```
package librarysystem;


import java.sql.*;
import java.util.ArrayList;
public class JDBC_Connection {


    private static final String URL = "jdbc:mysql://127.0.0.1:3306/library_db";
    private static final String USER = "root";
    private static final String PASSWORD = ""; // Replace "password" with your actual MySQL
password



    private Connection connection;


    public JDBC_Connection() {
        try {
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


    public Connection getConnection() {
        return connection;
    }


    // Login Function
    public boolean login(String username, String password) {
        String query = "SELECT * FROM users WHERE username = ? AND password = ?";
        try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, password);

            ResultSet resultSet = preparedStatement.executeQuery();

            return resultSet.next();

        } catch (SQLException e) {

            e.printStackTrace();

        }

        return false;

    }

      //check user name

    public boolean isUsernameTaken(String username) {

    String query = "SELECT * FROM users WHERE username = ?";

    try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

        preparedStatement.setString(1, username);

        ResultSet resultSet = preparedStatement.executeQuery();

        return resultSet.next(); // Returns true if username exists

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return false;

}


    // Signup Function

    public boolean signup(String name, String username, String password, String securityQuestion,
String answer) {

    String query = "INSERT INTO users (name, username, password, security_question, answer)
VALUES (?, ?, ?, ?, ?)";

    try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {

        preparedStatement.setString(1, name);

        preparedStatement.setString(2, username);

        preparedStatement.setString(3, password);

        preparedStatement.setString(4, securityQuestion);

        preparedStatement.setString(5, answer);
        int rowsInserted = preparedStatement.executeUpdate();
        System.out.println("Signup successful. Rows inserted: " + rowsInserted); // Debugging
        return rowsInserted > 0;
    } catch (SQLException e) {
        System.out.println("Signup failed due to SQL error.");
        e.printStackTrace();
    }
    return false;
}


    // Add Book Function
    public boolean addBook(int bookId, String category, String name, String author, int copies) {
        String query = "INSERT INTO books (book_id, category, name, author, copies) VALUES (?, ?, ?, ?, 
?)";
        try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
            preparedStatement.setInt(1, bookId);
            preparedStatement.setString(2, category);
            preparedStatement.setString(3, name);
            preparedStatement.setString(4, author);
            preparedStatement.setInt(5, copies);
            int rowsInserted = preparedStatement.executeUpdate();
            return rowsInserted > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    // Remove Book Function
    public boolean removeBookByIdOrName(int bookId, String bookName) {
        String query = "DELETE FROM books WHERE book_id = ? OR name = ?";
        try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
            preparedStatement.setInt(1, bookId);
            preparedStatement.setString(2, bookName);
            int rowsDeleted = preparedStatement.executeUpdate();
            return rowsDeleted > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    // Function to update an existing book's details
public boolean updateBook(int bookId, String category, String name, String author, int copies) {
    String query = "UPDATE books SET category = ?, name = ?, author = ?, copies = ? WHERE book_id =
?";
    try (PreparedStatement preparedStatement = connection.prepareStatement(query)) {
        preparedStatement.setString(1, category);
        preparedStatement.setString(2, name);
        preparedStatement.setString(3, author);
        preparedStatement.setInt(4, copies);
        preparedStatement.setInt(5, bookId);

        int rowsUpdated = preparedStatement.executeUpdate();
        return rowsUpdated > 0;
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return false;
}
```

```
    public ResultSet getBookDetailsByIdOrName(int bookId, String bookName) {
    String query = "SELECT * FROM books WHERE book_id = ? OR name = ?";
    try {
        PreparedStatement preparedStatement = connection.prepareStatement(query);
        preparedStatement.setInt(1, bookId);
        preparedStatement.setString(2, bookName);
        return preparedStatement.executeQuery();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}
}
```