CS 6314.501

Web Programming Languages - F17

Project Report

Submitted by:

Priyank Shah (pss160530)

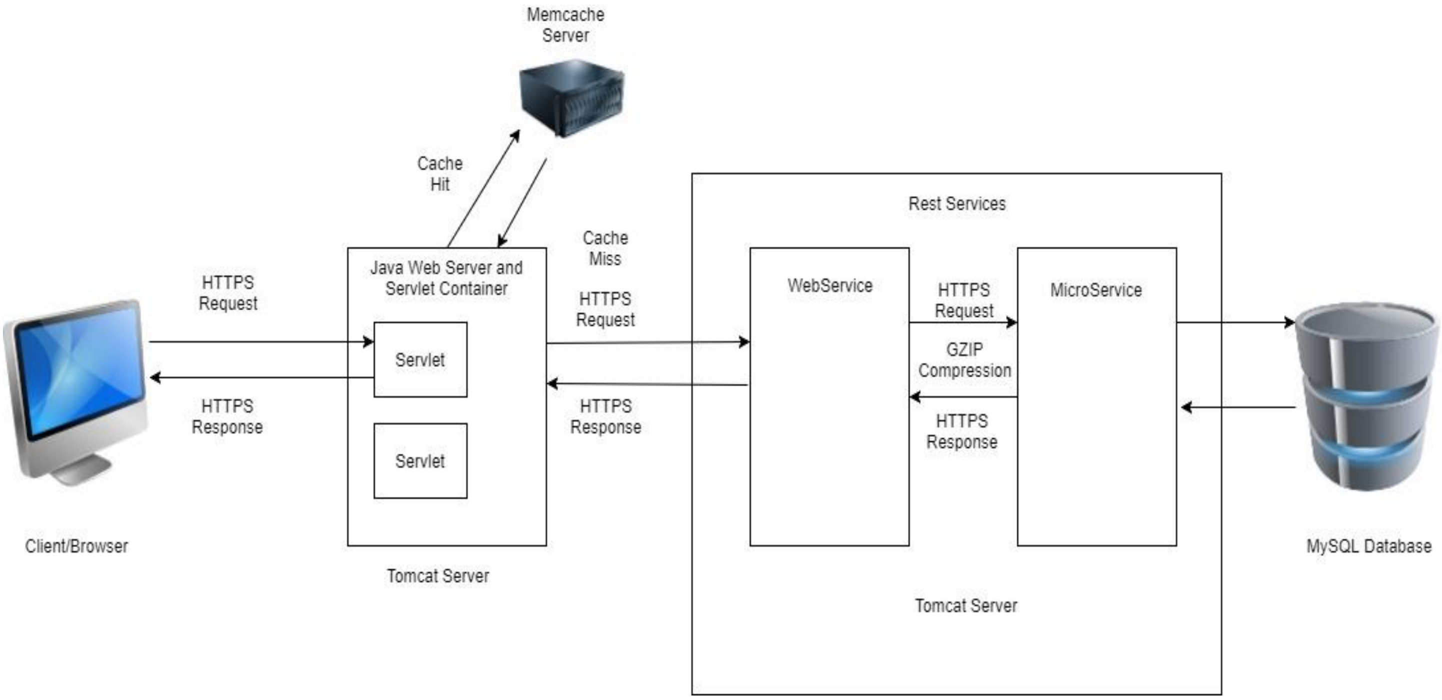
Jaiminee Kataria (jxk172330)

DhwanibenKaneria (drk170130)

Objective:

The objective of this project is to develop a web application that can be used to know the type of the items my friends want to buy, so that I can gift them on their birthday or any other special occasion.

Architectural Design:



Description of the modules:
The following are the technologies used in the modules for the development of our website.

1) Client/Browser:
   Bootstrap, HTML, CSS, JavaScript, JSP and Servlets

2) Webserver: Apache Tomcat v 8.0.33
   Other Options: Apache Http Server.
   Tomcat was preferred because of the following reasons:
   - Provides the Java Servlet and JSP support for dynamically served pages
   - Works as a light-weight testing server
   - Can be run in different modes to promote better performance

3) Rest Service Framework: JAX-RS Jersey Rest framework.
   Other Options: Spring MVC
   JAX-RS was considered because
   - JAX-RS has the advantage of creating APIs that are simpler to create and digest messages for in different browsers and mobile devices.
   - The Rest Services are deployed on Apache Tomcat server.
   - JAX-RS targets the development of web services (as opposed to HTML web applications) while Spring MVC has its roots in web application development.

4) Cache: Memcache Distributed Cache.
   Other Options: Redis.
   - Memcache is a good option for implementing a distributed caching mechanism.

- It stores data in the format of key value pair. Currently, only single node is implemented using spymemcache.

5) Database: MySQL v5.7.11
   Other Options: Orcale, PostgreSQL.
   - MySQL is relatively light-weight, can be extremely fast when applications leverage architecture.
   - Lots of features stay free as the database servers grow such as replication and partitioning.
   - MySQL excels when high speed reads can be used for web, gaming and small/medium data warehouses and OLTP systems.

6) Development Environment: JDK 1.8(JSP/Servlet Framework)
   Other Options: Javascript, Python
   - Java is high performance compiled languages, with great support from a well-known vendors, and entire ecosystem of companies that sell everything - from IDEs to libraries to automation tools.

7) Compression: GZip compression.

Functionalities available to user on our website.
Website Overview:

- The website is for the users to view and select the favorite items of their friends to gift them on some special occasion.
- The user can search and assign their name besides the products by clicking I'll purchase it. Also, they can make their own multiple registry and have power to make it either public or shared with selected users of the website.

1) New regular user registration.
   New User can register through the registration page where he would provide email id, username and password. Once he clicks sign up, he would be redirected to home page.

2) Regular user should be able to do following things:
   a. Login
   b. Logout
   c. Can view and edit their profile information.
   d. Forgot password functionality
   e. Can create the multiple registry
   f. Can search for an item that the user would like to add to a registry
   g. Can place filters on the display products to ease their item search.
   h. Can add/remove an item from the registry
   i. Can share a registry to particular user or make it public
   j. Can self-assign an item on another user's registry.

1. New regular user registration
   New User can register through the registration page where he would provide email id, username and password. Once he clicks sign up, he would be redirected to home page.

2. Regular User
   i. Login

Registered user would be able to login through login page and once he clicks sign in. he would be redirected to home page.

ii. Logout

Logged in user would be able to log out by clicking Log Out button.

iii. User profile information display and editing

User can change his profile and add extra details by filling data in Profile page.

iv. Forgot password functionality

User can click on Forgot Password Link. It would redirect to page of security questions. Once User has filled answers, new password can be set.

v. Ability to create a registry

When User clicks “Create Registry” button, he would select the items from the inventory and add it into own registry.

vi. Ability to search for items that you would like to add into registry

1. Table display:

User can type a name of item in search bar and result would be displayed based on the name. Result would be displayed in tabular format.

a. Results (with at least four properties) should be displayed in a sortable table (i.e. allowing resulting to be sorted on any column)
   We have implemented four properties of product. Color, Price, Category, Brand. Data would be displayed in sorted manner with all properties.

2. Search results filtering capabilities on at least four result item properties
   Data can be filtered on selected property of product. For example, if price is selectedas low then, data with the lower price would be first in table.

vii. Ability to add/remove items from a registry

User can add or delete multiple items in registry by clicking two buttons.

viii. Ability to share registry to particular user or make it public

User can share registry with his specific friends if it is private and if it is public then it would be shared to all other users.

ix. Self-assign an item on another user’s registry

Once Other users share their registry with current user then he would be able to see multiple registries which are shared by other users to him and when he click add or buy button, he can self assign himself in other user’s registries.

3. Accessible any unavailable page should retrieve a pretty and generic 404 page
   When any error occurs like unregistered user tries to login, our system would redirect to error page showing that User does not exist.

4. Admin User:

i. Login

Admin can log in with username= admin and password=admin.

ii. Logout

Admin can logout once he clicks logout button.

iii. Add/remove items into/from inventory

- ➔ Extra functionality: Add/remove specific filters like color, price, category and brand of items in databases and that would be affected to the page where he can see filters in drop down while adding new items in the inventory.

iv. Display the items in the inventory

Admin can see added items in the inventory.

Other Required Features:

MemCache:

We implemented memcache using SpymemcacheMemcacheClient at login level. When user Logs in first time, there would be Cache miss because username and password are not Added in memcache. When he logs in second time, memcache will give hit as username Password (key value pair) are stored.

Request/Response Compression:

We have used server.xml file to enable Request/Response compression. We added Compression=”force”. It will show Content-encoding=gzip for response headers and Accept-Encoding=

Client-Server Communication Encryption:

We have used keytool command to create keystore and it was used to create certificate, we Added piece of code in server.xml. To enable our web application SSL configured, we Changed our web.xml file.

To verify, SSL configured web application u can see “https:” instead of “http:” in the URL.

Web Services:

Web Service	Description
RestService/loginservices/checkuservalidity	Will call username and userpass micro services by passing username and password to it. It will check for the validity of the user.
RestService/fetch/setpassword	This service will call the fetchprofile microservice and it will send the username and password to this microservice.
RestService/registerservices/checkuser	This service will call two micro services namely regusername microservice and insertusername microservice by passing username, email and password respectively.
RestService/GetFriendsService/GetFriends	This service will call the getfriends micro service to display all the

	users of the system so that the currently logged in user can select from the listed user with whom he/she want to share their registry.
RestService/CreateUserRegService/CreateUserReg	This service will call the createUserRegService microservice by passing username, registry name and all the users with whom the current user want to share this particular registry.
RestService/getItemService/searchItems	This service will call the search item microservice which will allow the user to search for the particular items.
RestService/getHomeService/getHome	This is a dedicated home page which will call the getHomevalues microservice in order to bring all the values required to be display on the home page.
RestService/deleteRegItemService/deleteRegitem	This service will call the deleteRegItem micro service by passing the product id and the registry id.
RestService/addRegistryValuesService/addRegistryValues	This service will call the addregistry micro service by passing product id, username and the registry id.
RestService/getUserService/searchUser	It will call dedicated micro service and get users typed in search bar
RestService/addfriend/addfriend	It will call microservice to insert the friend in the database
RestService/productFiltersAndDetailsService/addBrand	It will add brand in database thru micro service
RestService/productFiltersAndDetailsService/addColor	It will add color in database thru micro service
RestService/productFiltersAndDetailsService/addCategory	It will add category in database thru micro service
RestService/productFiltersAndDetailsService/addCategory	It will add brand in database thru micro service
RestService/insertAndDeleteProductsService/deleteProduct	It will call micro service to delete product from registry
RestService/fetchdata/alldatae/fetchdata/alldata	It will fetch the all profile information
RestService/insertAndDeleteProductsService/insertProduct	It will call micro service to insert product from registry
RestService/getFilterService/getFilter	It gets the product filters thru

	micro service
RestService/fetch/updateprofile	It updates the profile thru micro service

Micro Services:

Micro Service	Description
RestMicroService/usermicroservice/checkusername	It checks username with the data already exists in userdetails table.
RestMicroService/usermicroservice/checkpassword	It checks password once user already entered in userdetails table. If it matches then it redirects to home page. Otherwise shows password does not match.
RestMicroService/regusermicroservice/checknamereg	It checks username of registering user with the username exists in table. If it matches then it shows user already exits.
RestMicroService/insertusermicroservice/insertusernamereg	If username does not exists then it inserts username, email and password in the table
RestMicroService/fetchprofile/fetchprofile	It fetches user profile information from the database
RestMicroService/fetchprofile/updateprofile	It inserts updated values of user profile in the table.
RestMicroService/fetchprofile/fetchfriends	It fetches friends of current user from table
RestMicroService/fetchprofile/checkanswer	It validates the answer of security question already entered while updating profile.
RestMicroServices/fetchprofile/setpassword	It sets new passport for the user into database.
RestMicroServices/MicroCheckRegName/checkRegName	It checks if user alreadyhave registry name or not. If yes then It won't allowed to create registry with the same name.
RestMicroServices/MicroInsertRegName/insertRegName	It creates registry with given name and shared with specific friends or make it public.
RestMicroServices/deleteRegItemMicro/deleteRegItem	It deletes particular items from particular registry.
RestMicroServices/addRegistryMicro/addRegistryValuesMicro	It adds items into registry while creating registry.

RestMicroServices/searchItemmicro/searchItem	It searches items with partial name or product description of Items.
RestMicroServices/Fetchdatamicro/alldata	It fetches all Items from database.
RestMicroServices/addAndDeleteProductDetailsMicroService/insertProductDetails	It adds Items details into database.

problems encountered during the project and how these issueswere resolved

1. MemCache Issue:
   Previously we used whalinmemcache but it was not working because it was not able to connect to memcache server with specific port. To resolve it, we installed memcache, it was also working as a daemon process, but memcacheclient from whalin was not giving correct result.

   Solution: We used spy memcacheMemcache client and tried to connect to memcache server using `InetSocketAddress Class.`

2. TL SSL Issue:
   Previously we tried to execute keytool command to generate keystore and certificate but it was not generating it with zero error.

   Solution:
   we tried different paramaeters to keytool command to configure SSL in Tomcat.

3. Compression Issue:
   We followed specific steps to enable compression/ content encoding in response header. It worked for one server but as we have two different servers one for Web Client and one for Web Service and Micro Services, it is not compressing response for web service or micro service call.