Team Avengers – Sathwik, Santosh, Sourabh, Arundhati

Project Setup Instructions:

1.Download the Application:

  Obtain the application source files from the provided repository or download link.

2. Add JavaFX Library:
a) Download the JavaFX jar from https://gluonhq.com/products/javafx/
b) Navigate to Project -> Properties -> Java Build Path -> Module Path
c) Add Library -> User Library -> JavaFX Jar Files
d) Go to Project -> src -> application -> Main.java
e) Click on Run As -> Run Configurations -> Arguments
f) Add the following to VM arguments: --add-modules javafx.controls,javafx.fxml

3. Add MySQL Connector Jar:
a) Go to Project -> Properties -> Java Build Path -> Class Path b) Add MySQL Jar (8.2.0) to the classpath

Note: Ensure that a **MySQL server** is running with the configured credentials before starting the application.

Additonal Note: Used **MySQL Workbench** for Manipulating of data.

4. Update Database Credentials:
a) Navigate to Project -> src -> code -> LibraryDB.java b) Update the database credentials in the file

5. MySQL Database Setup:

CREATE DATABASE librarysql;

USE librarysql;

CREATE TABLE person (
   id INT PRIMARY KEY,
   name VARCHAR(255),
   surname VARCHAR(255),
   password VARCHAR(255),
   isAdmin INT,
   hasFine INT,

```sql
    BookReceived TEXT,
    BookReadBefore TEXT,
    deadline TEXT
);

CREATE TABLE book (
    name VARCHAR(255),
    author VARCHAR(255),
    pageNumber INT,
    ISBN INT PRIMARY KEY,
    quantity INT
);

SHOW TABLES;


INSERT INTO librarysql.book (name, author, pageNumber, ISBN, quantity) VALUES
('The Great Gatsby', 'F. Scott Fitzgerald', 300, 1001, 5),
('To Kill a Mockingbird', 'Harper Lee', 250, 1002, 3),
('1984', 'George Orwell', 150, 1003, 2),
('Pride and Prejudice', 'Jane Austen', 500, 1004, 4),
('The Catcher in the Rye', 'J.D. Salinger', 200, 1005, 6),
('One Hundred Years of Solitude', 'Gabriel García Márquez', 350, 1006, 1),
('The Hobbit', 'J.R.R. Tolkien', 280, 1007, 5),
('The Da Vinci Code', 'Dan Brown', 400, 1008, 7),
('The Hitchhiker\'s Guide to the Galaxy', 'Douglas Adams', 320, 1009, 2),
('The Great Expectations', 'Charles Dickens', 450, 1010, 3);


INSERT INTO librarysql.person (id, name, surname, password, isAdmin, hasFine, BookReceived,
BookReadBefore, deadline)
VALUES
(1, 'Sathwik', 'Reddy', 'Avengers1234', 1, 0, NULL, NULL, NULL),
(2, 'Arundhati', 'Pathrikar', 'Avengers1234', 1, 0, NULL, NULL, NULL),
(3, 'Santosh', 'Ratnakar', 'Avengers1234', 1, 0, NULL, NULL, NULL),
(4, 'Sourabh', 'Kumar', 'Avengers1234', 1, 0, NULL, NULL, NULL),
(5, 'Daniel', 'Brown', '1234', 1, 0, NULL, NULL, NULL),
(6, 'Alice', 'Clark', '1234', 0, 1, '1001,1002', '1003', '2024-12-01'),
(7, 'Robert', 'Miller', '1234', 0, 0, '1004,1005', NULL, '2024-12-15'),
(8, 'Sophia', 'Anderson', '1234', 0, 1, '1006', '1007,1008', '2024-11-30'),
(9, 'William', 'Taylor', '1234', 0, 0, NULL, '1009', NULL),
(10, 'Olivia', 'Thomas', '1234', 0, 1, '1010', NULL, '2024-12-20');
```

SELECT * FROM librarysql.book;
SELECT * FROM librarysql.person;

7. Build and Test the Application

- Compile and run the project in your IDE.
- Test the application functionalities

8. Optional Enhancements

- Use MySQL Workbench or any GUI client to visualize and manipulate the data during testing.
- Implement exception handling for database connectivity issues.

9. Run and Enjoy

Start the application, verify its functionalities, and ensure the database interactions are working smoothly!