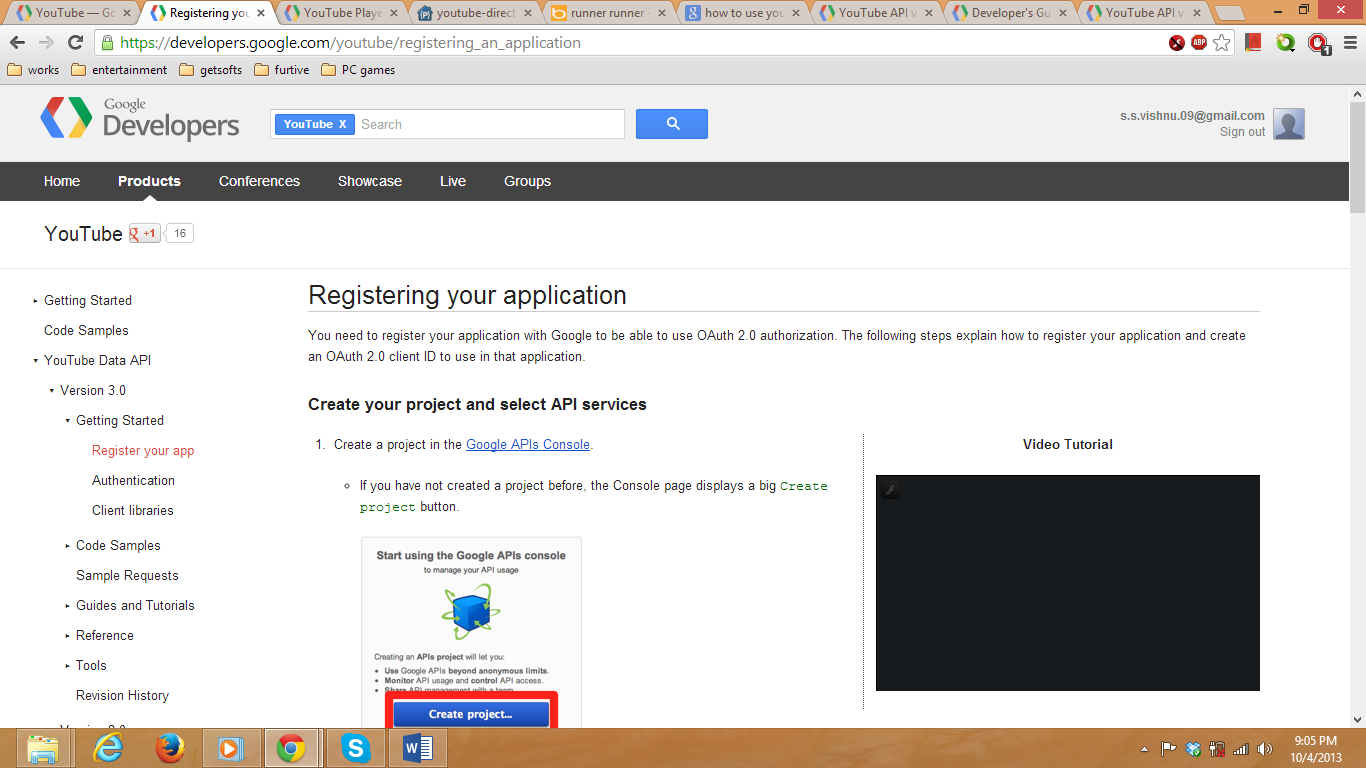
**First Increment Report   
Due Date: October 4 (F) 2013, 11:59PM**

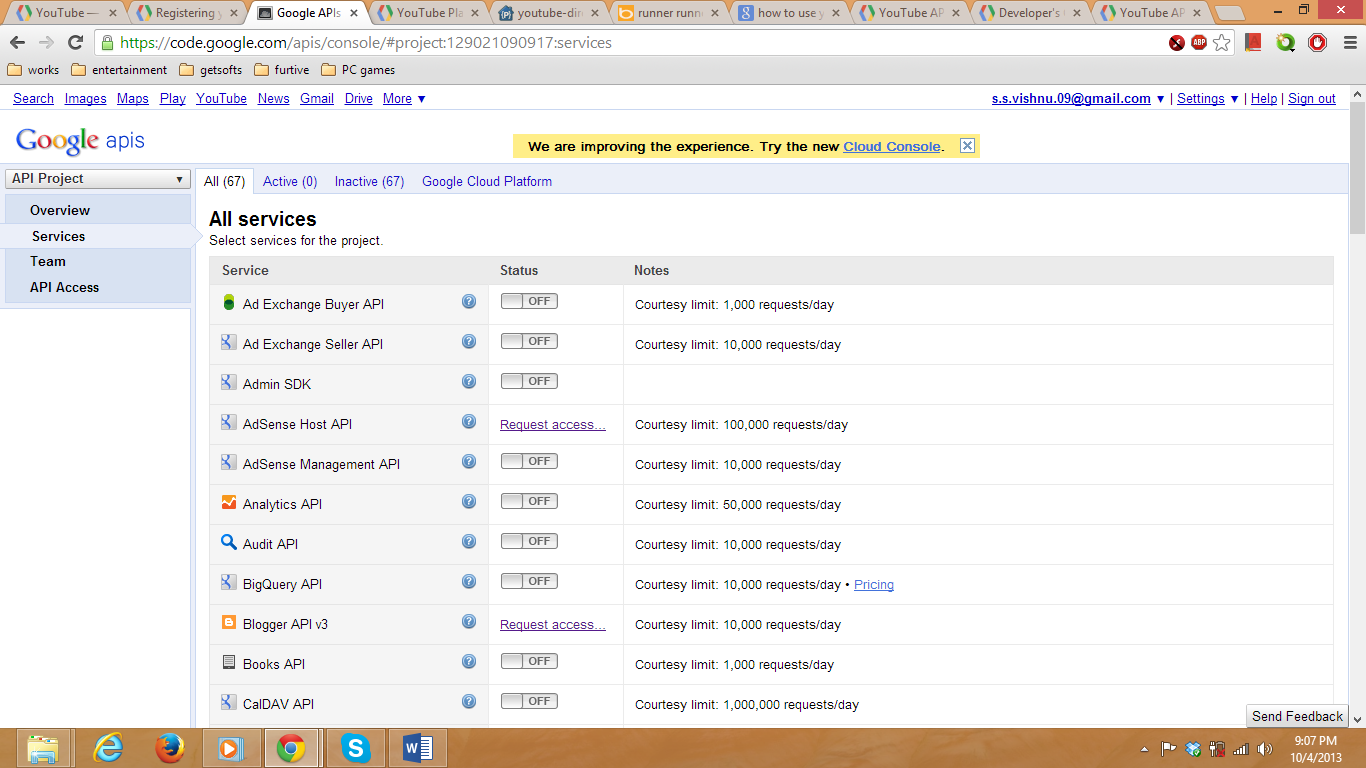
**IMPORT EXISTING SERVICES/API**

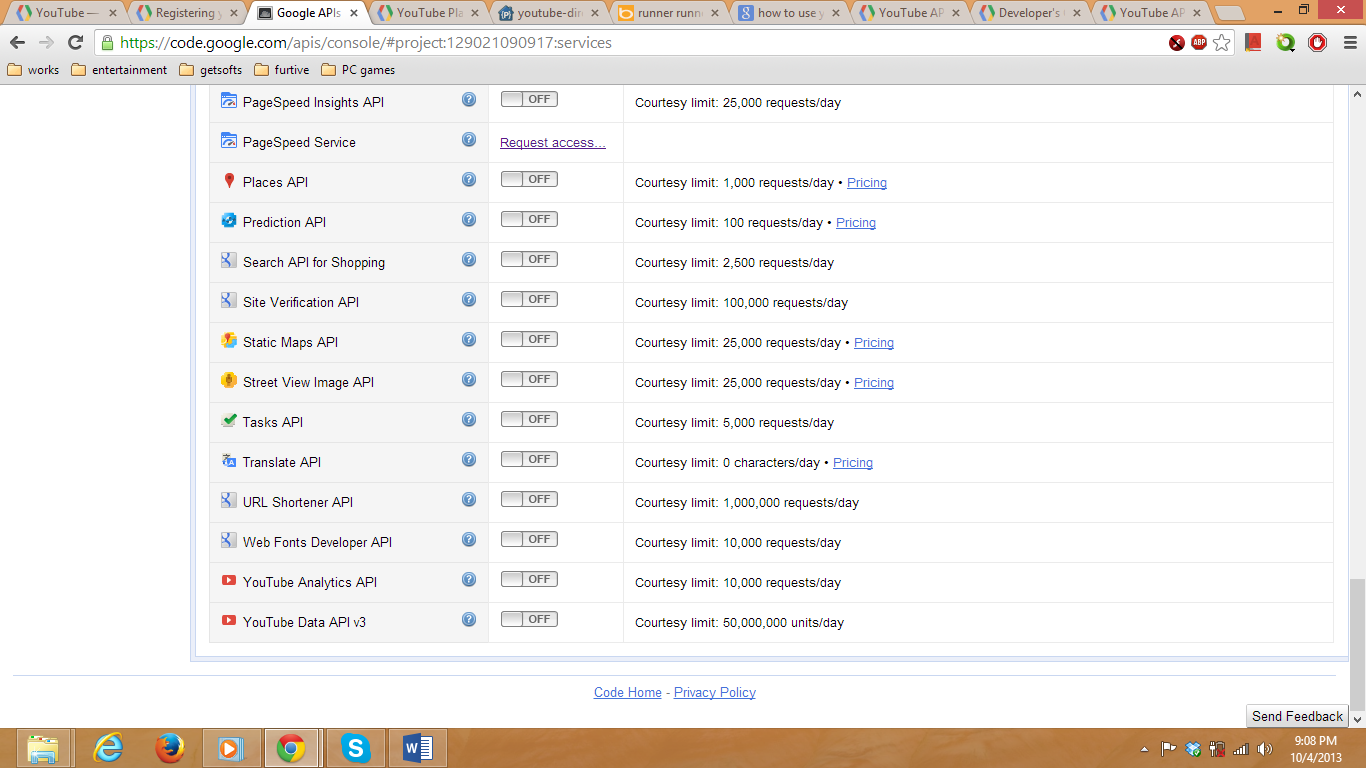
Using YouTube API for accessing, searching, and filtering of results are done following the steps:

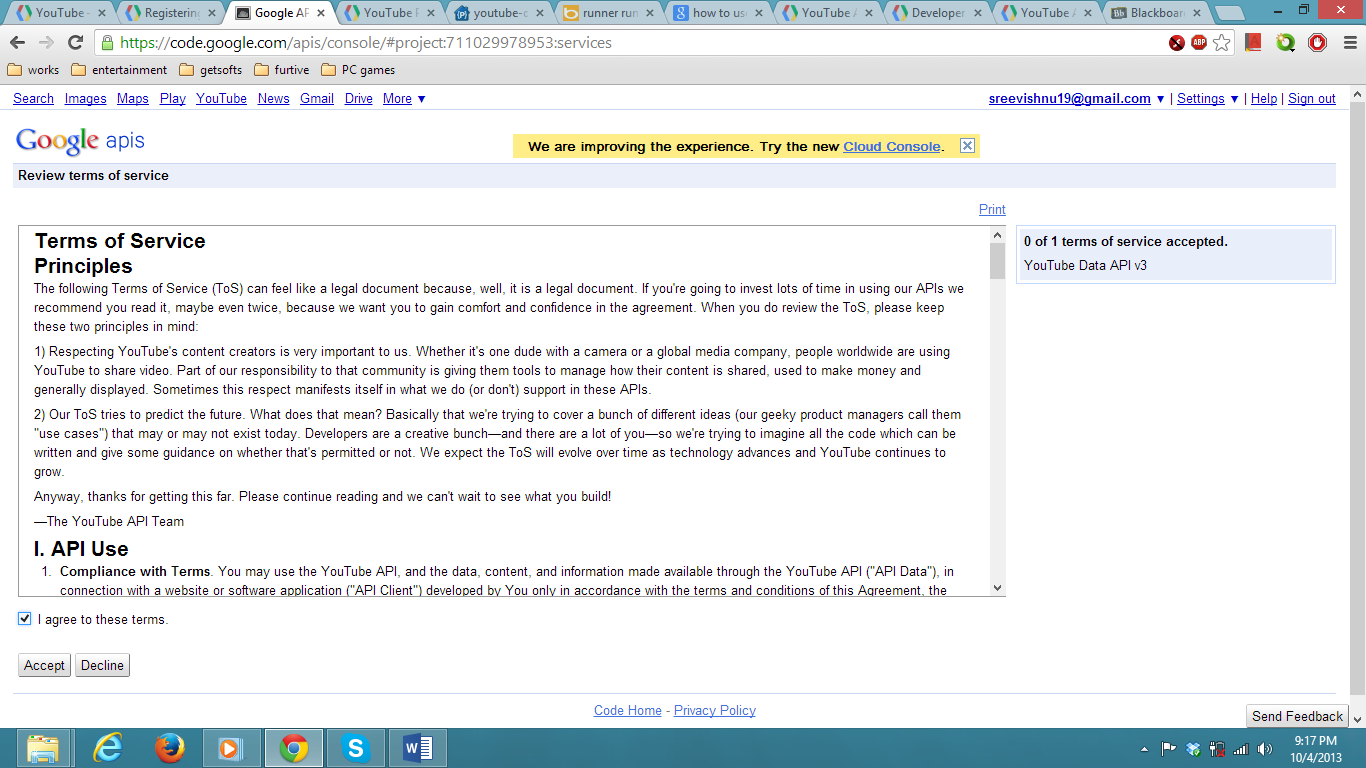
1. Registering the SmartTube Application at Youtube Developer.

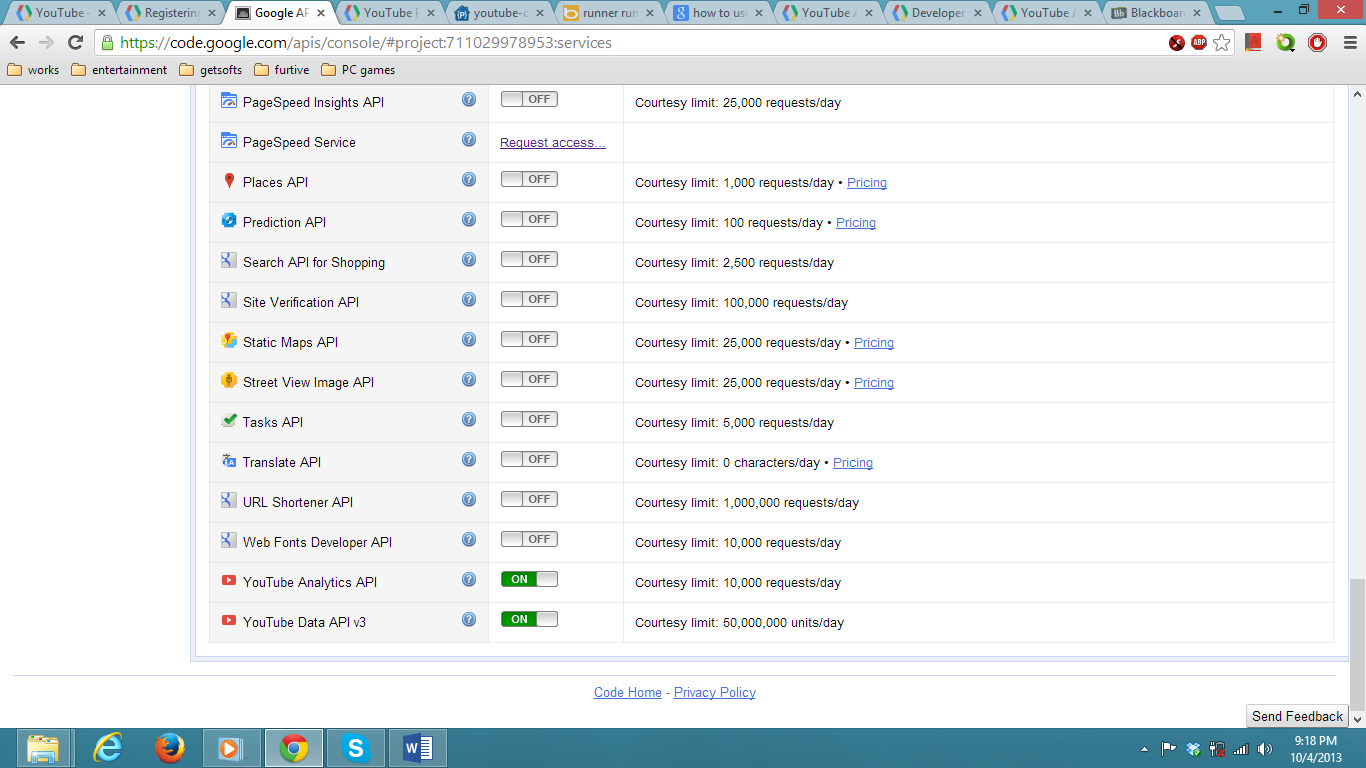


2. Click on Google APIs Console:

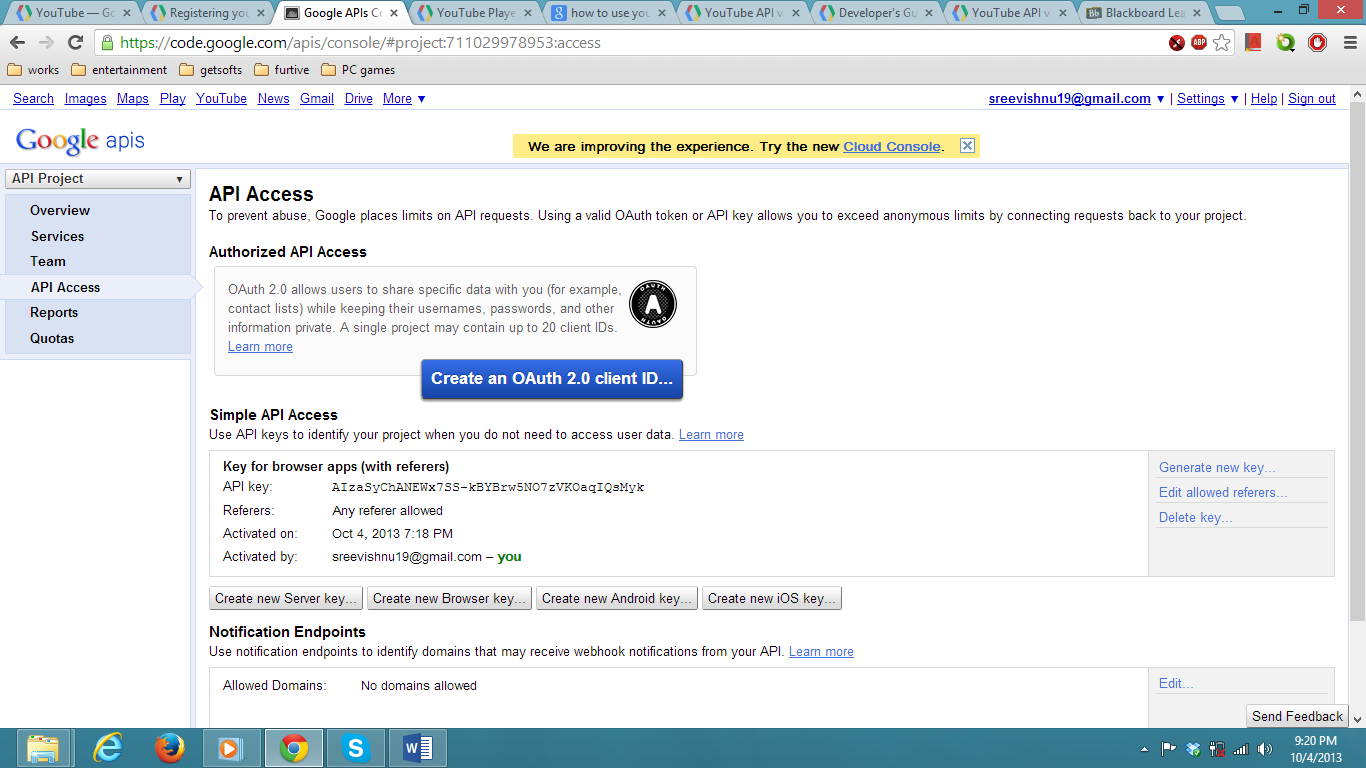
Click on Youtube DATA API

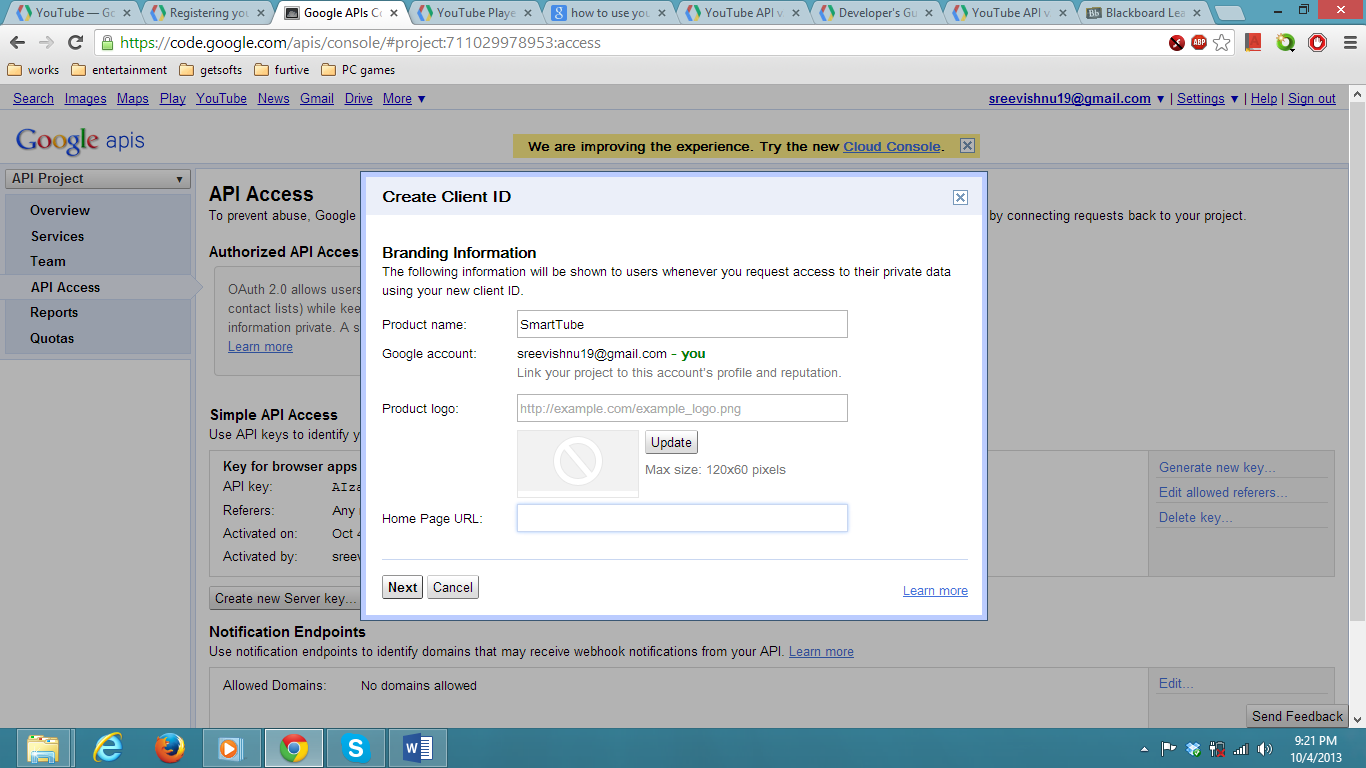




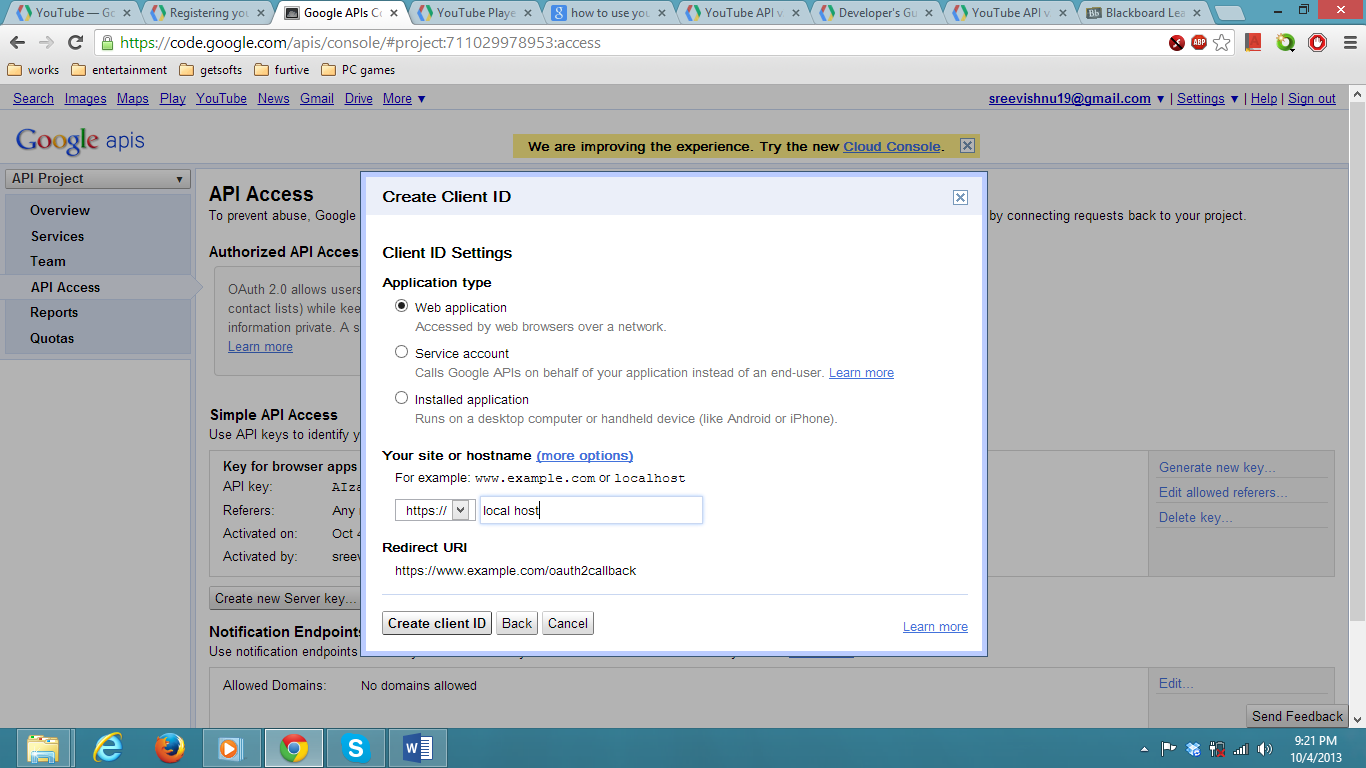


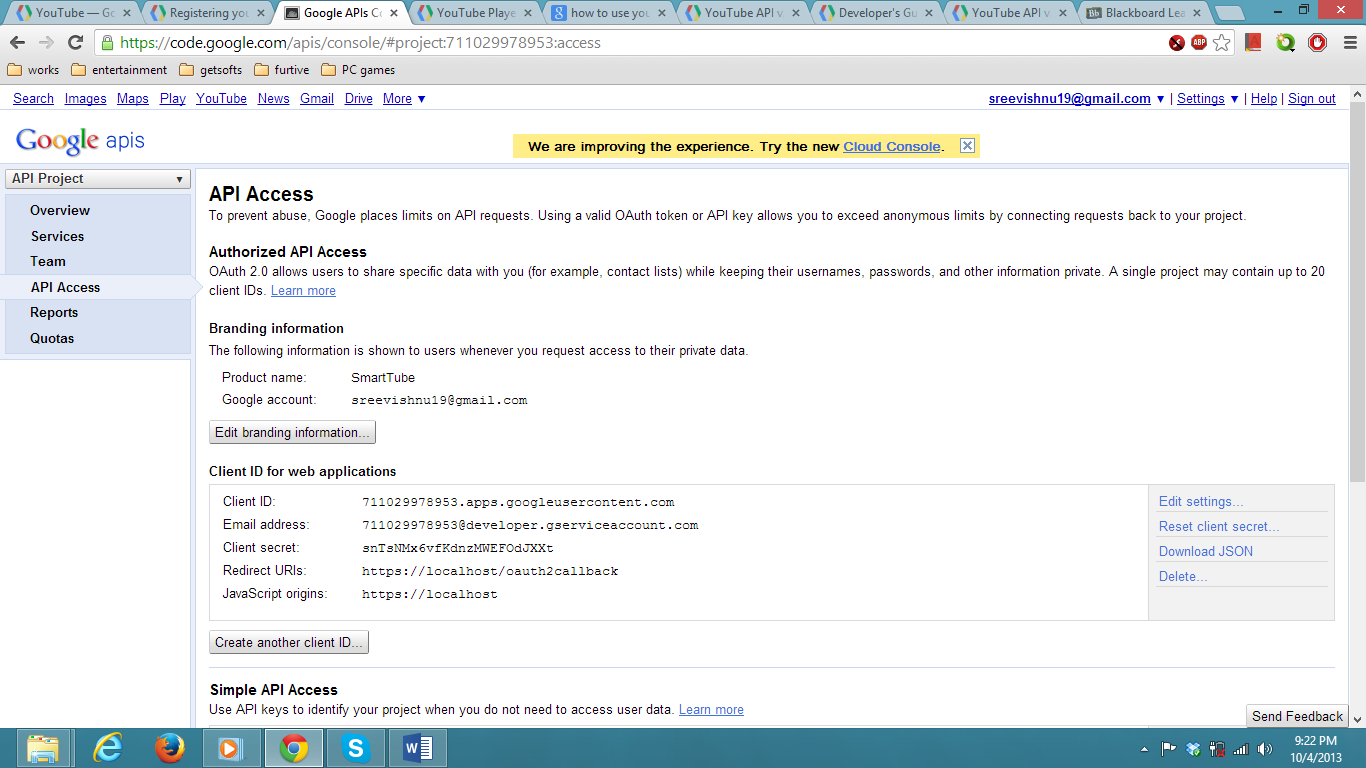
Click on Create an OAuth 2.0





Click on Create Client ID





We use the Simple API with the key provided access to use the youtube services and resources.

The following links are to be used as reference:

<https://developers.google.com/youtube/2.0/developers_guide_protocol_api_query_parameters>

<https://developers.google.com/api-client-library/javascript/>

<https://developers.google.com/youtube/v3/code_samples/>

<https://developers.google.com/youtube/v3/code_samples/java#search_by_keyword>

<https://developers.google.com/youtube/v3/code_samples/javascript>

<https://developers.google.com/youtube/v3/guides/searching_by_topic>

**DETAIL DESIGN OF SERVICES**

Service description (WSDL, Restful services): We are using Glassfish restful server to get these services done. We are having other three services such as

1) Search service which searches the videos according to the data filled by the group members.

2) View results service: It views the result from the sorting done according to the priorities of the group members and the views in the YouTube.

3) Registration service: It registers the group member when the user enters the data such as first name, last name, group name and group key.

Sequence diagram (using Visio): 

Class diagram (using Visio): 

Design of Mobile Client Interface: The screenshots of mobile client interface is present below

**First page of mobile client interface code**:

<!DOCTYPE html>

<html>

<body>

<body style="background-color:red;">

<h2 style="background-color:pink;">WELCOME TO SMART TUBE</h2>

<p style="background-color:yellow;">You can watch videos of your interest here.</p>

<p style="background-color:skyblue;">in order to proceed click on continue.</p>

<button type="button">Continue!</button><br>

<img style="-webkit-user-select: none; cursor: -webkit-zoom-in;" src="http://rtv\_news.s3.amazonaws.com/wp-content/uploads/2013/09/MJ-wallpapers-michael-jackson-31128130-1600-1200.jpg" width="200" height="100">

</body>

</html>

**Registration Page code**:

<!DOCTYPE html>

<html>

<body style="background-color:pink;">

<h1><i>REGISTRATION PAGE</i></h1>

<form>

First name: <input type="text" name="firstname"><br><br>

Last name: <input type="text" name="lastname"><br><br>

group name: <input type="text" name="groupname"><br><br>

group key :<input type="text" name="group key"><br><br>

<button type="button">Register!</button><br>

</form>

</body>

</html>

**Login Page Code**:

<!DOCTYPE html>

<html>

<body style="background-color:skyblue;">

<h1><i>LOGIN PAGE</i></h1>

<form>

group name: <input type="text" name="groupname"><br><br>

group key :<input type="text" name="group key"><br><br>

<button type="button">login!</button><br>

</form>

</body>

</html>

**Data that should be entered by the group members page code**:

<!DOCTYPE html>

<html>

<body style="background-color:brown;">

<h1><i>Fill your preferences</i></h1>

<form>

Language: <input type="text" name="language"><br><br>

genre :<input type="text" name="genre"><br><br>

artist:<input type="text" name="artist"><br><br>

<button type="button">submit!</button><br>

</form>

</body>

</html>

* + Design of Map Reduce Algorithm or usage of any other Hadoop Ecosystem: Here we are designing a map reduce algorithm which will sort according to the preferences given by the group members. After the videos come according to their preferences it has to again sort according to the number of views in the YouTube.

Design of Data: Data is already present in the YouTube. We are using that pre-defined service of YouTube. So, no need of design of new data.

**IMPLEMENTATION**

* Implementation of Services:

The source code implementing the map reduce algorithm is placed on the cloud instance of the IBM cloud. All the Restful services Register, Search, View Results of the search are also deployed on the instance using a glassfish server which is previously installed on the cloud instance.

* Generate Your Datasets:

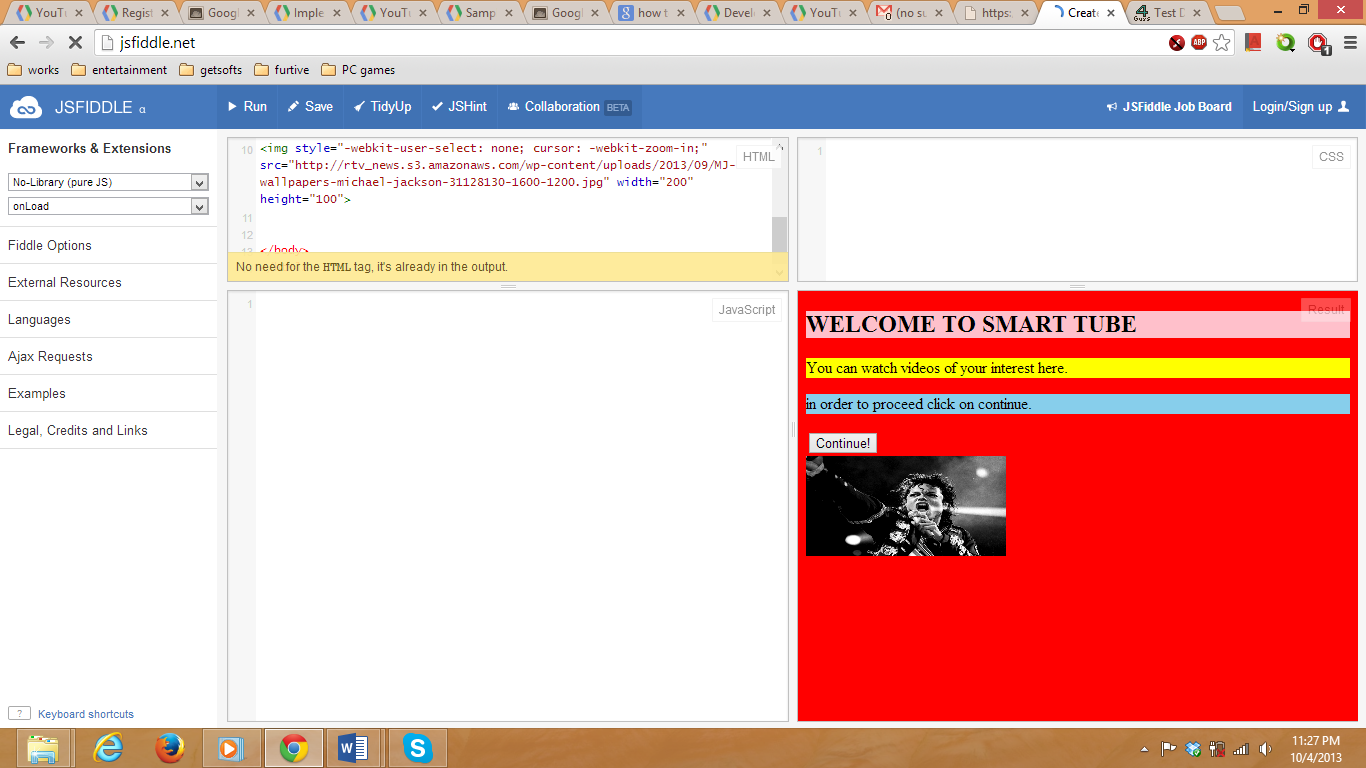
The data is obtained from the YouTube using the YouTube API and the data is then processed using the map reduce algorithm we implement on the cloud instance.

* Implementation of Mobile Interface:

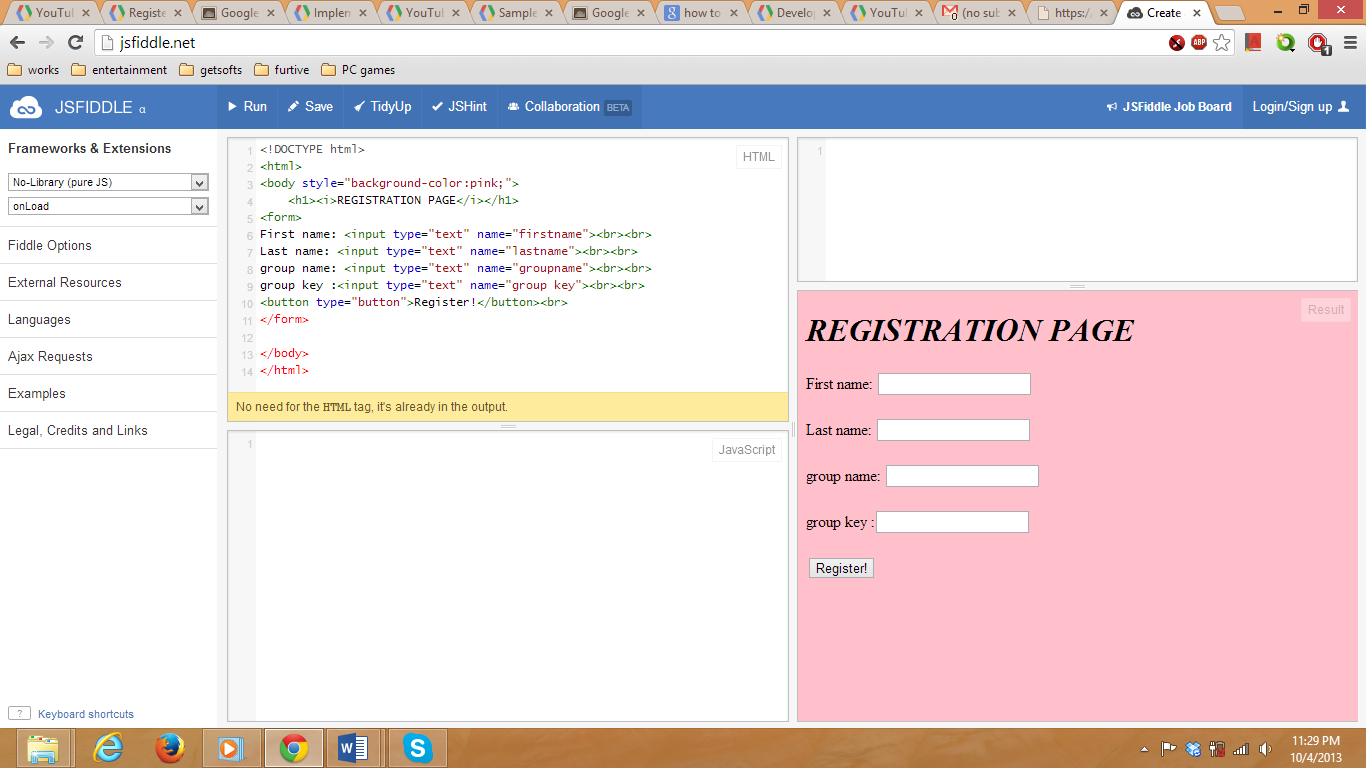
The mobile interface is implemented using html, css and jquery and with four main pages in interface as shown in the screen shots. The pages are used for registering, login and Setting preferences and finally searching and viewing the results.

Screenshots:

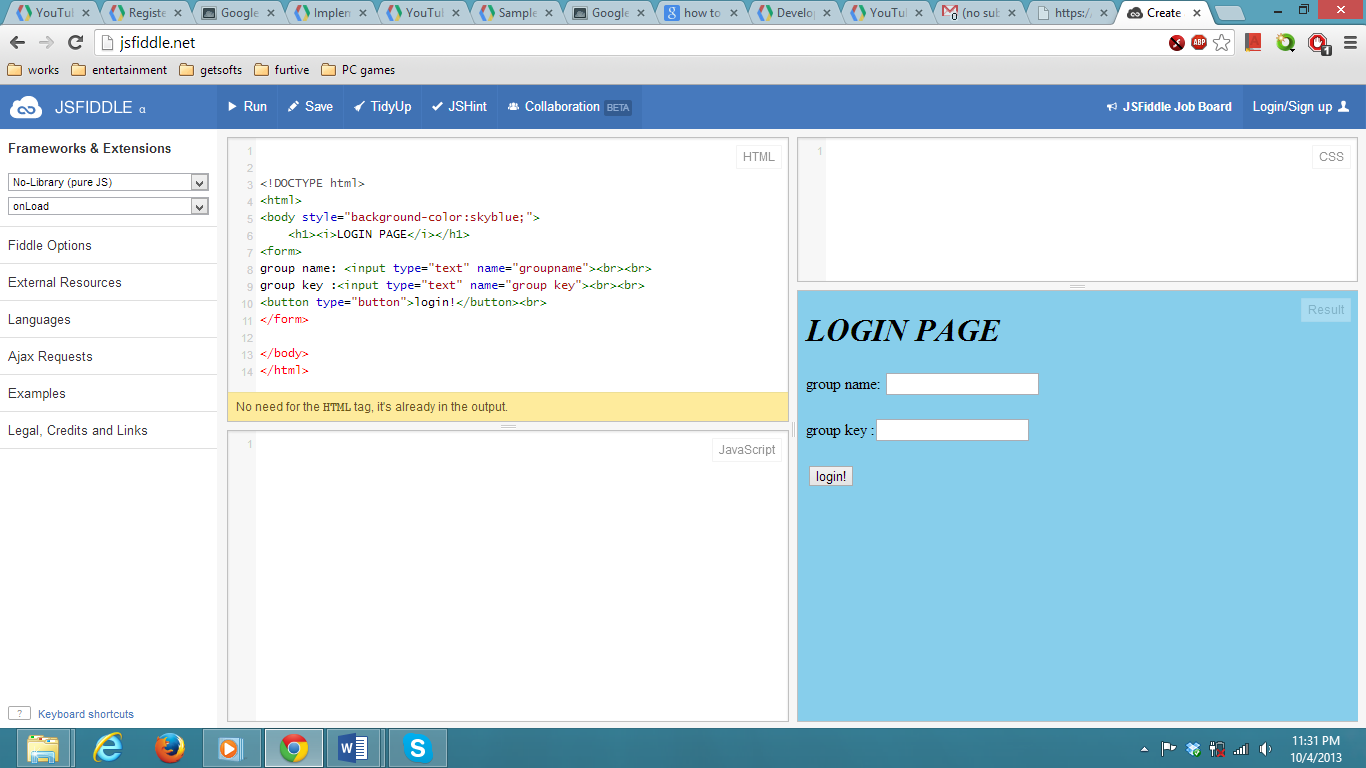
Page 1: Intro

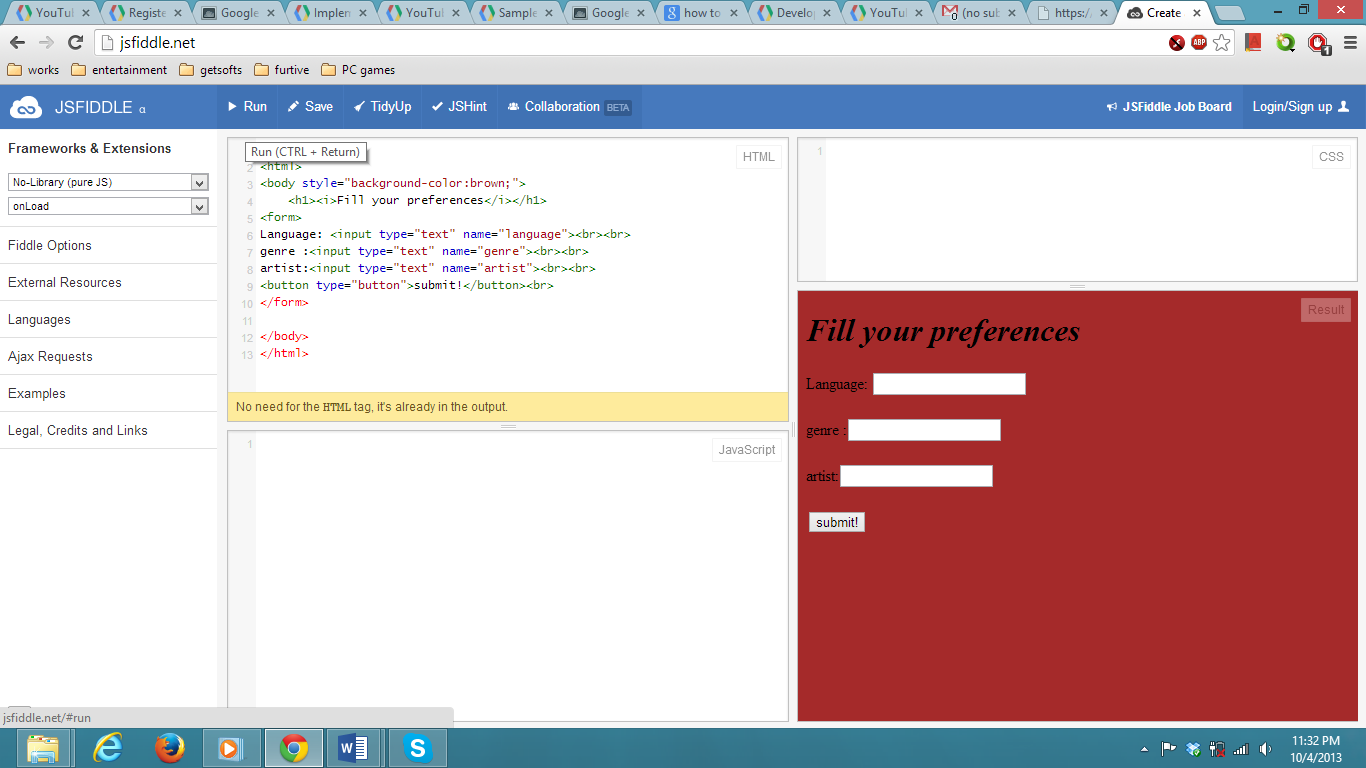


Page 2: Registration



Page 3: Login



Page4: Set Preferences 

**TESTING:**

Each of the pages in the interface in run and checked for errors individually with dummy values before implementing the services.

Each service, register service, search service and view results service are tested individually before connecting with each other by using different values. To test the register service the usernames and passwords are entered and checked whether they are stored in the database. The search service is tested by providing the preferences directly into the code without arguments and tested for the results. View service is tested based on the results of the algorithm.

**DEPLOYMENT:**

The interface was deployed on the cloud instance using the war file.

Cloud URL:<http://vhost0057.site1.compute.ihost.com:18080/smartube/>

Scrum Do URL: <https://www.scrumdo.com/projects/project/situation-aware-musicapp/>