**Report on Theatre Web project**

**Table of Contents**

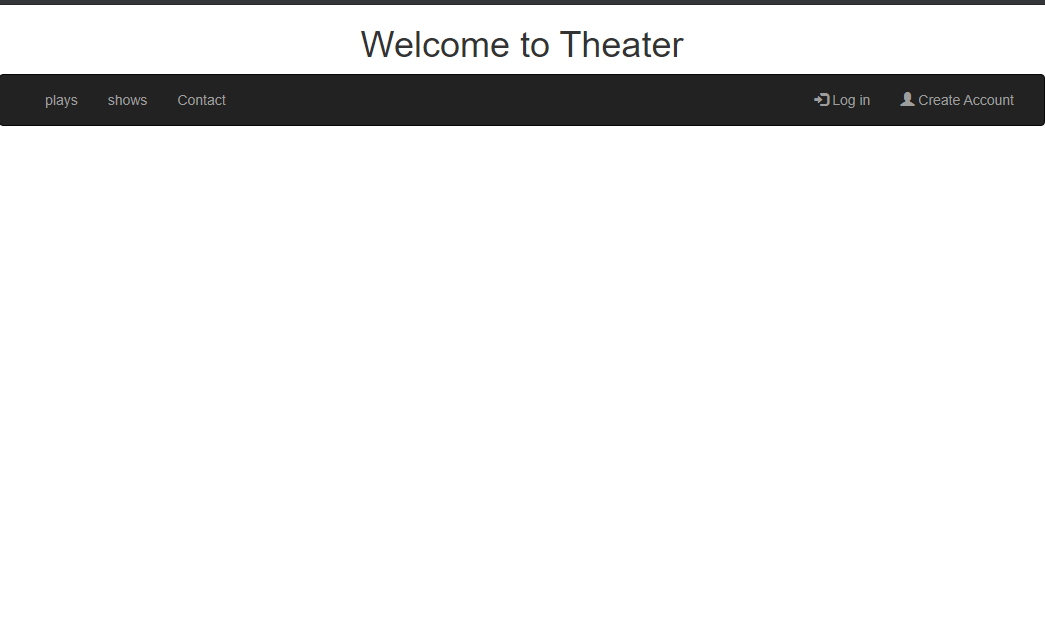
[**1.** **Project Configuration** 1](#_Toc114895752)

[**2.** **Database Schema** 2](#_Toc114895753)

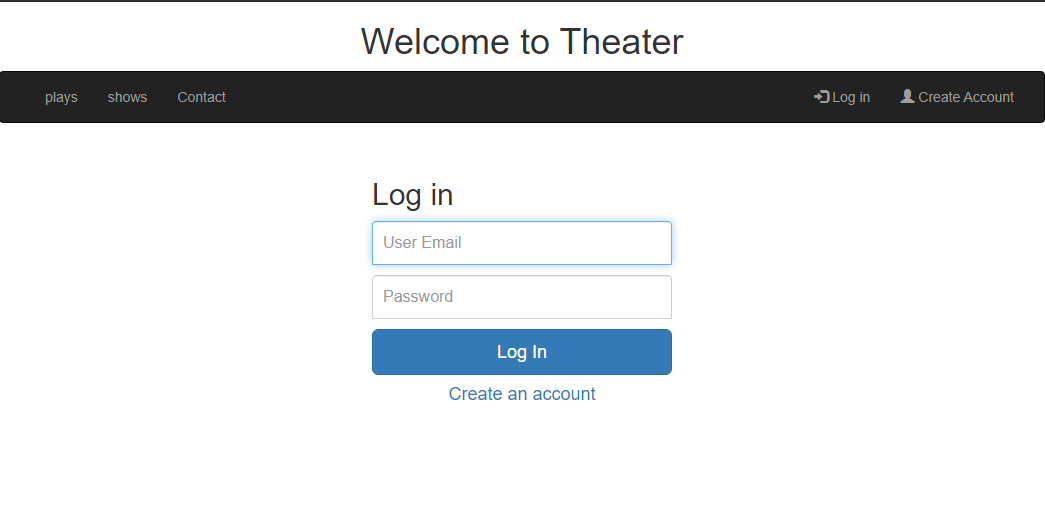
[**3.** **Deploying the project** 3](#_Toc114895754)

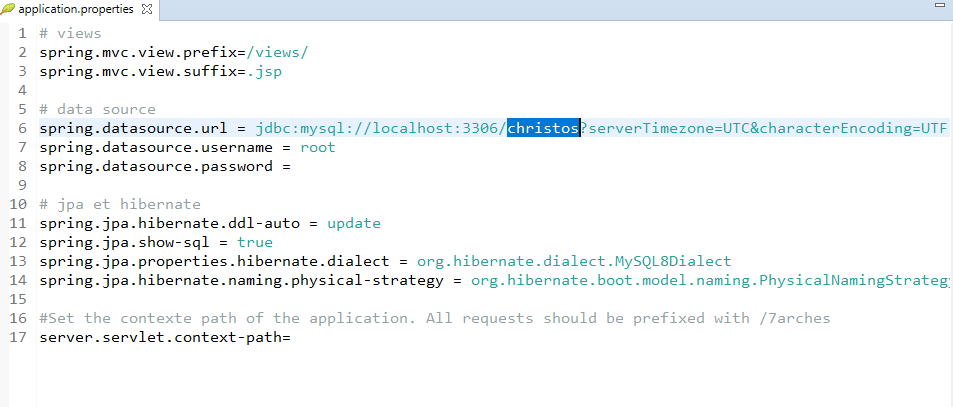
# **Project Configuration**

To solve the problems presented in the requirements, the project was organized as follows. The user interface was composed of JSP and integrated with Spring boot Security and JPA to perform user login and CRUD operations presented in the requirements. Apache Tomcat 9.0 was used as a server to run the project. Also, the project was configured as a Java Maven project. When starting the project with Apache Tomcat Server, the following Welcome page is displayed.



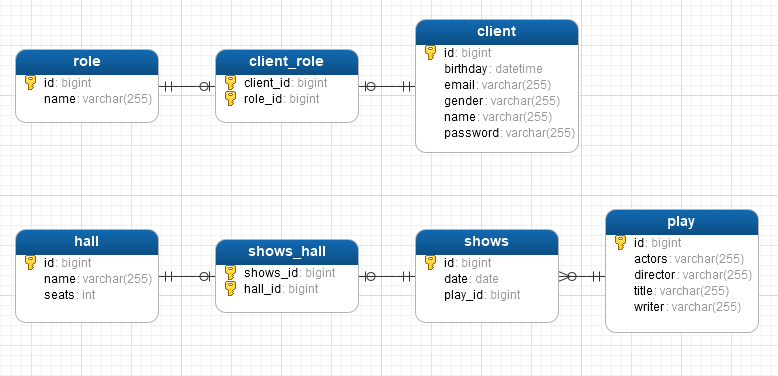
If you click the playstab in the Header section, the currently playing plays are displayed. Log In function was performed using Spring Security. In order to proceed with login, I made it to pass email and password. The login interface is as follows.



Face-to-face with the data base was realized by using JPA, and MySQL was used as the data base. You can set up the data base face in the application.properties file of the project. This file is included in the resources folder. The contents of the file are as follows.

As shown in the figure on the right, the url for datasource is set as mysql database, and a new database called christos needs to be created first to run the project.

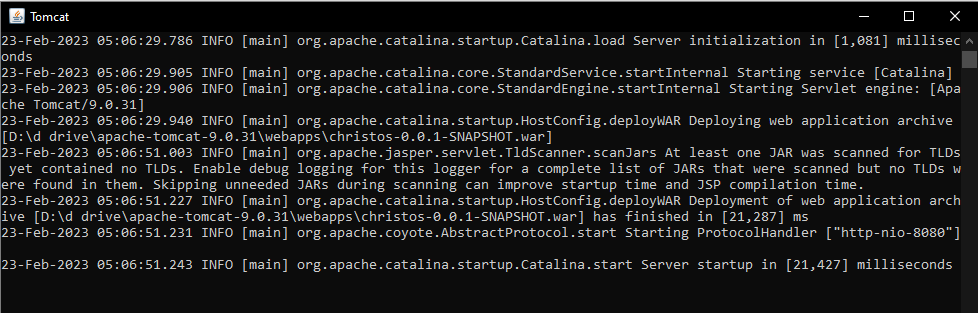
# **Database Schema**



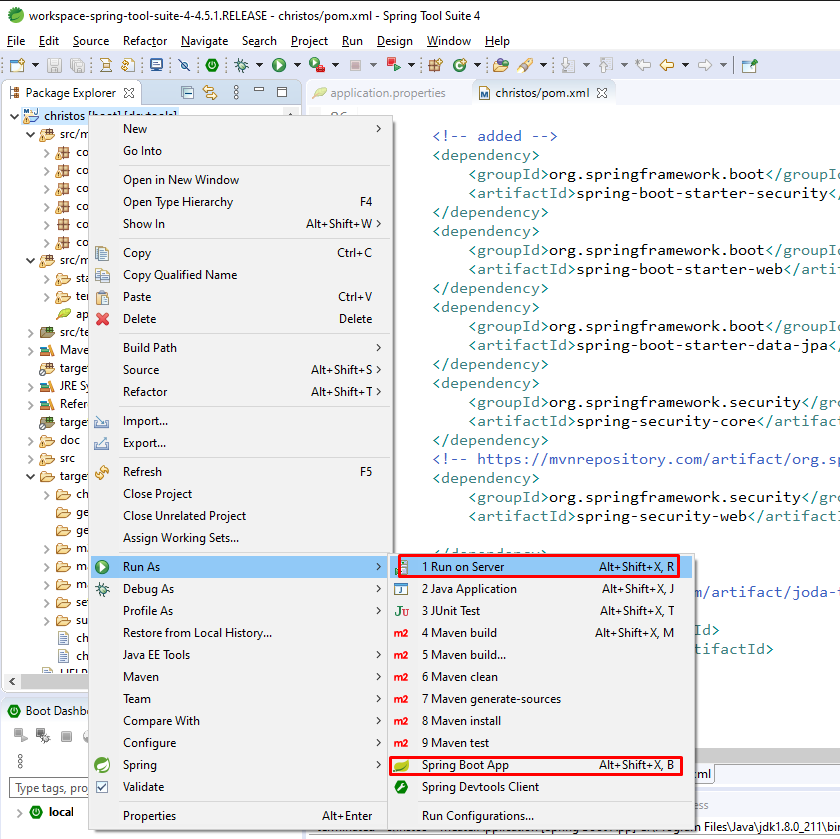
As the schema on the right shows, there is an N-N relationship between the Client (User) table and the Role table. And there is a 1-N relationship between the Play table and the Shows table, and an N-N relationship between the Hall table and the Shows table.

# **Deploying the project**

When the project is built, a .war file is created in the target folder. Copy the created war file to the webapps folder of Apache Tomcat Server. Next, move to the bin register of the tomcat server and execute the start.bat (start.sh) file to run the tomcat server. After the server runs successfully, try to access http://localhost:8080 through the browser.



Also, you can install Tomcat Server in Eclipse or Spring Tool Suite IDE and run the project through it.



# UML diagrams









