Abstract

This is a full-stack recommendation web app project which has three main functions

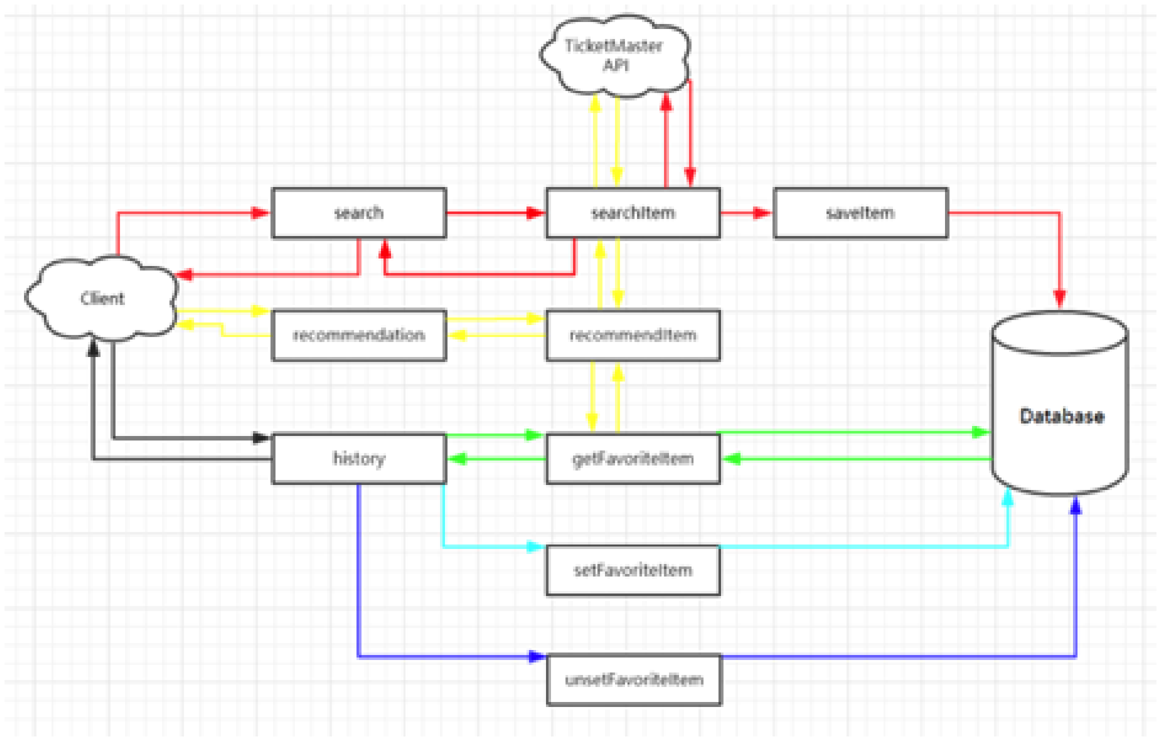
1. Search: Users can search events nearby based on their location.
2. Save: Users can save the events if they like them
3. Recommend: The recommendation system will recommend the nearby events based on users' favorite items and searching history

Data Source

For this project, the data are taken from external API. Then based on back-end logic, some of them will be stored into the mysql database tables.

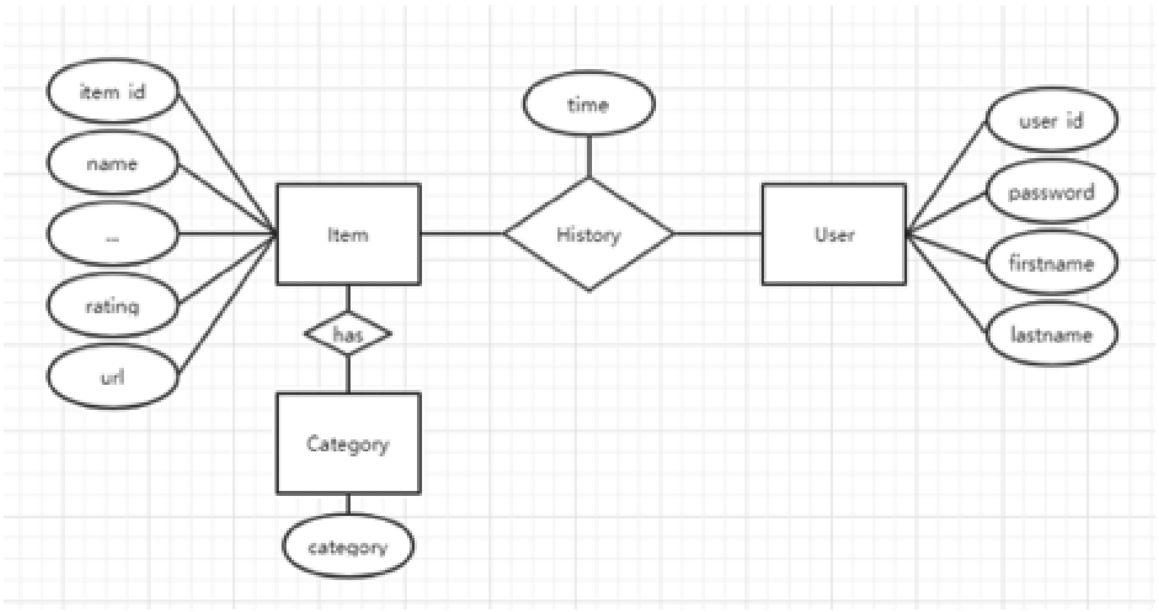
The data from both external API and Database's tables, when they are sent to the front-end, will be converted into raw JSON file. Then the JSON file will be presented to users in the forms of images, fancy font, and url links.

Architecture of this project



In the picture above, users can actually use this app through Client-end, and Client will talk to Server through HTTP request. In the Back-end, Java Servlets, where there are several RESTful APIs, will handle the HTTP request and routes. Based on service logic, the RESTful APIs will decide which service-methods, such as getItemById, searchItem, setFavoriteItem, and many many other methods, to be called. These service-methods are the methods which will be directly connected with database and external APIs, then query or insert or delete data from them.

Database design and ER diagram



There are four tables:

1. User table will store the user information, including their user_id, password, and user_name.
2. History table will store the user searching information, including user_id, item_id, and the time when user search the certain item.
3. Item table will store all the information of event, including item_id, name, distance from client to event place, event url, image url.
4. Category table will store the categories information of certain event

SQL

All the queries, as well as inserts, delete, and update SQL statements, are embedded into the web service.

Specifically, all the SQL statements are in the package ‘db’. The MySQLConnection class includes all the inserts, queries, update, and delete SQL statements, which are corresponded with

Java methods. For example, the ‘getFavoriteItems’ method include SQL query ‘SELECT * FROM items WHERE item_id = ?’, while the ‘saveItem’ method includes SQL insert ‘INSERT IGNORE INTO items VALUES(?, ?, ?, ?, ?, ?, ?)’

Codes

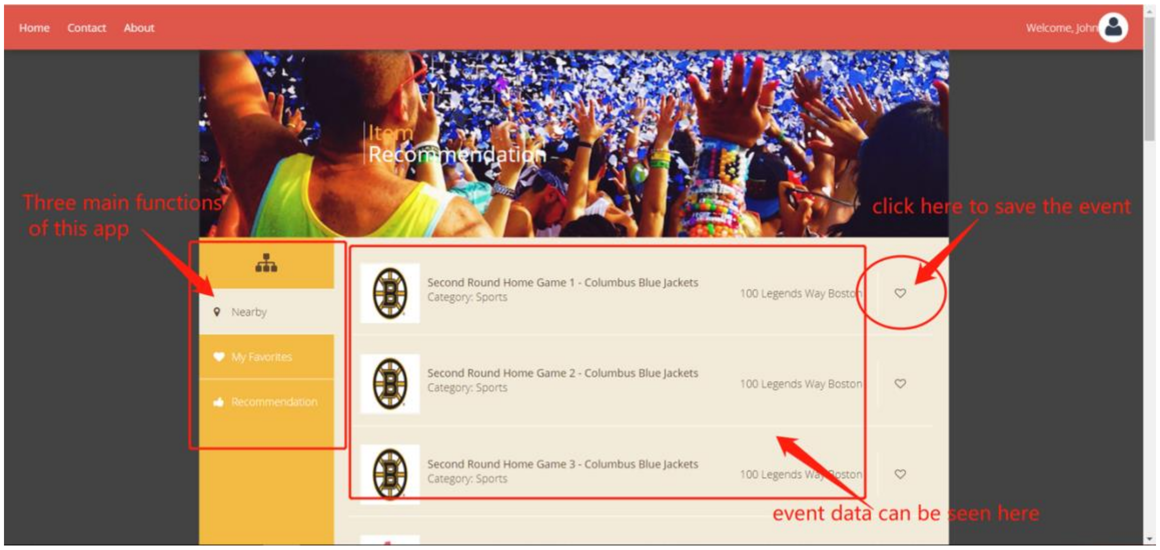
Since this is the full stack web app, the front-end codes includes HTML/CSS/JavaScript, back-end codes includes Java.

The front-end codes are in the WebContent folder and the back-end codes are in the src folder.

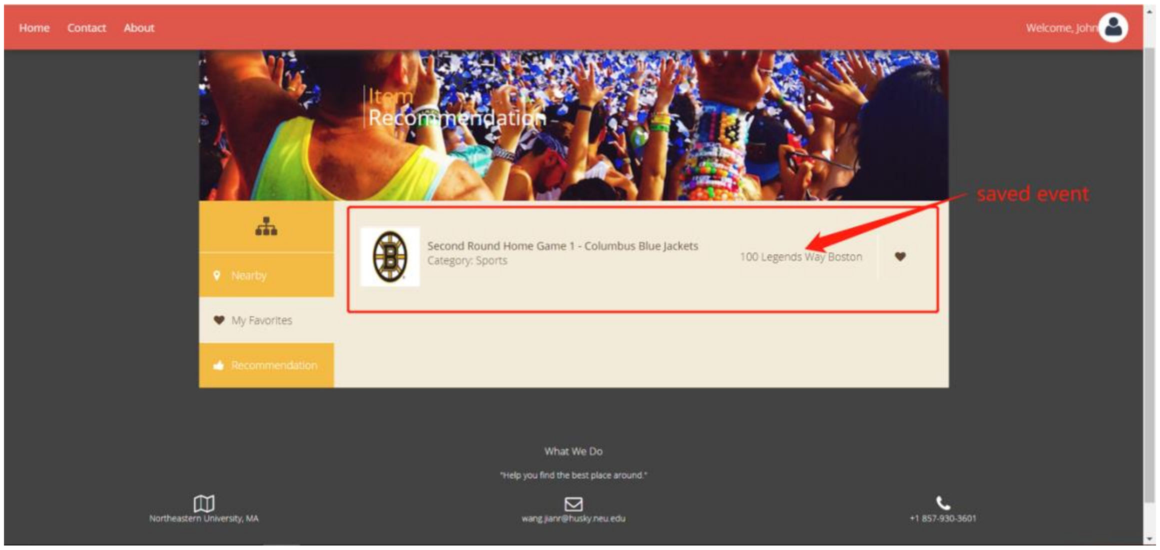
Demo of this project

This web app is now running at local machine, localhost:8080/Jupiter/

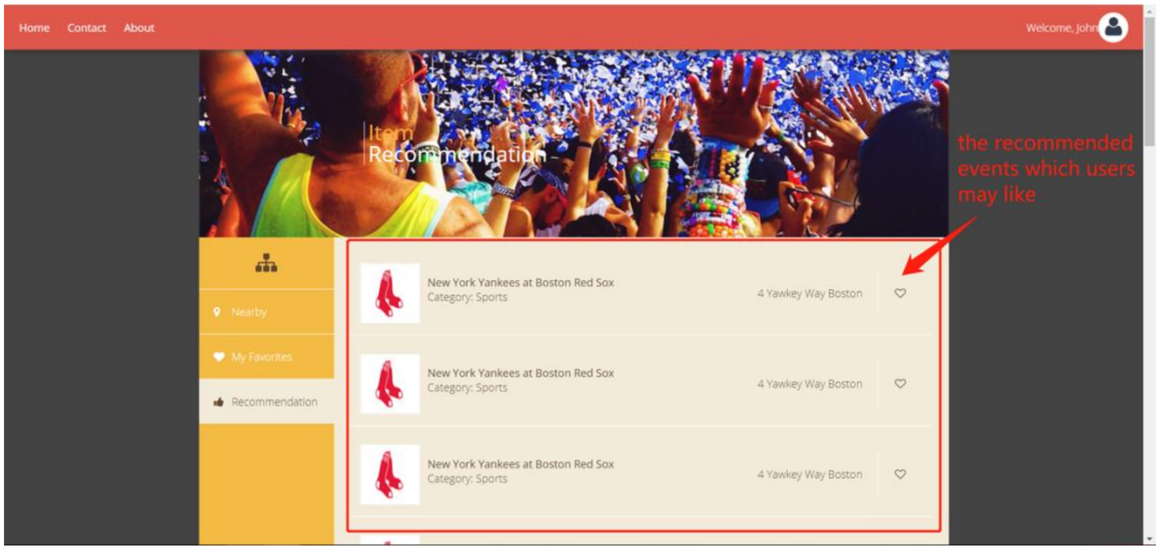
Users can use the app in the way which the pictures show below


Home
Contact
About
Welcome, John
Item Recommendation
Three main functions of this app
click here to save the event
Nearby
My Favorites
Recommendation
Second Round Home Game 1 - Columbus Blue Jackets
Category: Sports
100 Legends Way Boston
Second Round Home Game 2 - Columbus Blue Jackets
Category: Sports
100 Legends Way Boston
Second Round Home Game 3 - Columbus Blue Jackets
Category: Sports
100 Legends Way Boston
event data can be seen here

The User Interface will provide three main functions in the left nav bar; The User Interface will also provide events nearby based on users’ locations; Users can click what they like and save them.



Whatever the users save will show in the My Favorites part



Then the recommendation part will recommend events based on what users saved

How this project works in back-end? How to execute the SQL code?

The entire web service has already been running in the AWS. And since the SQL statements are embedded in the service, as a user you just need to click the buttons or links, which will trigger the corresponding back-end methods, inside which the SQL statements are.

For example, if you like an event and you want to save it, you can just click the 'love' icon. Then, the Eventlistener will trigger the REST API in the Servlet, and the REST API will call a method 'setFavoriteItems'. Inside this method, there is a SQL insert statement 'INSERT IGNORE INTO history (user_id, item_id) VALUES(?, ?)' and once this method is called, this SQL statement will be executed. At last, the event you want to save will be actually inserted into the Database table 'history'.

Also, if you want to see the events you save, you can just click the button 'Favorites'. Then, based on your userId, the method 'getFavoriteItemIds' and 'getFavoriteItems' will be called. And the SQL query statement 'SELECT item_id FROM history WHERE user_id = ?' and 'SELECT * FROM items WHERE item_id = ?' will also be executed. Thus, you can finally see the SQL query results, which are your favorite events, in the front-end pages.