Knowledge Discovery

and

Management

Report(Exam)

**Data Model**

**Features**: Here is an application developed using big data. The big data used in this application is movies information where they are a number of users, who watch a number of movies. The movies are given ratings based on a scale of 10.In this application there are 4 features. They are

* Firstly in the application we need to mention the user Id, where the Id is needed to get the other 3 features.
* Secondly, the user Id taken in the above step gives the result of the movies which are recommended to be watched by the User as mentioned above
* Thirdly, the movies watched by the user are displayed based on the given user Id
* Fourthly, An API is used that is the weather API which gives the information about the weather conditions in the respective given area you wish to search. The search consists of both state code and city name by which the temperature will be displayed.

The results in the second and third features are the movie Ids

**Recommendation Model and Algorithms**

**Model** : This model uses the recommendation algorithm. In this application the data set used is based on the movies, where a number of users have watched a number of movies and have given rating for them. The users are recommended to watch the movies when two users watch in common and there leaves movies which are uncommon, so these uncommon movies are suggested to be watched which have a good rating. Therefore these are the ones which are not watched by them. For example if we consider two users 1 and 2 where user 1 has watched A, B and C movies and user 2 has watched B, F, G movies. In this case what happens is when we run recommendation algorithm user 1 will be suggested movies A and C and user 2 will be suggested movies F and G. So this is how the recommendation algorithm works.

**Recommendation Algorithm:**

package com.unresyst;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.List;

import java.io.IOException;

import org.apache.commons.cli2.OptionException;

import org.apache.mahout.cf.taste.common.TasteException;

import org.apache.mahout.cf.taste.impl.model.file.FileDataModel;

import org.apache.mahout.cf.taste.impl.recommender.CachingRecommender;

import org.apache.mahout.cf.taste.impl.recommender.slopeone.SlopeOneRecommender;

import org.apache.mahout.cf.taste.model.DataModel;

import org.apache.mahout.cf.taste.recommender.RecommendedItem;

import org.apache.mahout.cf.taste.impl.common.LongPrimitiveIterator;

public class UnresystBoolRecommend {

public static void main(String... args) throws FileNotFoundException, TasteException, IOException, OptionException {

// create data source (model) - from the csv file

File ratingsFile = new File("datasets/moviesdata.csv");

DataModel model = new FileDataModel(ratingsFile);

// create a simple recommender on our data

CachingRecommender cachingRecommender = new CachingRecommender(new SlopeOneRecommender(model));

// for all users

for (LongPrimitiveIterator it = model.getUserIDs(); it.hasNext();){

long userId = it.nextLong();

// get the recommendations for the user

List<RecommendedItem> recommendations = cachingRecommender.recommend(userId, 10);

// if empty write something

if (recommendations.size() == 0){

System.out.print("User ");

System.out.print(userId);

System.out.println(": no recommendations");

}

// print the list of recommendations for each

for (RecommendedItem recommendedItem : recommendations) {

System.out.print("User ");

System.out.print(userId);

System.out.print(": ");

System.out.println(recommendedItem);

}

}

}

}

**Selection of datasets**

The datasets are selected based on whether it satisfies recommendation algorithm or not. It is a set of movies watched by a set of users mentioned by its movie Ids.

The data set I have chosen is posted in my Github link

**Mobile App/ Web Design**

This application front end that is the graphical user interface is created using jsfiddle under the restrictions of using of jquery 1.9.1 and jquery 1.3.01b

The features of this application are being mentioned above.

**Software Services**

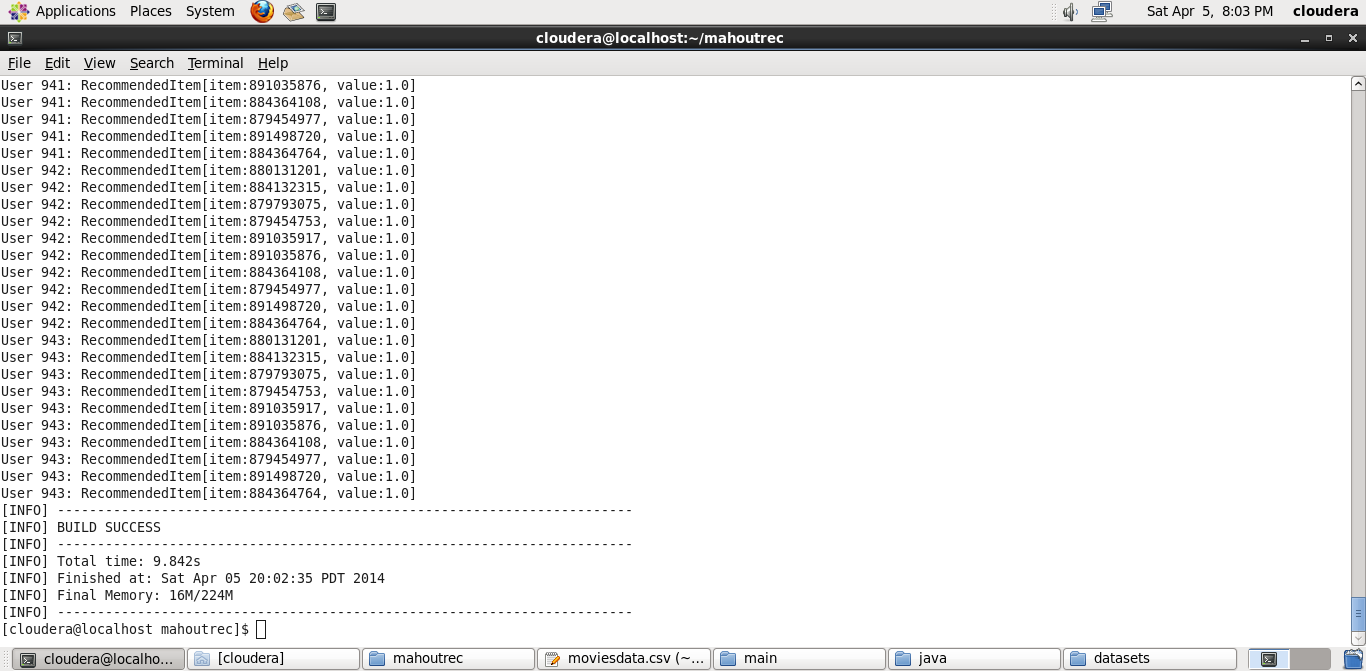
The weather services of the application are being implemented, where the user needs to provide the state code and the city name based on which the temperature of the city will be displayed.

**Mobile User Interface**

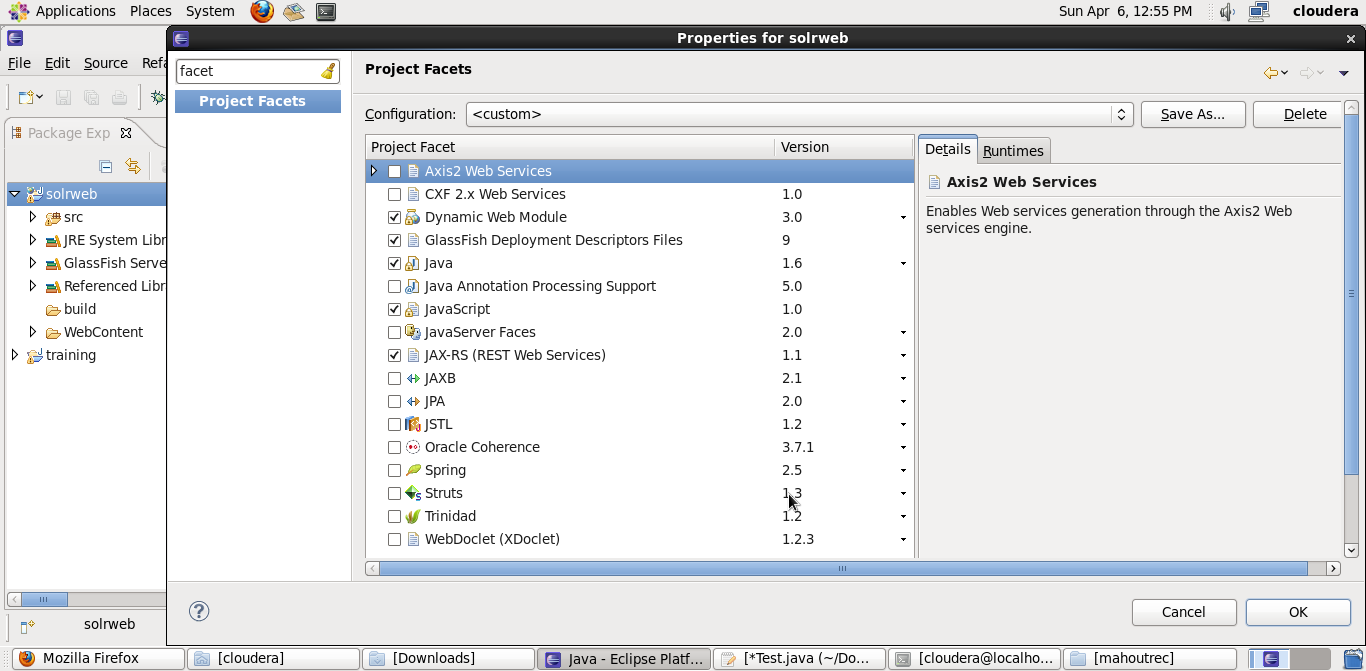
This is a mobile user interface where the interface is designed using the JSfiddle. It consists of 4 4 pages. One the home page , next the page

**Outputs**

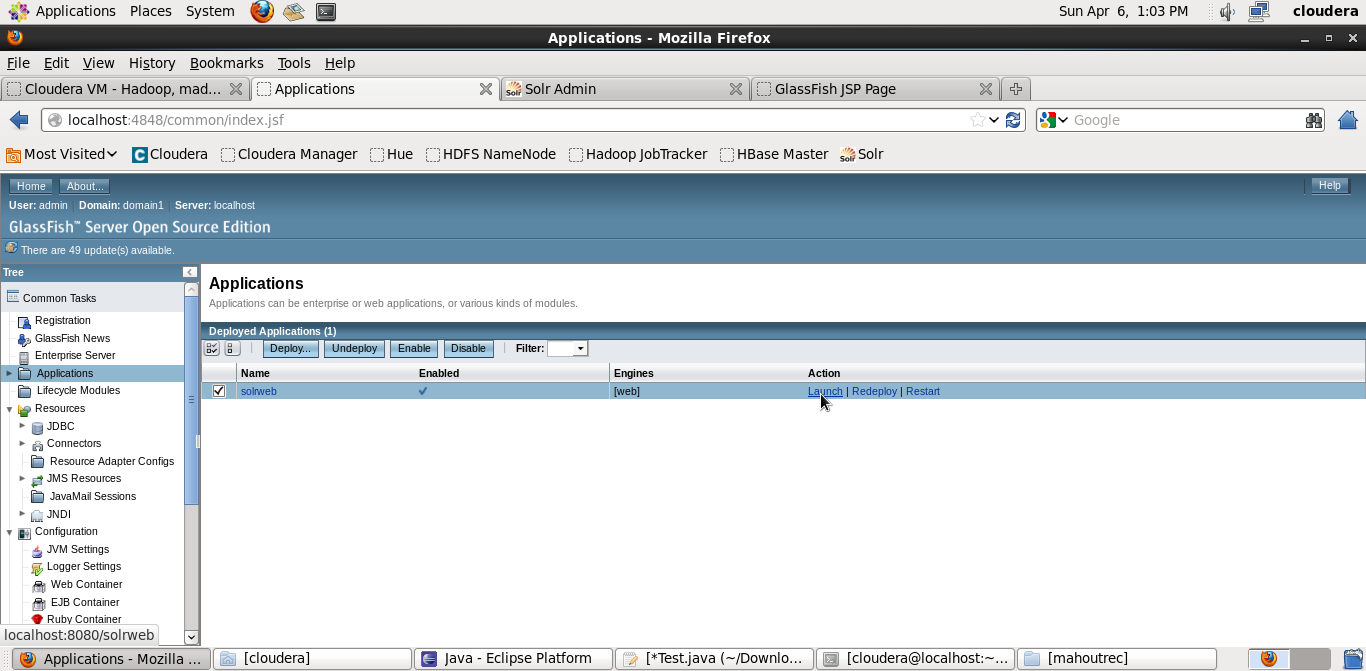
Firstly we need to convert the data sets with us into the JSON format. To convert the data we need to first run the recommender algorithm for it. This recommender algorithm is then saved in a text file which is then converted to the JSON format.



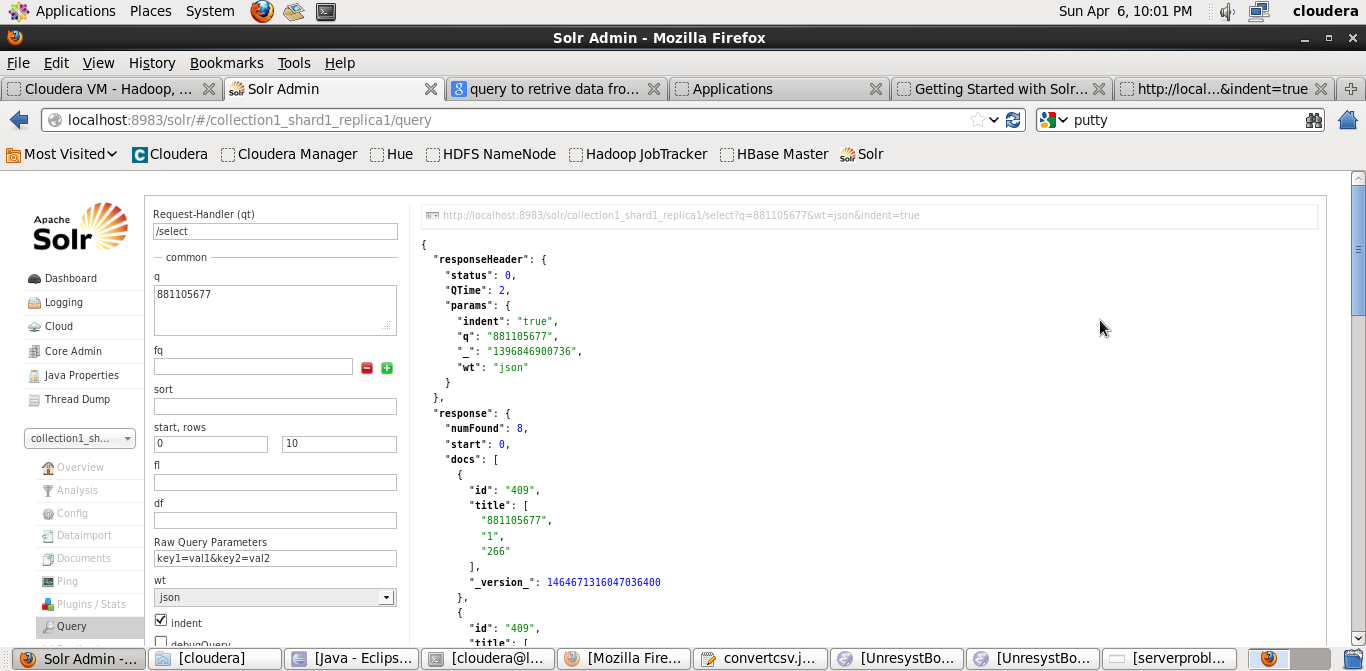
The next step is to work with the eclipse where we need to check whether the glass fis server is installed and properly working and then check the solr srvices. Once done with the above criteria we need to create a web application project in eclipse and then export a web service which is the SOLR web service.

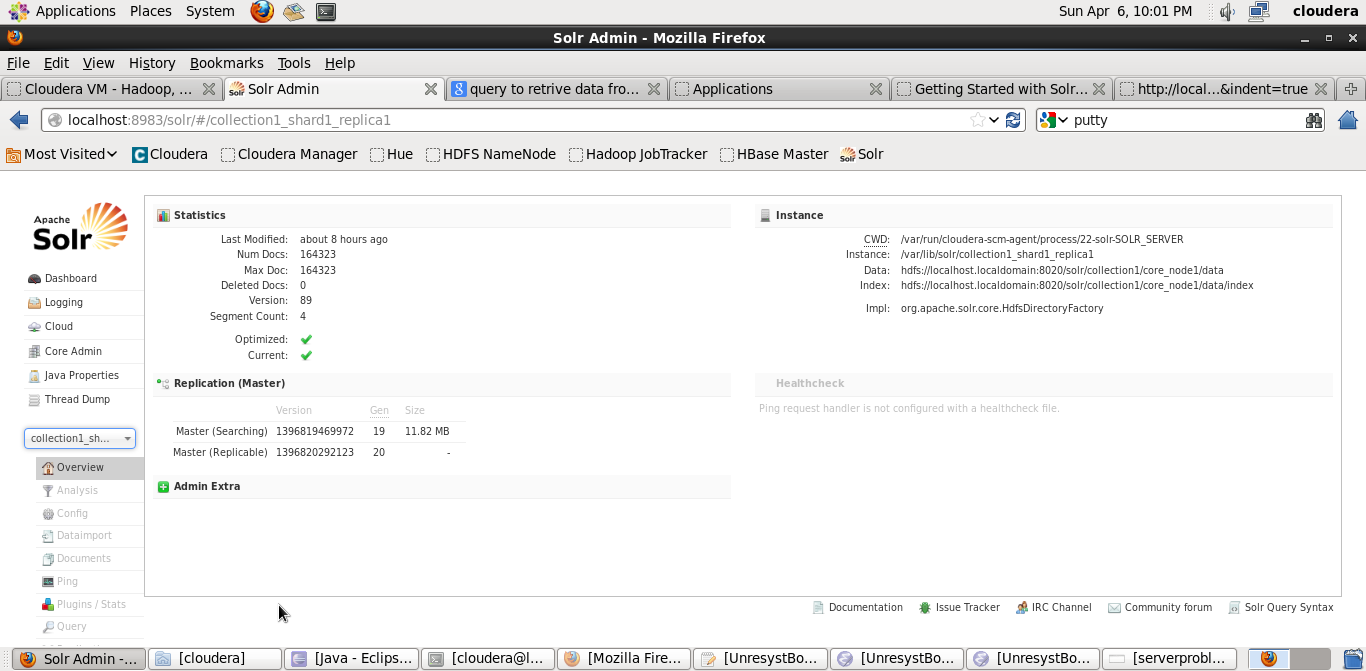


We need to launch the Solr web service to insert the data. After the following steps we can now insert data in the Solr either using the query or the Solr web service. I have used the Solr web service created to insert the data.



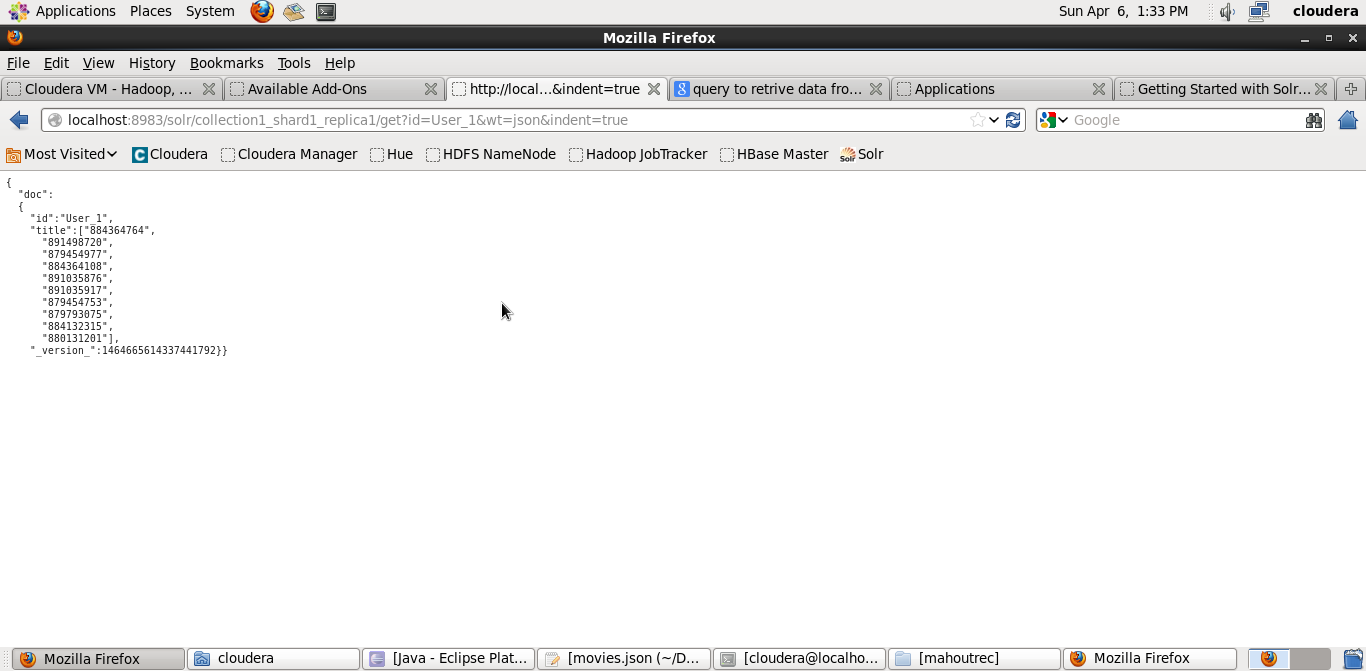
Once done with launching the Solr web service the next step is to insert the data which is converted in JSON format.



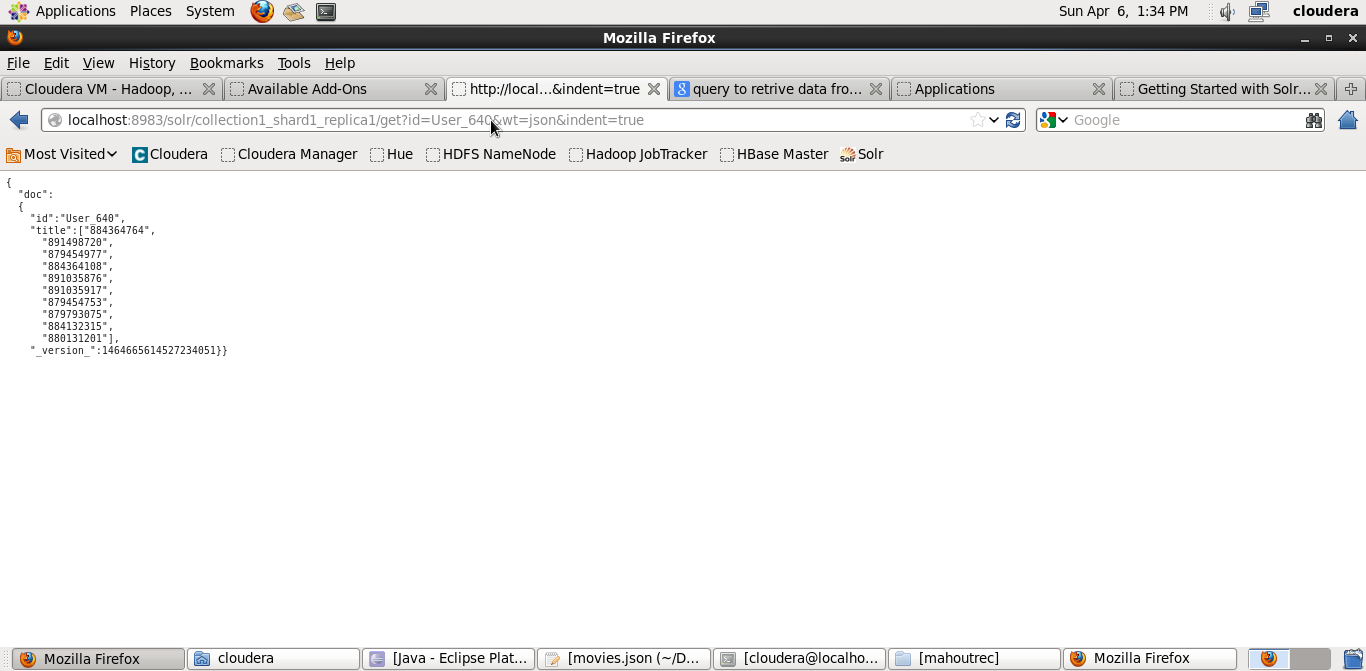


Here the back end part is done where the data is being inserted and we can now retrieve it also.

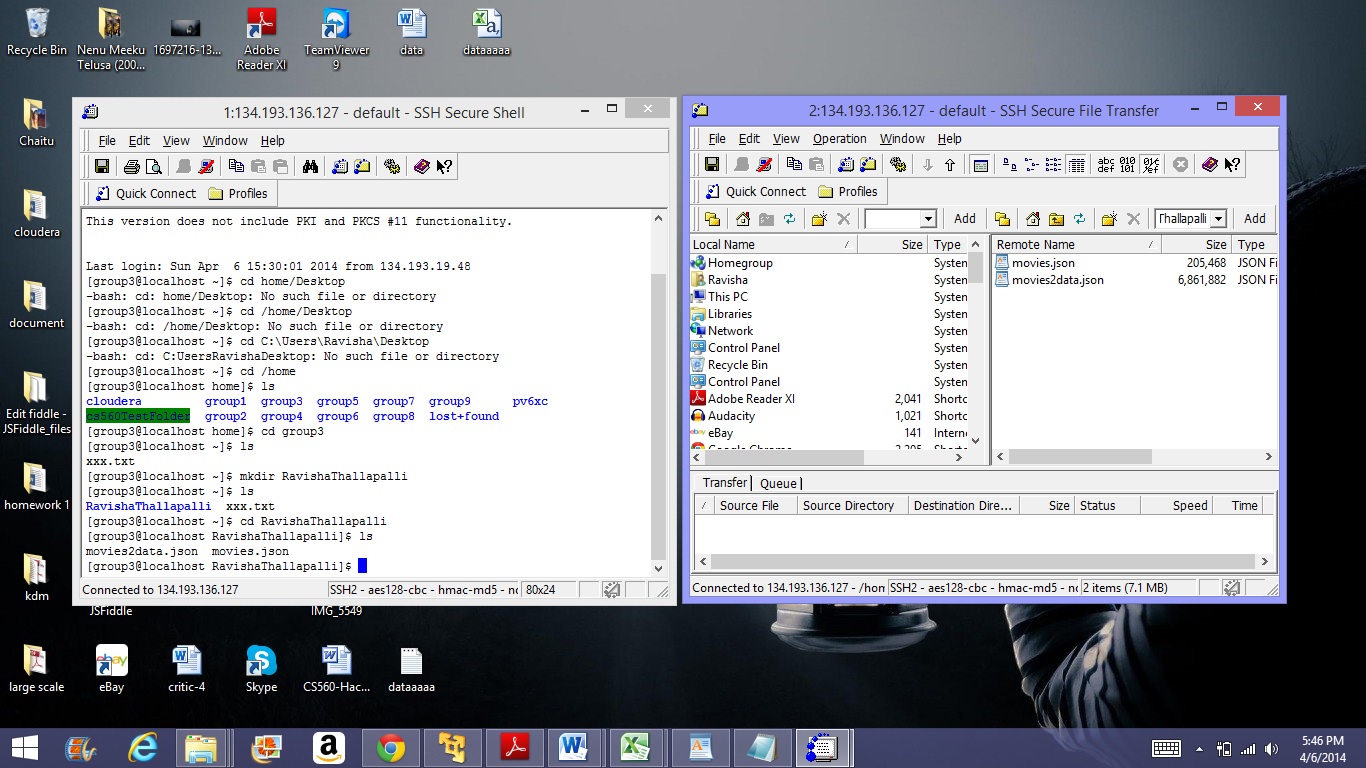
Recommendation Data of user 1



Recommendation data of user 648

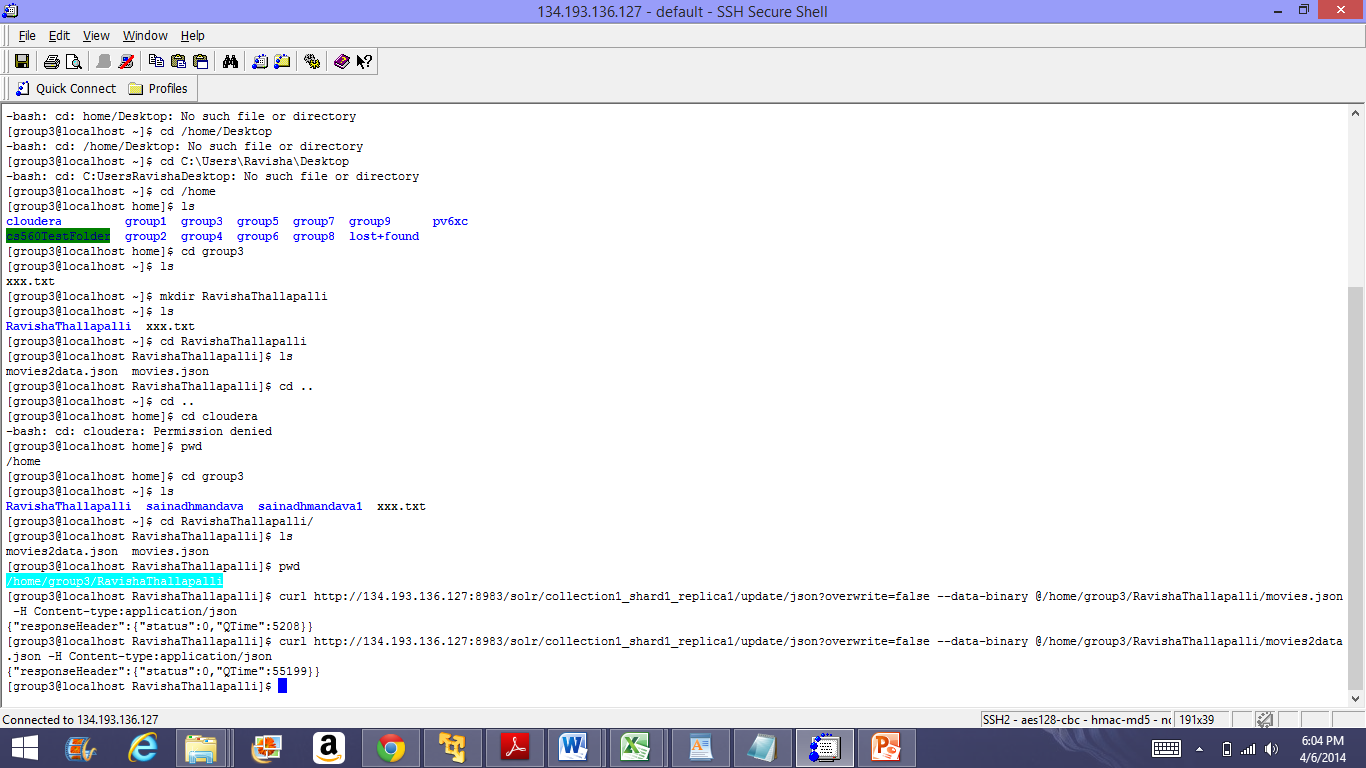


Now the next step is to launch them in the server, till now it was local server. We can do this by using putty from home.

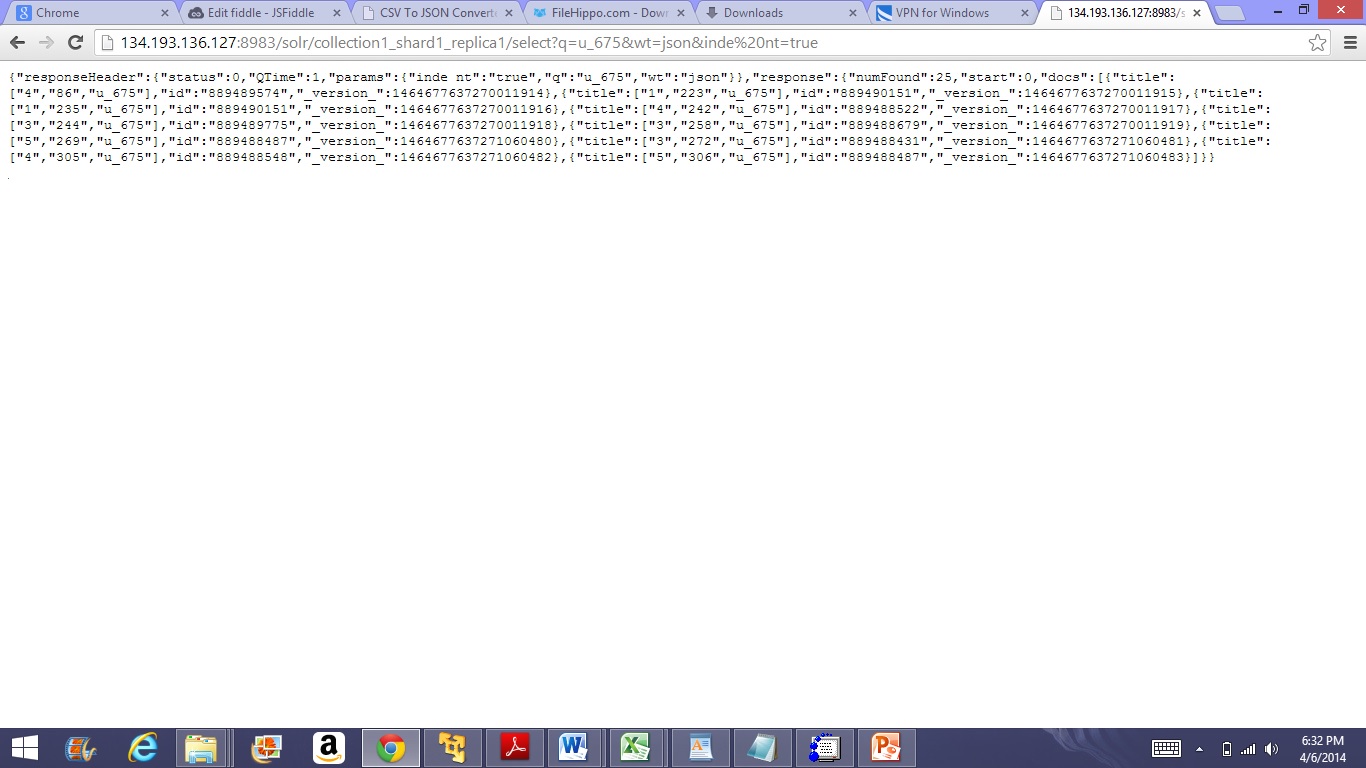


The two data converted to JSON files are being uploaded to the Server.

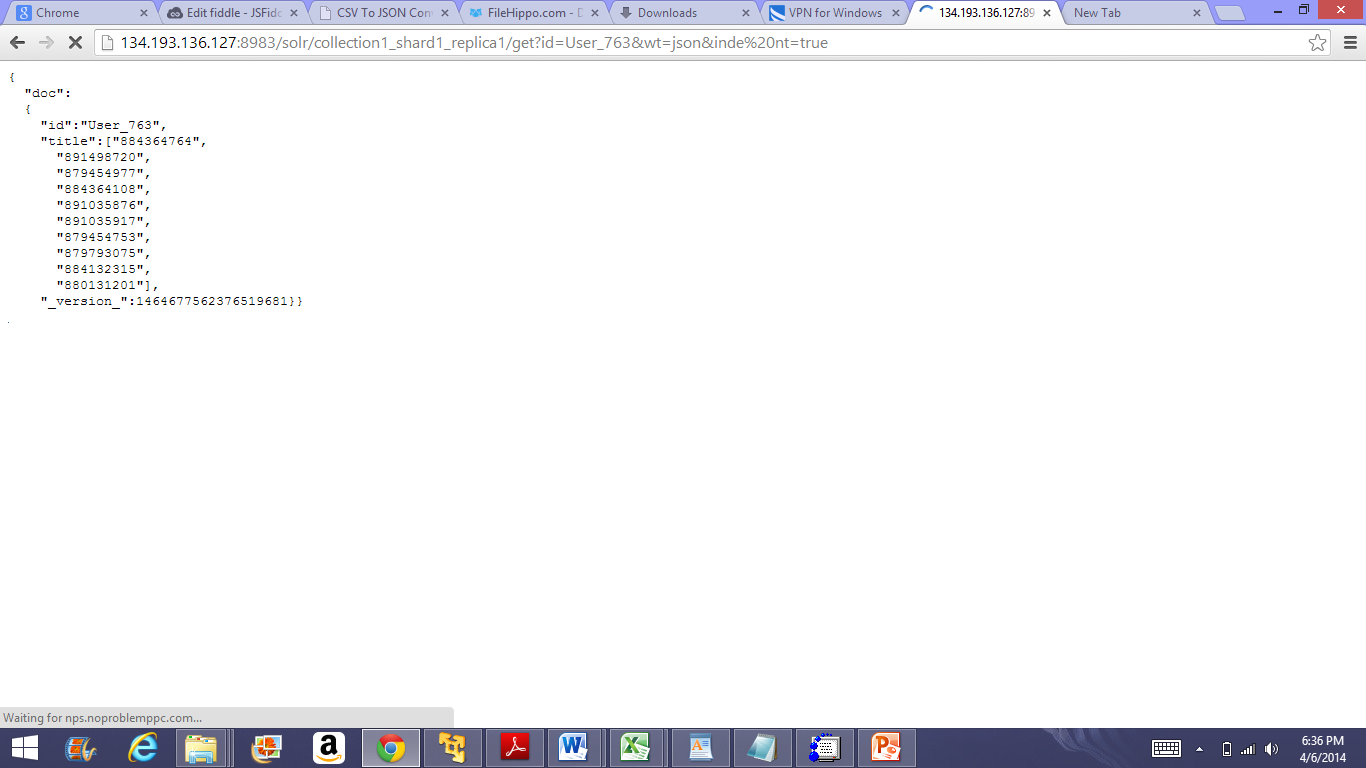
Data being uploaded to the server.



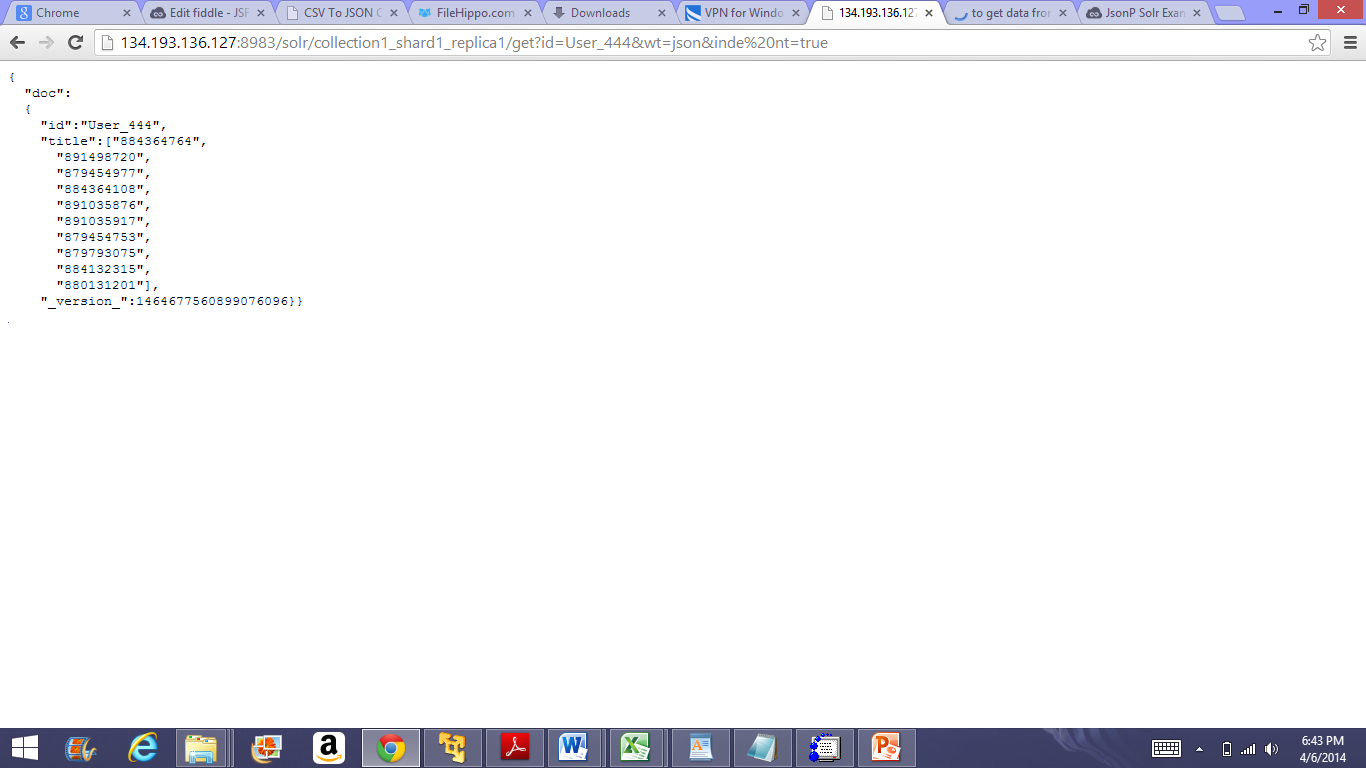
Data being retrieved from the server



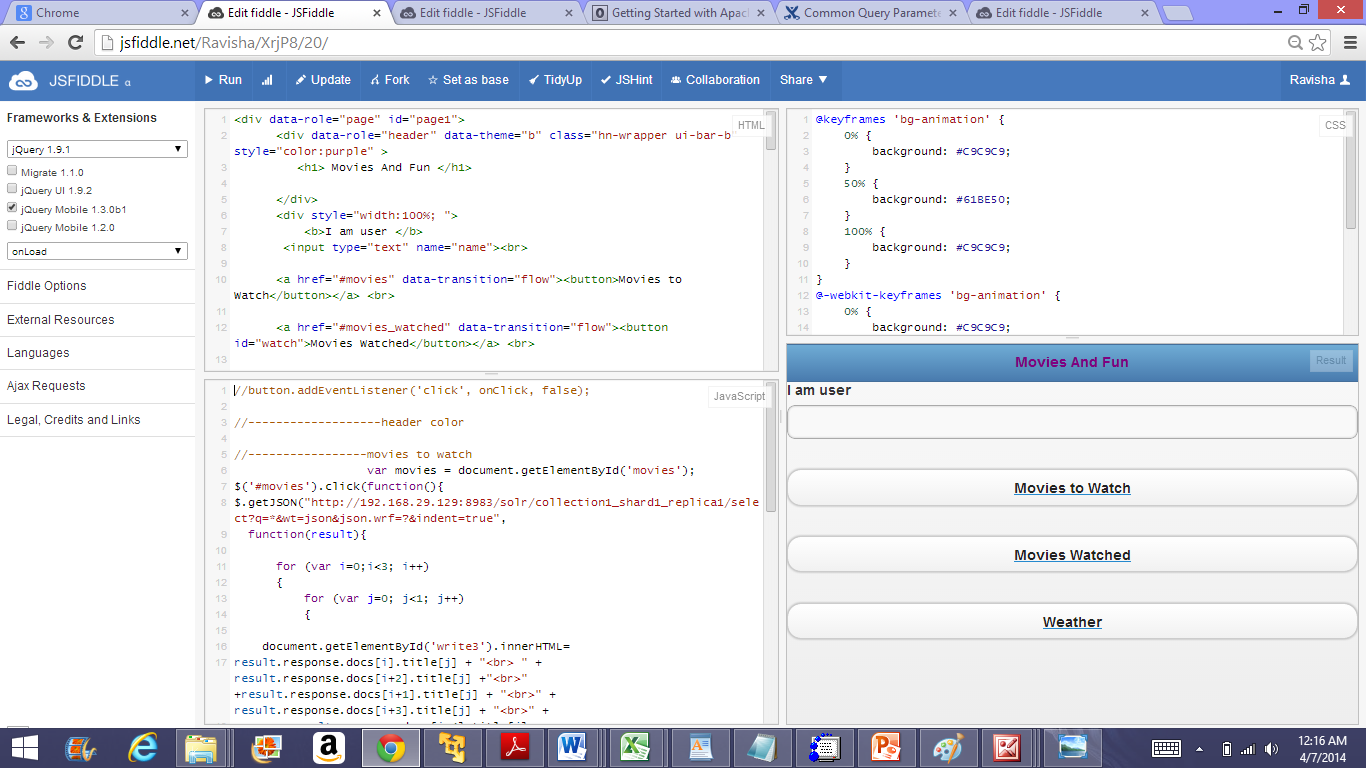
Output for the user 763 from server



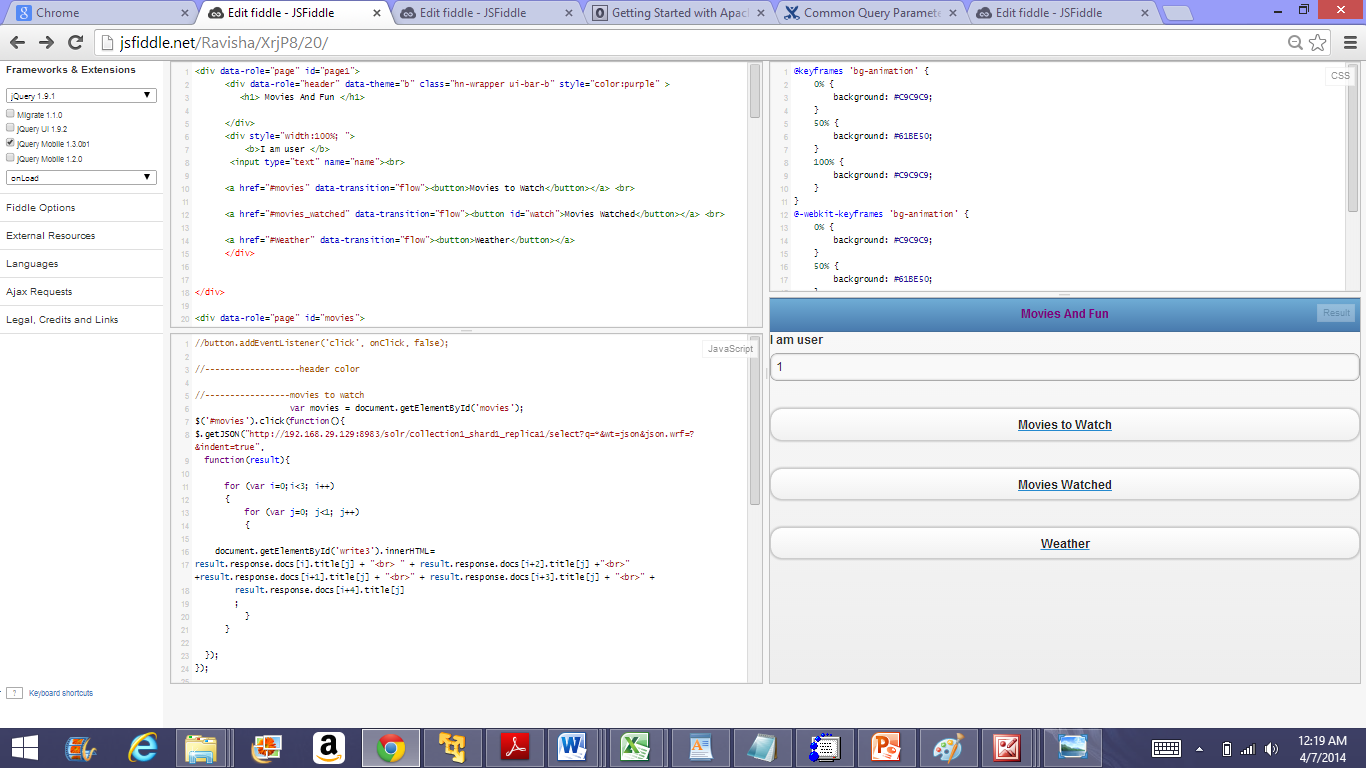
Output data from server for 444 user



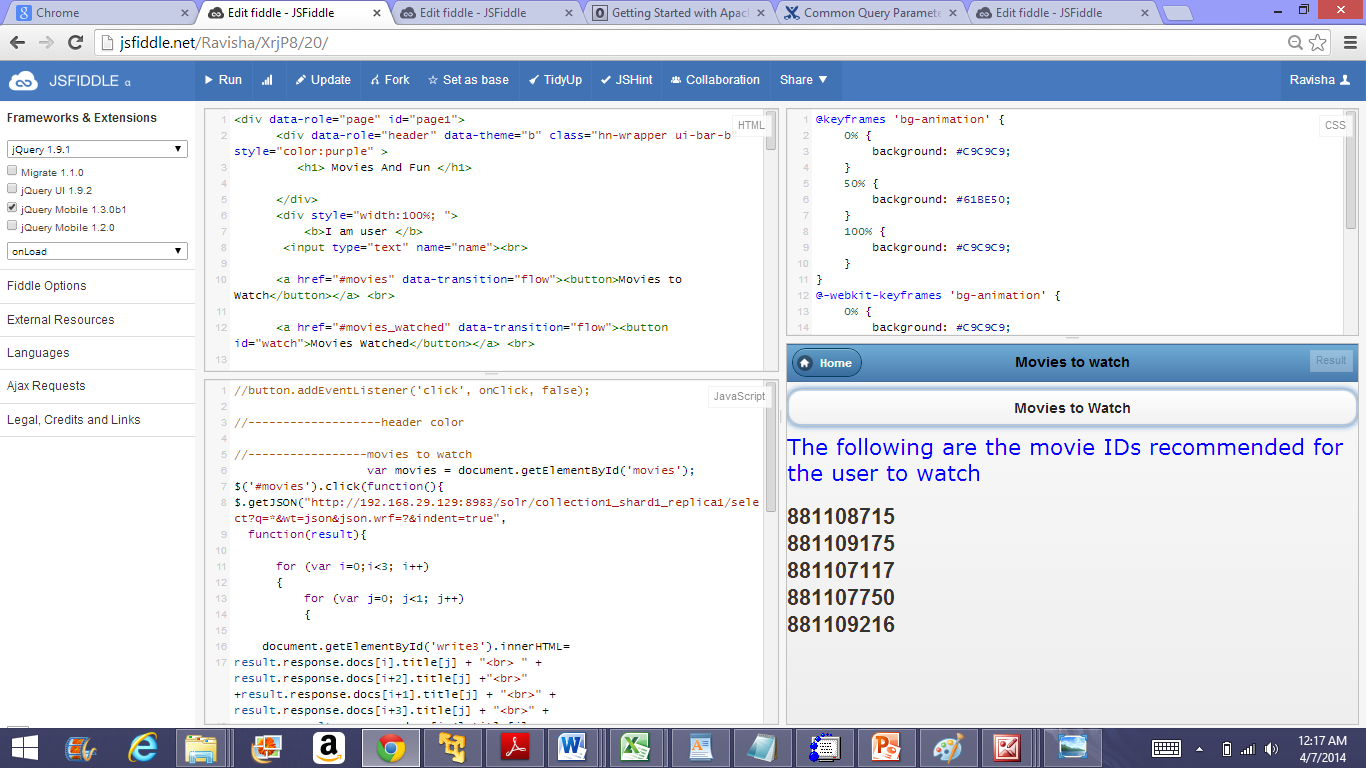
The user interface of the application goes as follows which is created using jquery, html and css



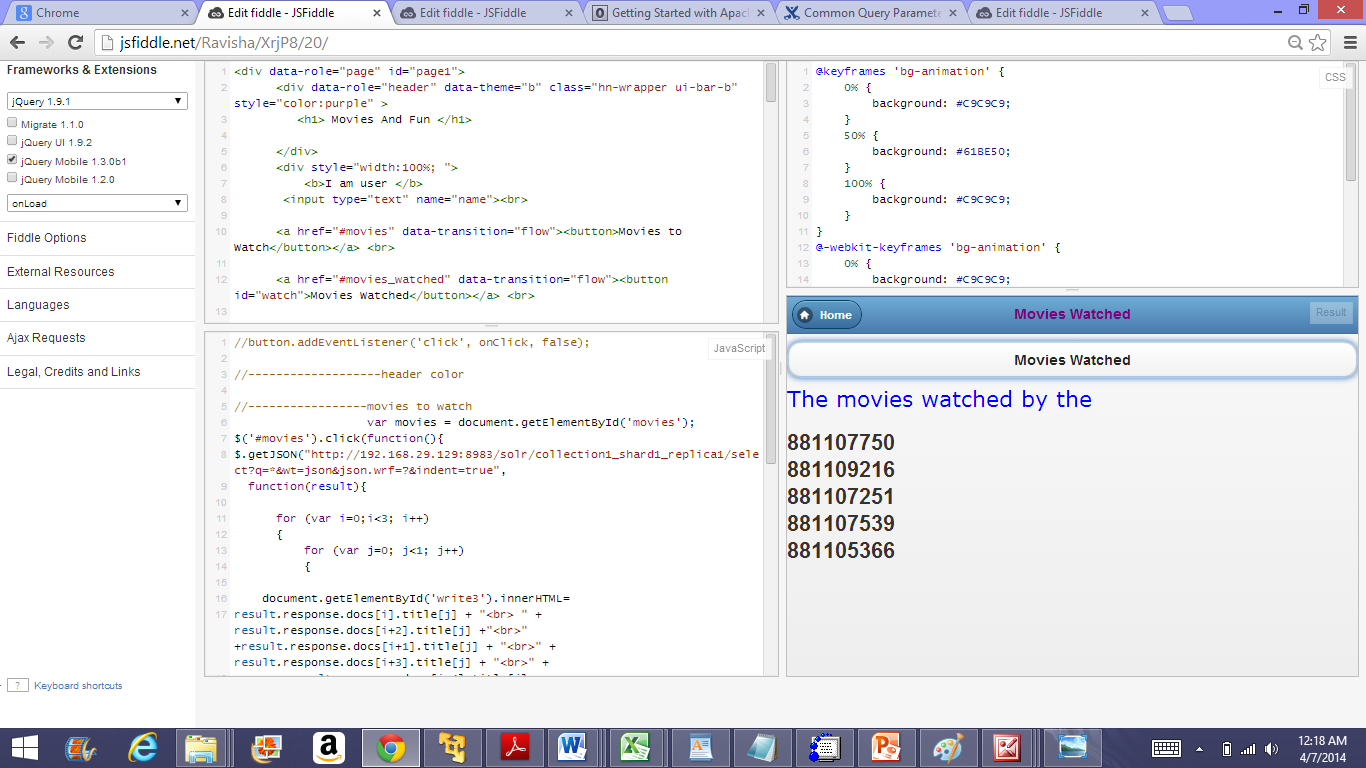
Next the user Id has to be entered



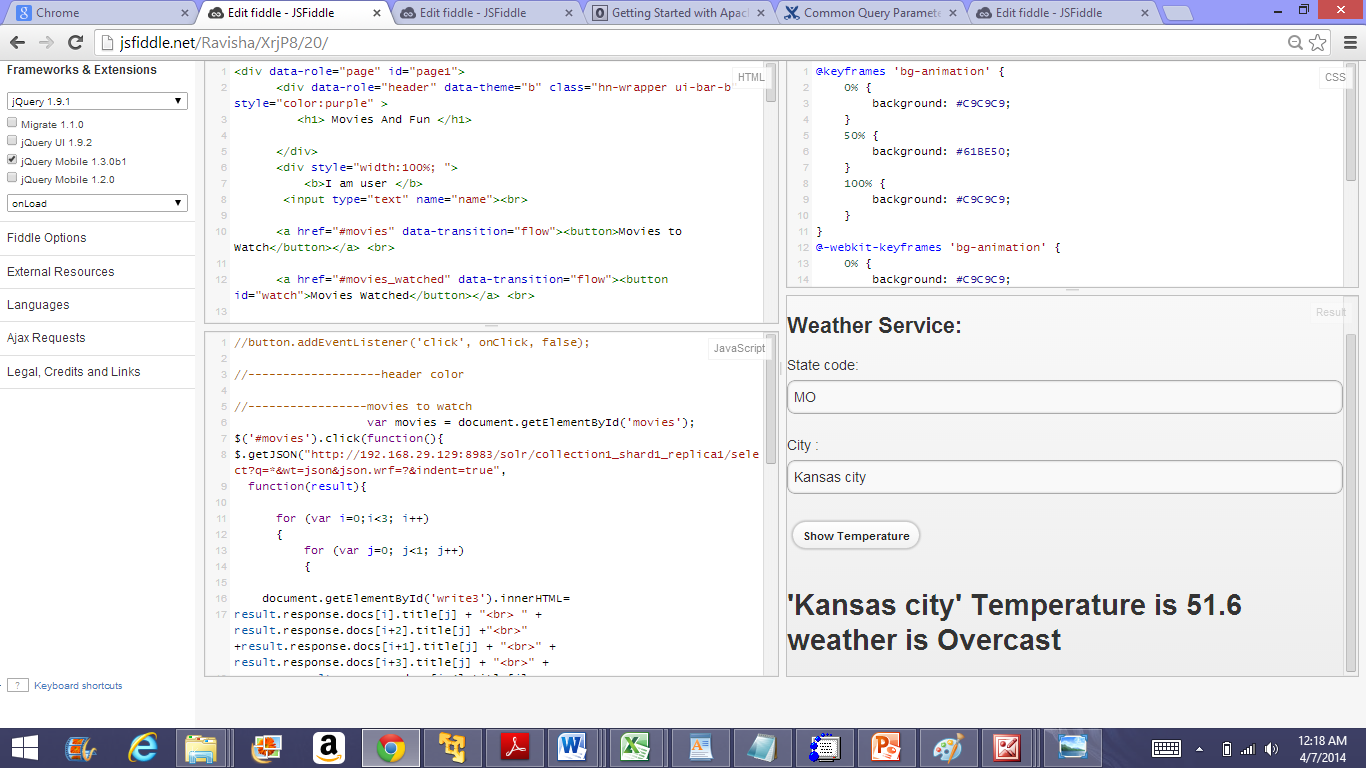
Now the movies recommended for User 1 are displayed once we press the movies to watch button



Next we can also output the movies watched by the user 1 as per the input given



The weather services can also be viewed



The web service I have used is the weather service, which shows the temperature based on the state code and city name,

**My Github link**

<https://github.com/rt6f8/Examm>

**References**

<https://cwiki.apache.org/confluence/display/solr/Common+Query+Parameters>

<http://anchetawern.github.io/blog/2013/06/15/getting-started-with-solr/>