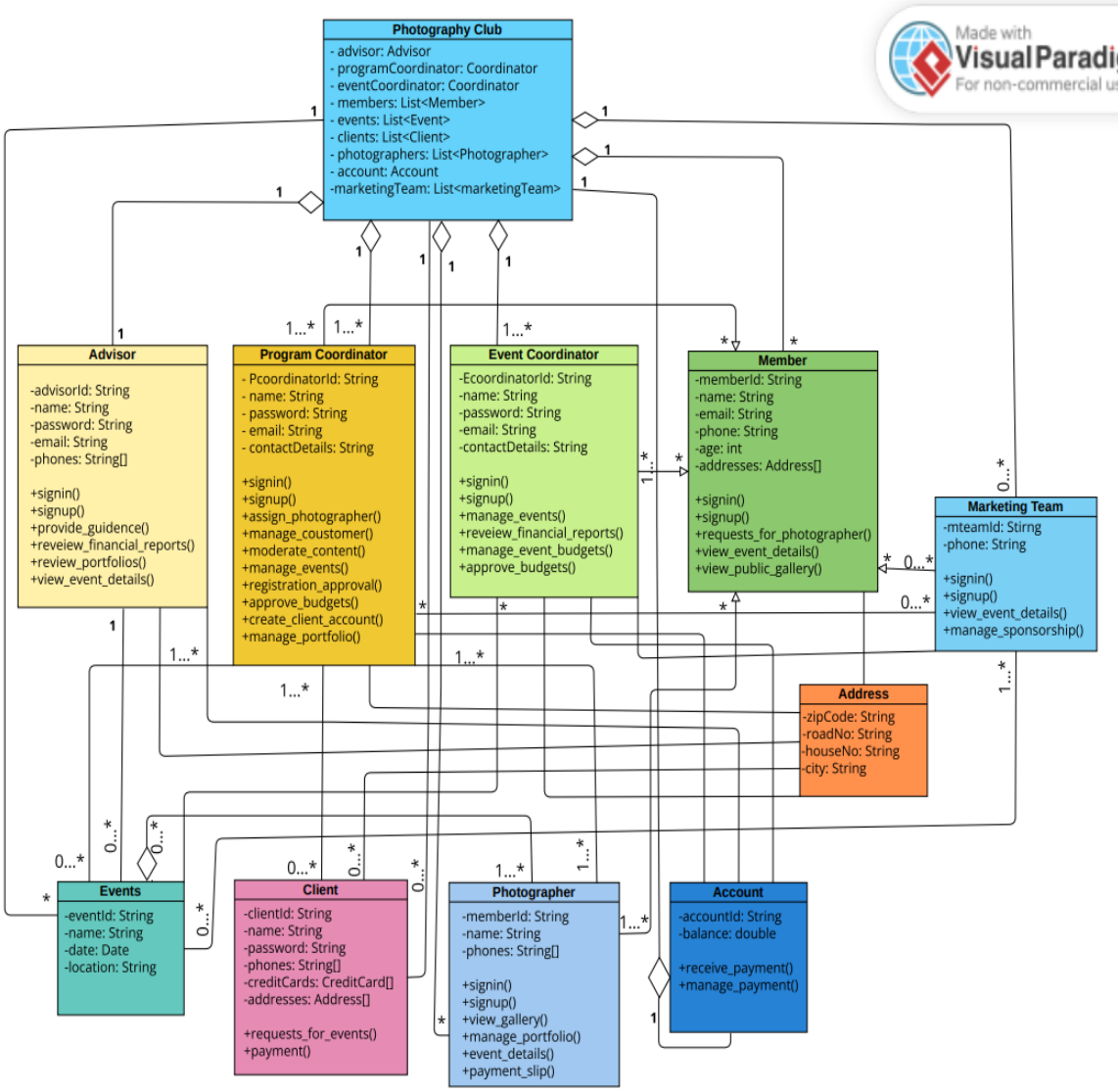
Benefits

- Improved Efficiency: Streamlined processes for membership management, event coordination, and resource allocation.
- Enhanced Communication: A centralized platform for effective communication among club members and administrators.

- Optimized Resource Usage: Efficient handling of equipment bookings, studio reservations, and inventory tracking.
- Data-Driven Decision-Making: Insights from attendance tracking and event analytics for informed decision-making.
- Streamlined Documentation: Centralized repository for photography competition entries, judging criteria, and results.
- User-Friendly Interface: Intuitive interfaces for members and administrators, ensuring ease of use.
- Automated Attendance Tracking: Reduced manual effort with an automated system for tracking member attendance.
- Increased Member Engagement: Easy access to event details, communication features, and simplified processes.
- Customization and Scalability: Tailor the system to fit the unique needs of different photography clubs and scale as necessary.

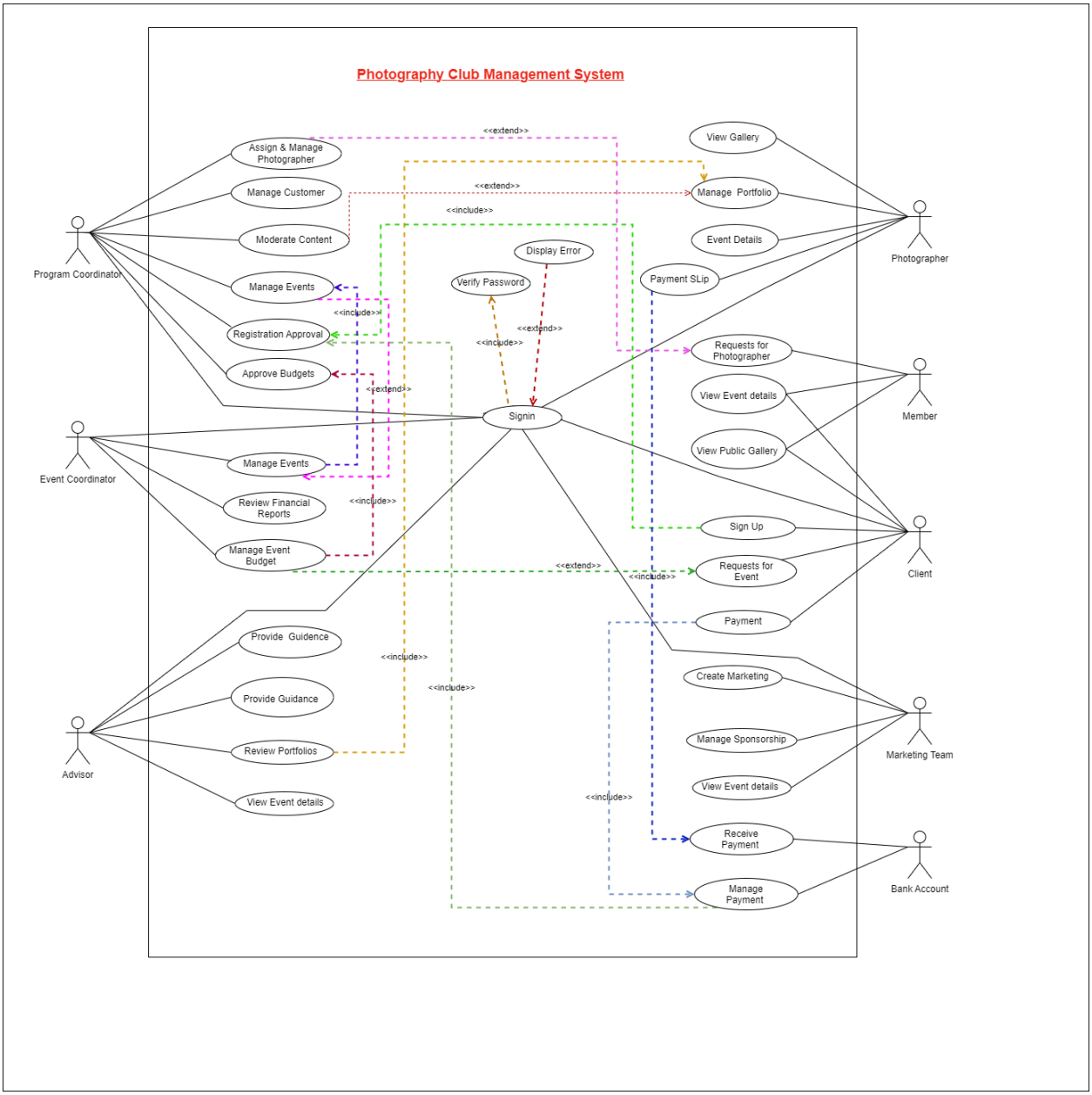
The Photography Club Management System aims to enhance the overall experience for photography enthusiasts and professionals within club settings.

Class Diagram



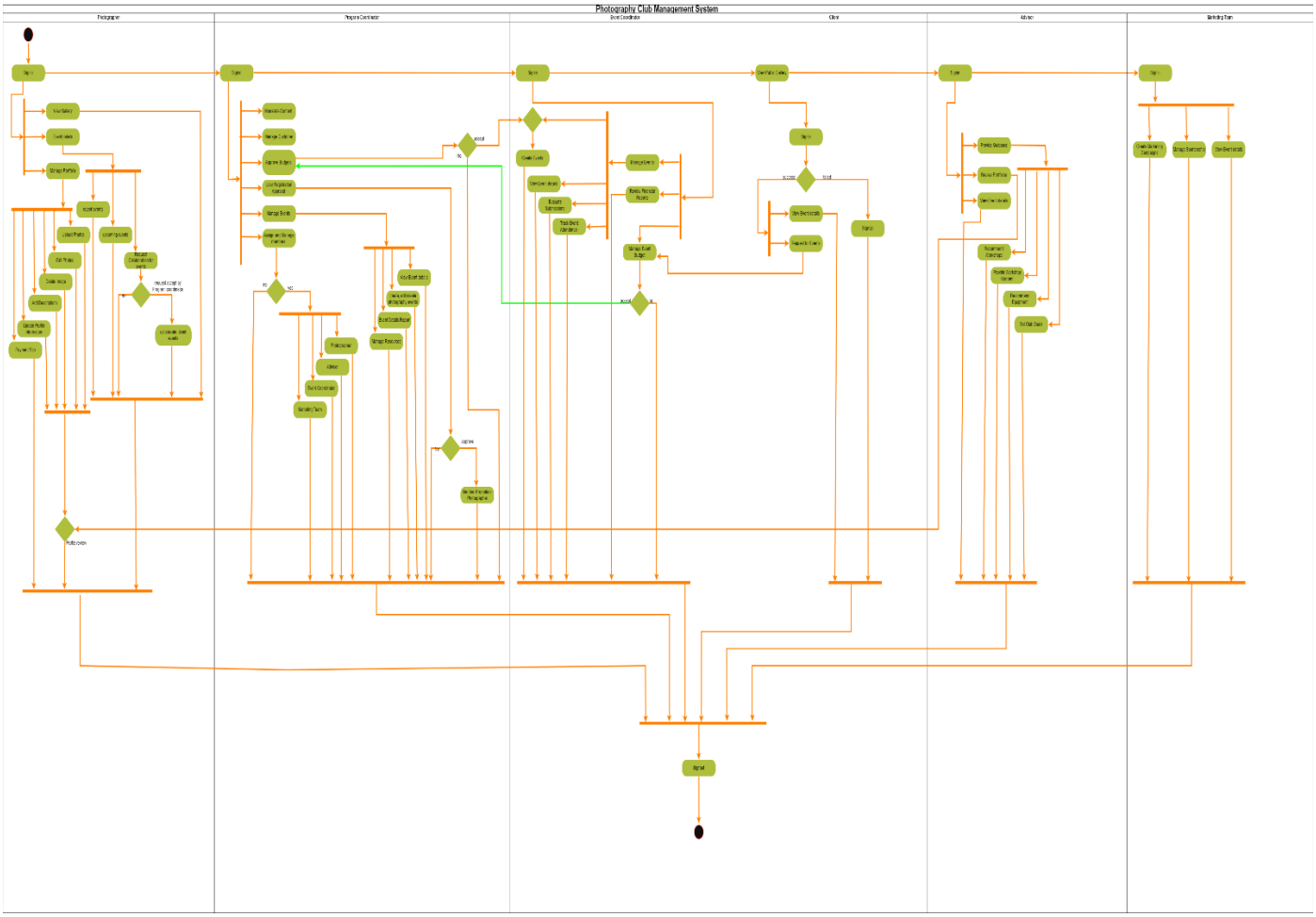
BACK

Use Case Diagram



BACK

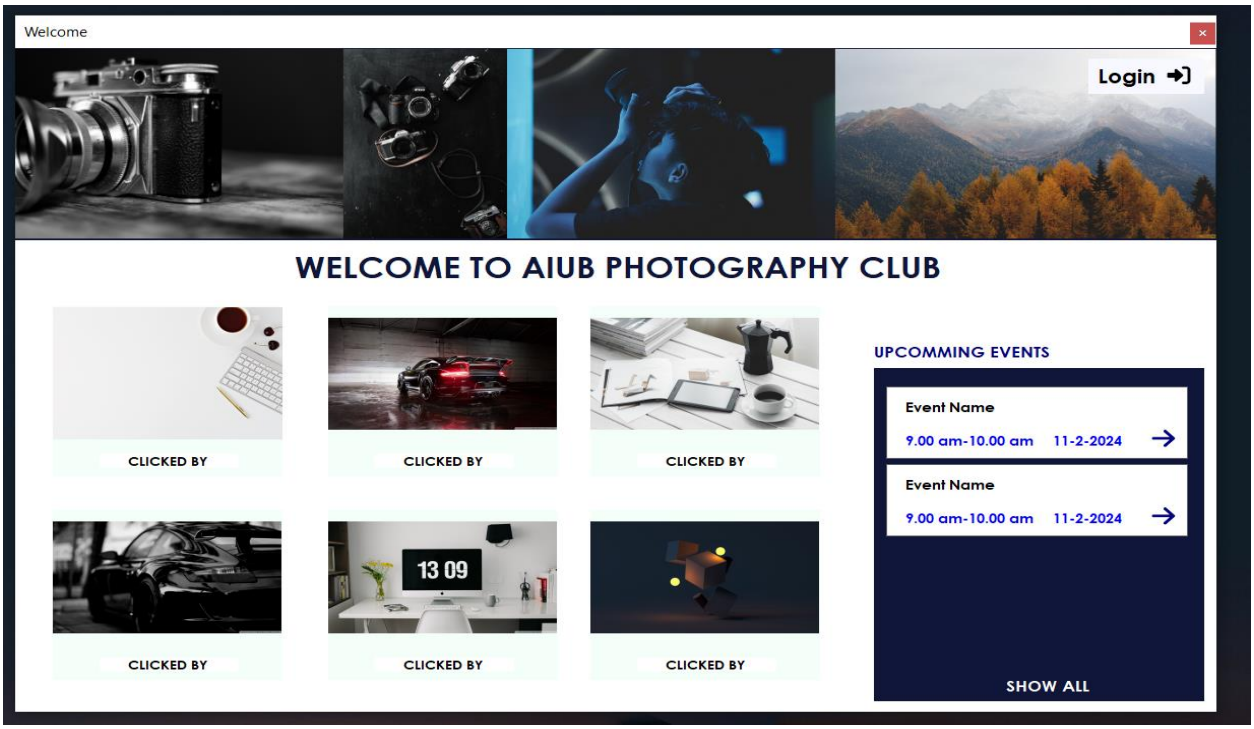
Activity Diagram

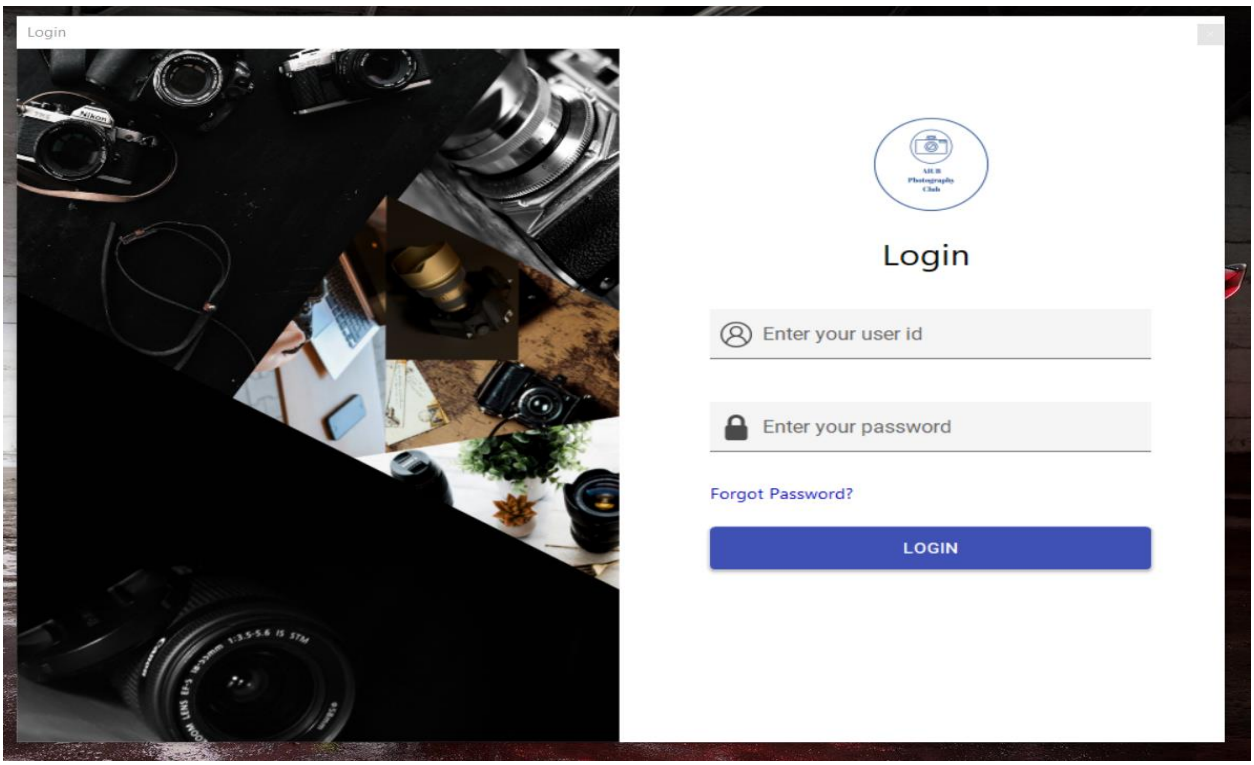


BACK

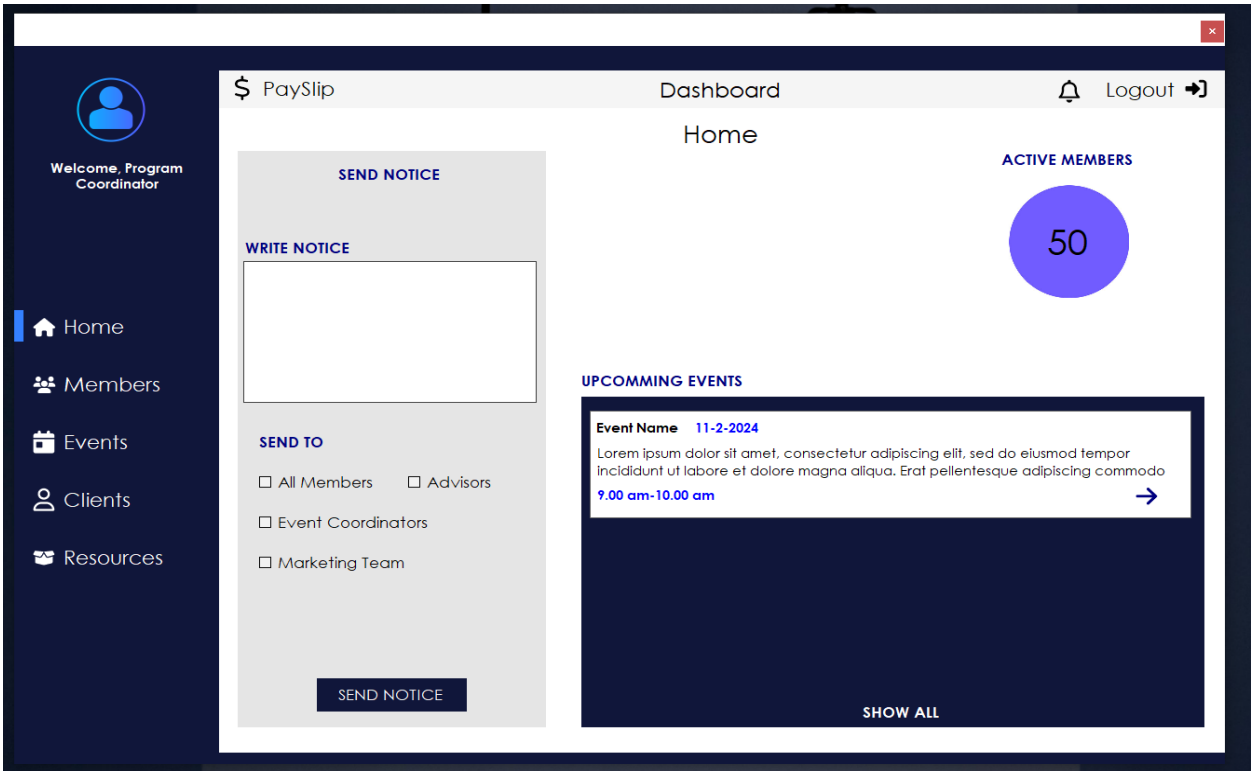
User Interface

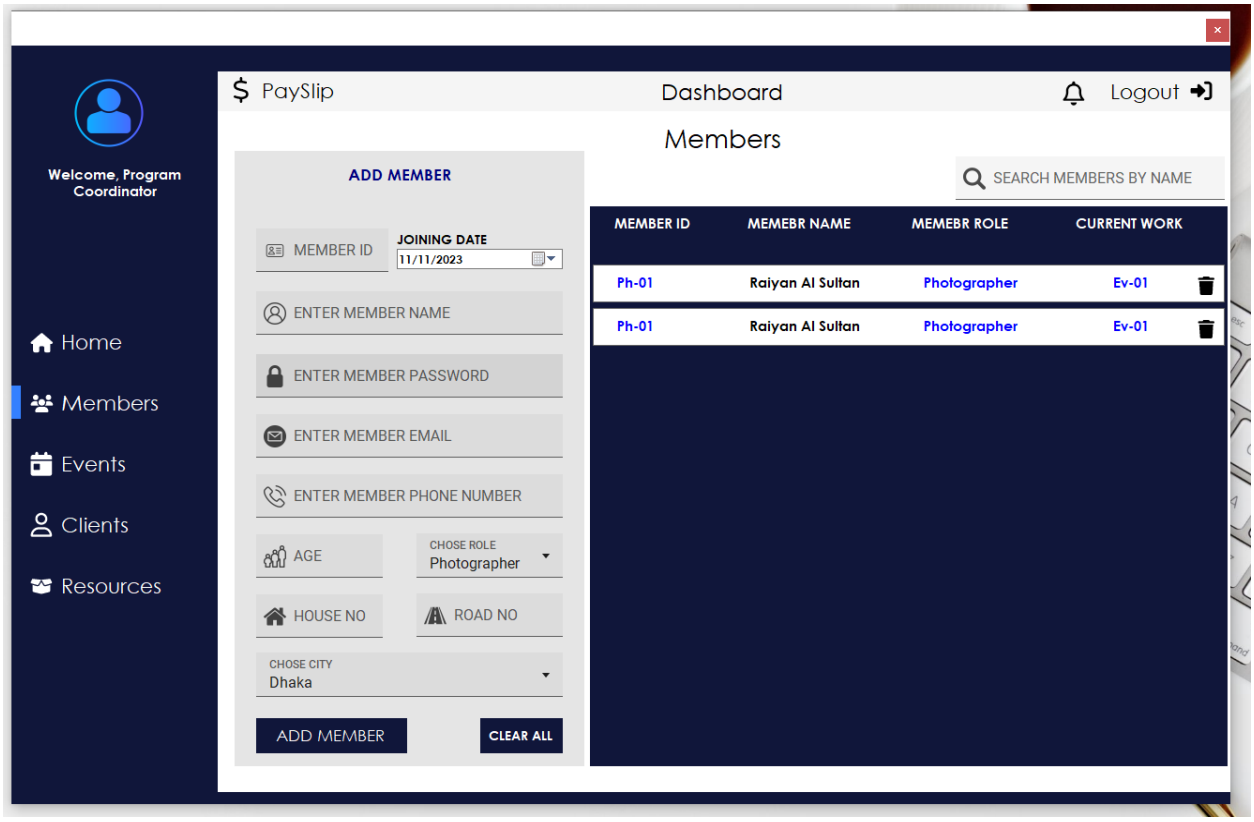
Video Demo: Click for view



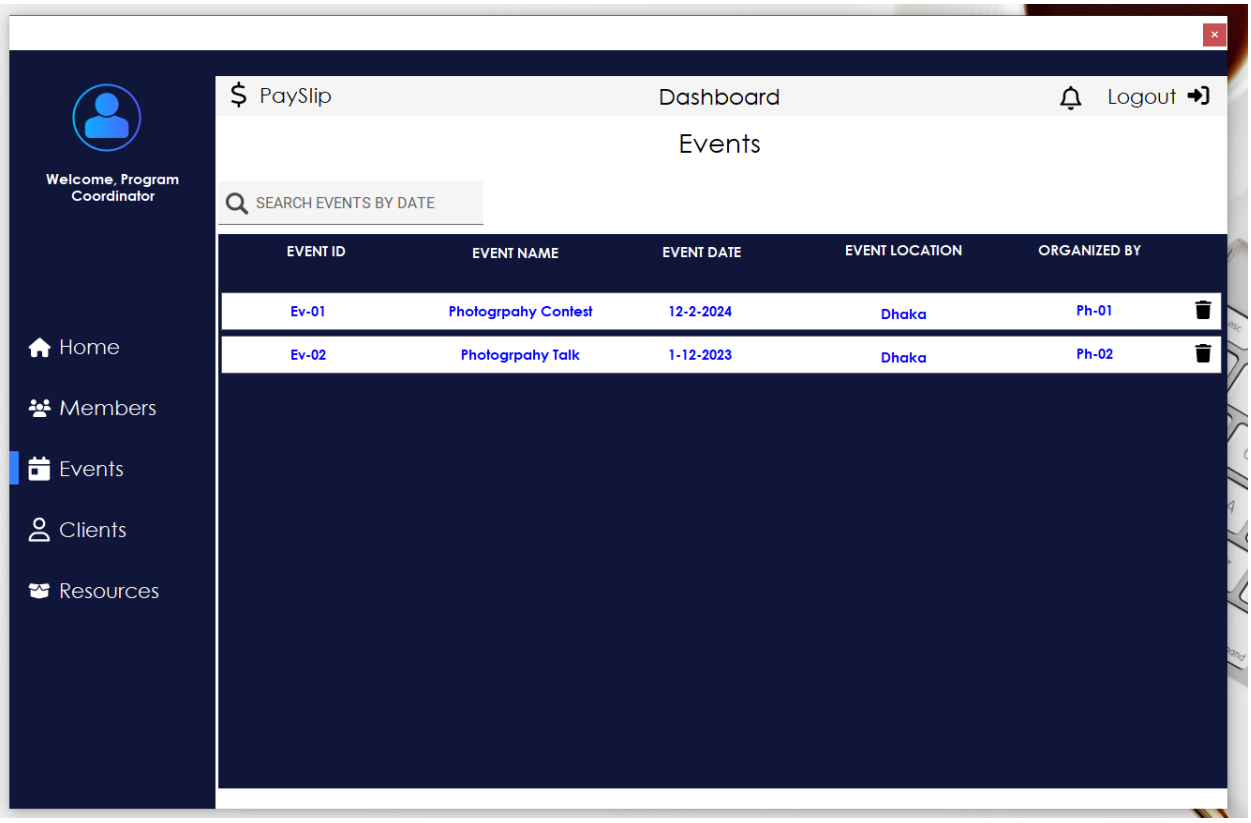


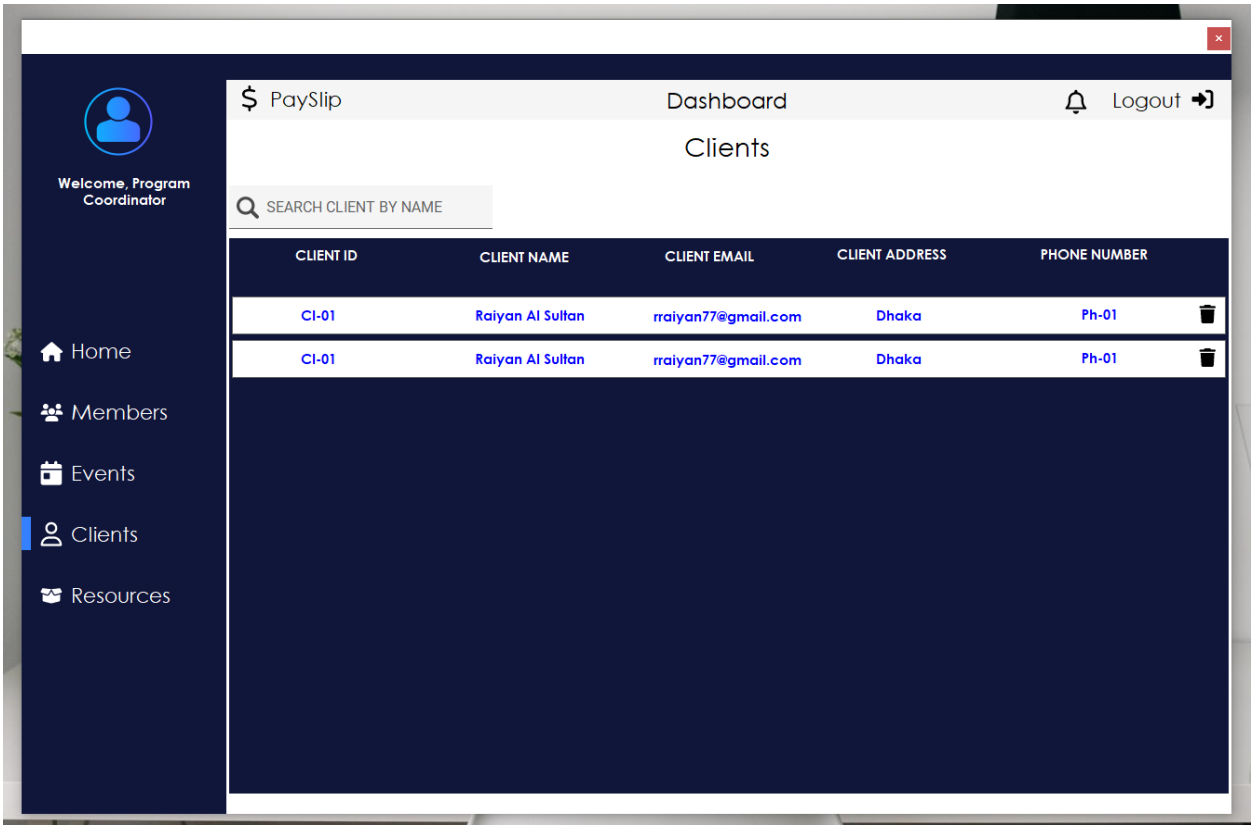
BACK


Welcome, Program Coordinator
Home
Members
Events
Clients
Resources
PaySlip
Dashboard
Logout
Home
SEND NOTICE
WRITE NOTICE
SEND TO
All Members
Advisors
Event Coordinators
Marketing Team
SEND NOTICE
ACTIVE MEMBERS
50
UPCOMMING EVENTS
Event Name 11-2-2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Erat pellentesque adipiscing commodo
9.00 am-10.00 am
SHOW ALL

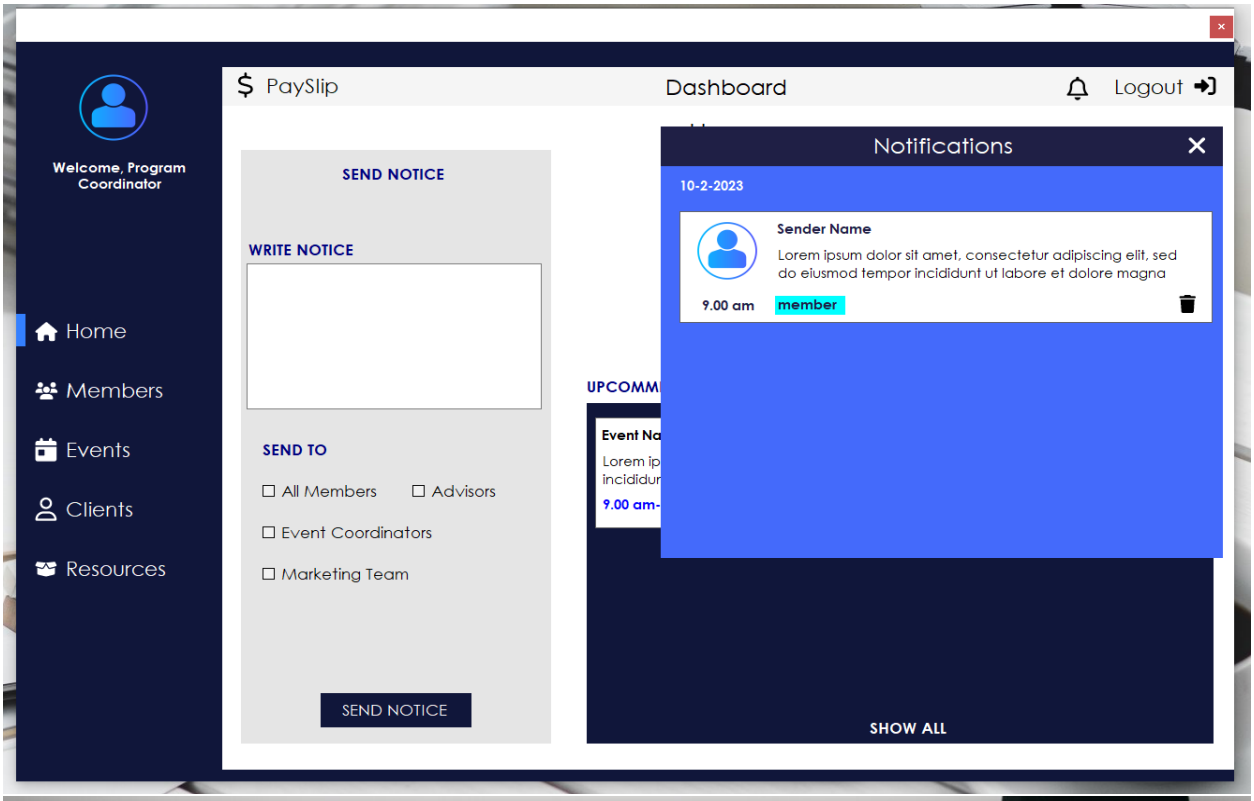

Welcome, Program Coordinator
Home
Members
Events
Clients
Resources
PaySlip
Dashboard
Logout
Members
ADD MEMBER
MEMBER ID
JOINING DATE
11/11/2023
ENTER MEMBER NAME
ENTER MEMBER PASSWORD
ENTER MEMBER EMAIL
ENTER MEMBER PHONE NUMBER
AGE
CHOSE ROLE
Photographer
HOUSE NO
ROAD NO
CHOSE CITY
Dhaka
ADD MEMBER
CLEAR ALL
SEARCH MEMBERS BY NAME
MEMBER ID
MEMEBR NAME
MEMEBR ROLE
CURRENT WORK
Ph-01
Raiyan Al Sultan
Photographer
Ev-01
Ph-01
Raiyan Al Sultan
Photographer
Ev-01

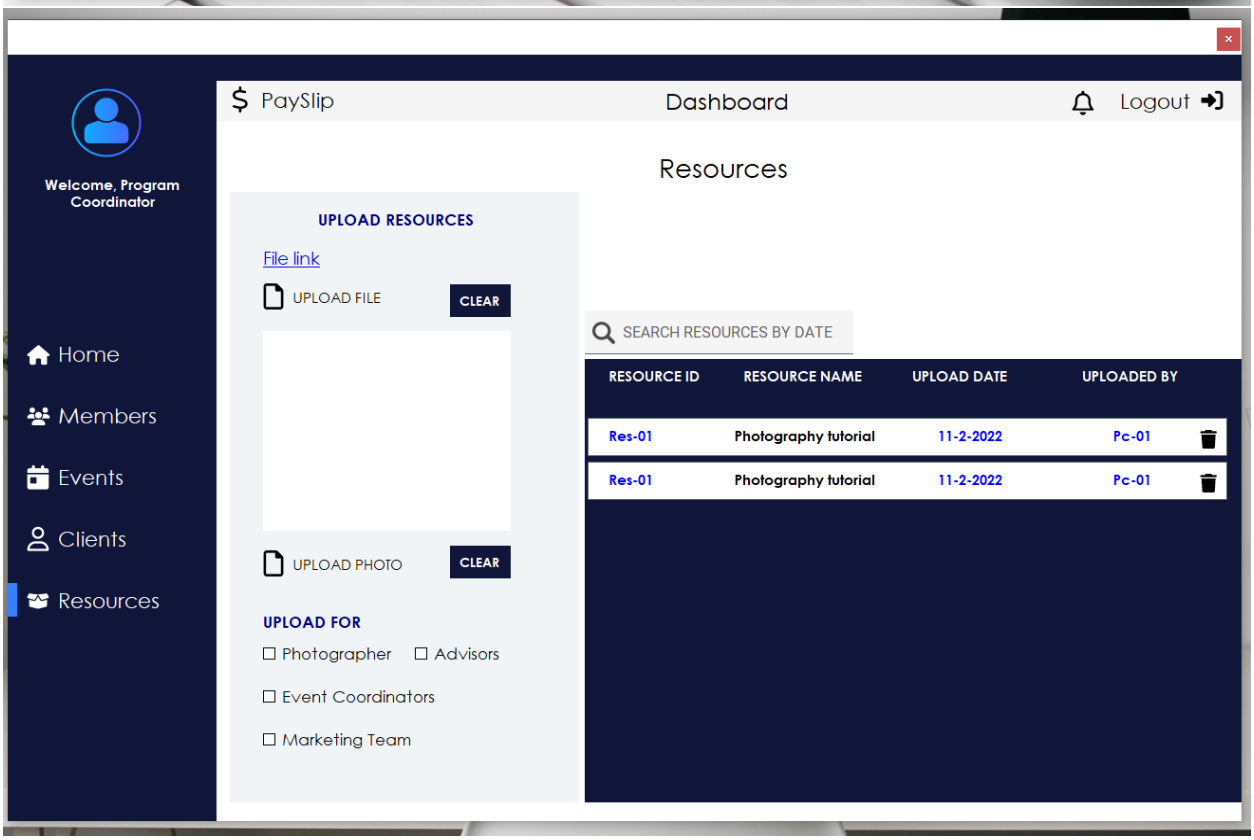
BACK


Welcome, Program Coordinator
PaySlip
Dashboard
Logout
Events
SEARCH EVENTS BY DATE
EVENT ID
EVENT NAME
EVENT DATE
EVENT LOCATION
ORGANIZED BY
Ev-01
Photogrpahy Contest
12-2-2024
Dhaka
Ph-01
Ev-02
Photogrpahy Talk
1-12-2023
Dhaka
Ph-02
Home
Members
Events
Clients
Resources

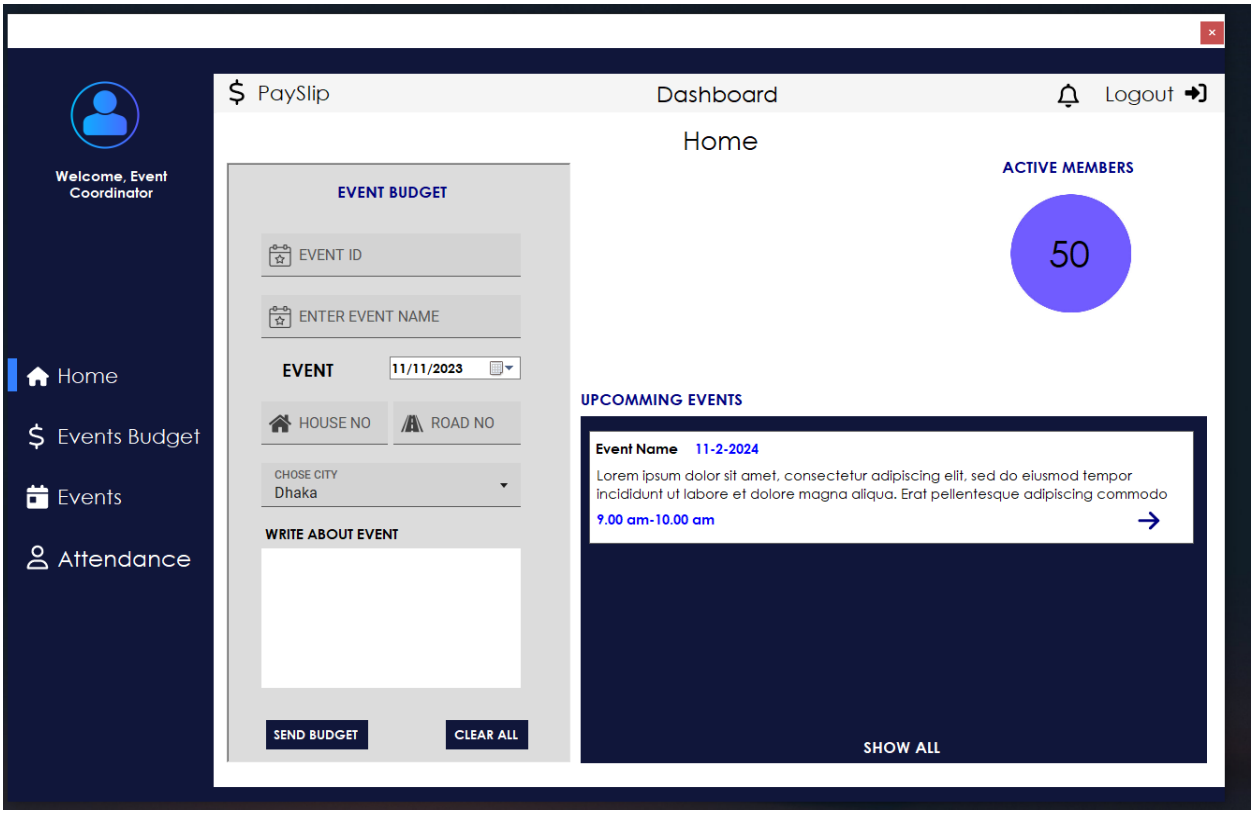

Welcome, Program Coordinator
PaySlip
Dashboard
Logout
Clients
SEARCH CLIENT BY NAME
CLIENT ID
CLIENT NAME
CLIENT EMAIL
CLIENT ADDRESS
PHONE NUMBER
Cl-01
Raiyan Al Sultan
rraiyan77@gmail.com
Dhaka
Ph-01
Cl-01
Raiyan Al Sultan
rraiyan77@gmail.com
Dhaka
Ph-01
Home
Members
Events
Clients
Resources

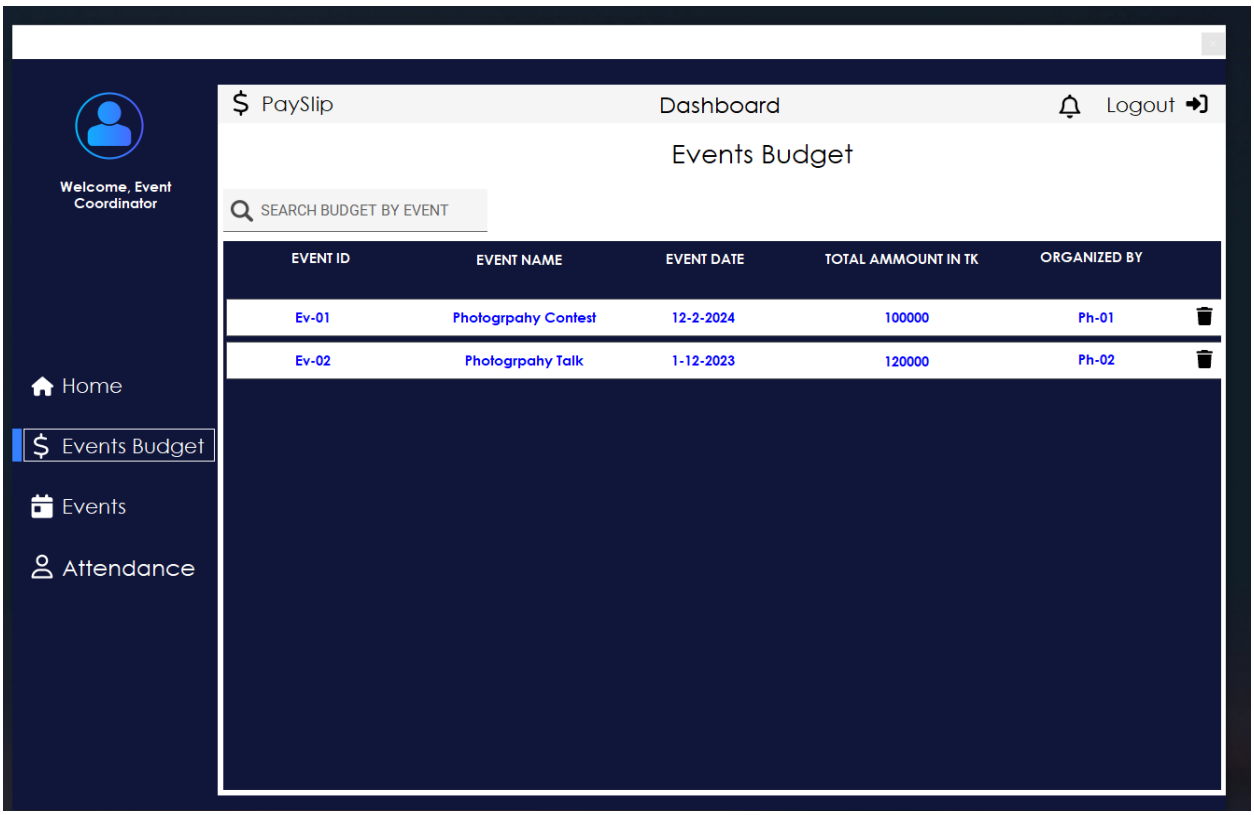
BACK


PaySlip
Dashboard
Logout
Welcome, Program Coordinator
Home
Members
Events
Clients
Resources
SEND NOTICE
WRITE NOTICE
SEND TO
All Members
Advisors
Event Coordinators
Marketing Team
SEND NOTICE
Notifications
10-2-2023
Sender Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
9.00 am
member
SHOW ALL

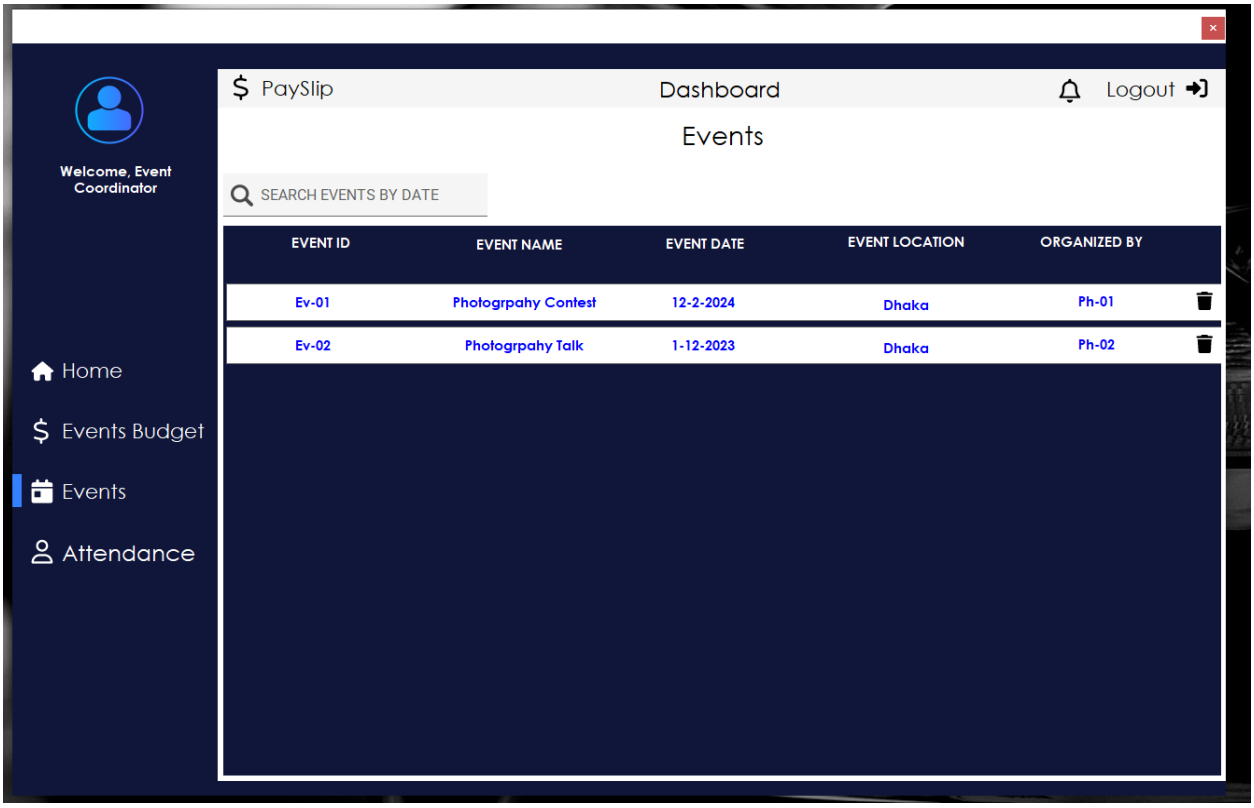

PaySlip
Dashboard
Logout
Welcome, Program Coordinator
Home
Members
Events
Clients
Resources
Resources
UPLOAD RESOURCES
File link
UPLOAD FILE
CLEAR
UPLOAD PHOTO
CLEAR
UPLOAD FOR
Photographer
Advisors
Event Coordinators
Marketing Team
SEARCH RESOURCES BY DATE
RESOURCE ID
RESOURCE NAME
UPLOAD DATE
UPLOADED BY
Res-01
Photography tutorial
11-2-2022
Pc-01
Res-01
Photography tutorial
11-2-2022
Pc-01

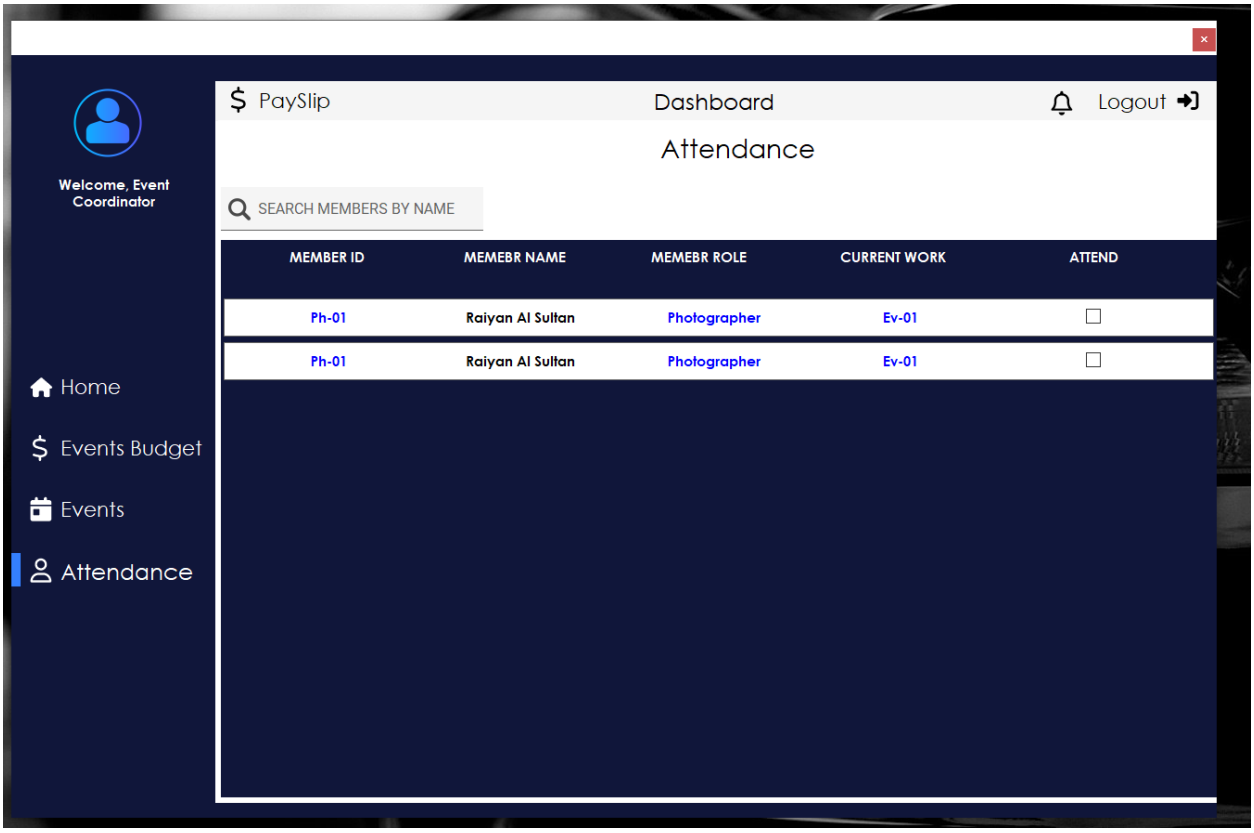
BACK


Welcome, Event Coordinator
PaySlip
Dashboard
Logout
Home
EVENT BUDGET
EVENT ID
ENTER EVENT NAME
EVENT
11/11/2023
HOUSE NO
ROAD NO
CHOSE CITY
Dhaka
WRITE ABOUT EVENT
SEND BUDGET
CLEAR ALL
ACTIVE MEMBERS
50
UPCOMMING EVENTS
Event Name 11-2-2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Erat pellentesque adipiscing commodo
9.00 am-10.00 am
SHOW ALL
Home
Events Budget
Events
Attendance

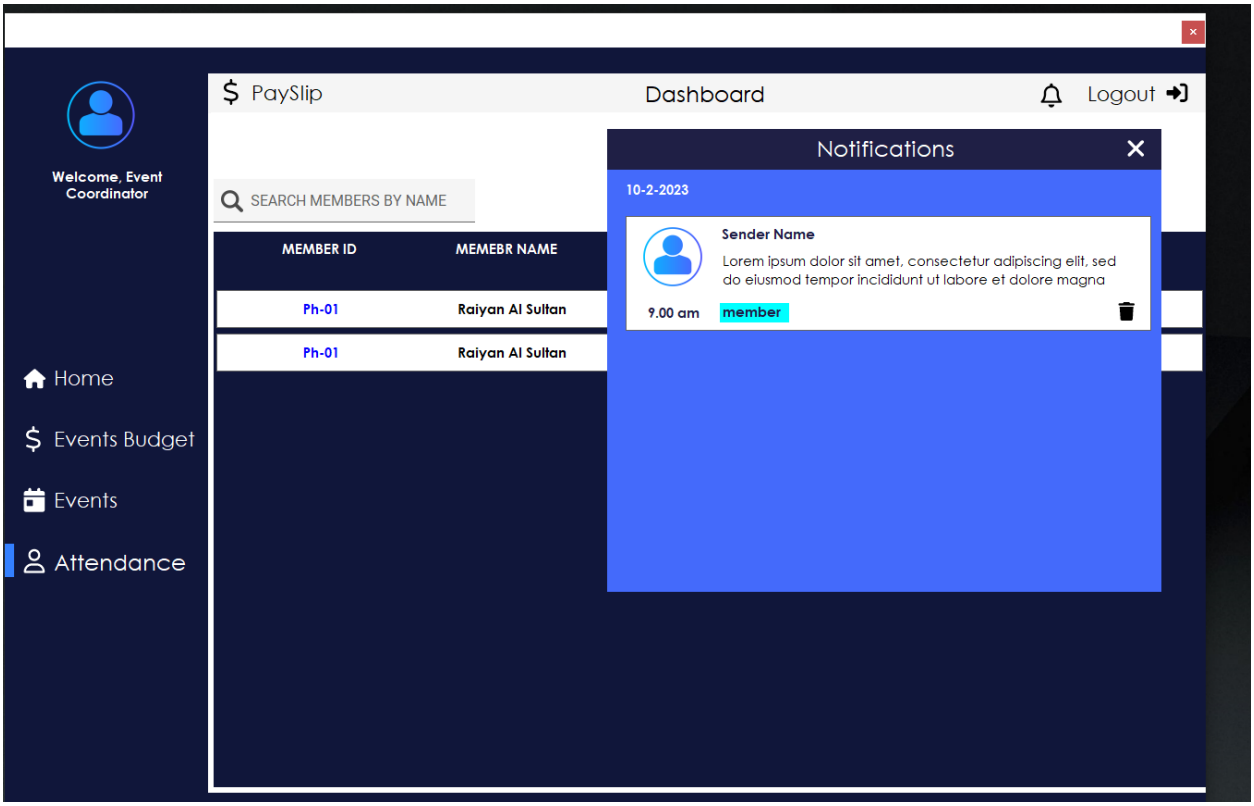

Welcome, Event Coordinator
PaySlip
Dashboard
Logout
Events Budget
SEARCH BUDGET BY EVENT
EVENT ID
EVENT NAME
EVENT DATE
TOTAL AMMOUNT IN TK
ORGANIZED BY
Ev-01
Photogrpahy Contest
12-2-2024
100000
Ph-01
Ev-02
Photogrpahy Talk
1-12-2023
120000
Ph-02
Home
Events Budget
Events
Attendance

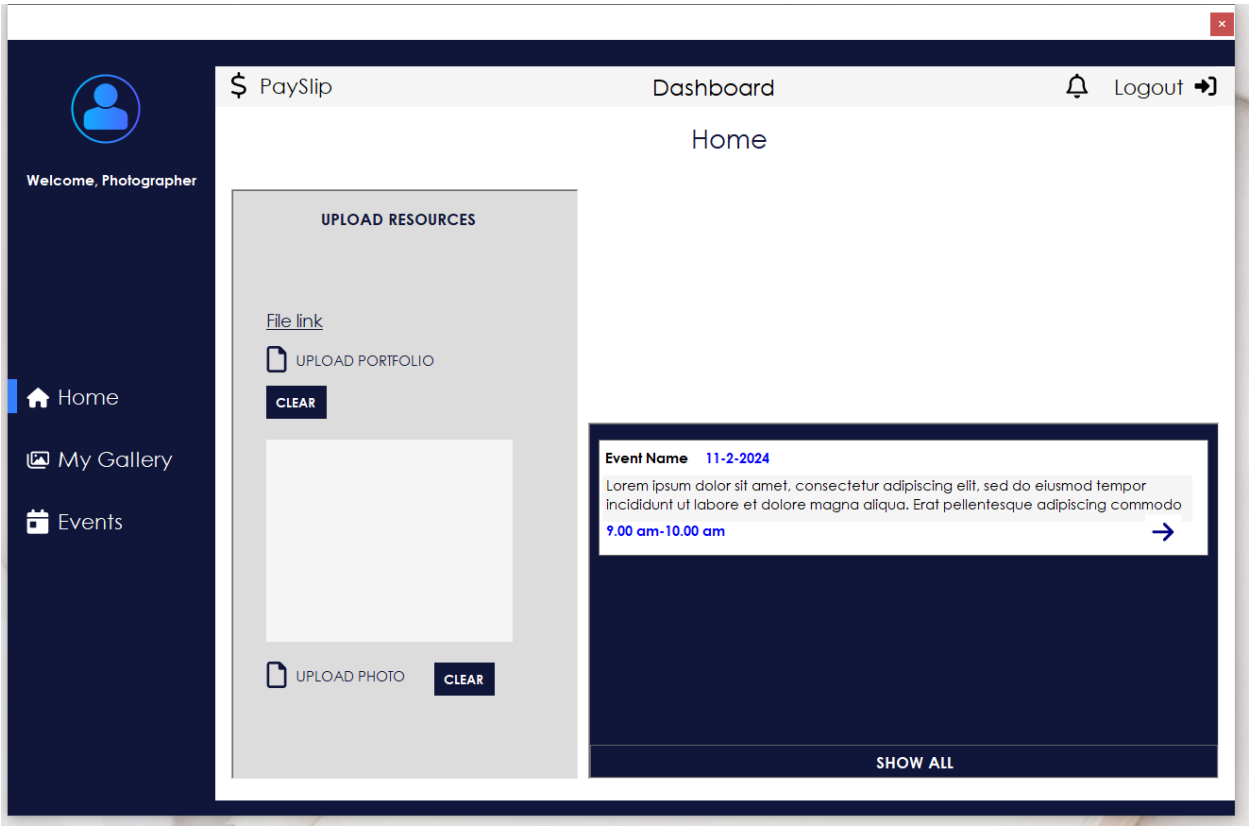
BACK


Welcome, Event Coordinator
PaySlip
Dashboard
Logout
Events
SEARCH EVENTS BY DATE
EVENT ID
EVENT NAME
EVENT DATE
EVENT LOCATION
ORGANIZED BY
Ev-01
Photogrpahy Contest
12-2-2024
Dhaka
Ph-01
Ev-02
Photogrpahy Talk
1-12-2023
Dhaka
Ph-02
Home
Events Budget
Events
Attendance

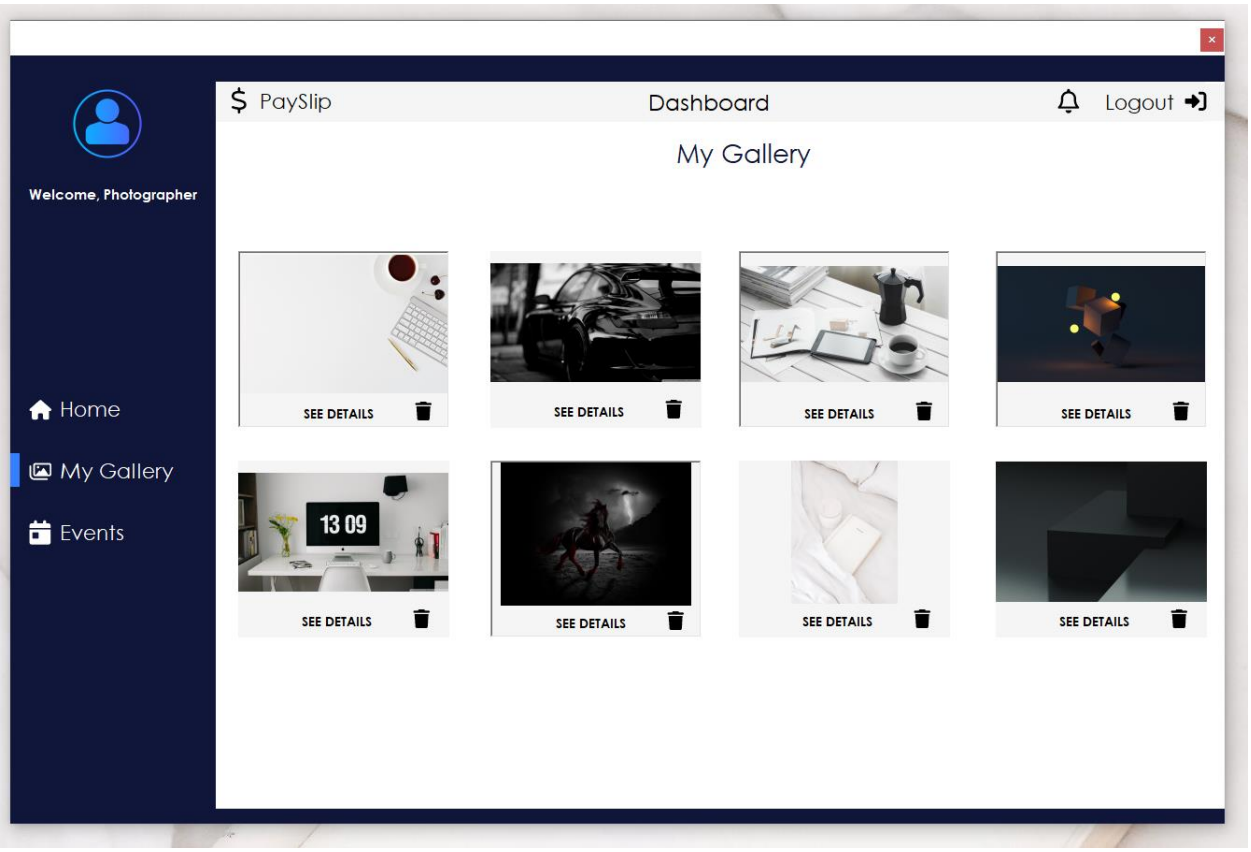

Welcome, Event Coordinator
PaySlip
Dashboard
Logout
Attendance
SEARCH MEMBERS BY NAME
MEMBER ID
MEMEBR NAME
MEMEBR ROLE
CURRENT WORK
ATTEND
Ph-01
Raiyan Al Sultan
Photographer
Ev-01
Ph-01
Raiyan Al Sultan
Photographer
Ev-01
Home
Events Budget
Events
Attendance

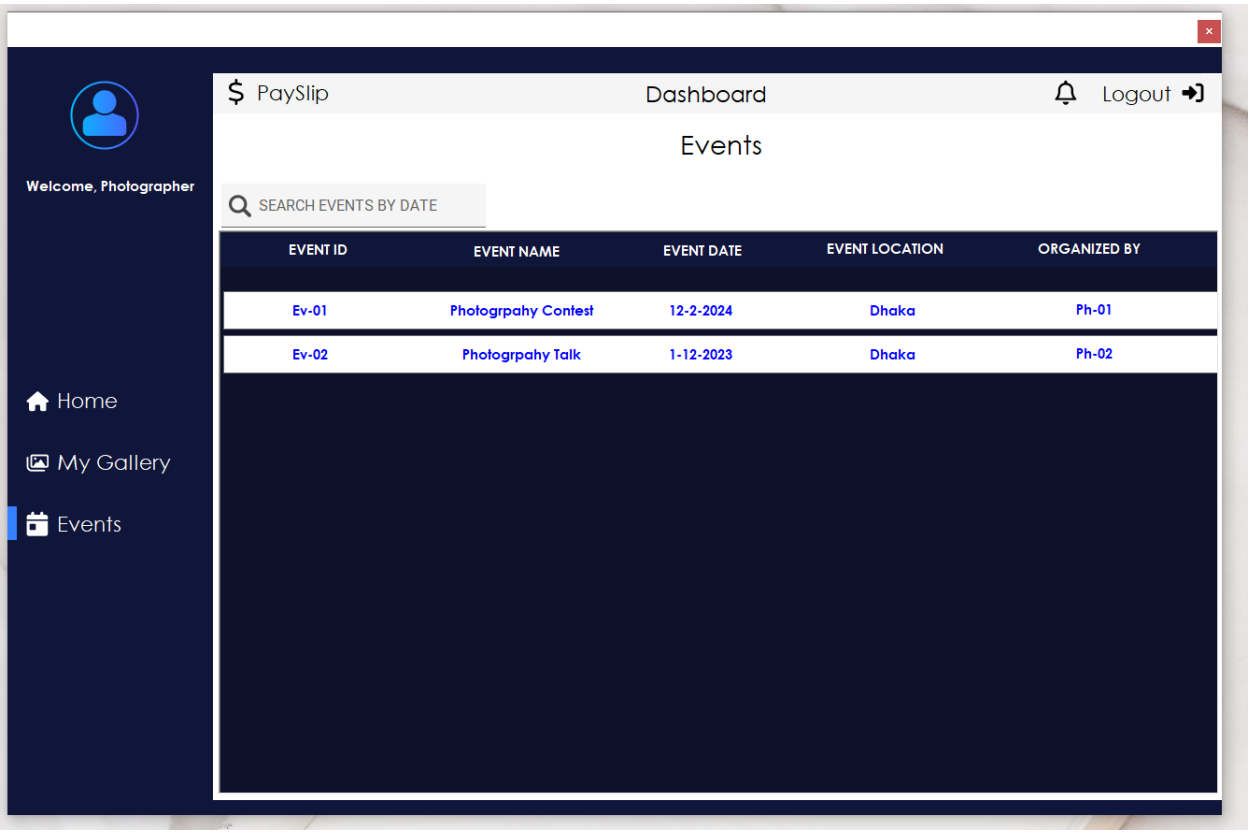
BACK


Welcome, Event Coordinator
PaySlip
Dashboard
Logout
Notifications
10-2-2023
Sender Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
9.00 am
member
SEARCH MEMBERS BY NAME
MEMBER ID
MEMEBR NAME
Ph-01
Raiyan Al Sultan
Ph-01
Raiyan Al Sultan
Home
Events Budget
Events
Attendance

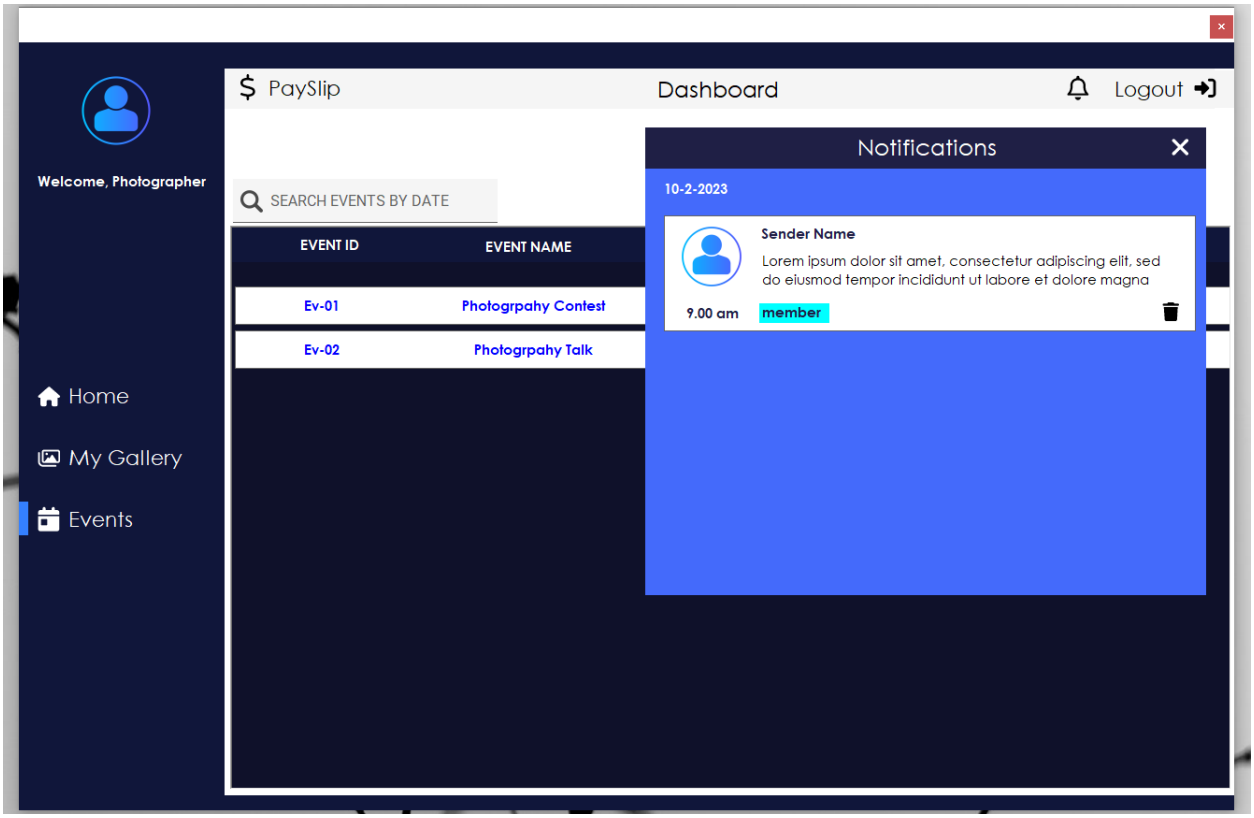

Welcome, Photographer
PaySlip
Dashboard
Logout
Home
UPLOAD RESOURCES
File link
UPLOAD PORTFOLIO
CLEAR
UPLOAD PHOTO
CLEAR
Event Name
11-2-2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Erat pellentesque adipiscing commodo
9.00 am-10.00 am
SHOW ALL
Home
My Gallery
Events

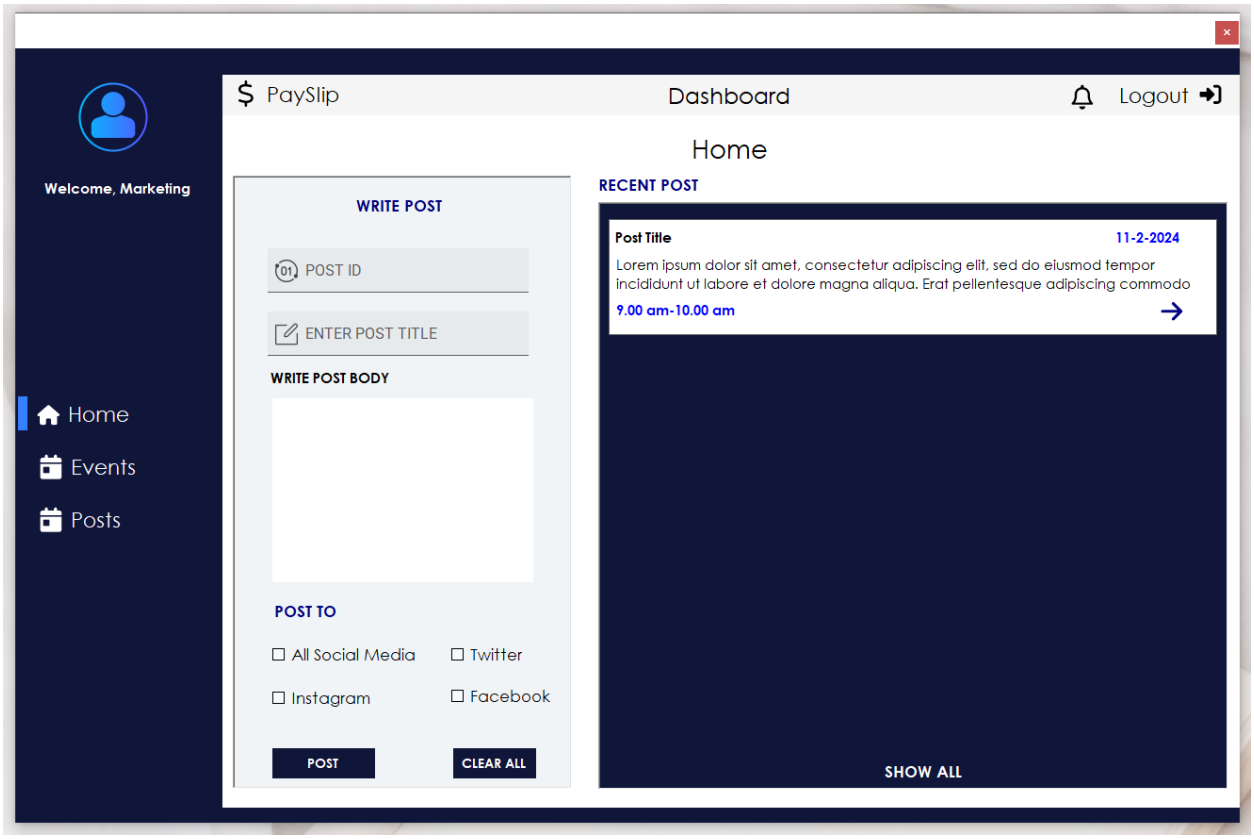
BACK


Welcome, Photographer
PaySlip
Dashboard
Logout
My Gallery
Home
My Gallery
Events
SEE DETAILS
13 09

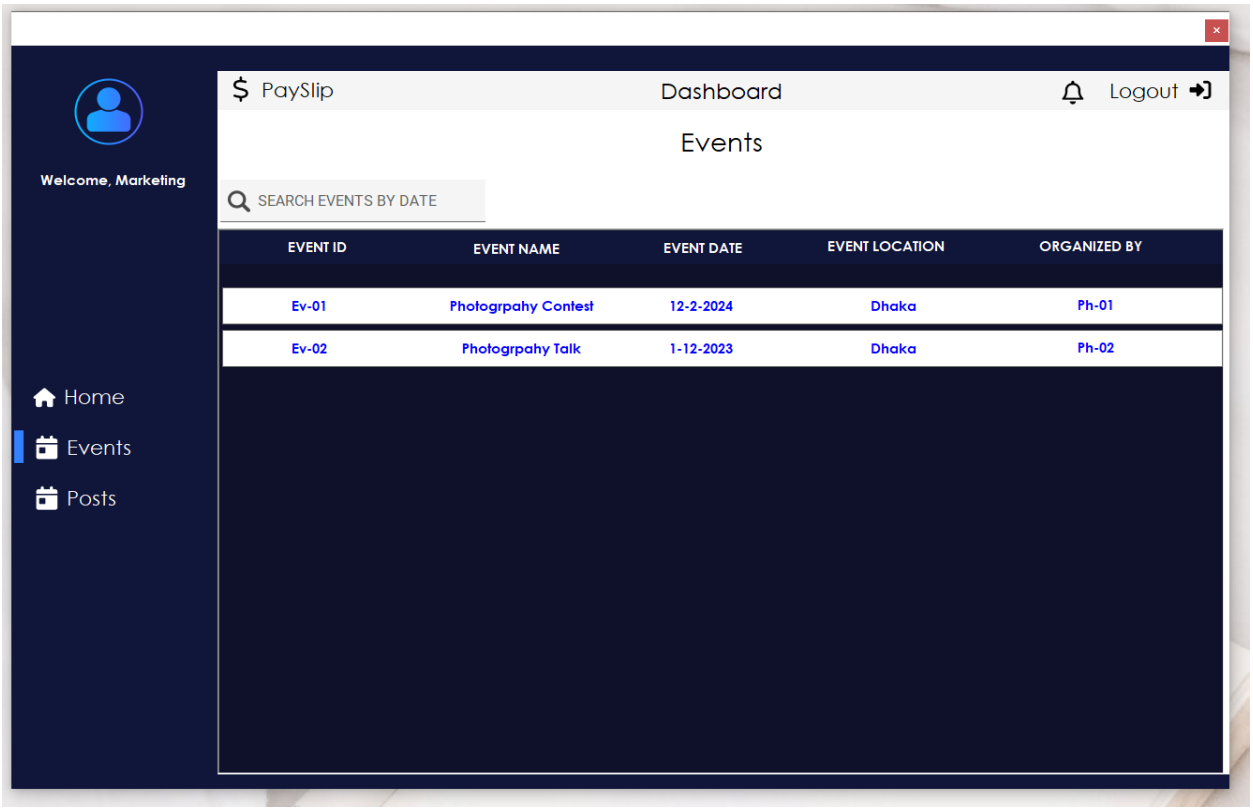

Welcome, Photographer
PaySlip
Dashboard
Logout
Events
SEARCH EVENTS BY DATE
EVENT ID
EVENT NAME
EVENT DATE
EVENT LOCATION
ORGANIZED BY
Ev-01
Photogrpahy Contest
12-2-2024
Dhaka
Ph-01
Ev-02
Photogrpahy Talk
1-12-2023
Dhaka
Ph-02
Home
My Gallery
Events

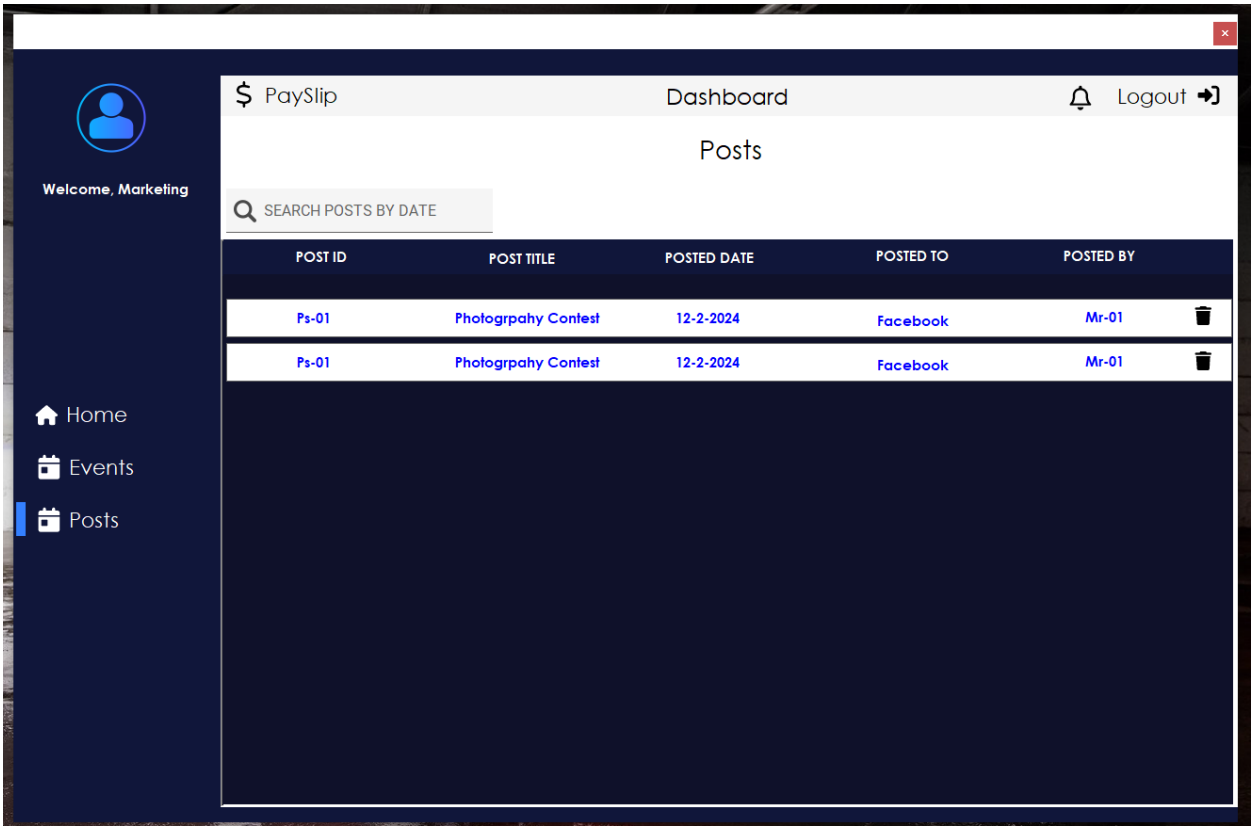
BACK


Welcome, Photographer
PaySlip
Dashboard
Logout
Notifications
10-2-2023
Sender Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
9.00 am
member
SEARCH EVENTS BY DATE
EVENT ID
EVENT NAME
Ev-01
Photogrpahy Contest
Ev-02
Photogrpahy Talk
Home
My Gallery
Events

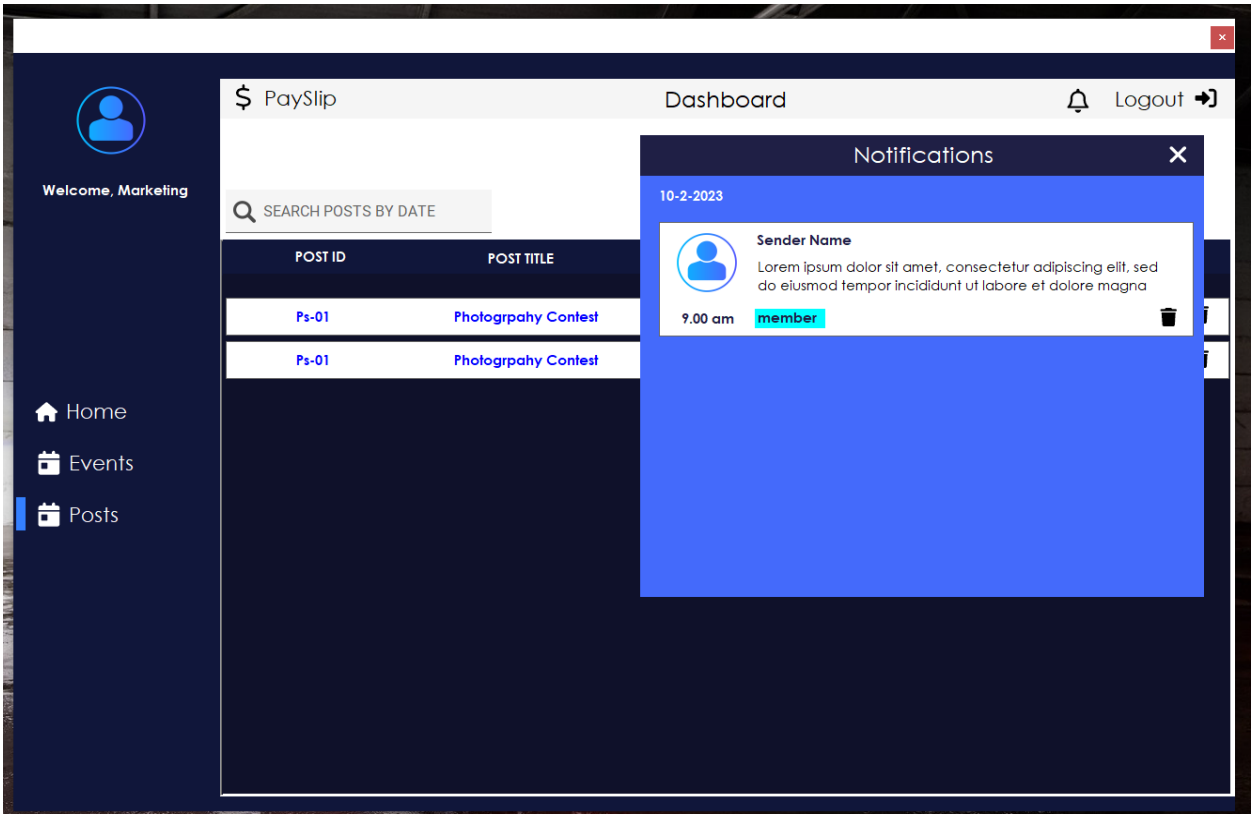

Welcome, Marketing
PaySlip
Dashboard
Logout
Home
WRITE POST
POST ID
ENTER POST TITLE
WRITE POST BODY
POST TO
All Social Media
Twitter
Instagram
Facebook
POST
CLEAR ALL
RECENT POST
Post Title
11-2-2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Erat pellentesque adipiscing commodo
9.00 am-10.00 am
SHOW ALL
Home
Events
Posts

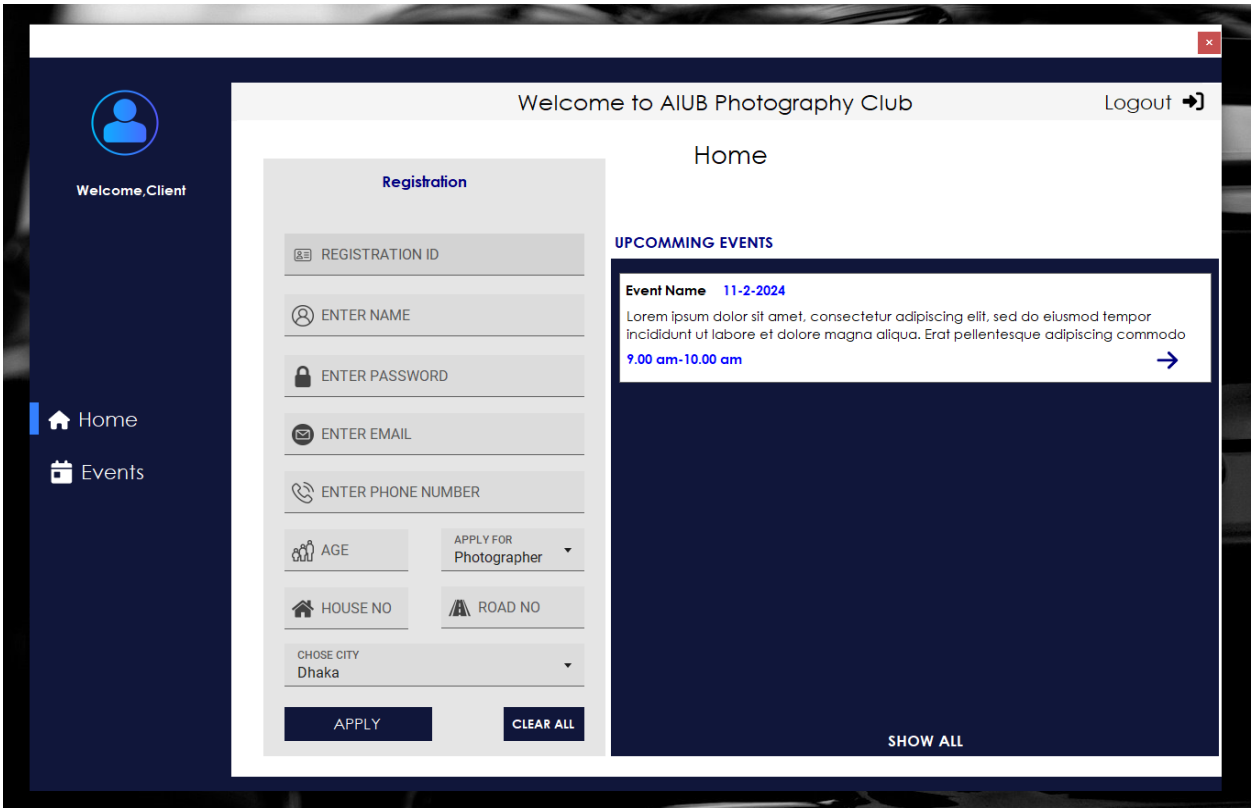
BACK


PaySlip
Dashboard
Logout
Welcome, Marketing
Events
SEARCH EVENTS BY DATE
EVENT ID
EVENT NAME
EVENT DATE
EVENT LOCATION
ORGANIZED BY
Ev-01
Photogrpahy Contest
12-2-2024
Dhaka
Ph-01
Ev-02
Photogrpahy Talk
1-12-2023
Dhaka
Ph-02
Home
Events
Posts


PaySlip
Dashboard
Logout
Welcome, Marketing
Posts
SEARCH POSTS BY DATE
POST ID
POST TITLE
POSTED DATE
POSTED TO
POSTED BY
Ps-01
Photogrpahy Contest
12-2-2024
Facebook
Mr-01
Ps-01
Photogrpahy Contest
12-2-2024
Facebook
Mr-01
Home
Events
Posts

BACK


Welcome, Marketing
PaySlip
Dashboard
Logout
Notifications
10-2-2023
Sender Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna
9.00 am
member
SEARCH POSTS BY DATE
POST ID
POST TITLE
Ps-01
Photogrpahy Contest
Ps-01
Photogrpahy Contest
Home
Events
Posts


Welcome,Client
Welcome to AIUB Photography Club
Logout
Home
Registration
REGISTRATION ID
ENTER NAME
ENTER PASSWORD
ENTER EMAIL
ENTER PHONE NUMBER
AGE
APPLY FOR
Photographer
HOUSE NO
ROAD NO
CHOSE CITY
Dhaka
APPLY
CLEAR ALL
UPCOMMING EVENTS
Event Name
11-2-2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Erat pellentesque adipiscing commodo
9.00 am-10.00 am
SHOW ALL
Home
Events

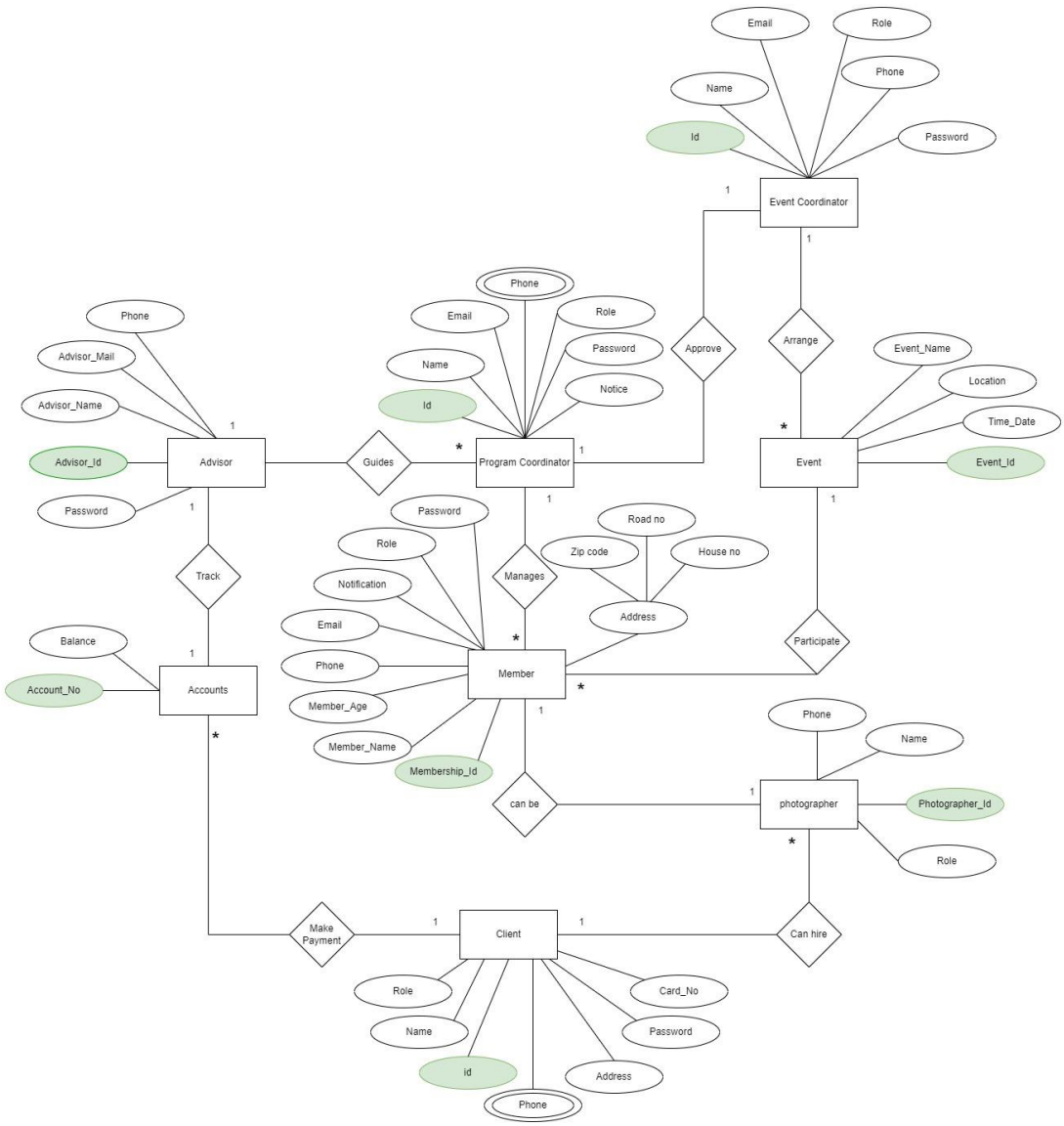
BACK

Scenario Description

A Photography Club is a vibrant community of photography enthusiasts those students gather to share their passion for capturing moments through the lens. In this photography club, the heart of operations revolves around a dedicated Advisor who provides invaluable guidance and expertise. Advisor has unique Advisor_Id. Also have Name, email, and phone and password. Each user with their specific roles and responsibilities. The club boasts program coordinate and event coordinator, ensuring the smooth functioning of the club. Program coordinator will manage the members and the can give the notice about club activities. Event coordinator will arrange the event. Both have ID, Name, Email. Password and phone number. Program coordinator can have multiple phone number. The system allows the program coordinator to approve all kind of events and also add the members, capturing their basic information such as Id, name, email, phone, age, role, password, and their addresses. Address can have zip code, road no, house no, city. They will get notified in every notice. Each member will identify by their unique id's. The members vary in their level of involvement. The club is a place of activity, where hosting many Events. These events are organized by the Event Coordinator, who ensure that every detail like EventID, Name, Date, Location. Moreover, the club offers a distinctive service wherein Clients can hire talented Photographers. Photographer is identified by their own unique member ID. Each client has a unique client id. Client data such as Id, Name, phone number, credit card no, address. A client can have multiple Phone number. Client can make payment. The club maintains a single account where balance is recorded. It has unique Account no and Balance track by advisor.

BACK

ER Diagram



BACK

Normalization

GUIDES (A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE)

UNF: (A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE)

1NF: PC_PHONE IS A MULTIVALUED ATTRIBUTE.

(A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE)

2NF:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE

2.PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

3NF:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE

2.PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

FINAL LIST FORM GUIDES:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, PC_ID, P_ID

2. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

3.P_ID, PC_PHONE -> COMPOSIT PK

APPROVE (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE)

UNF: (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE)

1NF: PC_PHONE IS A MULTIVALUED ATTRIBUTE.

(PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE)

2NF:

1.PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

2.EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE

3NF:

FINAL TABLE

1. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

2.EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE

FINAL LIST FORM **APPROVE**:

1. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, EC_ID, P_ID

2.EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE

3.P_ID, PC_PHONE -> COMPOSIT PK

MANAGES (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO)

UNF: (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO)

1NF: PC_PHONE IS A MULTIVALUED ATTRIBUTE.

(PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO)

2NF:

1.PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

2.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

3NF:

1. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE

2.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

FINAL LIST FORM **MANAGES**

FINAL TABLE

1. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, MEMBER_ID, D_I, P_ID

2.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, D_ID

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

4.P_ID, PC_PHONE -> COMPOSIT PK

ARRANGE (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

UNF: (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

1NF: THERE IS NO MULTIVALUED ATTRIBUTE.

(EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

2NF:

1. EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

3NF:

1. EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

FINAL LIST FORM **ARRANGE**

1.EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE, EVENT_ID

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

TRACK (A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, ACCOUNT_ID, BALANCE)

UNF: (A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, ACCOUNT_ID, BALANCE)

1NF: THERE IS NO MULTIVALUED ATTRIBUTE.

(A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, ACCOUNT_ID, BALANCE)

2NF:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE

FINAL TABLE

2. ACCOUNT_ID, BALANCE

3NF:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE

2. ACCOUNT_ID, BALANCE

FINAL LIST FORM **TRACK**

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, ACCOUNT_ID

2. ACCOUNT_ID, BALANCE

PATICIPATE (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

UNF: (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

1NF: THERE IS NO MULTIVALUED ATTRIBUTE.

(MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)

2NF:

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

3NF:

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

FINAL TABLE

FINAL LIST FORM **PATICIPATE**

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, D_ID

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

CAN BE (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE)

UNF:(MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE)

1NF: THERE IS NO MULTIVALUED ATTRIBUTE.

((MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE)

2NF:

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE

3NF:

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

FINAL TABLE

FINAL LIST FORM **CAN BE:**

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, PHOTOGRAPHER_ID, D_ID

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE, D_ID

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

MAKE PAYMENNT(CLIENT_ID,CLIENT_NAME, CLIENT_PHONE, CLIENT_ADDRESS, CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT ROLE, ACCOUNT_ID, BALANCE)

UNF:(CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARD NO,CLIENT_PASSWORD,CLIENT ROLE, ACCOUNT_ID, BALANCE)

1NF: CLIENT_PHONE IS MULTIVALUED ATTRIBUTE.

(CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO,C LIENT_PASSWORD,CLIENT ROLE, ACCOUNT_ID, BALANCE)

2NF:

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS, CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT ROLE

2. ACCOUNT_ID, BALANCE

3NF:

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE

2. ACCOUNT_ID, BALANCE

FINAL LIST FORM **MAKE PAYMENNT:**

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE, ACCOUNT_ID, C_ID

2. ACCOUNT_ID, BALANCE

3. C_ID, CLIENT_PHONE -> COMPOSIT PK

FINAL TABLE

CAN HIRE (CLIENT_ID, CLIENT_NAME, CLIENT_PHONE, CLIENT_ADDRESS, CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT ROLE, PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE, PHOTOGRAPHER_PHONE)

UNF:(CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS, CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT_ROLE,PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE, PHOTOGRAPHER_PHONE)

1NF: CLIENT_PHONE IS MULTIVALUED ATTRIBUTE.

(CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT ROLE, PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE, PHOTOGRAPHER_PHONE)

2NF:

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE,PHOTOGRAPHER_PHONE

3NF:

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE,PHOTOGRAPHER_PHONE

<u>FINAL LIST FORM **CAN HIRE:**</u>

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE, PHOTOGRAPHER_ID, C_ID

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLEPHOTOGRAPHER_PHONE

3. C_ID, CLIENT_PHONE -> COMPOSIT PK

FINAL TABLE

FINAL LIST OF TABLES

FINAL LIST FORM GUIDES:

1. A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, PC_ID, P_ID, ACCOUNT_ID

2. ~~PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE~~

3.P_ID, PC_PHONE -> COMPOSIT PK

FINAL LIST FORM **APPROVE**:

1. ~~PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, EC_ID, P_ID~~

2.~~EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE~~

3.~~P_ID, PC_PHONE -> COMPOSIT PK~~

FINAL LIST FORM **MANAGES**:

1. PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_PHONE, PC_ROLE, PC_NOTICE, MEMBER_ID, D_I, P_ID,EC_ID

~~2. MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, D_ID~~

3. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.

4.~~P_ID, PC_PHONE -> COMPOSIT PK~~

FINAL LIST FORM **ARRANGE**:

1. EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EC_ROLE, EC_PHONE, EVENT_ID

2. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION

FINAL LIST FORM **TRACK**:

1. ~~A_ID, A_NAME, A_EMAIL, A_PASSWORD, A_PHONE, ACCOUNT_ID~~

2. ACCOUNT_ID, BALANCE

FINAL TABLE

FINAL LIST FORM **PATICIPATE:**

~~1. MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, D_ID~~

2. ~~EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION~~

3. ~~D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.~~

FINAL LIST FORM CAN BE:

1.MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE,MEMBER_PHONE, NOTIFICATION, MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, PHOTOGRAPHER_ID, D_ID, EVENT_ID.

2.PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLEPHOTOGRAPHER_PHONE, D_ID

3. ~~D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.~~

FINAL LIST FORM **MAKE PAYMENNT:**

1.CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARDNO, CLIENT_PASSWORD,CLIENT ROLE, ACCOUNT_ID, C_ID, PHOTOGRAPHER_ID

2. ~~ACCOUNT_ID, BALANCE~~

3. C_ID, CLIENT_PHONE -> COMPOSIT PK

FINAL LIST FORM **CAN HIRE:**

1. ~~CLIENT_ID, CLIENT_NAME, CLIENT_PHONE, CLIENT_ADDRESS, CLIENT_CARDNO,CLIENT_PASSWORD,CLIENT ROLE, PHOTOGRAPHER_ID, C_ID~~

~~2. PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_ROLE PHOTOGRAPHER_PHONE~~

3. ~~C_ID, CLIENT_PHONE -> COMPOSIT PK~~

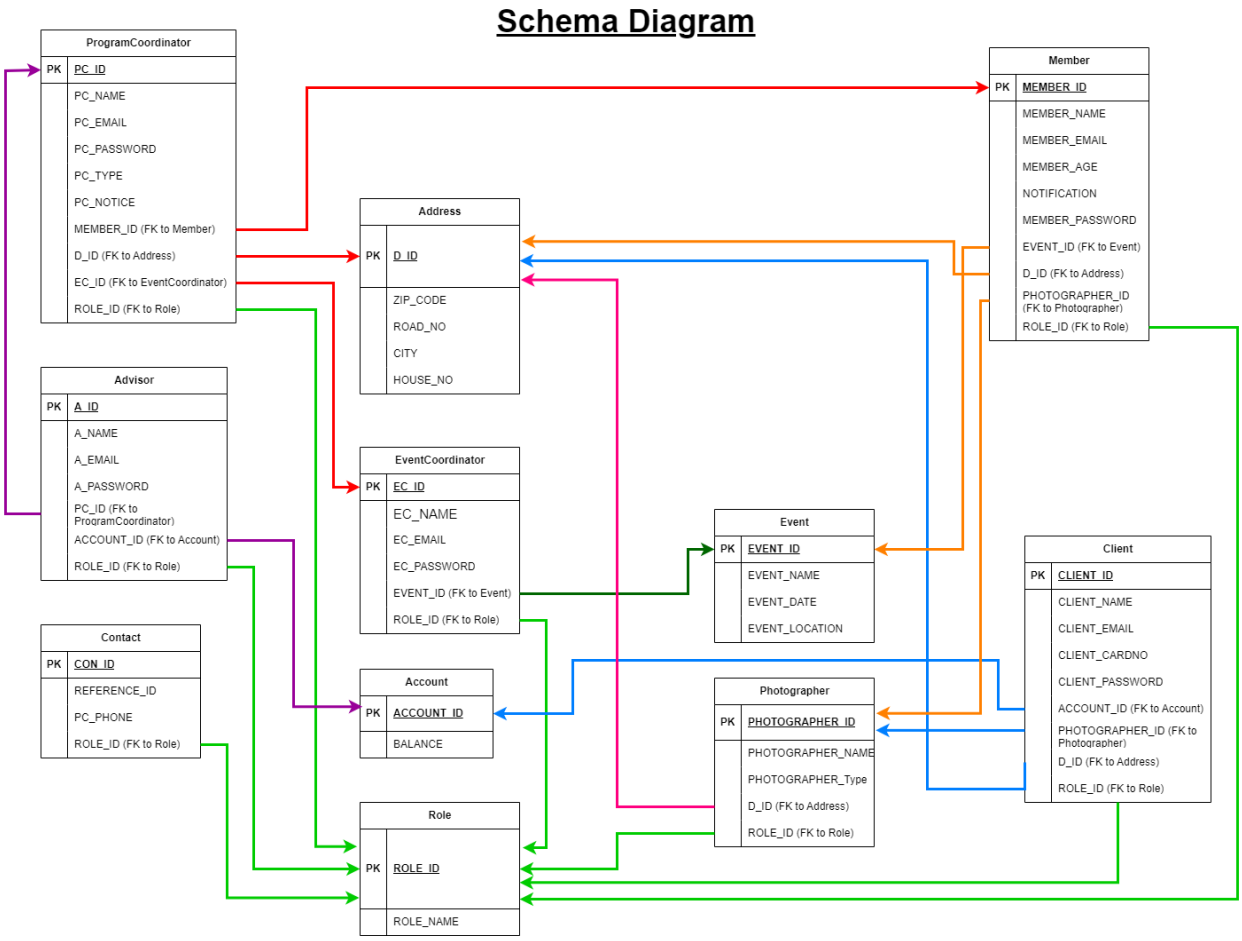
FINAL TABLE

FINAL TABLE

1. A_ID,A_NAME,A_EMAIL,A_PASSWORD, A_PHONE, PC_ID, P_ID, ACCOUNT_ID
2. P_ID, PC_PHONE -> COMPOSIT PK
3. PC_ID,PC_NAME,PC_EMAIL,PC_PASSWORD,PC_PHONE,PC_ROLE,C_NOTICE, MEMBER_ID, D_I, P_ID, EC_ID
4. D_ID, ZIP CODE, ROAD NO, CITY, HOUSE NO.
5. EC_ID,EC_NAME,EC_EMAIL,EC_PASSWORD,EC_ROLE, EC_PHONE, EVENT_ID
6. EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION
7. ACCOUNT_ID, BALANCE
8. MEMBER_ID,MEMBER_NAME,MEMBER_EMAIL,MEMBER_AGE,MEMBER_PHONE,NOTIFICATION,MEMBER_ROLE, MEMBER_PASSWORD, ZIP CODE, ROAD NO, HOUSE NO, EVENT_ID, D_ID, PHOTOGRAPHER_ID
9. PHOTOGRAPHER_ID,PHOTOGRAPHER_NAME,PHOTOGRAPHER_ROLE,PHOTOGRAPHER_PHONE, D_ID
10. CLIENT_ID,CLIENT_NAME,CLIENT_PHONE,CLIENT_ADDRESS,CLIENT_CARD NO,CLIENT_PASSWORD,CLIENT_ROLE,ACCOUNT_ID,C_ID,PHOTOGRAPHER_ID
11. CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID
12. ROLE_ID, ROLE_TYPE

BACK

Schema Diagram



BACK

Table Creation

System

```
CREATE ROLE  Advisor;
CREATE ROLE  PC;
CREATE ROLE  EC;
CREATE ROLE  Client;
CREATE ROLE  Member;
CREATE ROLE  Photographer;
```

Results Explain Describe Saved SQL His

Role created.

```
CREATE USER ADV  IDENTIFIED BY ADV;
CREATE USER PCO IDENTIFIED BY PCO;
CREATE USER ECO IDENTIFIED BY ECO;
CREATE USER CLI  IDENTIFIED BY CLI;
CREATE USER MEM  IDENTIFIED BY MEM;
CREATE USER PHOG  IDENTIFIED BY PHOG;
```

Results Explain Describe Saved SQL Histo

User created.

----------next----------

```
GRANT ALL PRIVILEGES TO PC WITH ADMIN OPTION;
GRANT CREATE TABLE TO PC;
GRANT CREATE VIEW TO PC;
GRANT CREATE PROCEDURE TO PC;
GRANT CREATE SEQUENCE TO PC;
GRANT CREATE TRIGGER TO PC;
GRANT CREATE ANY INDEX TO PC;
GRANT SELECT ANY TABLE TO PC;
GRANT INSERT ANY TABLE TO PC;
GRANT UPDATE ANY TABLE TO PC;
GRANT DELETE ANY TABLE TO PC;
GRANT DROP ANY INDEX TO PC;
GRANT DROP ANY TABLE TO PC;
GRANT DROP ANY VIEW TO PC;
GRANT ALTER ANY TABLE TO PC;
GRANT ALTER ANY INDEX TO PC;
-- Grant the PC role to PC_USER
GRANT PC TO PCO;
```

Results Explain Describe Saved SQL History

Statement processed.

BACK

```
SELECT ROLE FROM DBA_ROLES;
```

Results Explain Describe Saved SQL Histo

ADVISOR
PC
EC
CLIENT
MEMBER
PHOTOGRAPHER

```
SELECT username FROM dba_users;
```

Results Explain Describe Saved SQL H

USERNAME
PCO
SAMPLE
PHOG
ADV
ECO
CLI
MEM

----------next----------

PCO USER

```
CREATE SEQUENCE seq_role START WITH 1201 INCREMENT BY 1;
CREATE SEQUENCE seq_address START WITH 9101 INCREMENT BY 1;
CREATE SEQUENCE seq_account START WITH 10101 INCREMENT BY 1;
CREATE SEQUENCE seq_event START WITH 4101 INCREMENT BY 1;
CREATE SEQUENCE seq_photographer START WITH 6101 INCREMENT BY 1;
CREATE SEQUENCE seq_member START WITH 5101 INCREMENT BY 1;
CREATE SEQUENCE seq_eventcoordinator START WITH 3101 INCREMENT BY 1;
CREATE SEQUENCE seq_programcoordinator START WITH 2101 INCREMENT BY 1;
CREATE SEQUENCE seq_client START WITH 7101 INCREMENT BY 1;
CREATE SEQUENCE seq_advisor START WITH 1101 INCREMENT BY 1;
CREATE SEQUENCE seq_contact START WITH 8101 INCREMENT BY 1;
```

Results Explain Describe Saved SQL History

```
Sequence created.
```

----------next----------

```
CREATE TABLE Role (
    ROLE_ID NUMBER PRIMARY KEY,
    ROLE_NAME VARCHAR(100) NOT NULL
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

```
-- Account Table
CREATE TABLE Account (
    ACCOUNT_ID NUMBER PRIMARY KEY,
    BALANCE NUMBER NOT NULL
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

BACK

```
-- Address Table
CREATE TABLE Address (
    D_ID NUMBER PRIMARY KEY,
    ZIP_CODE NUMBER(10) NOT NULL,
    ROAD_NO VARCHAR(50) NOT NULL,
    CITY VARCHAR2(100) NOT NULL,
    HOUSE_NO VARCHAR(50) NOT NULL
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

```
-- Event Table
CREATE TABLE Event (
    EVENT_ID NUMBER PRIMARY KEY,
    EVENT_NAME VARCHAR(100) NOT NULL,
    EVENT_DATE DATE NOT NULL,
    EVENT_LOCATION VARCHAR(255) NOT NULL
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

----------next----------

```
-- Photographer Table
CREATE TABLE Photographer (
    PHOTOGRAPHER_ID NUMBER PRIMARY KEY,
    PHOTOGRAPHER_NAME VARCHAR(100) NOT NULL,
    PHOTOGRAPHER_Type VARCHAR(100) NOT NULL,

    D_ID NUMBER NOT NULL,
    ROLE_ID NUMBER,

    FOREIGN KEY (D_ID) REFERENCES Address(D_ID),
    FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

```
ALTER TABLE Photographer
ADD PHOTOGRAPHER_EMAIL VARCHAR(255);
```

Results Explain Describe Saved SQL History

```
Table altered.
```

----------next----------

```
CREATE TABLE Member (
    MEMBER_ID NUMBER PRIMARY KEY,
    MEMBER_NAME VARCHAR(100) NOT NULL,
    MEMBER_EMAIL VARCHAR(100) NOT NULL,
    MEMBER_AGE NUMBER CHECK (MEMBER_AGE >= 15) NOT NULL,
    NOTIFICATION VARCHAR(255) NOT NULL,
    MEMBER_PASSWORD VARCHAR(100) NOT NULL,

    EVENT_ID NUMBER NOT NULL,
    D_ID NUMBER NOT NULL,
    PHOTOGRAPHER_ID NUMBER NOT NULL,
    ROLE_ID NUMBER,

    FOREIGN KEY (EVENT_ID) REFERENCES Event(EVENT_ID) ,
    FOREIGN KEY (D_ID) REFERENCES Address(D_ID),
    FOREIGN KEY (PHOTOGRAPHER_ID) REFERENCES Photographer(PHOTOGRAPHER_ID),
    FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

BACK

```sql
CREATE TABLE EventCoordinator (
    EC_ID NUMBER PRIMARY KEY,
    EC_NAME VARCHAR(100) NOT NULL,
    EC_EMAIL VARCHAR(100) NOT NULL,
    EC_PASSWORD VARCHAR(100) NOT NULL,

    EVENT_ID NUMBER NOT NULL,
    ROLE_ID NUMBER,

    FOREIGN KEY (EVENT_ID) REFERENCES Event(EVENT_ID),
    FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

----------next----------

```sql
-- ProgramCoordinator Table
CREATE TABLE ProgramCoordinator (
    PC_ID       NUMBER PRIMARY KEY,
    PC_NAME     VARCHAR(100) NOT NULL,
    PC_EMAIL    VARCHAR(100) NOT NULL,
    PC_PASSWORD VARCHAR(100) NOT NULL,
    PC_TYPE     VARCHAR(100) NOT NULL,
    PC_NOTICE   VARCHAR(255) NOT NULL,

    MEMBER_ID NUMBER NOT NULL,
    D_ID      NUMBER NOT NULL,
    EC_ID     NUMBER NOT NULL,
    ROLE_ID   NUMBER,

    FOREIGN KEY (MEMBER_ID) REFERENCES Member(MEMBER_ID),
    FOREIGN KEY (D_ID)      REFERENCES Address(D_ID),
    FOREIGN KEY (EC_ID)     REFERENCES EventCoordinator(EC_ID),
    FOREIGN KEY (ROLE_ID)   REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

BACK

```sql
-- Client Table
CREATE TABLE Client (
    CLIENT_ID NUMBER PRIMARY KEY,
    CLIENT_NAME VARCHAR(100) NOT NULL,
    CLIENT_EMAIL VARCHAR(100) NOT NULL,
    CLIENT_CARDNO VARCHAR(20) NOT NULL,
    CLIENT_PASSWORD VARCHAR(100) NOT NULL,

    ACCOUNT_ID NUMBER NOT NULL,
    PHOTOGRAPHER_ID NUMBER NOT NULL,
    D_ID NUMBER NOT NULL,
    ROLE_ID NUMBER,

    FOREIGN KEY (ACCOUNT_ID) REFERENCES Account(ACCOUNT_ID),
    FOREIGN KEY (PHOTOGRAPHER_ID) REFERENCES Photographer(PHOTOGRAPHER_ID),
    FOREIGN KEY (D_ID) REFERENCES Address(D_ID),
    FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

----------next----------

```sql
-- Advisor Table
CREATE TABLE Advisor (
    A_ID NUMBER PRIMARY KEY,
    A_NAME VARCHAR(100) NOT NULL,
    A_EMAIL VARCHAR(100) NOT NULL,
    A_PASSWORD VARCHAR(100) NOT NULL,

    PC_ID NUMBER NOT NULL,
    ACCOUNT_ID NUMBER NOT NULL,
    ROLE_ID NUMBER,

    FOREIGN KEY (PC_ID) REFERENCES ProgramCoordinator(PC_ID),
    FOREIGN KEY (ACCOUNT_ID) REFERENCES Account(ACCOUNT_ID),
    FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);

-- Create the Contact
```

Results Explain Describe Saved SQL History

```
Table created.
```

----------next----------

```sql
-- Create the Contact
CREATE TABLE Contact (
    CON_ID NUMBER PRIMARY KEY,
    REFERENCE_ID NUMBER NOT NULL,
    PC_PHONE VARCHAR(15) NOT NULL,

    ROLE_ID NUMBER,

   FOREIGN KEY (ROLE_ID) REFERENCES Role(ROLE_ID)
);
```

Results Explain Describe Saved SQL History

```
Table created.
```

BACK

Index

```
--CREATE Index
CREATE INDEX idx_address_city ON Address(CITY);
CREATE INDEX idx_account_balance ON Account(BALANCE);
CREATE INDEX idx_event_name ON Event(EVENT_NAME);
CREATE INDEX idx_photographer_name ON Photographer(PHOTOGRAPHER_NAME);
CREATE INDEX idx_member_email ON Member(MEMBER_EMAIL);
CREATE INDEX idx_eventcoordinator_email ON EventCoordinator(EC_EMAIL);
CREATE INDEX idx_programcoordinator_email ON ProgramCoordinator(PC_EMAIL);
CREATE INDEX idx_client_email ON Client(CLIENT_EMAIL);
CREATE INDEX idx_advisor_email ON Advisor(A_EMAIL);
CREATE INDEX idx_contact_phone ON Contact(PC_PHONE);
```

Results Explain Describe Saved SQL History

```
Index created.
```

----------next----------

INSERTION

```
-- Inserts Data Role
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Advisor');
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Program Coordinator');
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Event Coordinator');
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Client');
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Member');
INSERT INTO Role (ROLE_ID, ROLE_NAME) VALUES (seq_role.NEXTVAL, 'Photographer');
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

----------next----------

```
-- Inserts Data Address
INSERT INTO Address (D_ID, ZIP_CODE, ROAD_NO, CITY, HOUSE_NO) VALUES (seq_address.NEXTVAL, 12345, '101', 'New York', '1A');
INSERT INTO Address (D_ID, ZIP_CODE, ROAD_NO, CITY, HOUSE_NO) VALUES (seq_address.NEXTVAL, 23456, '102', 'Los Angeles', '2B');
INSERT INTO Address (D_ID, ZIP_CODE, ROAD_NO, CITY, HOUSE_NO) VALUES (seq_address.NEXTVAL, 34567, '103', 'Chicago', '3C');
INSERT INTO Address (D_ID, ZIP_CODE, ROAD_NO, CITY, HOUSE_NO) VALUES (seq_address.NEXTVAL, 45678, '104', 'Houston', '4D');
INSERT INTO Address (D_ID, ZIP_CODE, ROAD_NO, CITY, HOUSE_NO) VALUES (seq_address.NEXTVAL, 56789, '105', 'Phoenix', '5E');
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

BACK

```sql
-- Inserts Data Account Table
INSERT INTO Account (ACCOUNT_ID, BALANCE) VALUES (seq_account.NEXTVAL, 1000);
INSERT INTO Account (ACCOUNT_ID, BALANCE) VALUES (seq_account.NEXTVAL, 1500);
INSERT INTO Account (ACCOUNT_ID, BALANCE) VALUES (seq_account.NEXTVAL, 2000);
INSERT INTO Account (ACCOUNT_ID, BALANCE) VALUES (seq_account.NEXTVAL, 2500);
INSERT INTO Account (ACCOUNT_ID, BALANCE) VALUES (seq_account.NEXTVAL, 3000);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

----------next----------

```sql
-- Inserts Data Event Table
INSERT INTO Event (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION) VALUES (seq_event.NEXTVAL, 'Event 1', TO_DATE('2023-12-01', 'YYYY-MM-DD'), 'Location 1');
INSERT INTO Event (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION) VALUES (seq_event.NEXTVAL, 'Event 2', TO_DATE('2023-12-02', 'YYYY-MM-DD'), 'Location 2');
INSERT INTO Event (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION) VALUES (seq_event.NEXTVAL, 'Event 3', TO_DATE('2023-12-03', 'YYYY-MM-DD'), 'Location 3');
INSERT INTO Event (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION) VALUES (seq_event.NEXTVAL, 'Event 4', TO_DATE('2023-12-04', 'YYYY-MM-DD'), 'Location 4');
INSERT INTO Event (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION) VALUES (seq_event.NEXTVAL, 'Event 5', TO_DATE('2023-12-05', 'YYYY-MM-DD'), 'Location 5');
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

----------next----------

```sql
--Inserts Data Photographer Table
INSERT INTO Photographer (PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_Type, D_ID, ROLE_ID) VALUES (seq_photographer.NEXTVAL, 'John Doe', 'Portrait', 9101, 1206);
INSERT INTO Photographer (PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_Type, D_ID, ROLE_ID) VALUES (seq_photographer.NEXTVAL, 'Jane Smith', 'Wedding', 9102, 1206);
INSERT INTO Photographer (PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_Type, D_ID, ROLE_ID) VALUES (seq_photographer.NEXTVAL, 'Emily Johnson', 'Nature', 9103, 1206);
INSERT INTO Photographer (PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_Type, D_ID, ROLE_ID) VALUES (seq_photographer.NEXTVAL, 'Michael Brown', 'Fashion', 9104, 1206);
INSERT INTO Photographer (PHOTOGRAPHER_ID, PHOTOGRAPHER_NAME, PHOTOGRAPHER_Type, D_ID, ROLE_ID) VALUES (seq_photographer.NEXTVAL, 'Linda Davis', 'Event', 9105, 1206);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

BACK

```sql
-- Insert Data  Member Table
INSERT INTO Member (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, NOTIFICATION, MEMBER_PASSWORD, EVENT_ID, D_ID, PHOTOGRAPHER_ID, ROLE_ID) VALUES
(seq_member.NEXTVAL, 'John Doe',    'john@example.com',   25, 'Text',  'password123', 4101, 9101, 6102, 1205);
INSERT INTO Member (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, NOTIFICATION, MEMBER_PASSWORD, EVENT_ID, D_ID, PHOTOGRAPHER_ID, ROLE_ID) VALUES
(seq_member.NEXTVAL, 'Emily Clark',   'emily@example.com',   30, 'Email', 'password321', 4102, 9102, 6103, 1205);
INSERT INTO Member (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, NOTIFICATION, MEMBER_PASSWORD, EVENT_ID, D_ID, PHOTOGRAPHER_ID, ROLE_ID) VALUES
(seq_member.NEXTVAL, 'Michael Brown',  'michael@example.com', 40, 'Phone', 'password456', 4103, 9103, 6104, 1205);
INSERT INTO Member (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, NOTIFICATION, MEMBER_PASSWORD, EVENT_ID, D_ID, PHOTOGRAPHER_ID, ROLE_ID) VALUES
(seq_member.NEXTVAL, 'Linda Smith',   'linda.@example.com',  22, 'App',   'password789', 4104, 9104, 6105, 1205);
INSERT INTO Member (MEMBER_ID, MEMBER_NAME, MEMBER_EMAIL, MEMBER_AGE, NOTIFICATION, MEMBER_PASSWORD, EVENT_ID, D_ID, PHOTOGRAPHER_ID, ROLE_ID) VALUES
(seq_member.NEXTVAL, 'Robert Johnson', 'robert@example.com',  35, 'Mail',  'password654', 4105, 9105, 6101, 1205);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

----------next----------

```sql
-- Insert Data  EventCoordinator Table
INSERT INTO EventCoordinator (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EVENT_ID, ROLE_ID) VALUES (seq_eventcoordinator.NEXTVAL, 'Jane Smith',
'jane@example.com',  'password123', 4101, 1203);
INSERT INTO EventCoordinator (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EVENT_ID, ROLE_ID) VALUES (seq_eventcoordinator.NEXTVAL, 'Paul Brooks',
'paul@example.com',  'password234', 4102, 1203);
INSERT INTO EventCoordinator (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EVENT_ID, ROLE_ID) VALUES (seq_eventcoordinator.NEXTVAL, 'Susan Hill',
'susan@example.com', 'password345', 4103, 1203);
INSERT INTO EventCoordinator (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EVENT_ID, ROLE_ID) VALUES (seq_eventcoordinator.NEXTVAL, 'Gary White',
'gary@example.com',  'password456', 4104, 1203);
INSERT INTO EventCoordinator (EC_ID, EC_NAME, EC_EMAIL, EC_PASSWORD, EVENT_ID, ROLE_ID) VALUES (seq_eventcoordinator.NEXTVAL, 'Lisa Turner',
'lisa@example.com',  'password567', 4105, 1203);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

----------next----------

```sql
-- Insert Data  ProgramCoordinator Table
INSERT INTO ProgramCoordinator (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_TYPE, PC_NOTICE, MEMBER_ID, D_ID, EC_ID, ROLE_ID) VALUES
(seq_programcoordinator.NEXTVAL, 'Alice Brown',  'alice@example.com',  'password123', 'Coordinator',   'Notice',      5102, 9101, 3101, 1202);
INSERT INTO ProgramCoordinator (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_TYPE, PC_NOTICE, MEMBER_ID, D_ID, EC_ID, ROLE_ID) VALUES
(seq_programcoordinator.NEXTVAL, 'Frank Moore',  'frank@example.com',  'password234', 'Lead',          'Announcement', 5103, 9102, 3102, 1202);
INSERT INTO ProgramCoordinator (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_TYPE, PC_NOTICE, MEMBER_ID, D_ID, EC_ID, ROLE_ID) VALUES
(seq_programcoordinator.NEXTVAL, 'Nancy Taylor', 'nancy@example.com',  'password345', 'Supervisor',    'Alert',       5104, 9103, 3103, 1202);
INSERT INTO ProgramCoordinator (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_TYPE, PC_NOTICE, MEMBER_ID, D_ID, EC_ID, ROLE_ID) VALUES
(seq_programcoordinator.NEXTVAL, 'Larry King',   'larry@example.com',  'password456', 'Administrator    ', 'Message',     5106, 9104, 3104, 1202);
INSERT INTO ProgramCoordinator (PC_ID, PC_NAME, PC_EMAIL, PC_PASSWORD, PC_TYPE, PC_NOTICE, MEMBER_ID, D_ID, EC_ID, ROLE_ID) VALUES
(seq_programcoordinator.NEXTVAL, 'Sandra Lee',   'sandra@example.com', 'password567', 'Administrator    ', 'Update',      5101, 9105, 3105, 1202);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

BACK

```
-- Insert Data  Client Table
INSERT INTO Client (CLIENT_ID, CLIENT_NAME, CLIENT_EMAIL, CLIENT_CARDNO, CLIENT_PASSWORD, ACCOUNT_ID, PHOTOGRAPHER_ID, D_ID, ROLE_ID) VALUES
(seq_client.NEXTVAL, 'Bob Johnson',  'bob@gmail.com',    '1234567890123456', 'password123', 10101, 6102, 9101, 1204);
INSERT INTO Client (CLIENT_ID, CLIENT_NAME, CLIENT_EMAIL, CLIENT_CARDNO, CLIENT_PASSWORD, ACCOUNT_ID, PHOTOGRAPHER_ID, D_ID, ROLE_ID) VALUES
(seq_client.NEXTVAL, 'Rachel Adams', 'rachel@gmail.com', '2345678901234567', 'password234', 10102, 6103, 9102, 1204);
INSERT INTO Client (CLIENT_ID, CLIENT_NAME, CLIENT_EMAIL, CLIENT_CARDNO, CLIENT_PASSWORD, ACCOUNT_ID, PHOTOGRAPHER_ID, D_ID, ROLE_ID) VALUES
(seq_client.NEXTVAL, 'Steven Hall',  'steven@gmail.com', '3456789012345678', 'password345', 10103, 6104, 9103, 1204);
INSERT INTO Client (CLIENT_ID, CLIENT_NAME, CLIENT_EMAIL, CLIENT_CARDNO, CLIENT_PASSWORD, ACCOUNT_ID, PHOTOGRAPHER_ID, D_ID, ROLE_ID) VALUES
(seq_client.NEXTVAL, 'Laura Bush',   'laura@gmail.com',  '4567890123456789', 'password456', 10104, 6105, 9104, 1204);
INSERT INTO Client (CLIENT_ID, CLIENT_NAME, CLIENT_EMAIL, CLIENT_CARDNO, CLIENT_PASSWORD, ACCOUNT_ID, PHOTOGRAPHER_ID, D_ID, ROLE_ID) VALUES
(seq_client.NEXTVAL, 'James Franco', 'james@gmail.com',  '5678901234567890', 'password567', 10105, 6105, 9105, 1204);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

----------next----------

```
-- Insert Data  Advisor Table
INSERT INTO Advisor (A_ID, A_NAME, A_EMAIL, A_PASSWORD, PC_ID, ACCOUNT_ID,ROLE_ID) VALUES (seq_advisor.NEXTVAL, 'Charles Green', 'charles@example.com',
'password123', 2101, 10101, 1201);
INSERT INTO Advisor (A_ID, A_NAME, A_EMAIL, A_PASSWORD, PC_ID, ACCOUNT_ID,ROLE_ID) VALUES (seq_advisor.NEXTVAL, 'Donna Red',    'donna@example.com',
'password234', 2102, 10102, 1201);
INSERT INTO Advisor (A_ID, A_NAME, A_EMAIL, A_PASSWORD, PC_ID, ACCOUNT_ID,ROLE_ID) VALUES (seq_advisor.NEXTVAL, 'Edward Blue',   'edward@example.com',
'password345', 2103, 10103, 1201);
INSERT INTO Advisor (A_ID, A_NAME, A_EMAIL, A_PASSWORD, PC_ID, ACCOUNT_ID,ROLE_ID) VALUES (seq_advisor.NEXTVAL, 'Gina Yellow',   'gina@example.com',
'password456', 2106, 10104, 1201);
INSERT INTO Advisor (A_ID, A_NAME, A_EMAIL, A_PASSWORD, PC_ID, ACCOUNT_ID,ROLE_ID) VALUES (seq_advisor.NEXTVAL, 'Henry White',   'henry@example.com',
'password567', 2107, 10105, 1201);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

----------next----------

```
-- Insert Data  Contact Table
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 1102,'123-456-7890', 1201);
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 2101,'123-456-7890', 1202);
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 3101,'123-456-7890', 1203);
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 7102,'123-456-7890', 1204);
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 6102,'123-456-7890', 1206);
INSERT INTO Contact (CON_ID, REFERENCE_ID, PC_PHONE, ROLE_ID) VALUES (seq_contact.NEXTVAL, 5102,'123-456-7890', 1205);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.
```

BACK

SELECT

```
SELECT * FROM ROLE
```

Results Explain Describe Save

ROLE_ID	ROLE_NAME
1201	Advisor
1202	Program Coordinator
1203	Event Coordinator
1204	Client
1205	Member
1206	Photographer

6 rows returned in 0.02 seconds

----------next----------

```
SELECT *FROM ADDRESS
```

Results Explain Describe Saved SQL History

D_ID	ZIP_CODE	ROAD_NO	CITY	HOUSE_NO
9101	12345	101	New York	1A
9102	23456	102	Los Angeles	2B
9103	34567	103	Chicago	3C
9104	45678	104	Houston	4D
9105	56789	105	Phoenix	5E

5 rows returned in 0.00 seconds CSV Export

----------next----------

```
SELECT * FROM ACCOUNT
```

Results Explain Describe Sav

ACCOUNT_ID	BALANCE
10101	1000
10102	1500
10103	2000
10104	2500
10105	3000

5 rows returned in 0.00 seconds

BACK

```
SELECT * FROM EVENT
```

Results Explain Describe Saved SQL History

EVENT_ID	EVENT_NAME	EVENT_DATE	EVENT_LOCATION
4101	Event 1	01-DEC-23	Location 1
4102	Event 2	02-DEC-23	Location 2
4103	Event 3	03-DEC-23	Location 3
4104	Event 4	04-DEC-23	Location 4
4105	Event 5	05-DEC-23	Location 5

----------next----------

```
SELECT * FROM Photographer
```

Results Explain Describe Saved SQL History

PHOTOGRAPHER_ID	PHOTOGRAPHER_NAME	PHOTOGRAPHER_TYPE	D_ID	ROLE_ID	PHOTOGRAPHER_EMAIL
6101	John Doe	Portrait	9101	1206	john@gmail.com
6102	Jane Smith	Wedding	9102	1206	jane@gmail.com
6103	Emily Johnson	Nature	9103	1206	emily@gmail.com
6104	Michael Brown	Fashion	9104	1206	michael@gmail.com
6105	Linda Davis	Event	9105	1206	linda@gmail.com

----------next----------

```
SELECT * FROM Member
```

Results Explain Describe Saved SQL History

MEMBER_ID	MEMBER_NAME	MEMBER_EMAIL	MEMBER_AGE	NOTIFICATION	MEMBER_PASSWORD	EVENT_ID	D_ID	PHOTOGRAPHER_ID	ROLE_ID
5101	John Doe	john@example.com	25	Text	password123	4101	9101	6102	1205
5102	Emily Clark	emily@example.com	30	Email	password321	4102	9102	6103	1205
5103	Michael Brown	michael@example.com	40	Phone	password456	4103	9103	6104	1205
5104	Linda Smith	linda.@example.com	22	App	password789	4104	9104	6105	1205
5106	Robert Johnson	robert@example.com	35	Mail	password654	4105	9105	6101	1205

5 rows returned in 0.02 seconds CSV Export

```
SELECT * FROM EventCoordinator
```

Results Explain Describe Saved SQL History

EC_ID	EC_NAME	EC_EMAIL	EC_PASSWORD	EVENT_ID	ROLE_ID
3101	Jane Smith	jane@example.com	password123	4101	1203
3102	Paul Brooks	paul@example.com	password234	4102	1203
3103	Susan Hill	susan@example.com	password345	4103	1203
3104	Gary White	gary@example.com	password456	4104	1203
3105	Lisa Turner	lisa@example.com	password567	4105	1203

BACK

```
SELECT * FROM ProgramCoordinator
```

Results Explain Describe Saved SQL History

PC_ID	PC_NAME	PC_EMAIL	PC_PASSWORD	PC_TYPE	PC_NOTICE	MEMBER_ID	D_ID	EC_ID	ROLE_ID
2101	Alice Brown	alice@example.com	password123	Coordinator	Notice	5102	9101	3101	1202
2102	Frank Moore	frank@example.com	password234	Lead	Announcement	5103	9102	3102	1202
2103	Nancy Taylor	nancy@example.com	password345	Supervisor	Alert	5104	9103	3103	1202
2106	Larry King	larry@example.com	password456	Administrator	Message	5106	9104	3104	1202
2107	Sandra Lee	sandra@example.com	password567	Administrator	Update	5101	9105	3105	1202

5 rows returned in 0.00 seconds CSV Export

----------next----------

```
SELECT * FROM Client
```

Results Explain Describe Saved SQL History

CLIENT_ID	CLIENT_NAME	CLIENT_EMAIL	CLIENT_CARDNO	CLIENT_PASSWORD	ACCOUNT_ID	PHOTOGRAPHER_ID	D_ID	ROLE_ID
7101	Bob Johnson	bob@gmail.com	1234567890123456	password123	10101	6102	9101	1204
7102	Rachel Adams	rachel@gmail.com	2345678901234567	password234	10102	6103	9102	1204
7103	Steven Hall	steven@gmail.com	3456789012345678	password345	10103	6104	9103	1204
7104	Laura Bush	laura@gmail.com	4567890123456789	password456	10104	6105	9104	1204
7106	James Franco	james@gmail.com	5678901234567890	password567	10105	6105	9105	1204

5 rows returned in 0.01 seconds CSV Export

----------next----------

```
SELECT * FROM Advisor
```

Results Explain Describe Saved SQL History

A_ID	A_NAME	A_EMAIL	A_PASSWORD	PC_ID	ACCOUNT_ID	ROLE_ID
1101	Charles Green	charles@example.com	password123	2101	10101	1201
1102	Donna Red	donna@example.com	password234	2102	10102	1201
1103	Edward Blue	edward@example.com	password345	2103	10103	1201
1105	Gina Yellow	gina@example.com	password456	2106	10104	1201
1106	Henry White	henry@example.com	password567	2107	10105	1201

5 rows returned in 0.00 seconds CSV Export

```
SELECT * FROM CONTACT
```

Results Explain Describe Saved SQL History

CON_ID	REFERENCE_ID	PC_PHONE	ROLE_ID
8101	1102	123-456-7890	1201
8102	2101	123-456-7890	1202
8103	3101	123-456-7890	1203
8104	7102	123-456-7890	1204
8105	6102	123-456-7890	1206
8106	5102	123-456-7890	1205

6 rows returned in 0.02 seconds CSV Export

BACK

Query Writing

Needed Privilege:

```
CREATE PUBLIC SYNONYM EVENT FOR EVENT;
```

Results Explain Describe Saved SQL History

```
ORA-00955: name is already used by an existing object
```

```
SYSTEM:
GRANT INSERT ANY TABLE TO EC;
GRANT EC TO ECO;
PCO USER:
GRANT INSERT ON PCO.EVENT TO ECO;
GRANT SELECT ON PCO.seq_event TO ECO;
```

----------next----------

```
PCO USER:
GRANT INSERT ON PCO.EVENT TO ECO;
GRANT SELECT ON PCO.seq_event TO ECO;
```

----------next----------

```
CREATE SYNONYM CLIENTS FOR PCO.CLIENT;
GRANT SELECT, INSERT, UPDATE, DELETE ON PCO.CLIENT TO ECO;
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

```
CREATE SYNONYM CLIENTS FOR PCO.CLIENT;
GRANT SELECT ON CLIENTS TO CLI;
GRANT SELECT, INSERT, UPDATE, DELETE ON PCO.CLIENT TO CLI;
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

BACK

```
SELECT * FROM PCO.CLIENTS
```

Results Explain Describe Saved SQL History

CLIENT_ID	CLIENT_NAME	CLIENT_EMAIL	CLIENT_CARDNO	CLIENT_PASSWORD	ACCOUNT_ID	PHOTOGRAPHER_ID	D_ID	ROLE_ID
7101	Bob Johnson	bob@gmail.com	1234567890123456	password123	10101	6102	9101	1204
7102	Rachel Adams	rachel@gmail.com	2345678901234567	password234	10102	6103	9102	1204
7103	Steven Hall	steven@gmail.com	3456789012345678	password345	10103	6104	9103	1204
7104	Laura Bush	laura@gmail.com	4567890123456789	password456	10104	6105	9104	1204
7106	James Franco	james@gmail.com	5678901234567890	password567	10105	6105	9105	1204

5 rows returned in 0.01 seconds CSV Export

----------next----------

User: ECO

Home > SQL > SQL Commands

Autocommit Display 10

```
INSERT INTO PCO.EVENT (EVENT_ID, EVENT_NAME, EVENT_DATE, EVENT_LOCATION)
VALUES (PCO.seq_event.NEXTVAL, 'Event 6', TO_DATE('2024-02-15', 'YYYY-MM-DD'), 'Location 6');

SELECT * FROM EVENT
```

Results Explain Describe Saved SQL History

EVENT_ID	EVENT_NAME	EVENT_DATE	EVENT_LOCATION
4101	Event 1	01-DEC-23	Location 1
4102	Event 2	02-DEC-23	Location 2
4103	Event 3	03-DEC-23	Location 3
4104	Event 4	04-DEC-23	Location 4
4105	Event 5	05-DEC-23	Location 5
4106	Event 6	15-FEB-24	Location 6

6 rows returned in 0.00 seconds CSV Export

----------next----------

```
CREATE SYNONYM MEMBERS FOR PCO.MEMBER;
GRANT SELECT ON  MEMBERS TO ECO;
```

Results Explain Describe Saved SQL History

Statement processed.

----------next----------

```
CREATE SYNONYM PHOTOGRAPHERS FOR PCO.PHOTOGRAPHER;
GRANT SELECT ON  PHOTOGRAPHER TO ECO;
```

Results Explain Describe Saved SQL History

Statement processed.

BACK

Single Row Function (Event Coordinator, Program Coordinator, (question 3 for all))

1. **Concatenate Event Name and Location.**

```
SELECT event_id, event_name || ' - ' || event_location AS event_details
FROM event;
```

Results Explain Describe Saved SQL History

EVENT_ID	EVENT_DETAILS
4101	Event 1 - Location 1
4102	Event 2 - Location 2
4103	Event 3 - Location 3
4104	Event 4 - Location 4
4105	Event 5 - Location 5

5 rows returned in 0.00 seconds CSV Export

----------next----------

2. **Formatting Client Email Addresses.**

```
SELECT client_id, LOWER(client_email) AS email_lowercase
FROM client;
```

Results Explain Describe Saved SQL History

CLIENT_ID	EMAIL_LOWERCASE
7101	bob@gmail.com
7102	rachel@gmail.com
7103	steven@gmail.com
7104	laura@gmail.com
7106	james@gmail.com

BACK

3. **Calculate the total number of events organized by the club.**

```
SELECT COUNT(*) AS total_events
FROM event;
```

Results Explain Describe Saved SQL History

TOTAL_EVENTS
5

1 rows returned in 0.00 seconds CSV Export

----------next----------

4. **Retrieve the event name, date, and location for the next scheduled event.**

```
SELECT event_name, event_date, event_location
FROM (
    SELECT event_name, event_date, event_location
    FROM event
    WHERE event_date > SYSDATE
    ORDER BY event_date
)
WHERE ROWNUM = 1;
```

Results Explain Describe Saved SQL History

EVENT_NAME	EVENT_DATE	EVENT_LOCATION
Event 1	01-DEC-23	Location 1

BACK

Group Function (Program Coordinator, (question 3 for Event coordinator and progragram coordinator also))

1. **List all upcoming events coordinated by a specific event coordinator, including event details.**

```
SELECT e.event_name, e.event_date, e.event_location
FROM event e
JOIN eventcoordinator ec ON e.event_id = ec.event_id
WHERE ec.ec_name = 'Susan Hill'
AND e.event_date >= CURRENT_DATE;
```

Results Explain Describe Saved SQL History

EVENT_NAME	EVENT_DATE	EVENT_LOCATION
Event 3	03-DEC-23	Location 3

----------next----------

2. **Calculate and display the total number of events coordinated by each event coordinator.**

```
SELECT ec.ec_name, COUNT(e.event_id) AS total_events_coordinated
FROM eventcoordinator ec
JOIN event e ON ec.event_id = e.event_id
GROUP BY ec.ec_name;
```

Results Explain Describe Saved SQL History

EC_NAME	TOTAL_EVENTS_COORDINATED
Gary White	1
Jane Smith	1
Susan Hill	1
Paul Brooks	1
Lisa Turner	1

5 rows returned in 0.01 seconds CSV Export

BACK

3. **Write a SQL query to find the address of members.**

```
SELECT m.member_name, a.city, a.road_no, a.zip_code, a.house_no
FROM member m
JOIN address a ON m.d_id = a.d_id;
```

Results Explain Describe Saved SQL History

MEMBER_NAME	CITY	ROAD_NO	ZIP_CODE	HOUSE_NO
John Doe	New York	101	12345	1A
Emily Clark	Los Angeles	102	23456	2B
Michael Brown	Chicago	103	34567	3C
Linda Smith	Houston	104	45678	4D
Robert Johnson	Phoenix	105	56789	5E

----------next----------

Subquery (Event coordinator and program coordinator)

1. **Find all events where a photographer of type Portrait Wedding participated.**

```
SELECT e.event_name
FROM event e
WHERE e.event_id IN (
    SELECT m.event_id
    FROM member m
    JOIN photographer p ON m.photographer_id = p.photographer_id
    WHERE p.photographer_type = 'Portrait'
);
```

Results Explain Describe Saved SQL History

EVENT_NAME
Event 5

BACK

2. **List the names of members who were assigned to photographers of type WEDDING**

```
SELECT DISTINCT m.member_name
FROM member m
JOIN photographer p ON m.photographer_id = p.photographer_id
WHERE p.photographer_type = 'Wedding'
AND m.event_id IN (
    SELECT DISTINCT m.event_id
    FROM member m
);
```

Results Explain Describe Saved SQL History

MEMBER_NAME
John Doe

1 rows returned in 0.00 seconds CSV Export

----------next----------

3. **Calculate and display the number of events in which members were assigned to photographers of type NATURE.**

```
SELECT m.member_name, COUNT(m.event_id) AS events_participated
FROM member m
JOIN photographer p ON m.photographer_id = p.photographer_id
WHERE p.photographer_type = 'NATURE'
GROUP BY m.member_name;
```

Results Explain Describe Saved SQL History

no data found

BACK

Joining query (Program and event coordinator)

1. **Retrieve a list of clients along with the names and contact information of their associated photographers.**

```
SELECT C.Client_ID, C.Client_Name, P.Photographer_Id
FROM CLIENT C
LEFT JOIN PHOTOGRAPHER P ON C.photographer_id = P.Photographer_ID;
```

Results Explain Describe Saved SQL History

CLIENT_ID	CLIENT_NAME	PHOTOGRAPHER_ID
7101	Bob Johnson	6102
7102	Rachel Adams	6103
7103	Steven Hall	6104
7104	Laura Bush	6105
7106	James Franco	6105

----------next----------

2. **Identify photographers who do not have any clients.**

```
SELECT P.Photographer_ID, P.Photographer_Name, P.D_ID
FROM Photographer P
LEFT JOIN Client C ON P.Photographer_ID = C.photographer_id
WHERE C.Client_ID IS NULL;
```

Results Explain Describe Saved SQL History

PHOTOGRAPHER_ID	PHOTOGRAPHER_NAME	D_ID
6101	John Doe	9101

----------next----------

3. **Find the addresses of all members who participated in Event 1**.

```
SELECT m.member_name, a.*
FROM member m
JOIN address a ON m.d_id = a.d_id
WHERE m.event_id IN (
    SELECT event_id
    FROM event
    WHERE event_name = 'Event 1'
);
```

Results Explain Describe Saved SQL History

MEMBER_NAME	D_ID	ZIP_CODE	ROAD_NO	CITY	HOUSE_NO
John Doe	9101	12345	101	New York	1A

BACK

##Views:(Program coordinator)

1. **Create a view named MemberEventView that displays member names and their corresponding event names.**

```
CREATE VIEW MemberEventView AS
SELECT m.member_name, e.event_name
FROM member m
JOIN event e ON m.event_id = e.event_id;
```

Results Explain Describe Saved SQL History

```
View created.
```

----------next----------

```
SELECT * FROM MemberEventView
```

Results Explain Describe Saved SQL Histo

MEMBER_NAME	EVENT_NAME
John Doe	Event 1
Emily Clark	Event 2
Michael Brown	Event 3
Linda Smith	Event 4
Robert Johnson	Event 5

----------next----------

2. **Create a view named PhotographerClientView that displays photographers and their associated clients.**

```
CREATE VIEW PhotographerClientView AS
SELECT p.photographer_name, c.client_name
FROM photographer p
JOIN client c ON p.photographer_id = c.photographer_id;
```

Results Explain Describe Saved SQL History

```
View created.
```

BACK

```
SELECT * FROM PhotographerClientView
```

Results Explain Describe Saved SQL History

PHOTOGRAPHER_NAME	CLIENT_NAME
Jane Smith	Bob Johnson
Emily Johnson	Rachel Adams
Michael Brown	Steven Hall
Linda Davis	Laura Bush
Linda Davis	James Franco

----------next----------

3. **Create a view named EventCoordinatorView that displays event coordinators and their upcoming events.**

```
CREATE VIEW UpcomingEventsView AS
SELECT ec.ec_name, e.event_name, e.event_date
FROM eventcoordinator ec
JOIN event e ON ec.event_id = e.event_id
WHERE e.event_date >= SYSDATE;
```

Results Explain Describe Saved SQL History

```
View created.
```

```
SELECT * FROM UpcomingEventsView
```

Results Explain Describe Saved SQL History

EC_NAME	EVENT_NAME	EVENT_DATE
Jane Smith	Event 1	01-DEC-23
Paul Brooks	Event 2	02-DEC-23
Susan Hill	Event 3	03-DEC-23
Gary White	Event 4	04-DEC-23
Lisa Turner	Event 5	05-DEC-23

BACK

Synonym

```
CREATE SYNONYM PRCO FOR PROGRAMCOORDINATOR;
CREATE SYNONYM EVCO FOR EVENCOORDINATOR;
CREATE SYNONYM PROGRAM FOR EVEN;
```

Results Explain Describe Saved SQL History

```
Synonym created.
```

----------next----------

```
SELECT * FROM EVCO
```

Results Explain Describe Saved SQL History

EC_ID	EC_NAME	EC_EMAIL	EC_PASSWORD	EVENT_ID	ROLE_ID
3101	Jane Smith	jane@example.com	password123	4101	1203
3102	Paul Brooks	paul@example.com	password234	4102	1203
3103	Susan Hill	susan@example.com	password345	4103	1203
3104	Gary White	gary@example.com	password456	4104	1203
3105	Lisa Turner	lisa@example.com	password567	4105	1203

----------next----------

```
SELECT * FROM PROGRAM;
```

Results Explain Describe Saved SQL History

EVENT_ID	EVENT_NAME	EVENT_DATE	EVENT_LOCATION
4101	Event 1	01-DEC-23	Location 1
4102	Event 2	02-DEC-23	Location 2
4103	Event 3	03-DEC-23	Location 3
4104	Event 4	04-DEC-23	Location 4
4105	Event 5	05-DEC-23	Location 5

BACK

```
SELECT * FROM PRCO
```

Results Explain Describe Saved SQL History

PC_ID	PC_NAME	PC_EMAIL	PC_PASSWORD	PC_TYPE	PC_NOTICE	MEMBER_ID	D_ID	EC_ID	ROLE_ID
2101	Alice Brown	alice@example.com	password123	Coordinator	Notice	5102	9101	3101	1202
2102	Frank Moore	frank@example.com	password234	Lead	Announcement	5103	9102	3102	1202
2103	Nancy Taylor	nancy@example.com	password345	Supervisor	Alert	5104	9103	3103	1202
2106	Larry King	larry@example.com	password456	Administrator	Message	5106	9104	3104	1202
2107	Sandra Lee	sandra@example.com	password567	Administrator	Update	5101	9105	3105	1202

Relational Algebra

1. List of Event names, id, date and location (Using projection)

Ans: πEvent_ID, Event_Name, Event_Date, Event_Location(Event)

2. List of all members who are Photographer

Ans: πMember_Name(σ*Member*_Role='Photographer'(Member))

3. List of all Upcoming events

Ans: πEvent_ID, Event_Name, Event_Date(σEvent_Date > CurrentDate(Event))

4. List of clients who have photographer assigned

Ans: πClient_ID, Client_Name, Photographer_ID (σPhotographer_ID is not null(Client))

5. List of Event coordinator name, id and event_id who are currently involved in an event

Ans: π*EC_ID,EC_Name,Event_ID*(σEvent_ID is not null (EVENTCOORDINATOR))

Conclusion

In conclusion, the Photography Club Management System has successfully addressed the challenges faced by photography clubs, providing an efficient solution for membership management, event coordination, event marketing, communication, documentation, and attendance tracking. The implementation of this system has resulted in improved efficiency, enhanced communication, optimized resource usage, streamlined documentation, a user-friendly interface, increased member engagement, and customization/scalability options.

Through the development process, the system's key features, such as member management, event planning, marketing, communication platform, document repository, and attendance tracking, carefully design and will implement using C# for interactive fronted and backend processing.

Future Work

To further enhance the "Photography Club Management System" and ensure its continued relevance and effectiveness, the following areas of improvement and future work are proposed:

- Mobile Application Development: Develop a dedicated mobile application to extend the accessibility of the system, allowing members and administrators to manage club activities on-the-go, enhancing convenience and user experience.
- Integration with External Platforms: Enable integration with popular photography and social media platforms to promote events, share achievements, and foster a broader community beyond the confines of the club.
- Feedback Mechanism: Implement a feedback system to gather input from members, administrators, and coordinators. This will aid in identifying areas of improvement and refining the system based on user experiences and preferences.
- Enhanced User Customization: Provide additional customization options for clubs to tailor the system to their specific needs, allowing for a more personalized and adaptable user experience.
- Incorporation of Virtual Events: With the rise of virtual participation, enhance the system to support and manage virtual photography events, workshops, and exhibitions, expanding the scope of club activities.
- Continuous Training and Support: Establish a comprehensive training program and support system for club administrators and members to ensure efficient onboarding and continued successful utilization of the system.

The Photography Club Management System will evolve into an even more robust and versatile platform, staying ahead of the evolving needs and trends within the dynamic realm of photography clubs. This commitment to continuous improvement ensures that the system remains a valuable asset for photography enthusiasts and professionals in fostering vibrant and engaged club communities.

END