

Project Title: Songs Catalogue Web App

Description:

A responsive web application that allows users to easily keep track of their favourite songs of all time. These songs are maintained in the user's songs catalogue, and catalogue songs can be added to as well as filtered.

Technologies Used: Java, Spring Boot, React JS

Installation & Usage Instructions:

- Install JDK & set JAVA_HOME environment variable
 - Get a Linux terminal application and open it (Terminal app for Mac, or see link for Windows: <https://itsfoss.com/run-linux-commands-in-windows/>)
 - Enter in terminal: `curl -s "https://get.sdkman.io" | bash`
 - Open a new terminal or run the 'source' command outputted by the previous command inside the current terminal
 - Enter in terminal: `sdk install java 17.0.5-tem`
 - Enter in terminal: `which java`
 - Copy the path shown containing "sdkman" up to /current
 - Enter in terminal: `export JAVA_HOME=/path/to/java/up/to/current`
- Install Node JS
 - Download the latest LTS version for your PC from here: <https://nodejs.org/en/download>
- Start the web application:
 - Enter in terminal: `cd path/to/Take-home project`
 - Start the web application by entering in terminal: `./mvnw spring-boot:run`
 - *Run the backend tests by entering in the terminal: `./mvnw test`*
 - Go to the website by opening a browser application (e.g. Chrome) and entering: <http://localhost:8080/login>
- To run React UI separately:
 - Enter in terminal: `cd path/to/Take-home project/frontend`
 - Start the UI by entering in terminal: `npm start`
 - Go to the website by opening a browser application (e.g. Chrome) and entering: <http://localhost:8081>
- To access the H2 DB:

- Enter in terminal: `cd path/to/Take-home project`
- Start the web application by entering in terminal: `./mvnw spring-boot:run`
- Go to the H2 console by opening a browser application (e.g. Chrome) and entering: <http://localhost:8080/h2>
- Enter login credentials which can be found in: `path/to/Take-home project/src/main/resources/application.properties`
- **Note:**
 - If any H2 DB or data source changes are made, will need to rebuild project first before running application. Similar for any updates to Java files.
 - Always run app from console not IDE (sometimes IDEs like IntelliJ are not recognising when Spring Security is disabled it seems).
 - Spring Security blocks access to H2 console in browser, so need to configure if using it.

Project Considerations & Future Improvements:

- Would focus more on validation of form input fields beyond selection of 'type' in real case scenario (e.g. incorporate FormFeedback, etc)
- Could add a route for No page match (e.g. 404) with a custom UI page
- Add SSL certificate (for https) and further security (e.g. OAuth logins with Google, FB, etc and 2FA)
- Choose from existing songs in DB for 'Add Song' as opposed to manual additions through form entry
- Remove song from catalogue
- Update user details (e.g. change password, etc)
- More tests & further improve UI
- Implement Spring Security and JWT for Login authentication & authorisation
- Further configuration of Spring Actuator for app monitoring / tracking purposes
- Integrate BE & FE into one using frontend-maven-plugin or exec-maven-plugin (so it can be fully run from just one url)

Additional Notes:

Did not manage to complete implementation of Spring Security with JWT for user login page process. Have left the WIP files in 'security_wip' folders for both FE & BE. All related code has been disabled for now.

