NGO SCHOLARSHIP MANAGEMENT SYSTEM

CLI-Based Application using Java, JDBC & MySQL

1. INTRODUCTION

The **NGO Scholarship Management System** is a command-line based application developed to automate the management of educational aid and donations. The system facilitates the coordination between students seeking financial help, members collecting donations, and administrators overseeing the approval process. By replacing manual record-keeping, it ensures transparency in fund allocation and faster processing of student applications.

2. OBJECTIVES

- To automate the scholarship application and approval workflow.
- To maintain transparent records of donations collected by various members.
- To implement secure CRUD (Create, Read, Update, Delete) operations for user profiles and applications.
- To manage data persistence using a centralized MySQL database.
- To provide a role-based system for Students, Members, and Admins.

3. SCOPE OF THE PROJECT

The system covers user registration (with a secure PIN for members), scholarship application submission by students, donation record management by members, and a centralized administrative panel for user approval and message monitoring. It is designed for small to medium-sized NGOs and can be extended to web or mobile platforms in the future.

4. TECHNOLOGIES USED

- **Language:** Java (Core Java 8 or higher)
- **Database:** MySQL (Relational Database Management System)
- **Connectivity:** JDBC (Java Database Connectivity)
- **Architecture:** Layered DAO (Data Access Object) Pattern
- **Input Handling:** Scanner-based CLI with custom `InputUtil` for validation
- **Build Tool:** Compatible with Maven or standard Java builds

5. SYSTEM ARCHITECTURE

The system follows a layered architecture to ensure modularity:

- **Presentation Layer:** Command Line Interface (CLI) handled by `Main`, `AuthService`, and individual Service classes.
- **Business Logic Layer:** Service classes that process user inputs and coordinate with DAOs.
- **Data Access Layer (DAO):** Dedicated classes (`UserDAO`, `ApplicationDAO`, `DonationDAO`, `AdminDAO`, `ContactDAO`) that execute SQL queries.
- **Database Layer:** MySQL database storing tables for users, students, members, applications, and donations.

6. DATABASE DESIGN

The database utilizes relational integrity with several interconnected tables:

- **users:** Stores credentials and `is_active` status for all roles.
- **students:** Stores specific academic and contact details of student users.
- **members:** Stores designation and profile info for NGO volunteers.
- **applications:** Tracks student requests, subjects, descriptions, and statuses (PENDING/APPROVED/REJECTED).
- **donations:** Records donor names, amounts, payment modes, and the member who collected it.
- **contacts:** Stores messages sent from users to the administrator.

7. MODULE DESCRIPTION

- **Authentication Module:** Handles login and registration. Includes a transaction-based registration process that ensures data is saved in both `users` and role-specific tables simultaneously.
- **Student Module:** Allows students to apply for help, check their application status, and message the admin.
- **Member Module:** Enables members to record new donations, view their collection history, and review pending student applications for status updates.
- **Admin Module:** Provides a dashboard to approve new user accounts (specifically for members) and read messages in the inbox.

8. CRUD OPERATIONS & JDBC

The system implements robust CRUD operations using `PreparedStatement` to prevent SQL injection:

- **Create:** Registering users, submitting applications, and adding donation records.
- **Read:** Fetching pending approvals, viewing application status, and donor history.
- **Update:** Admin approving a user (setting `is_active = 1`) or a member updating an application status.
- **Delete:** While the current version focuses on archiving, the architecture supports record removal if required.

9. FEATURES

- **Role-Based Dashboards:** Unique menus for Students, Members, and Admins.
- **Transactional Integrity:** Uses `con.setAutoCommit(false)` during registration to prevent partial data entry.
- **Secure Member Registration:** Requires a specific NGO PIN ("MWT2026") to register as a member.
- **Input Validation:** Custom `InputUtil` to handle `NumberFormatException` and prevent application crashes during data entry.
- **Admin Approval Workflow:** New member accounts remain "Locked" until a Super Admin approves them.

10. ADVANTAGES

- **Transparency:** Clear tracking of who collected which donation.
- **Efficiency:** Faster processing of student aid requests compared to paper forms.
- **Data Integrity:** Relational database prevents orphaned records.
- **Scalability:** The DAO pattern makes it easy to add new features or migrate to a GUI.

11. LIMITATIONS & FUTURE ENHANCEMENTS

- **Current Limitations:** Command-line interface only, lacks automated email notifications, and requires manual admin approval for all members.
- **Future Scope:** Integration of a Graphical User Interface (GUI), password encryption (hashing), online payment gateway integration for donations, and automated report generation for tax purposes.

12. CONCLUSION

The **NGO Scholarship Management System** effectively demonstrates the use of Java and MySQL to solve real-world organizational challenges. By utilizing JDBC and a modular design, the system provides a reliable platform for managing educational charity operations securely and efficiently.