

# DESCRIPTION: FedhaGroupSystem

NAME: OLOO, ASIRA

REGISTRATION: J17/6128/2023

## Project Setup Guide

### 1. Prerequisites

Ensure you have the following installed on your machine:

1. **Java Development Kit (JDK) 17 or later:** Required for JavaFX and the project.
2. **Apache Maven:** To manage dependencies and build the project.
3. **MySQL Server:** To host the database.

### 2. Set Up the Database

1. **Create the Database Schema:**
2. Open your MySQL client or workbench.
3. Execute the provided SQL scripts in `src/main/resources/sql` or as shown in the code snippet:
4. `members`, `deposits`, `loans`, `guarantors`, `shares`, `staff_members`, `users`.
5. Ensure the tables are created successfully.
6. **Configure Database Connection:**
7. Locate the `db.properties` file in the `resources` directory.
8. Update the file with your MySQL database credentials:

```
db.url=jdbc:mysql://localhost:3306/your_database_name
db.username=your_username
db.password=your_password
maximumPoolSize=10
minimumIdle=2
idleTimeout=30000
connectionTimeout=30000
poolName=HikariCP
```

1. **Optional: Import/Export Schema:**
2. Uncomment and use the methods in `DB_Config` to export or import schemas when needed.

### 3. Add Local Dependencies

This project depends on two locally available JARs:

1. `coreAuth-1.0.jar`
2. `DB-Initializer-1.0.jar`

To add these:

1. Place the JAR files in the `lib/` directory in the project root.

2. The Maven `pom.xml` file is already configured to reference these files:

```
<systemPath>${basedir}/lib/coreAuth-1.0.jar</systemPath>
<systemPath>${basedir}/lib/DB-Initializer-1.0.jar</systemPath>
```

#### 4. Build the Project

1. Open a terminal or IDE with Maven support.
2. Run the following command to install dependencies and build the project:

```
mvn clean install
```

#### 5. Run the Application

1. Run the application using Maven:

```
mvn javafx:run
```

1. This will start the JavaFX-based UI.
2. Alternatively, you can run the application directly from the IDE:
3. Locate the `FedhaApp` main class in `ac.ku.ooloo`.
4. Run it using the IDE's "Run" or "Debug" functionality.

#### 6. Application Overview

1. **Start Screen:** Displays a splash screen with a logo and loading animation.
2. **Login Screen:**
3. Enter your username and password to log in.
4. **Staff Dashboard:**
  1. Manage members, loans, and deposits.
  2. Includes a menu bar with additional features.
5. **Member Dashboard:**
  1. View member-specific information, including deposits and loans.
6. **Admin Actions:**
  1. Accessible features depend on the role: STAFF or MEMBER.

#### 7. Troubleshooting

1. **Database Connection Issues:**
  1. Check if MySQL is running and credentials in `db.properties` are correct.
2. **JavaFX Errors:**
  1. Ensure you have the correct JDK version (17+) and JavaFX SDK installed.
  2. Add JavaFX SDK to your `PATH` if required.
3. **Missing JAR Errors:**
  1. Verify the local JAR files (`coreAuth` and `DB-Initializer`) are placed correctly in the `lib` directory.

#### 8. Additional Notes

1. Logs are managed using SLF4J and Logback. Check the logs for detailed application activity and errors.
2. To modify UI styles, update `stylesheet.css` in the `resources` directory.

3. Ensure all dependencies, including external libraries, are correctly configured.