Project Proposal

Guided By:

Trainer Name: **Anuj Kumar**

Created By:

Student Name: Arbaj Ali

AFid: **AF04991332**

Student Name: Kartik

AFid: **AF04991231**

Table of Content

<u>Title of the Project:</u> - NGO Scholarship Management

<u>Introduction</u>

Non-Governmental Organizations (NGOs) play an important role in supporting underprivileged and deserving students by providing financial assistance in the form of scholarships. Many talented students are unable to continue their education due to financial constraints, even though they are academically eligible. To address this problem, NGOs collect funds from donors and utilize them to provide scholarships to eligible students.

The **Scholarship Management System** is developed to support an NGO that collects donations from various donors and efficiently distributes these funds as scholarships to deserving students. The system helps in managing donor information, student details, scholarship applications, and fund allocation in a systematic and transparent manner.

This digital system reduces manual work, ensures proper record maintenance, improves transparency, and helps the NGO in selecting eligible students and managing scholarships effectively.

Objectives

- To design and develop a computerized management system for an NGO that provides scholarships to eligible students.
- To collect, store, and manage donation details received from donors in a secure and organized manner.
- To maintain accurate records of students, members, and users with role-based access control.
- To enable students to apply for scholarships and track the status of their applications.
- To ensure transparent and efficient allocation of donated funds to deserving and eligible students.
- To reduce manual paperwork, minimize errors, and improve overall efficiency of scholarship management.

Poject Category

- Database Management/Application Development
- Frontend: Core Java (Console/Command Line)
- Backend: MySQL Database
- Connectivity: JDBC (Java Database Connectivity).

Analysis

(a):-Modules and Description

1. Authentication Module

This module is responsible for user login and security. It verifies user credentials such as username and password and provides access based on user roles like admin, student, or member. This module prevents unauthorized access to the system.

2. Admin Module

The admin module manages the overall functioning of the system. Admin can view registered users, manage student applications, monitor donation records, and handle contact queries. This module acts as the central controlling unit of the application.

3. Student Module

This module allows students to register and submit applications. Students can view their application status and manage their personal details. It helps in maintaining organized student information within the system.

4. Member Module

The member module is designed for donors or members who contribute to donations. Members can add donation details and view their donation history. This module ensures proper recording of donor information.

5. Donation Management Module

This module manages all donation-related records. It stores donation amounts, donor details, and donation dates. The module helps maintain transparency and accuracy in donation tracking.

6. Application Management Module

This module handles student applications submitted for approval. Admin can review, approve, or reject applications, while students can track the current status of their applications.

7. Contact Management Module

This module stores messages or queries submitted by users. Admin can view and respond to these queries, helping maintain proper communication between users and the system.

8. Database Management Module

This module manages database connectivity using JDBC. It ensures secure communication between the Java application and the MySQL database and maintains data consistency.

(b):- <u>Database Design</u>

1. APPLICATION Table Design

Field Name	Datatype	Properties
app_id	int	primary key, auto increment
student_id	int	foreign key → student(student_id)
subject	varchar(100)	not null
description	text	nullable
status	varchar(20)	default 'PENDING'
applied_date	date	default current date

2. CONTACT Table Design

Field Name	Datatype	Properties
contact_id	int	primary key, auto increment
user_id	int	foreign key → users(user_id)
name	varchar(100)	nullable
phone	varchar(15)	nullable
subject	varchar(100)	nullable
message	text	nullable
created_at	timestamp	default current timestamp

3. DONATION Table Design

Field Name	Datatype	Properties
donation_id	int	primary key, auto increment
donor_name	varchar(100)	not null
amount	decimal(10,2)	not null
donation_date	date	not null
payment_mode	varchar(50)	nullable
collected_by	int	foreign key → users(user_id)

4. MEMBER Table Design

Field Name	Datatype	Properties
member_id	int	primary key
full_name	varchar(100)	nullable
designation	varchar(50)	nullable
phone	varchar(15)	nullable

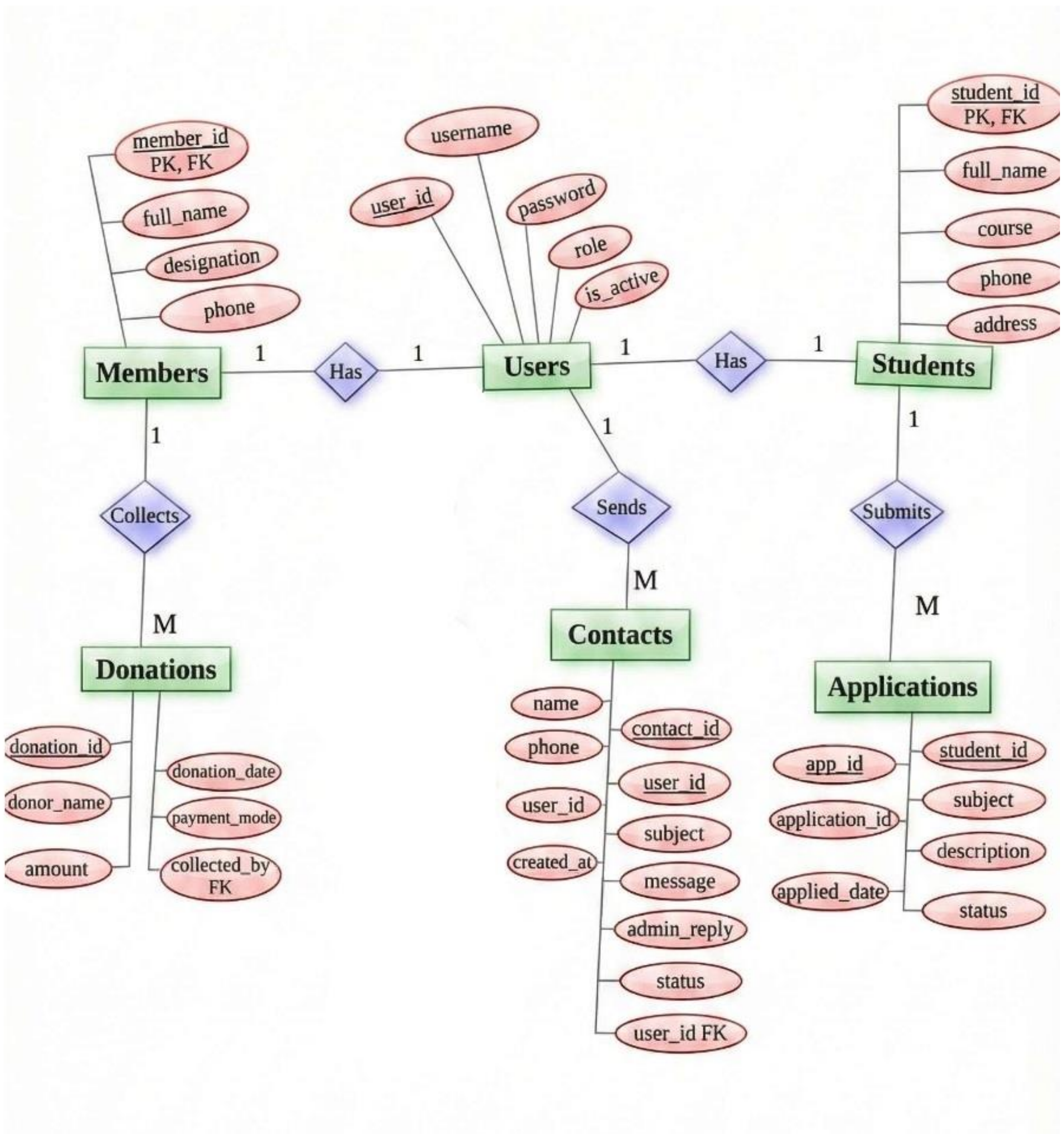
5. STUDENT Table Design

Field Name	Datatype	Properties
student_id	int	primary key
full_name	varchar(100)	nullable
course	varchar(50)	nullable
phone	varchar(15)	nullable
address	varchar(255)	nullable

6. USER Table Design

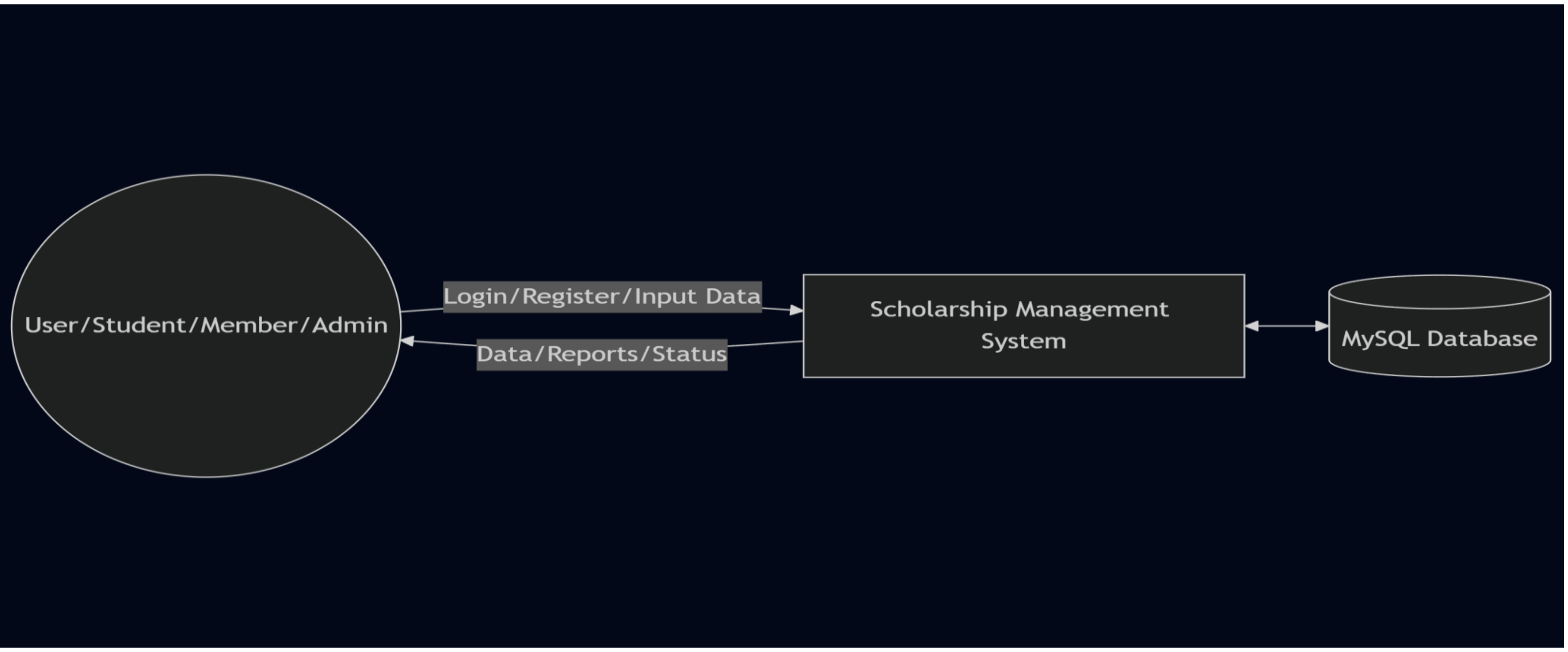
Field Name	Datatype	Properties
user_id	int	primary key, auto increment
username	varchar(50)	unique, not null
password	varchar(50)	not null
role	varchar(20)	not null
is_active	tinyint(1)	default 1

(c):-Entity Relationship diagram

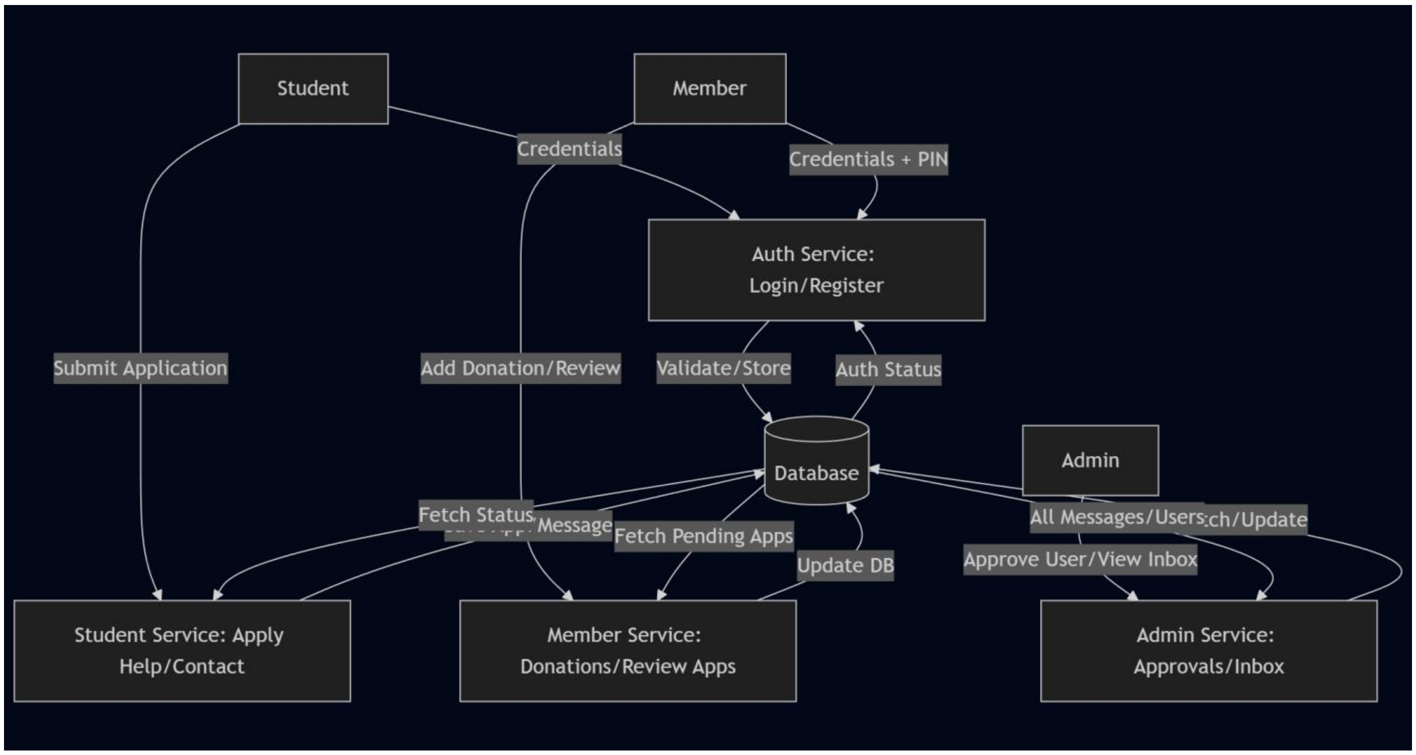


(d):- Data Flow Diagram

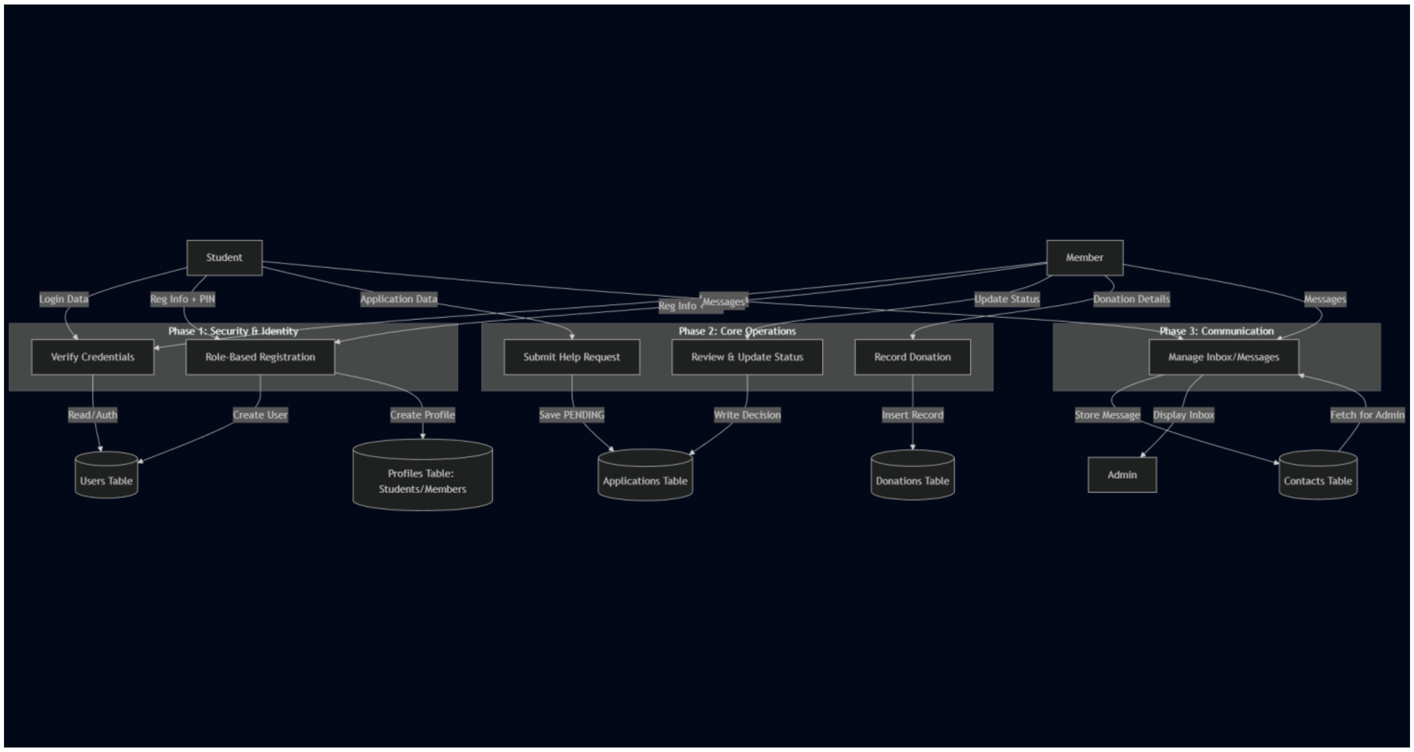
(i):- 0 Level DFD


User/Student/Member/Admin
Login/Register/Input Data
Data/Reports/Status
Scholarship Management System
MySQL Database

(ii):- 1 Level DFD

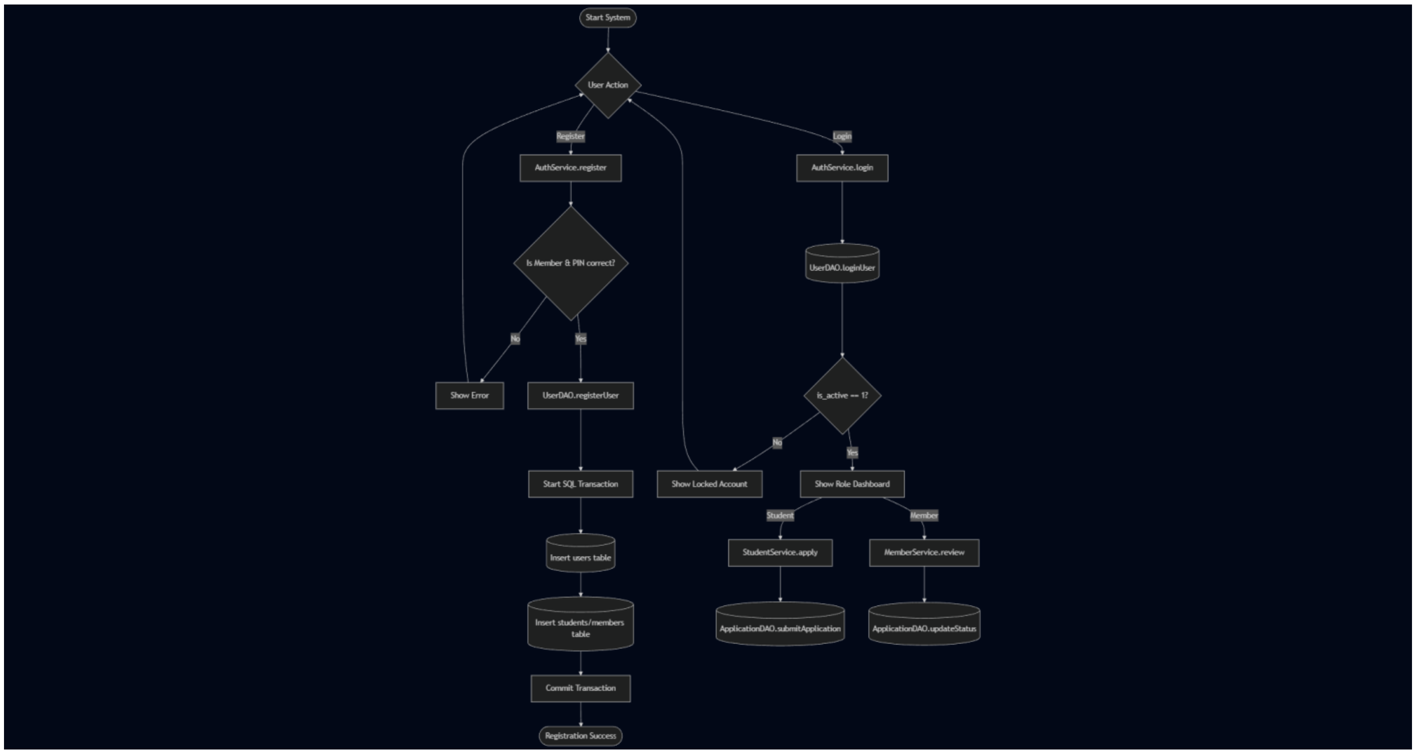


(iii):- 3 Level DFD



6:- Complete Structure

(a):- Process Logic Diagram



<u>(7):-Platform Used</u>

This project is a **Java Command-Line Interface (CLI)** application designed for a tiered user management system (Admin, Member, Student). The technical stack ensures that the system is lightweight yet robust for database operations.

1. Hardware Requirements

- **Processor:** Any standard PC with 1.0 GHz or higher (e.g., Intel i3/i5 or AMD Ryzen) is sufficient for running the Java Virtual Machine and MySQL server.
- **Memory (RAM):** A minimum of 1 GB to 2 GB of RAM is required, as CLI applications have low memory overhead compared to GUI-based systems.
- **Storage:** 500 MB of free disk space is adequate to store the project files, Java binaries, and the MySQL database records.

2. Software Requirements

- **Operating System:** Cross-platform compatibility; the system can run on **Windows, Linux, or macOS** provided by the Java Runtime is installed.
- **Java Development Kit (JDK):** JDK 8 or a later version is required to support the Object-Oriented features and the try-with-resources blocks used for database connection management.
- **Database Management System: MySQL Community Server** is used to store persistent data such as user credentials, donation records, and scholarship applications.
- **JDBC Connector:** The **MySQL Connector/J** library is necessary to enable communication between the Java DAO (Data Access Object) layer and the MySQL database.
- **IDE/Text Editor:** Development can be performed using **Eclipse, IntelliJ IDEA, or VS Code**, which provide debugging and build tools for Java projects.

(8):-Future Scope

The current implementation of the Scholarship Management System provides a foundational architecture using the DAO design pattern. The following enhancements are planned for future iterations:

1. Graphical User Interface (GUI) Integration

The existing Command-Line Interface (CLI) can be replaced with a modern **JavaFX or Swing-based GUI**. This will improve user experience by providing buttons, forms, and tables instead of manual text input.

2. Enhanced Security Features

- **Password Hashing:** Currently, passwords are stored in plain text. Future versions will implement **Crypt or SHA-256 hashing** to ensure user data security even in the event of a database breach.
- **Two-Factor Authentication (2FA):** Implementing OTP-based logins for Admin and Member accounts will add an extra layer of protection for sensitive financial records.

3. Advanced Scholarship Logic

- **Automated Eligibility Engine:** A module can be developed to automatically match student profiles (GPA, income, course) with available scholarship criteria to suggest the most relevant aid.
- **Document Uploading:** Integration with cloud storage (like AWS S3) or local BLOB storage to allow students to upload identity proofs and marksheets directly.

4. Integrated Communication Tools

- **Real-time Notifications:** While the system currently allows sending messages to the admin, future updates will include automated **Email or**

SMS notifications to alert students when their application status is updated (Approved/Rejected).

5. Financial & Reporting Modules

- **Online Payment Gateway:** Future versions can integrate payment of APIs (like Razor pay or PayPal) to allow Members or Donors to contribute funds directly through the application instead of manual record entry.
- **Automated Report Generation:** Generating PDF receipts and monthly donation reports using libraries like **intext or Jasper Reports** for auditing purposes.

(9):-Bibliography

- **Bloch, J.** (2018). *Effective Java* (3rd ed.). Addison-Wesley Professional. (Reference for best practices in Java programming and Object-Oriented Design used in the Model and DAO layers).
- **Oracle Corporation.** (2023). *Java SE Documentation*. [Online]. Available: https://docs.oracle.com/javase/. (Technical reference for `java.sql` package, `PreparedStatement`, and `ResultSet` interfaces).
- **MySQL AB.** (2024). *MySQL 8.0 Reference Manual*. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/. (Documentation for SQL syntax, data types, and the MySQL Connector/J driver used in `DBConnection`).
- **Horstmann, C. S.** (2022). *Core Java Volume I – Fundamentals* (12th ed.). Pearson. (Reference for the Scanner class, Exception handling, and CLI input/output management).

- **Sierra, K., & Bates, B.** (2022). *Head First Java* (3rd ed.). O'Reilly Media. (Reference for the implementation of the Data Access Object (DAO) design pattern and JDBC connectivity).
- **Gamma, E., Helm, R., Johnson, R., & Vlissides, J.** (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. (Theoretical background for the Transaction management logic implemented in the `registerUser` method).