**IDEXX SEARCH APPLICATION ASSIGNMENT**

**Introduction:**

* This document provides a high level overview and explains the architecture of the Search application.
* The document defines the architecture components that have been selected.

**Architectural Representation**

**Presentation**

3rd Party APIs

REST API (JAVA)

Javascript Function

HTML

**Logic**

**Model**

**Data**

**Fig: Application Architecture**

**Presentation:**

* This is the topmost level of the application.
* The presentation layer provides the application's user interface (UI)
* This involves the use of Graphical User Interface for client interaction and web based technologies for browser-based interaction.
* This layer communicates with other layers to fetch or add the data.
* For Search Application, we are using HTML as a presentation layer.

**Logic:**

* Typically, the logic tier contains the business logic of the application.
* It is the layer where all the critical business problems are solved.
* For Search Application, we are using Javascript function, which further have the jquery ajax function to make the Java REST API call.
* Similarly, for server side, we are mentioning the business logic in service layer.

**Data:**

* This is the tier where information is stored or retrieved .
* This layer keeps the data neutral from the other layers of application.
* For Search application, we are calling the 3rd party apis to fetch the data from this data layer.

**Justification for the above approach:**

* During the application's life cycle, the 3-tier architecture provides the benefits such as scalability, reusability, flexibility, manageability and maintainability.
* Each layer have independent files which contains the specific code.
* This approach helps to keep the code more readable and keeps the layers separated from each other.

**Application Development Environment (Technologies)**

* **Operating System:** Windows 10
* **Frontend** :- HTML, Bootstrap 3.3.7, Jquery, Javascript
* **Backend** :- Java 8, Spring boot 2.3.3.RELEASE, Eclipse
* **Application Server** :- Apache Tomcat 9.0.37 (Embedded tomcat provided by framework)

**Functional Specification Details of REST API**

1. In Search Application, we are using swagger for the REST API representation.
2. Below is the link to open the swagger.

<http://localhost:8080/idexx/swagger-ui.html>

**Application Monitoring Details**

We have used the actuator to monitor the application status.

Below are the URL's that would help to find further details of the application .

* To check the metrics of the application

<http://localhost:8080/idexx/actuator/metrics>

* To check the metrics of the HTTP server request

<http://localhost:8080/idexx/actuator/metrics/http.server.requests>

* To check the custom logs generated in log file

<http://localhost:8080/idexx/actuator/logfile>

* For health check of the application

http://localhost:8080/idexx/actuator/health

**How to run the Application?**

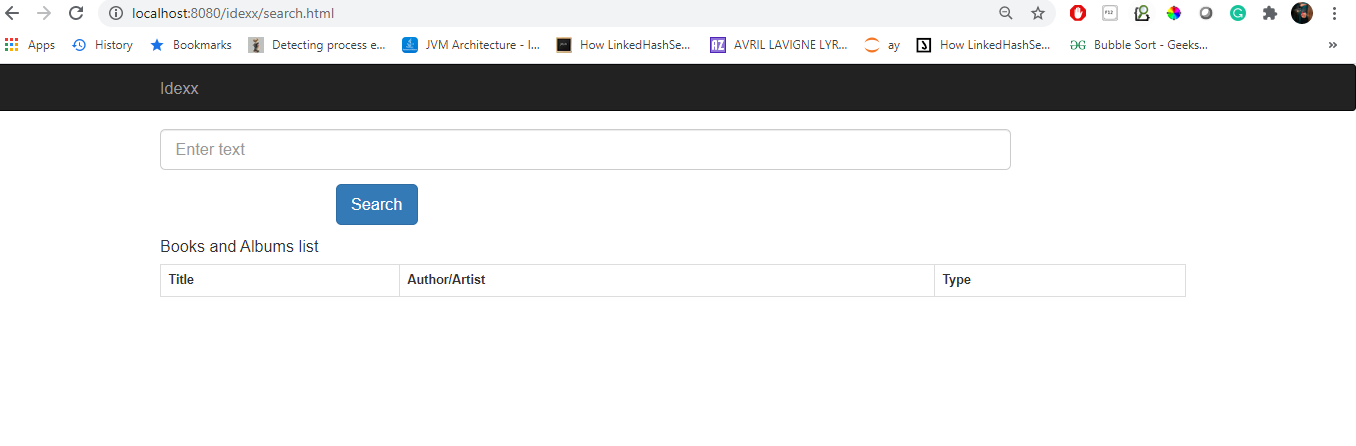
1. Clone the code or download the zip file from the git repository.

$ git clone https://github.com/samiksha-warkari/search-application.git

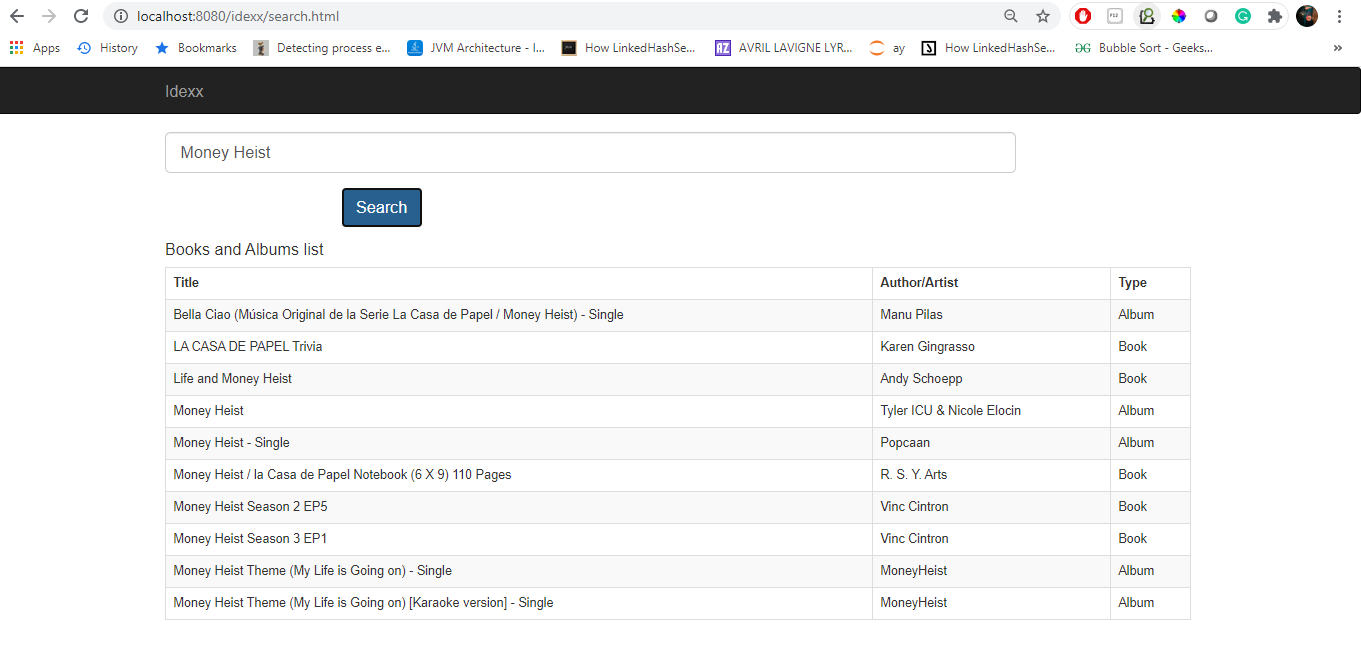
1. Extract the downloaded code.
2. Import the code in Eclipse IDE.

(Go to file -> Click Import -> Select Existing Maven Projects -> Click Next -> Provide the path of the extracted project -> Finish)

1. Right click on project -> select run as -> Spring Boot App.
2. Once the application started successfully, go to the web browser and paste the URL <http://localhost:8080/idexx/search.html>



1. Enter text in the input field.



NOTE:- The maximum result each of book and album is configured in the application. properties file, which is default set as 5.

In order to increase the limit of the result, following changes need to be done.

**application. properties**

maxResults=5

change the value of the maxResults parameter

limit=5

change the value of the limit parameter