



DEPARTMENT OF  
COMPUTER  
APPLICATIONS

# A Project Report on YSA NGO

---

By:

**Abbas Ghoghari  
Yash Dhokiya**

Submitted in partial fulfillment of the requirements  
for the degree

Bachelor of Computer Applications  
October 2025

Guided By:

Dr. Riddhi Tanna

# TABLE OF CONTENTS

|    |                              |       |
|----|------------------------------|-------|
| 01 | Introduction                 |       |
|    | Introduction                 | ..... |
|    | Objectives                   | ..... |
| 02 | System Analysis              |       |
|    | Problem Definition           | ..... |
|    | Feasibility Study            | ..... |
|    | SDLC                         | ..... |
| 03 | System Design                |       |
|    | DFD                          | ..... |
|    | ER Diagram                   | ..... |
|    | Activity Diagram             | ..... |
|    | Use Case and Process Diagram | ..... |
| 04 | Data Dictionary              | 23    |
| 05 | Implementation               |       |
|    | Screenshots                  | ..... |
|    | Code Explanation             | ..... |
| 06 | Testing                      |       |
|    | Test Case                    | ..... |
|    | Result                       | ..... |
| 07 | Conclusion & Future Scope    | 43    |

## CERTIFICATE

This is to certify that the work presented in the BCA project report entitled “**YSA NGO**”, submitted to the Department of Computer Applications, Christ College, Rajkot, in partial fulfilment of the requirements for the award of the Bachelor of Computer Applications, is an authentic record of the work carried out by **Yash Dhokiya and Abbas Ghoghari** during the period from June 2025 to October 2025 under the supervision of **Dr. Riddhi Tanna**.

We also certify that this work has not been submitted, either in part or full, for the award of any other degree elsewhere.

**Signature of Yash Dhokia** : \_\_\_\_\_

**Signature of Abbas Ghoghari** : \_\_\_\_\_

This is to certify that the above statement made by the candidate(s) is/are correct to the best of my knowledge.

**Head**

Dr. Shailendrasinh Jadeja

Department of Computer  
Applications

Christ College, Rajkot

**Guide**

Dr.Riddhi Tanna

Department of Computer  
Applications

Christ College, Rajkot

**Place:** Rajkot

**Date:** \_\_\_\_\_

## ACKNOWLEDGEMENT

When we embarked on this project, it initially appeared to us as an onerous task. However, as we progressed, we realized that we were not alone in this journey. We wish to express our sincere gratitude to **Rev. Fr. (Dr.) Jomon Thommana**, Director, Christ Campus, and **Dr. Yvonne Fernandes**, Principal, Christ College, for their constant support and encouragement.

We are deeply thankful to **Dr. Riddhi Tanna**, our project guide, for their invaluable guidance, suggestions, and the freedom they provided us throughout this project. Without their mentorship, the completion of this project would not have been possible.

We also extend our heartfelt thanks to **Ms.Pintukumari Bera**, our friends for offering their unwavering support, encouragement, and assistance during the preparation of this project.

Our gratitude also goes to our group members, classmates, and well-wishers, whose generous help made this work considerably easier. Finally, our deepest thanks to our parents, who patiently bore with us through the challenges and supported us with a smile throughout our study period and beyond.

**Yash Dhokiya**

**Abbas Ghoghari**

Christ College, Rajkot

**October 2025**

## **INTRODUCTION**

### **Definition**

The NGO Website is a web-based application designed to provide a digital platform for Non-Governmental Organizations (NGOs). It allows NGOs to showcase their mission and activities, manage volunteer registrations, accept donations, and share updates about events. The system uses HTML, CSS, and JavaScript for the frontend to ensure an interactive user interface, and PHP with MySQL (via XAMPP) for the backend to handle data storage and dynamic content management.

### **Goal**

The goal of this project is to design and develop a dynamic website for a NonGovernmental Organization (NGO) to improve its online presence, manage volunteers, promote causes, collect donations, and spread awareness about its initiatives.

### **Need**

Many NGOs operate without a proper digital platform, limiting their outreach and fundraising potential. Current methods such as paper-based systems or static websites do not support dynamic interaction with users or data management.

## **OBJECTIVES**

### **1. Health & Wellness**

- To improve community health by organizing regular medical check-up camps, with a focus on children and the elderly in underserved areas of Navapur.
- To conduct awareness programs on crucial health topics such as nutrition, hygiene, and preventive care to reduce the incidence of common illnesses.

### **2. Education & Skill Development**

- To provide after-school academic support and career guidance to students to reduce school dropout rates.
- To empower local youth and women with vocational training and digital literacy skills to enhance their livelihood opportunities.

### **3. Environmental Sustainability**

- To promote a cleaner and greener Navapur by leading community-driven tree plantation and clean-up campaigns.
- To raise awareness about sustainable practices like waste management, water conservation, and recycling among local households.

### **4. Social Empowerment & Livelihood**

- To facilitate the formation of Women's Self-Help Groups (SHGs) to foster financial independence and entrepreneurship.
- To connect community members with relevant government welfare schemes and ensure they can access their rights and entitlements.

## **SYSTEM ANALYSIS**

### **PROBLEM DEFINITION**

The YSA NGO website system is designed to streamline the process of managing donations, events, and volunteer activities for the organization. It analyzes user roles into donors, volunteers, and administrators, each with specific access rights and functionalities. The system collects, stores, and manages data related to donation campaigns, event schedules, and user information securely. It provides real-time updates on ongoing drives and upcoming programs to ensure transparency and active engagement.

The system architecture incorporates a responsive frontend for accessibility across devices and a robust backend to handle data processing and user authentication. Security measures such as encrypted logins and role-based access control protect sensitive data. The system ensures smooth navigation and efficient workflow for stakeholders, minimizing manual operations. Overall, it integrates technology and social service goals to create a reliable and user-centric platform for the NGO's mission.

## FEASIBILITY STUDY

Before developing the proposed system, it is important to evaluate its feasibility in terms of practicality, cost, and usefulness. A feasibility study helps determine whether the project is viable and worth pursuing. The study can be categorized into the following aspects:

### 1. Technical Feasibility

The project is technically feasible since the required hardware and software resources are readily available. The development tools, programming languages, and database systems selected are reliable, widely used, and well-supported, ensuring smooth implementation.

### 2. Economic Feasibility

The system is economically viable because it is developed as part of an academic project, requiring minimal cost. The software tools and platforms used are either open-source or provided by the institution, which reduces expenses. The potential benefits of automation, efficiency, and time-saving outweigh the minimal development cost.

### 3. Operational Feasibility

The proposed system is user-friendly, efficient, and designed to meet the end-user's requirements. It simplifies existing manual processes, reduces errors, and ensures accurate results. Users can easily adapt to the system with minimal training, ensuring smooth operation.

### 4. Schedule Feasibility

The project is scheduled within the academic timeframe (June 15 to October 25). By following the Software Development Life Cycle (SDLC), the phases of analysis, design, implementation, testing, and deployment are well-planned to ensure timely completion.

## SDLC

### SDLC TIMELINE FOR STUDENT PROJECT (JUNE 15 – OCT 25)

#### 1. PLANNING (JUNE 15 – JUNE 25) → 10 DAYS

- DEFINE PROJECT TOPIC & OBJECTIVES.
- IDENTIFY SCOPE (WHAT TO INCLUDE/EXCLUDE).
- ASSIGN ROLES IN THE TEAM.
- DECIDE TOOLS/TECH (LANGUAGE, DB, IDE).

#### 2. REQUIREMENT ANALYSIS (JUNE 26 – JULY 5) → 10 DAYS

- COLLECT REQUIREMENTS FROM GUIDES/MENTORS.
- PREPARE SRS (SOFTWARE REQUIREMENT SPECIFICATION).
- FINALIZE PROJECT MODULES & FEATURES.

#### 3. SYSTEM DESIGN (JULY 6 – JULY 20) → 15 DAYS

- CREATE ER DIAGRAMS, DFDs, UML DIAGRAMS.
- DESIGN DATABASE SCHEMA.
- SKETCH USER INTERFACE (WIREFRAMES).

#### 4. IMPLEMENTATION / CODING (JULY 21 – SEPT 10) → ~50 DAYS

- DIVIDE MODULES AMONG TEAM MEMBERS.
- START CODING (FRONTEND + BACKEND + DB).
- CONDUCT UNIT TESTING OF EACH MODULE.

#### 5. TESTING & INTEGRATION (SEPT 11 – SEPT 30) → 20 DAYS

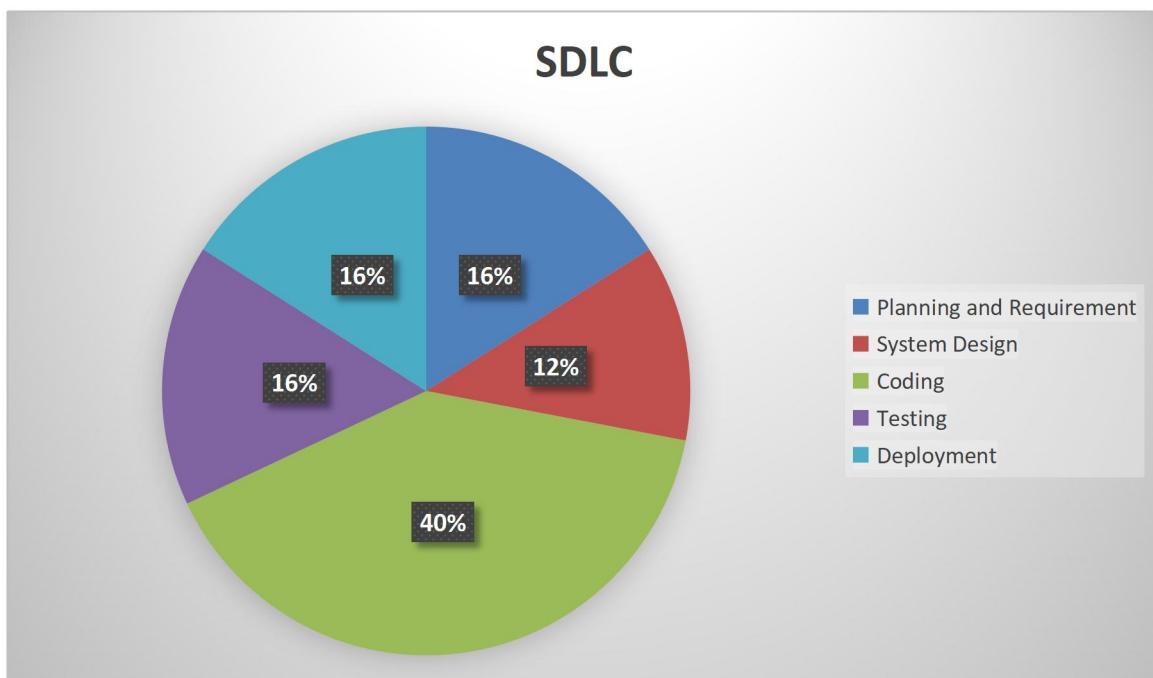
- INTEGRATE ALL MODULES.
- DO SYSTEM TESTING (FUNCTIONAL, USABILITY, SECURITY).
- DEBUG & FIX ERRORS.
- PREPARE TEST REPORT.

**6. DEPLOYMENT / FINALIZATION (OCT 1 – OCT 10) → 10 DAYS**

- DEPLOY ON SYSTEM (OR DEMO ENVIRONMENT).
- ENSURE ALL FEATURES WORK AS EXPECTED.
- TAKE FEEDBACK FROM FACULTY/PEERS.

**7. DOCUMENTATION & SUBMISSION PREP (OCT 11 – OCT 25) → 15 DAYS**

- COMPLETE FINAL PROJECT REPORT.
- ADD DIAGRAMS, SCREENSHOTS, TESTING RESULTS.
- PREPARE PPT FOR VIVA/DEMO.
- FINAL REVIEW & SUBMISSION.



## SYSTEM DESIGN

### **DFD**

A DFD also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it presents a picture. The DFD may be partitioned into levels that represent increasing information flow and functional detail. Four simple notations are used to complete a DFD.

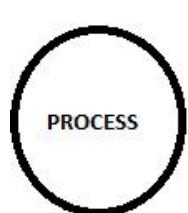
### **DATA FLOW**



The data flow is used to describe the movement of information from one part of the system to another part. Flows represent data in motion. It is a pipe line through which information flows. Data flow is represented by an arrow.

### **PROCESS**

A circle or bubble represents a process that transforms incoming data to outgoing data.



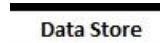
Process shows a part of the system that transforms inputs to outputs.

### **EXTERNAL ENTITY**

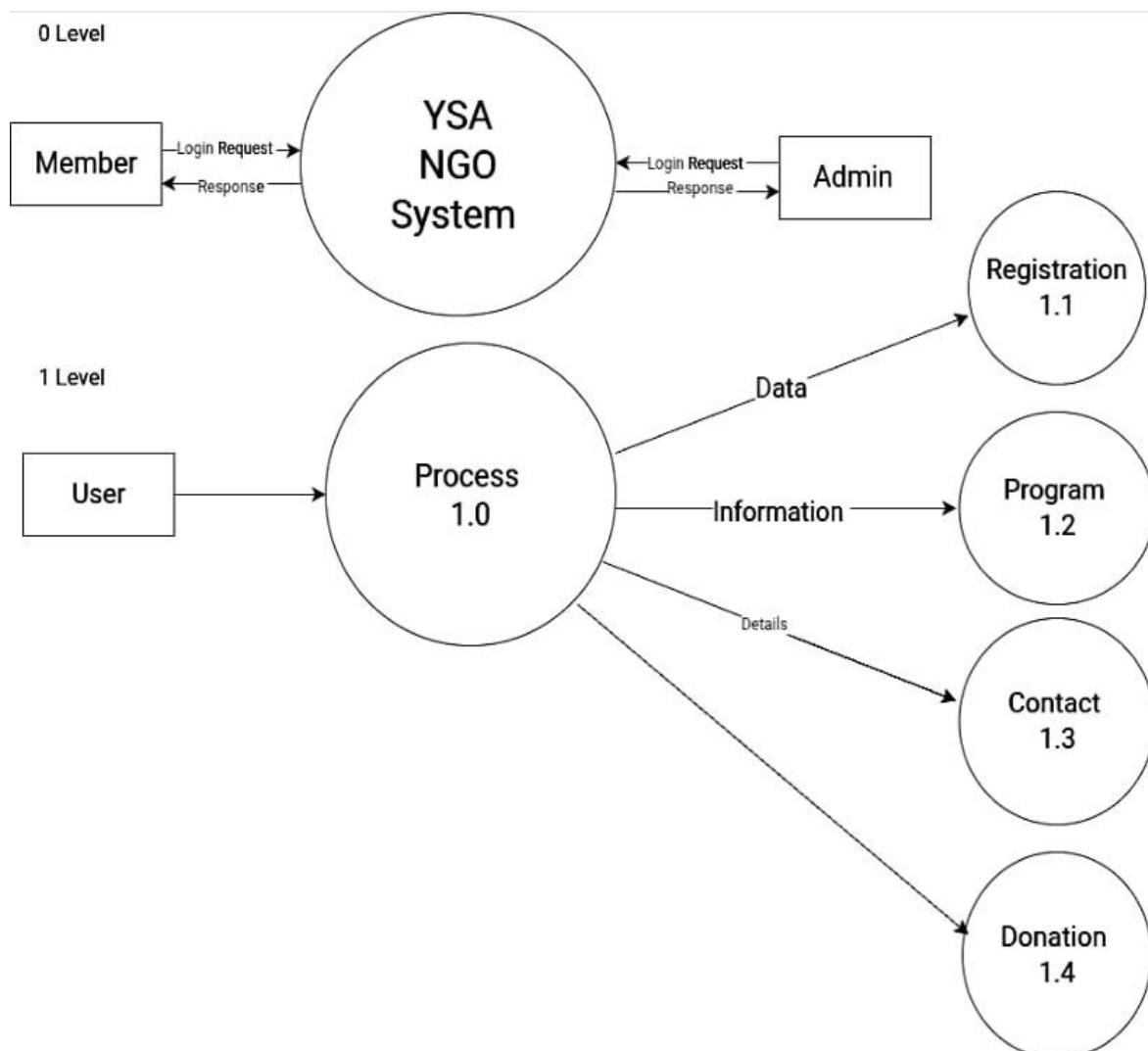


A square defines a source or destination of system data. External entities represent any entity that supplies or receive information from the system but is not a part of the system.

## **DATA STORE**

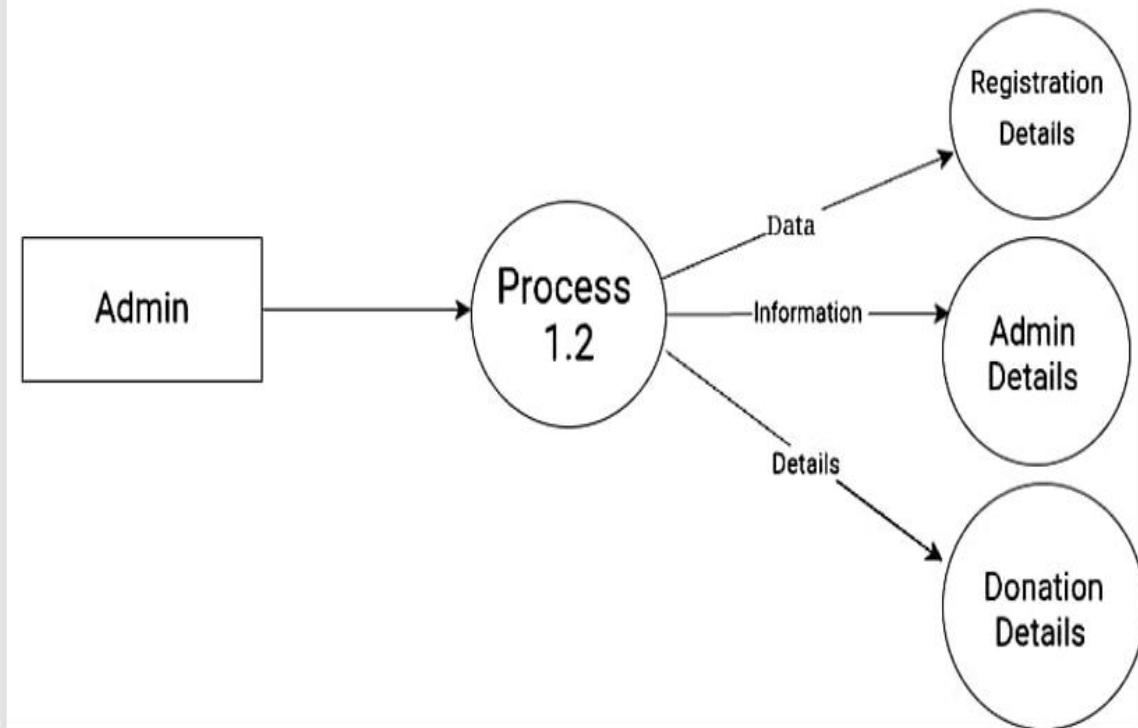


The data store represents a logical file. A logical file can represent either a data store symbol which can represent either a data structure or a physical file on disk. The data store is used to collect data at rest or a temporary repository of data. It is represented by open rectangle.

**DFD****USER**

**ADMIN**

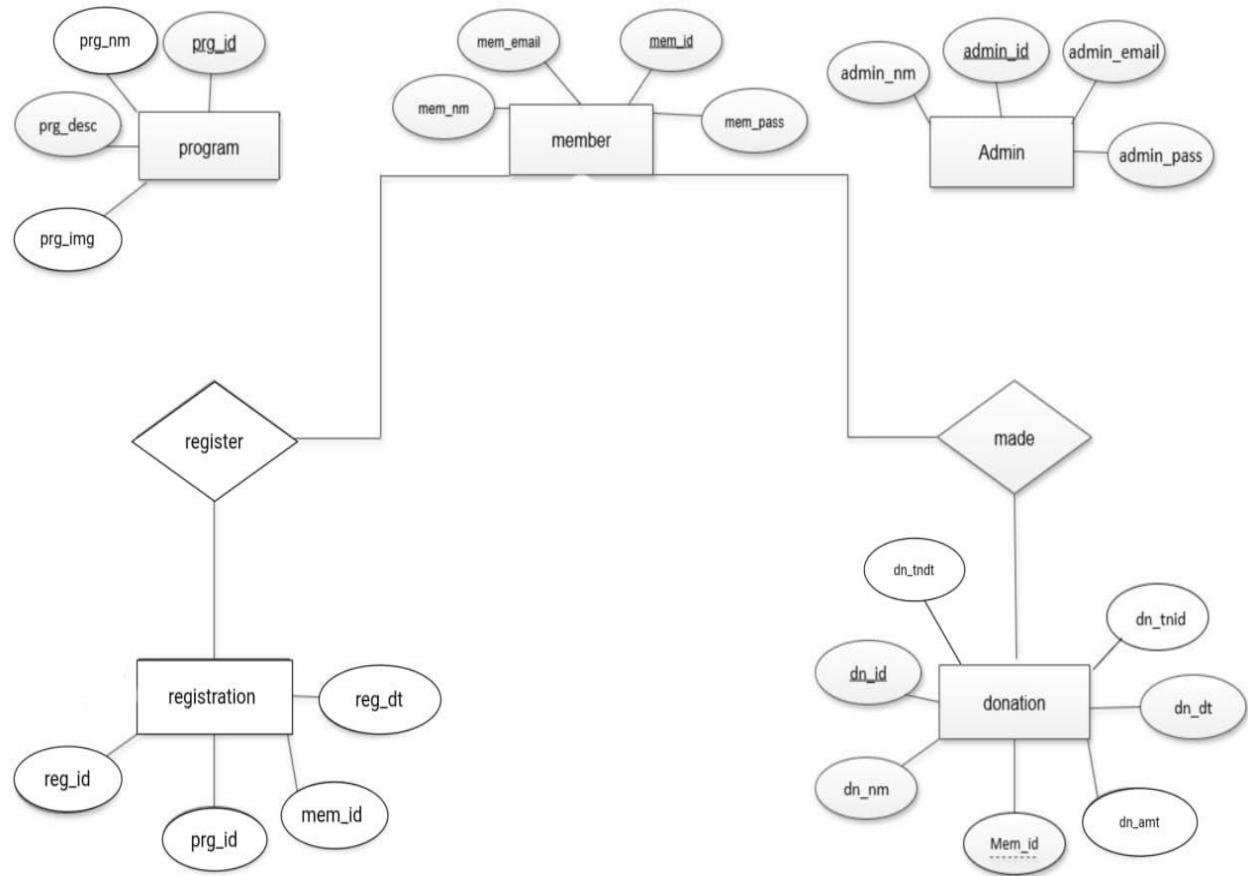
**1 Level Admin**



## ER DIAGRAM

### LIST OF FIGURES





## ACTIVITY DIAGRAM -

### LIST OF FIGURES



State



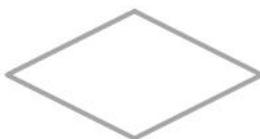
Initial state



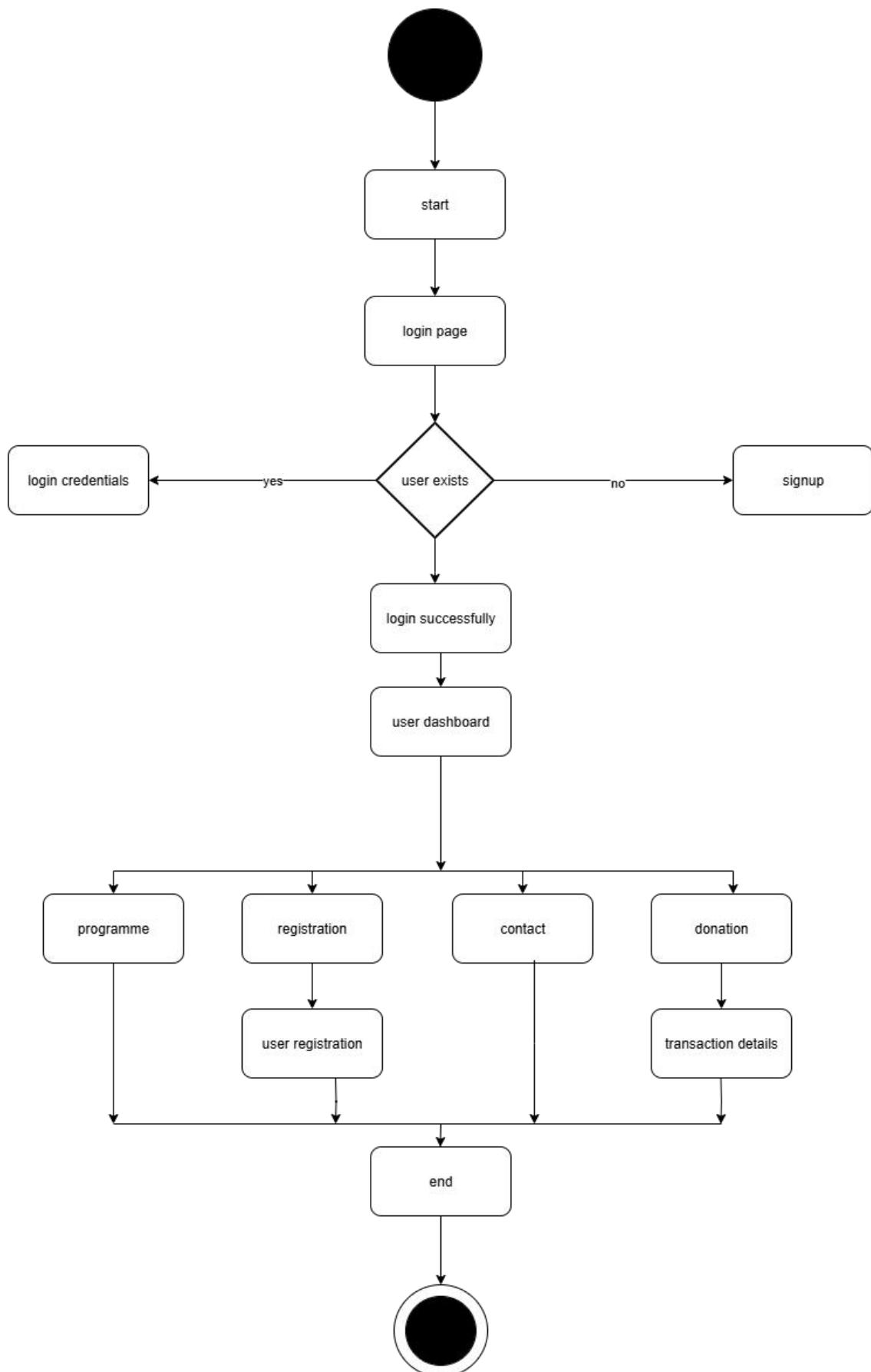
Final state



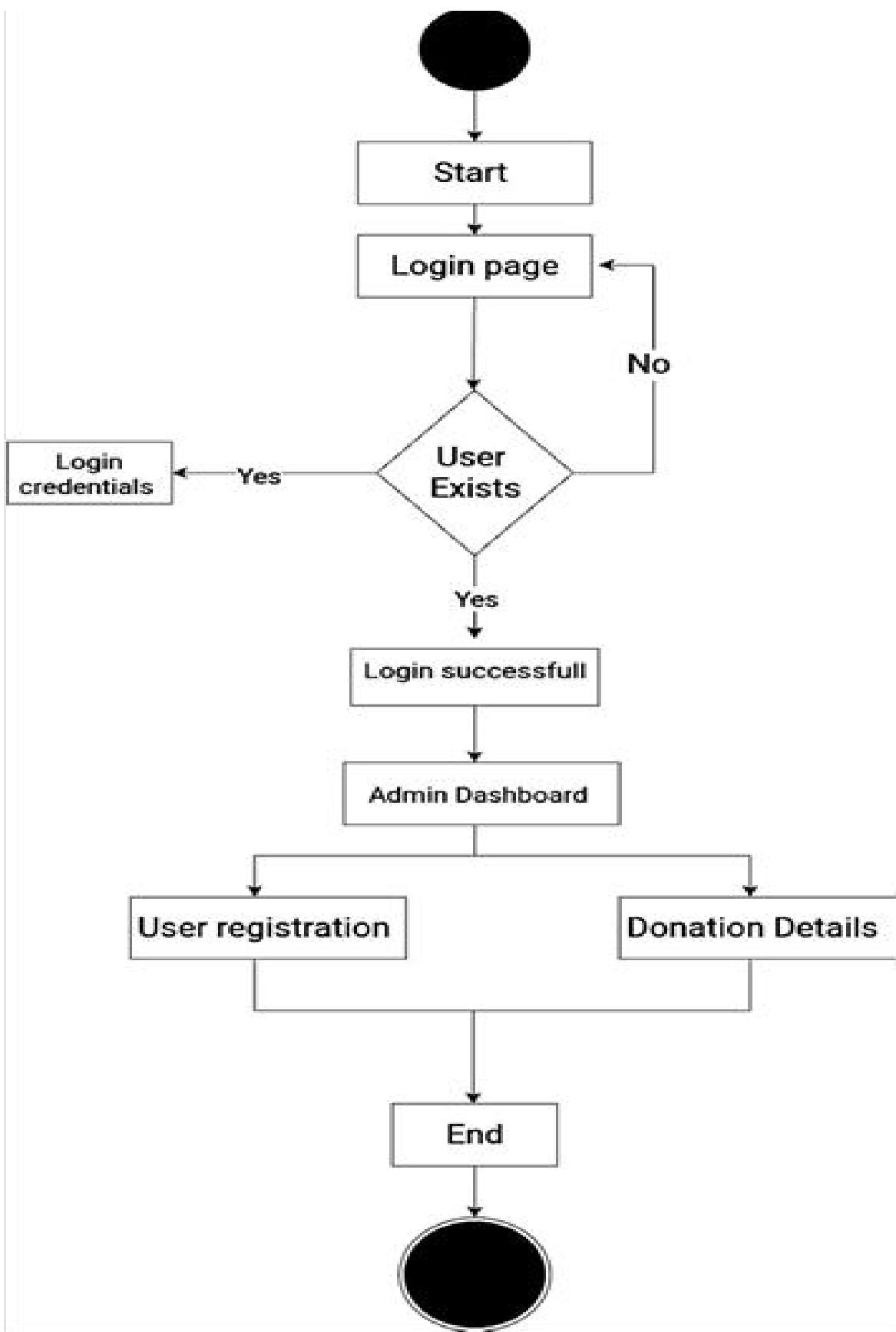
Control Flow



Decision

**ADMIN**

USER -&gt;



## USE CASE

### LIST OF FIGURES



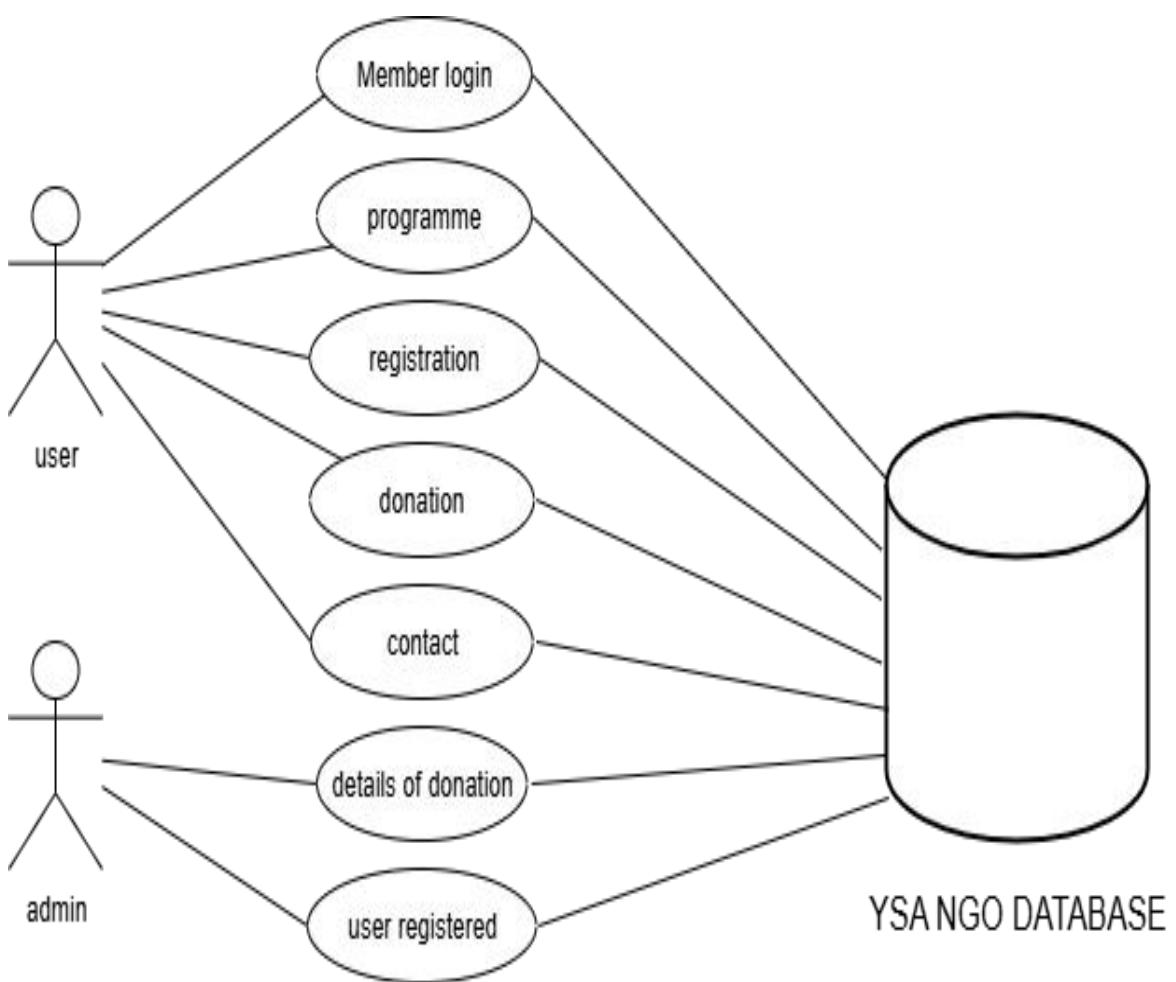
Actor



Use Case

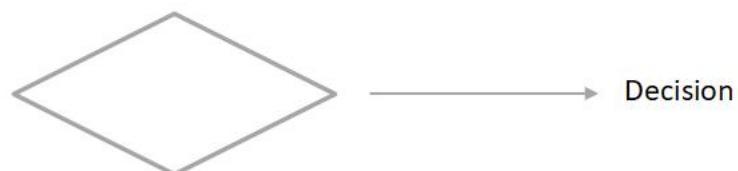


Connector



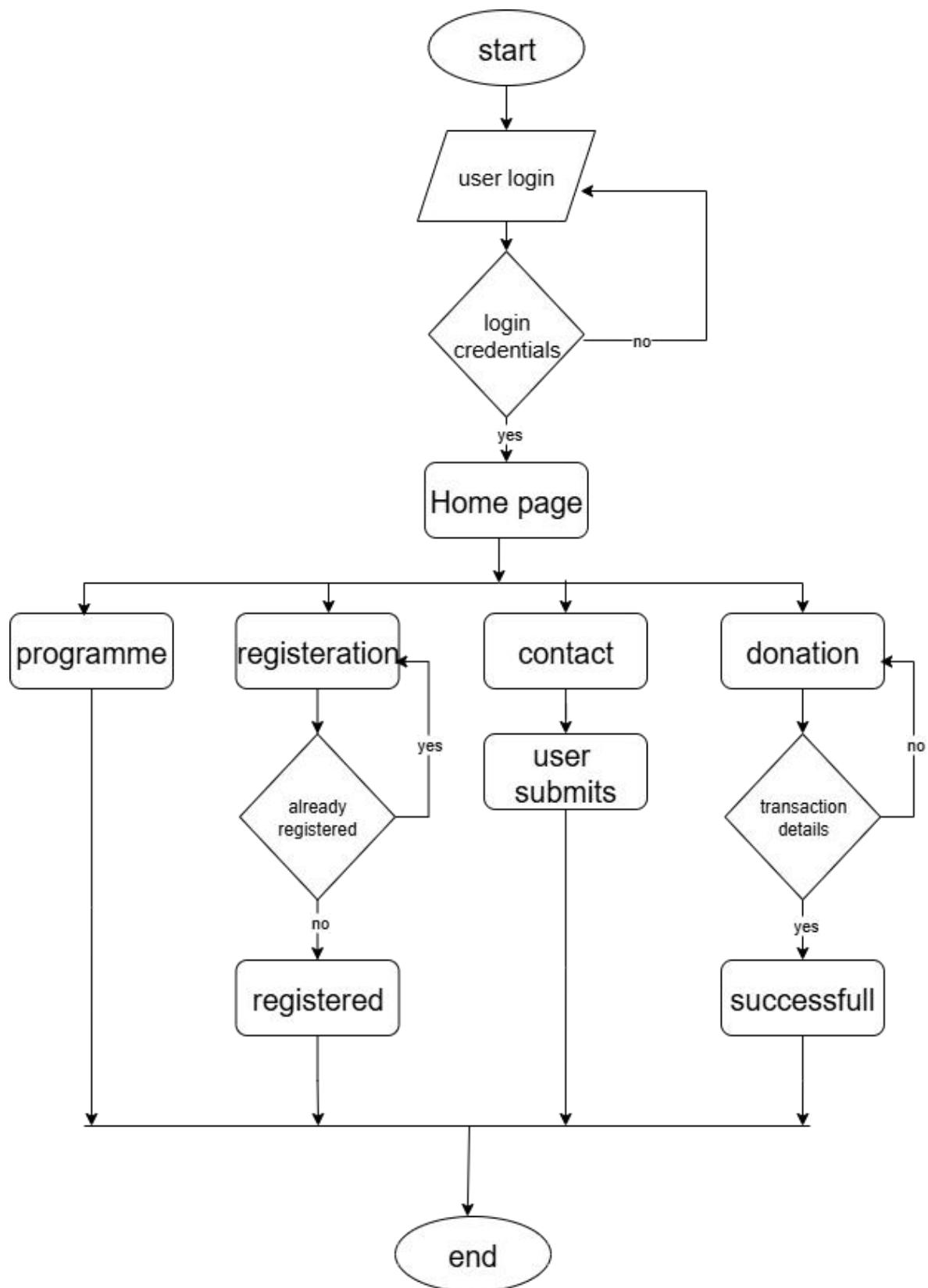
## **PROCESS FLOW ->**

### **LIST OF FIGURES**

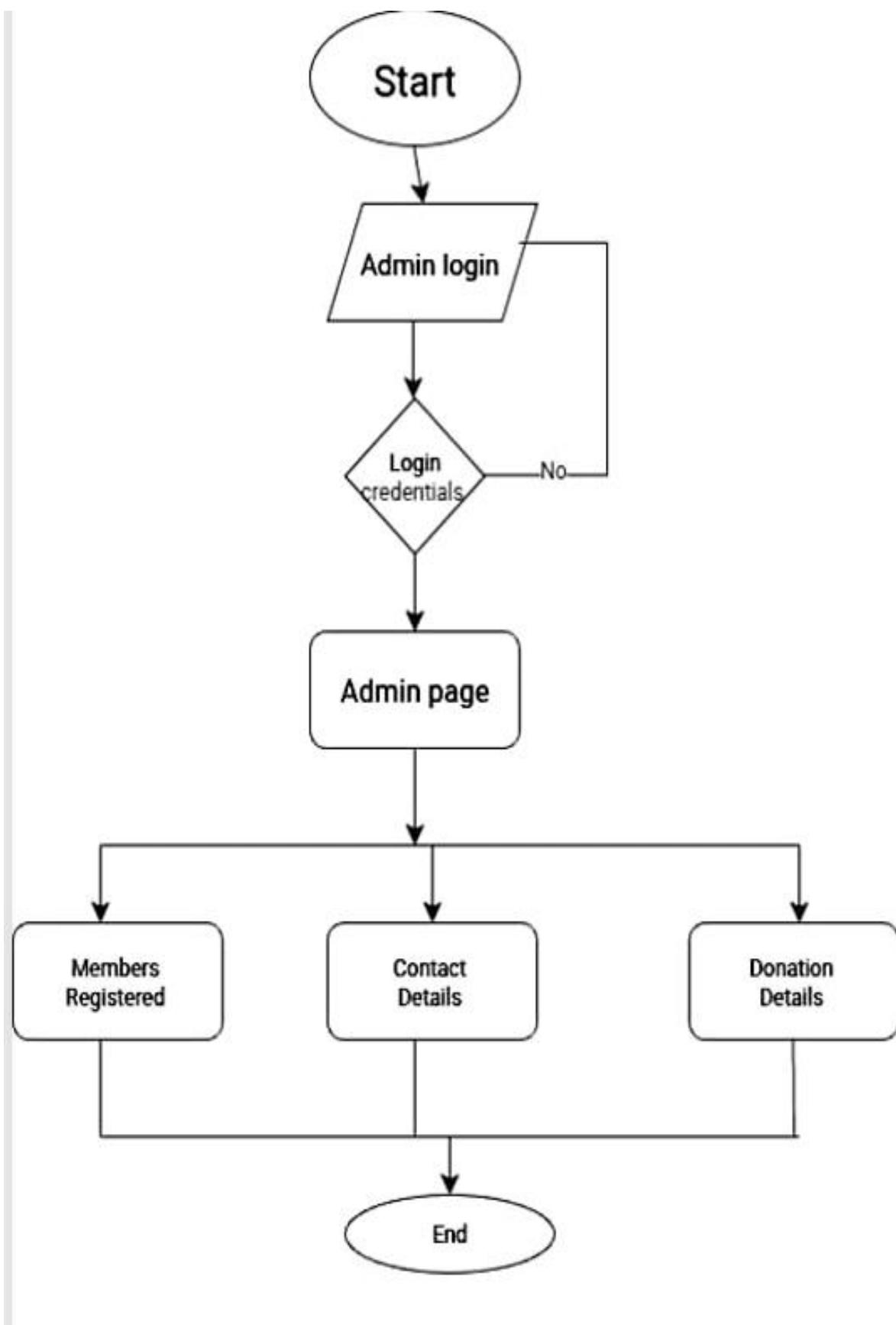


— — — — — → Connector

## USER



## PROCESS FLOW -&gt; ADMIN



## DATA DICTIONARY

### **CLIENT**

| Field Name   | Data Type | Size |
|--------------|-----------|------|
| Member_id    | int       |      |
| Member_nm    | varchar   | 100  |
| Member_pass  | varchar   | 100  |
| Member_email | varchar   | 100  |

### **ADMIN**

| Field Name  | Data Type | Size |
|-------------|-----------|------|
| Admin_id    | int       |      |
| Admin_nm    | varchar   | 100  |
| Admin_email | varchar   | 100  |
| Admin_pass  | varchar   | 255  |

### **PROGRAM**

| Field_name | Data type | length |
|------------|-----------|--------|
| Prg_id     | Int       |        |
| Prg_title  | Varchar   | 100    |
| Prg_desc   | Text      |        |
| Prg_img    |           |        |

**RAGISTRATION**

| Field_name | Data type | Length |
|------------|-----------|--------|
| Reg_id     | Int       |        |
| Mem_id     | Int       |        |
| Prg_id     | Int       |        |
| Reg_dt     | Date time |        |

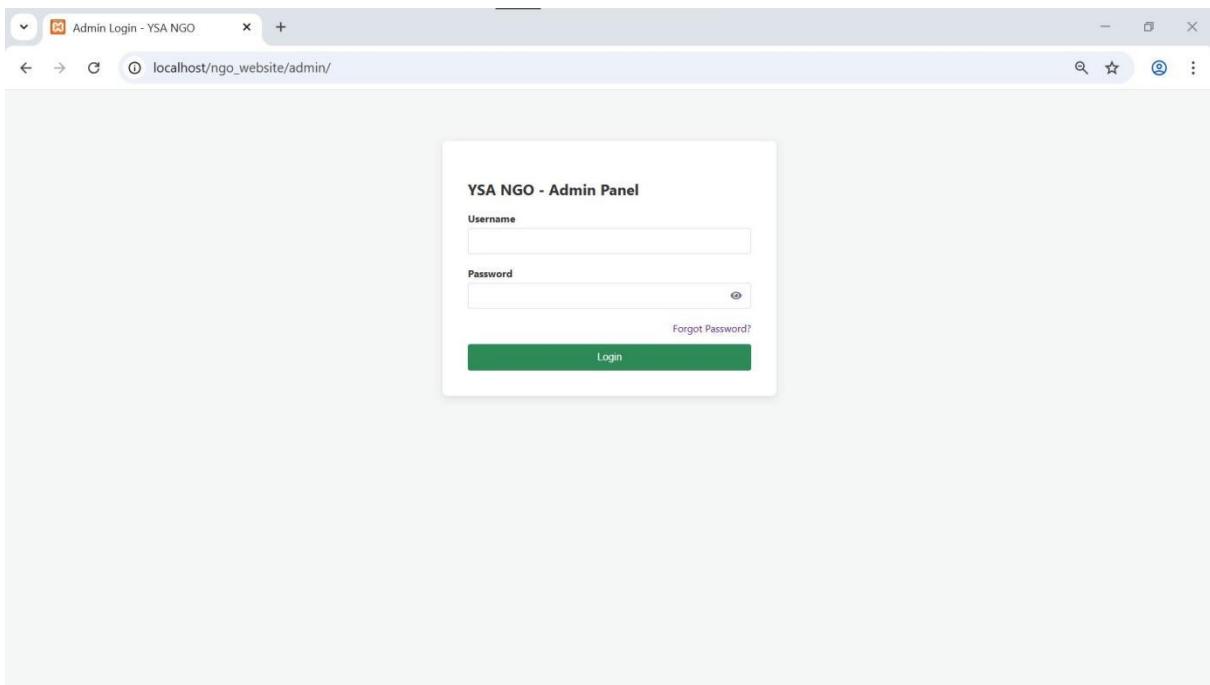
**DONATION**

| Field name | Data type | length |
|------------|-----------|--------|
| Dn_no      | Int       |        |
| Dn_txn_id  | Varchar   | 100    |
| Dn_id      | Varchar   | 100    |
| Dn_name    | Date time |        |
| Dn_amount  | Varchar   | 100    |
| Mem_tndt   | int       |        |

## **IMPLEMENTATION**

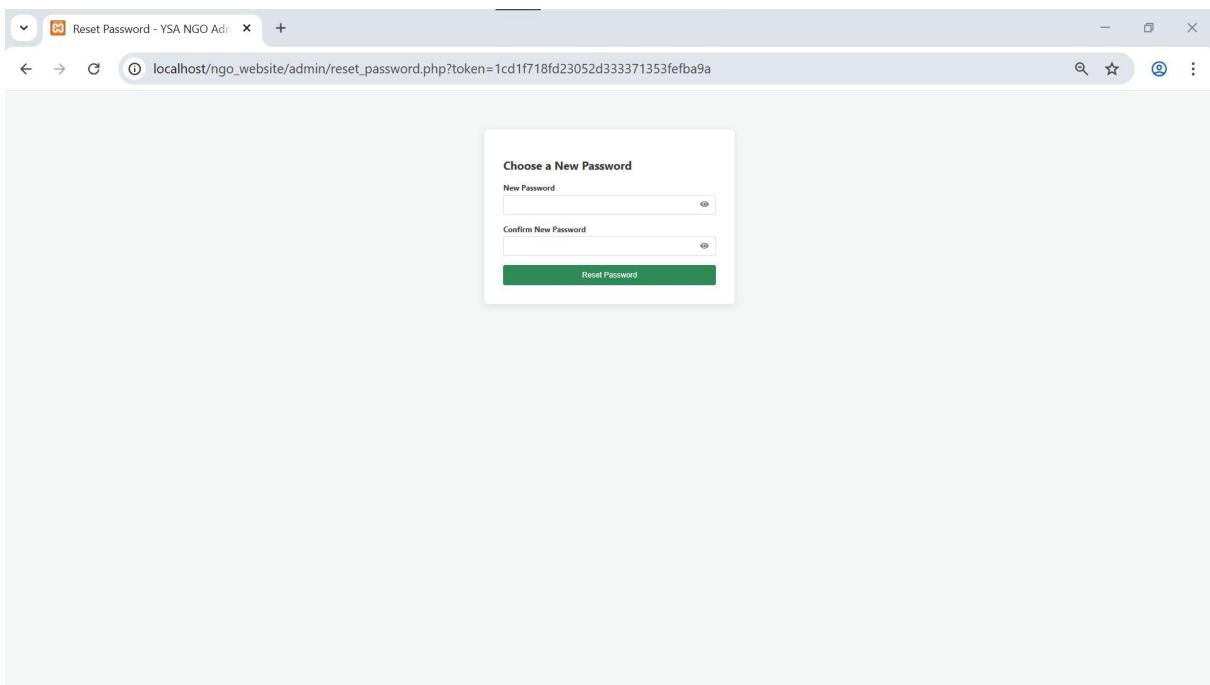
### **ADMIN:**

#### **ADMIN LOGIN**



A screenshot of a web browser window showing the 'Admin Login - YSA NGO' page. The URL in the address bar is 'localhost/ngo\_website/admin/'. The page title is 'YSA NGO - Admin Panel'. It features two input fields: 'Username' and 'Password', both with placeholder text. Below the password field is a 'Forgot Password?' link. A large green 'Login' button is at the bottom.

#### **FORGET PASSWORD**



A screenshot of a web browser window showing the 'Reset Password - YSA NGO Admin' page. The URL in the address bar is 'localhost/ngo\_website/admin/reset\_password.php?token=1cd1f718fd23052d333371353fefba9a'. The page title is 'Choose a New Password'. It has two input fields: 'New Password' and 'Confirm New Password', both with placeholder text. A green 'Reset Password' button is at the bottom.

## DASHBOARD

Welcome, admin!

**Dashboard**

- Total Members **7**
- Total Donations **₹133,311.00**
- Program Registrations **8**

Logout

## REPORTS

**Reports & Statistics**

All Time All Modules Generate Report Clear

Showing All Time Data

| New Members | Total Programs | Registrations | Donation Amount    |
|-------------|----------------|---------------|--------------------|
| <b>7</b>    | <b>6</b>       | <b>8</b>      | <b>₹133,311.00</b> |

**Registrations by Program**

| Program Name                | No. of Registrations |
|-----------------------------|----------------------|
| Environmental Cleanup Drive | 2                    |
| Skill Development Workshop  | 2                    |
| Education for All           | 2                    |
| blood donation              | 1                    |
| marathon                    | 1                    |
| Community Health            | 0                    |

**Top Donors**

| Donor Name | Total Donated |
|------------|---------------|
| dushyant   | ₹124,366.00   |
| mustafa    | ₹6,635.00     |
| mustad     | ₹1,000.00     |
| shiv       | ₹678.00       |
| om         | ₹340.00       |
| yash123    | ₹242.00       |
|            | ₹50.00        |

## MANAGE PROGRAM

The screenshot shows the 'Manage Programs' page of the YSA NGO Admin website. On the left, a sidebar menu lists various admin functions: Dashboard, Members, Registrations, Donations, Reports, Manage Programs, Add Admin, and Manage Admins. A red 'Logout' button is at the bottom. The main content area has a title 'Manage Programs'. It features a 'Add New Program' form with fields for Program Name, Program Description, and Program Image (with a 'Choose File' button). Below the form is a table titled 'Existing Programs' with columns for Program Name, Description, and Actions (Edit and Delete buttons). The table lists several programs:

| Program Name                | Description   | Actions                                     |
|-----------------------------|---|---|
| blood donation              | blood donation camp for the world blood bank..  | <a href="#">Edit</a> <a href="#">Delete</a> |
| Community Health            | organizing weekly health camps in the slums and rural areas ...   | <a href="#">Edit</a> <a href="#">Delete</a> |
| Education for All           | A drive to enroll children in schools and provide them with necessary learning materials...             | <a href="#">Edit</a> <a href="#">Delete</a> |
| Environmental Cleanup Drive | Organizing weekly cleanup activities in local parks and rivers to promote a cleaner environment...      | <a href="#">Edit</a> <a href="#">Delete</a> |
| marathon                    | marathon for collection of money for blood cancer...  | <a href="#">Edit</a> <a href="#">Delete</a> |
| Skill Development Workshop  | Conducting workshops to teach vocational skills like tailoring, computer basics, and carpentry to em... | <a href="#">Edit</a> <a href="#">Delete</a> |

## ADD NEW ADMIN

The screenshot shows the 'Add New Admin' page of the YSA NGO Admin website. The sidebar menu is identical to the previous page. The main content area has a title 'Add New Admin'. It features a form with fields for New Admin Username, New Admin Email, Password, and Confirm Password. A green 'Create Admin' button is at the bottom. The form fields have placeholder text and password strength indicators.

## MANAGE ADMINS

The screenshot shows a web browser window titled "Manage Admins - YSA NGO". The URL is "localhost/ngo\_website/admin/manage\_admins.php". On the left, there is a dark sidebar with the title "YSA NGO Admin" and a list of navigation items: Dashboard, Members, Registrations, Donations, Reports, Members, Manage Programs, Add Admin, Manage Admins, and Logout. The main content area is titled "Manage Administrators" and displays a table with the heading "Existing Administrators". The table has columns for Admin ID, Username, Email, and Actions. It contains two rows of data:

| Admin ID | Username | Email            | Actions                |
|----------|----------|------------------|------------------------|
| 1        | admin    | admin@ysango.org | No actions available   |
| 3        | admin1   | admin1@gmail.com | <a href="#">Delete</a> |

## USER:

### USER LOGIN

The screenshot shows a web browser window titled "Sign Up - YSA NGO". The URL is "localhost/ngo\_website/signup.php". At the top, there is a navigation bar with links for Home, About Us, Programs, Donate, and Login. Below the navigation bar, the main content area is titled "Create an Account" with the sub-instruction "Join our community to make a difference". There is a form with the following fields:

- Full Name: (input field)
- Email ID: (input field)
- Member ID (Optional): (input field)
- Password (min. 8 characters): (input field)
- Confirm Password: (input field)

At the bottom of the form is a green "Sign Up" button. Below the form, there is a link "Already have an account? [Log In](#)". At the very bottom of the page, there is a footer section with small text and links.

## MEMBER LOGIN

### Member Login

Welcome back! Please enter your credentials to access your account.

Member ID:

Password:

 (

**Login**

[Forgot Password?](#)

New to our NGO? [Create an account.](#)

## FORGOT PASS

Reset Password - YSA NGO Admin

localhost/ngo\_website/admin/reset\_password.php?token=1cd1f718fd23052d333371353fefba9a

Choose a New Password

New Password

 (

Confirm New Password

 (

**Reset Password**

## DASHBOARD

Welcome - YSA NGO

localhost/ngo\_website/homepage.php

YSA NGO

Home About Us Programs Donate Login

**Bringing Smiles to Every Community**

Join us in our mission to create lasting positive change through education, healthcare, and community development.

[Learn More](#)

### Our Mission

To empower underprivileged communities by providing access to quality education, healthcare, and sustainable development opportunities.

## ABOUT US

About Us - YSA NGO

localhost/ngo\_website/about.php

YSA NGO

Home About Us Programs Donate Login

**Our Mission**

To empower underprivileged communities by providing access to quality education, essential healthcare, and sustainable development opportunities that foster self-reliance and growth.

**Our Vision**

We envision a world where every individual has the opportunity to achieve their fullest potential, contributing to a just, equitable, and thriving society for all.

**Our Core Values**

- Community**: We work hand-in-hand with local communities, ensuring our programs are collaborative and meet real needs.
- Sustainability**: We focus on long-term solutions that empower individuals and create lasting, positive change.
- Integrity**: We operate with transparency and accountability, ensuring every resource is used effectively for our mission.

**Meet Our Team**

| John Doe      | Jane Smith        | Peter Jones    |
|---------------|-------------------|----------------|
|               |                   |                |
| Founder & CEO | Programs Director | Lead Developer |

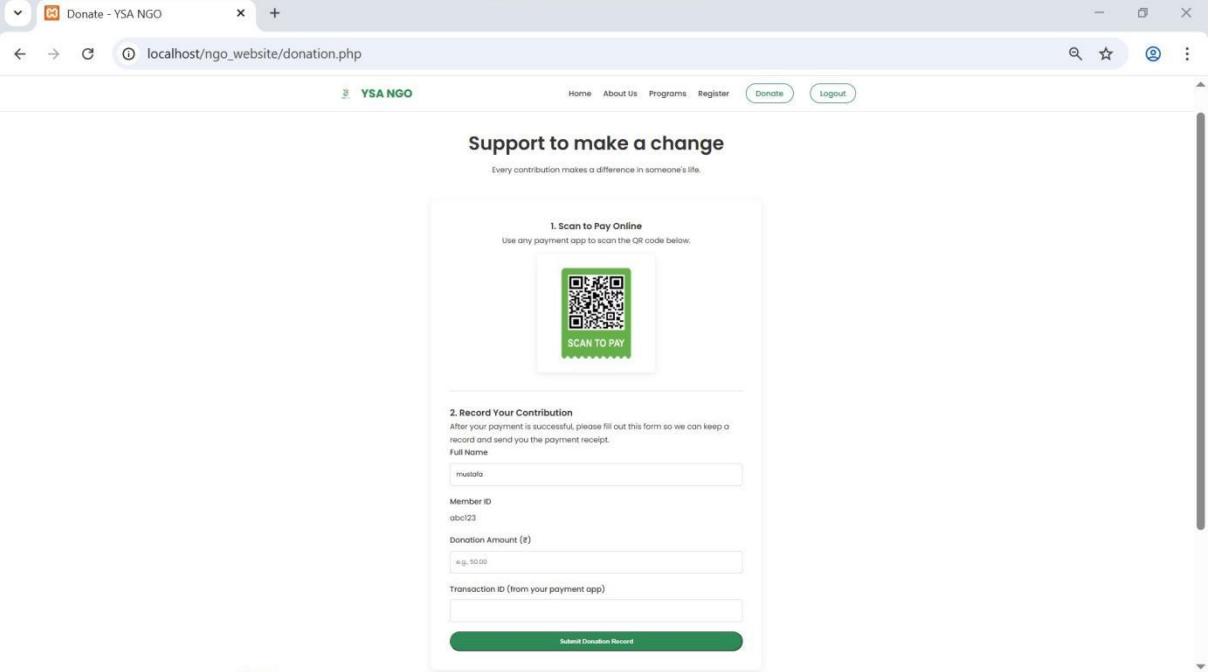
## PROGRAM

The screenshot shows a web browser window titled "Our Programs - YSA NGO". The URL in the address bar is "localhost/ngo\_website/programs.php". The page header includes the YSA NGO logo and navigation links for Home, About Us, Programs, Register, Donate, and Logout. Below the header, a section titled "Our Impactful Programs" features a sub-section titled "Creating sustainable change through focused initiatives.". It displays two program cards: "Education for All" (a group photo of children) and "Environmental Cleanup Drive" (a photo of people picking up trash). Both cards include a "Get involved" button.

## REGISTRATION

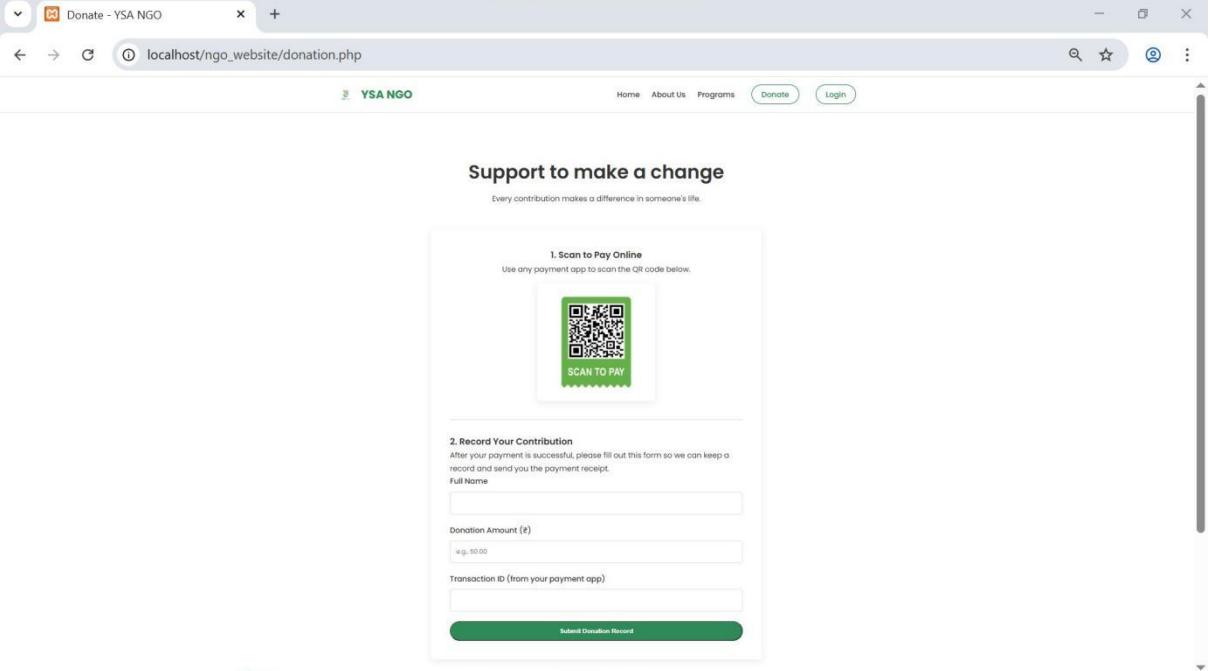
The screenshot shows a web browser window titled "Program Registration - YSA NGO". The URL in the address bar is "localhost/ngo\_website/registration.php". The page header includes the YSA NGO logo and navigation links for Home, About Us, Programs, Register, Donate, and Logout. Below the header, a section titled "Register for a Program" encourages users to "Join one of our initiatives and be a part of the change.". A registration form is displayed, asking for "Your Member ID:" (with "abcd123" entered), "Choose a Program:" (a dropdown menu showing "-- Select a Program --"), and a large green "Register Now" button. At the bottom of the page, a "Get In Touch" section provides address, phone number, and email information, along with a copyright notice: "© 2025 YSA NGO. All Rights Reserved."

## DONATION



The screenshot shows a web browser window titled "Donate - YSA NGO" with the URL "localhost/ngo\_website/donation.php". The page header includes the YSA NGO logo and navigation links for Home, About Us, Programs, Register, Donate, and Logout. The main content area is titled "Support to make a change" with the sub-instruction "Every contribution makes a difference in someone's life." Below this, there are two sections: "1. Scan to Pay Online" (with a QR code and "SCAN TO PAY" button) and "2. Record Your Contribution" (with fields for Full Name, Member ID, Donation Amount, and Transaction ID, followed by a "Submit Donation Record" button).

## DONATION WITHOUT LOGIN



This screenshot is identical to the one above, showing the "Scan to Pay Online" section. However, the header navigation bar does not contain the "Logout" link, indicating that the user is not logged in.

# CODE EXPLANATION

## 1. User Registration (Sign Up)

```
// Insert new member into the database
$sql = "INSERT INTO members (mem_id, mem_nm, mem_email, mem_pass)
VALUES ('$mem_id', '$mem_nm', '$mem_email', '$hashed_pass')";
if (mysqli_query($conn, $sql)) {
    header("Location: login.php?success=Account created successfully!
Please login.");
    exit();
}
```

## 2. Donation Without Login (POST Handler)

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $donor_name = trim($_POST['donor_name']);
    $amount = $_POST['amount'];
    $dn_txn_id = trim($_POST['dn_txn_id']);
}
```

## 3. Donation With Login (Prepare Statement)

```
// insertion
$stmt = $conn->prepare("INSERT INTO donations (donor_name, mem_id,
amount, dn_txn_id) VALUES (?, ?, ?, ?)");
$stmt->bind_param("ssds", $donor_name, $mem_id, $amount, $dn_txn_id);
```

## 4. User Login Authentication

```
$sql = "SELECT * FROM members WHERE mem_id = '$mem_id'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) == 1) {
    $row = mysqli_fetch_assoc($result);
    // Verify the password
    if (password_verify($mem_pass, $row['mem_pass'])) {
        // Password is correct, set session variables
        $_SESSION['mem_id'] = $row['mem_id'];
        $_SESSION['mem_nm'] = $row['mem_nm'];
        // Redirect to homepage
        header("Location: homepage.php");
        exit();
    } else {
        // Incorrect password
        header("Location: login.php?error=Invalid Member ID or
        
```

```

        Password.");
        exit();
    }
}

```

## 5. Registration Validation (Prevent Duplicate Registration)

```

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // mem_id is taken from session for security
    $mem_id = $_SESSION['mem_id'];
    $prog_id = mysqli_real_escape_string($conn, $_POST['prog_id']);

    // Check if user is already registered for this program
    $check_sql = "SELECT * FROM registrations WHERE mem_id =
    '$mem_id' AND prog_id = '$prog_id'";
    $check_result = mysqli_query($conn, $check_sql);
    if (mysqli_num_rows($check_result) > 0) {
        header("Location: registration.php?error=You are already
registered for this program.");
        exit();
    }
}

```

## 6. Register User to Program (Database Insert)

```

$sql = "INSERT INTO registrations (mem_id, prog_id) VALUES
    ('$mem_id', '$prog_id')";
if (mysqli_query($conn, $sql)) {
    header("Location: registration.php?success=You have been
successfully registered for the program!");
    exit();
} else {
    header("Location: registration.php?error=An error occurred.
Please try again.");
    exit();
}

```

## 7. Add Admin (Admin Registration)

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username']);
    $email = trim($_POST['email']);
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
}

```

```

    // Insert the new admin into the database
    $stmt = $conn->prepare("INSERT INTO admins (username, email,
password) VALUES (?, ?, ?)");
    $stmt->bind_param("sss", $username, $email, $hashed_password);

    if ($stmt->execute()) {
        header('Location: add_admin.php?success>New admin created
successfully.');
    } else {
        header('Location: add_admin.php?error=An error occurred.
Please try again.');
    }
}

```

## 8. Admin Password Recovery - Message Display

```

if (isset($_GET['error'])) {
    echo '<p class="error-msg">' . htmlspecialchars($_GET['error']) .
'</p>';
}
if (isset($_GET['success'])) {
    echo '<p class="success-msg">' . htmlspecialchars($_GET['success']) . '</p>';
}
?>
<form action="handle_forgot.php" method="post">
    <!-- form fields -->
</form>

```

## 9. Delete Admin (Admin Management)

```

$stmt = $conn->prepare("DELETE FROM admins WHERE id = ?");
$stmt->bind_param("i", $id_to_delete);
if ($stmt->execute()) {
    header('Location: manage_admins.php?success=Admin deleted
successfully.');
} else {
    header('Location: manage_admins.php?error=Error deleting
admin.');
}
$stmt->close();
exit;

```

**10. Reports Generation - Registration and Donations**

```

$regs_per_program_sql = "SELECT p.prog_name, COUNT(r.reg_id) as
registration_count
    FROM programs p
    LEFT JOIN registrations r ON p.prog_id = r.prog_id" .
    ("$start_date_sql ? " WHERE r.reg_date BETWEEN ? AND ?" : "") .
    " GROUP BY p.prog_id ORDER BY registration_count DESC";
$regs_per_program_stmt = $conn->prepare($regs_per_program_sql);
if ($start_date_sql) {
    $regs_per_program_stmt->bind_param("ss", $start_date_sql,
$end_date_sql);
}
$regs_per_program_stmt->execute();
$regs_per_program_result = $regs_per_program_stmt->get_result();

$top_donors_sql = "SELECT donor_name, SUM(amount) as total_donated
FROM donations" .
    ("$start_date_sql ? " WHERE dn_tndt BETWEEN ? AND ?" : "") .
    " GROUP BY donor_name ORDER BY total_donated DESC LIMIT 10";
$top_donors_stmt = $conn->prepare($top_donors_sql);
if ($start_date_sql) {
    $top_donors_stmt->bind_param("ss", $start_date_sql,
$end_date_sql);
}
$top_donors_stmt->execute();
$top_donors_result = $top_donors_stmt->get_result();
?>

```

## TESTING

### TEST CASE -> Test Case 1

**Test Case ID:** TC\_001

**Test Case Name:** User login with valid credentials

**Module:** User Authentication

**Preconditions:** User must be registered in the system.

**Test Steps:**

- Open the application.
- Go to the login page.
- Enter valid username: abc123.
- Enter valid password: 1234.
- Click the "Login" button.

**Expected Result:** User should be logged in and redirected to the dashboard.

**Actual Result:** (To be filled after execution)

**Status:** Pass/Fail

### Test Case 2

**Test Case ID:** TC\_002

**Test Case Name:** User login with invalid credentials

**Module:** User Authentication

**Preconditions:** User must be registered in the system.

**Test Steps:**

- Open the application.
- Go to the login page.
- Enter valid username: abc123.
- Enter invalid password: 1234.

- Click the "Login" button.

**Expected Result:** Login should fail, and an error message “Invalid username or password” should be displayed.

**Actual Result:** (To be filled after execution)

**Status:** Pass/Fail

## Test Case 3

**Test Case ID:** TC\_003

**Test Case Name:** Register with valid user details

**Module:** Registration

**Preconditions:** User should not already exist in the system.

**Test Steps:**

- Open the application.
- Go to the registration page.
- Enter details: Name, unique username, valid email, password.
- Click the "Register" button.

**Expected Result:** Registration should succeed, and the user should be notified of successful registration or redirected to login.

**Actual Result:** (To be filled after execution)

**Status:** Pass/Fail

## Test Case 4

**Test Case ID:** TC\_004

**Test Case Name:** Prevent duplicate user registration

**Module:** Registration

**Preconditions:** Username/email already exists in the system.

**Test Steps:**

- Open the application.
- Go to the registration page.
- Enter already registered username/email.
- Click the "Register" button.

**Expected Result:** Registration should fail, displaying “Username or email already exists.”

**Actual Result:** (To be filled after execution)

**Status:** Pass/Fail

## **Test Case 5**

**Test Case ID:** TC\_005

**Test Case Name:** Admin login with valid and invalid credentials

**Module:** Admin Authentication

**Preconditions:** Admin must be registered in the system.

**Test Steps:**

- Open the application.
- Go to the admin login page.
- Enter valid admin username and password, then click "Login."
- Log out.
- Enter admin username and invalid password, then click "Login."

**Expected Result:**

- With valid credentials, admin is logged in and taken to the admin dashboard.
- With invalid credentials, login fails and an error message appears.

**Actual Result:** (To be filled after execution)

**Status:** Pass/Fail

## RESULT

### Result Test Case -> 01

**Test Case ID:** TC\_001

**Test Case Name:** Login with valid credentials

**Module:** User Authentication

**Preconditions:** User must be registered in the system.

**Test Steps:**

- Open the application.
- Go to the login page.
- Enter valid username: testuser.
- Enter valid password: Test@123.
- Click on the "Login" button.

**Expected Result:** User should be logged in and redirected to the dashboard.

**Actual Result:** User successfully logged in and dashboard displayed.

**Status:** Pass ✓ .

### Result Test Case -> 02

**Test Case ID:** TC\_002

**Test Case Name:** Login with invalid credentials

**Module:** User Authentication

**Preconditions:** User must be registered in the system.

**Test Steps:**

- Open the application.
- Go to the login page.
- Enter valid username: testuser.
- Enter invalid password: WrongPass1.
- Click on the "Login" button.

**Expected Result:** Login should fail, and an error message “Invalid username or password” should be displayed.

**Actual Result:** Error message displayed. User not logged in.

**Status:** Pass ✓

**Result Test Case -> 03****Test Case ID:** TC\_003**Test Case Name:** Register with valid user details**Module:** Registration**Preconditions:** User should not already exist in the system.**Test Steps:**

- Open the application.
- Go to the registration page.
- Enter details: Name, unique username, valid email, password.
- Click the "Register" button.

**Expected Result:** Registration should succeed, and the user should be notified of successful registration or redirected to login.**Actual Result:** Registration successful. User prompted to log in.**Status:** Pass ✓**Result Test Case -> 04****Test Case ID:** TC\_004**Test Case Name:** Prevent duplicate user registration**Module:** Registration**Preconditions:** Username/email already exists in the system.**Test Steps:**

- Open the application.
- Go to the registration page.
- Enter already registered username/email.
- Click the "Register" button.

**Expected Result:** Registration should fail, displaying "Username or email already exists."**Actual Result:** Error message displayed. Registration not allowed.**Status:** Pass ✓

**Result Test Case -> 05****Test Case ID:** TC\_005**Test Case Name:** Admin login with valid and invalid credentials**Module:** Admin Authentication**Preconditions:** Admin must be registered in the system.**Test Steps:**

- Open the application.
- Go to the admin login page.
- Enter valid admin username and password, then click "Login."
- Log out.
- Enter admin username and invalid password, then click "Login."

**Expected Result:**

- With valid credentials, admin is logged in and taken to the admin dashboard.
- With invalid credentials, login fails and an error message appears.

**Actual Result:**

- With valid credentials, admin successfully logged in; dashboard displayed.
- With invalid credentials, error message shown; admin not logged in.

**Status:** Pass ✓

## **CONCLUSION AND FUTURE SCOPE**

### **Conclusion**

This project successfully developed a functional, data-driven web portal for a non-governmental organization (YSA NGO). It provides a complete ecosystem for both public-facing interaction and internal administrative management. The project effectively demonstrates a full-stack development cycle using core web technologies.

### **Key Accomplishments:**

- Comprehensive Member Portal: A secure frontend allows members to sign up, log in, reset their passwords, view dynamic program information, and register for programs.
- Donation Recording System: A donation module was implemented, allowing users to make payments via a QR code and subsequently record their transaction details, which are then stored in the database.
- Powerful Admin Panel: A secure, role-aware administrative backend provides full control over the website's core content and data. Key features include: Dashboard with at-a-glance statistics. Full CRUD (Create, Read, Update, Delete) functionality for managing NGO programs, including image uploads from the administrator's device. Management of administrator accounts, with restricted permissions for critical actions like deleting users. Modules to view member lists, program registrations, and donation records. Dynamic Reporting Engine: An advanced reports module was created, allowing administrators to generate precise, filterable reports based on specific date ranges (e.g., "This Month," custom dates) and by module (Members, Registrations, Donations).

### **Limitations**

While the project is functionally complete, it has several limitations inherent to its scope and development environment that would need to be addressed before a live, production deployment.

- **Payment Gateway Integration:**

The current donation system is not a real-time payment gateway. It relies on a user paying externally via a QR code and then manually entering their transaction details. This system is dependent on user honesty and can be prone to errors. A production-level site would require integration with a proper payment gateway API (like Stripe, PayPal, or Razorpay) for seamless and verified transactions.

- **Email Functionality:**

The password reset feature simulates sending an email by displaying the link on the screen. For the system to work in a live environment, the PHP server must be configured with SMTP settings to connect to a real email service (like SendGrid, Mailgun, or a Gmail account) to dispatch actual emails.

- **Security:**

While basic security is in place, the project has not undergone a formal security audit. It lacks advanced protections against vulnerabilities like Cross-Site Scripting (XSS) in all inputs and Cross-Site Request Forgery (CSRF) on form submissions.

The role management is minimal (only the admin with id=1 has delete privileges). A larger-scale application would require a more granular role-based access control (RBAC) system.

- **Development and Deployment Environment:**

The project is built and tested on a XAMPP local server. Migrating to a live hosting environment would require additional steps, including database migration, file transfer, and configuration of the db\_connect.php file with live credentials.

The code is written in raw PHP without a framework. While effective for learning, a production application would benefit from a PHP framework (like Laravel or Symfony) for better structure, maintainability, and built-in security features.