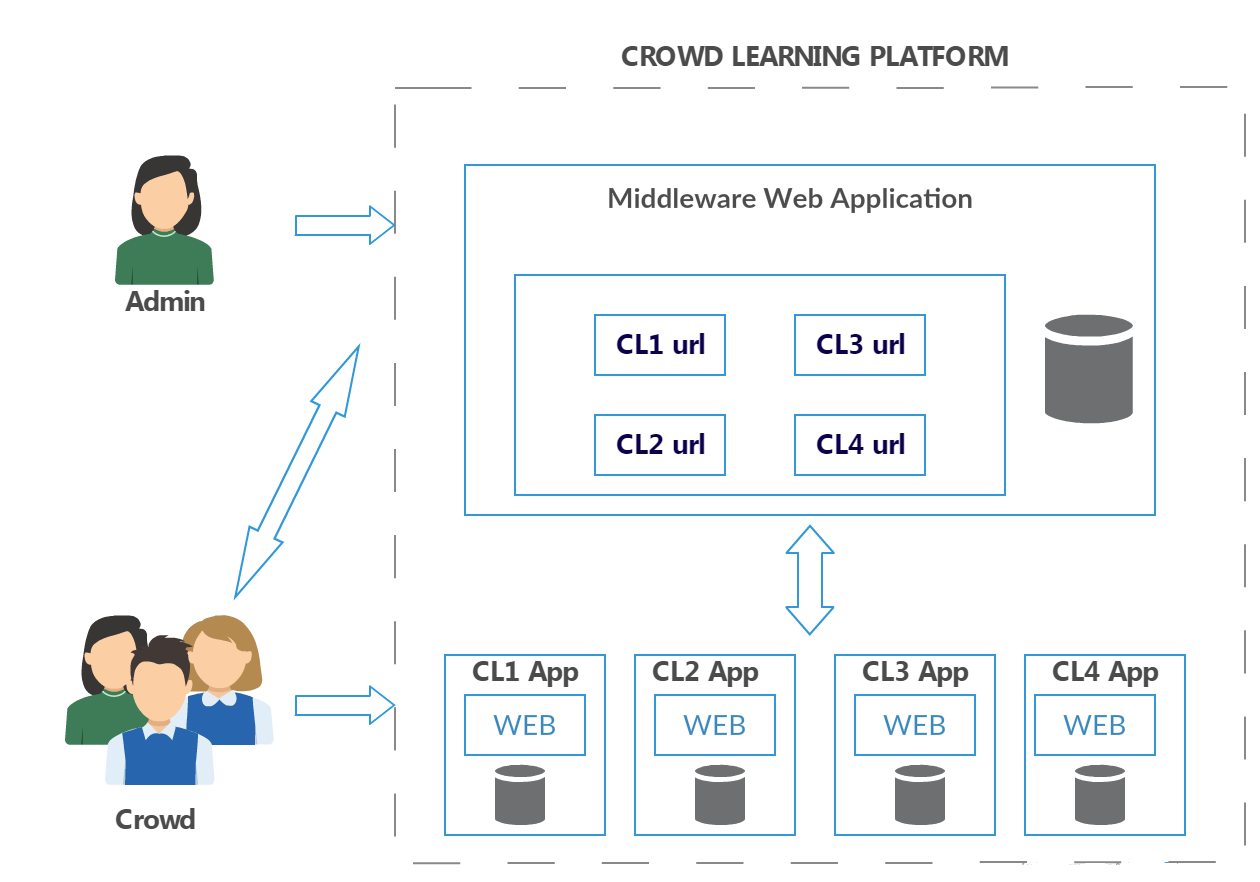
Cyber Infrastructure:



Middle ware is built using Java MVC and Hosted in Apache Tomcat server.

Educators can create an web application with any database and can host on any server and just give the respective url of their application into the middleware(Just a form entry).

Middleware Database(PostgreSQL) consist of

Users(userid,username,useremail,age,phone,deviceType,Device Config)

Tasks(taskid,taskname,taskurl,rest1url,rest2url,minCreditscore)

Userscore(UserId,TaskId,useremail,code,score)

Creditscore(userid,creditscore)

LoginUser(userid,useremail,password)

Hosts(hosted,hostname,hostemail,age,phone,deviceType,DeviceConfig)

HostUser(hosted,hostemail,password)

Auth(userid,code,taskid)

APPLICATION FLOW:

After login User will be able to see list of Crow Learning Application based on the credit score and selects the task depends on his interest

On selecting the task back end process update the middleware db with authentication doe for a respective task and respective user and calls the REST service of the CL application to update in its db with userid and authentication code.

I used the following Java Secure Random function (OTP) to generate the authentication code

public class SimpleOTPGenerator {

protected SimpleOTPGenerator() {

}

public static String random(int size) {

StringBuilder generatedToken = new StringBuilder();

try {

SecureRandom number = SecureRandom.getInstance("SHA1PRNG");

// Generate 20 integers 0..20

for (int i = 0; i < size; i++) {

generatedToken.append(number.nextInt(9));

}

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

}

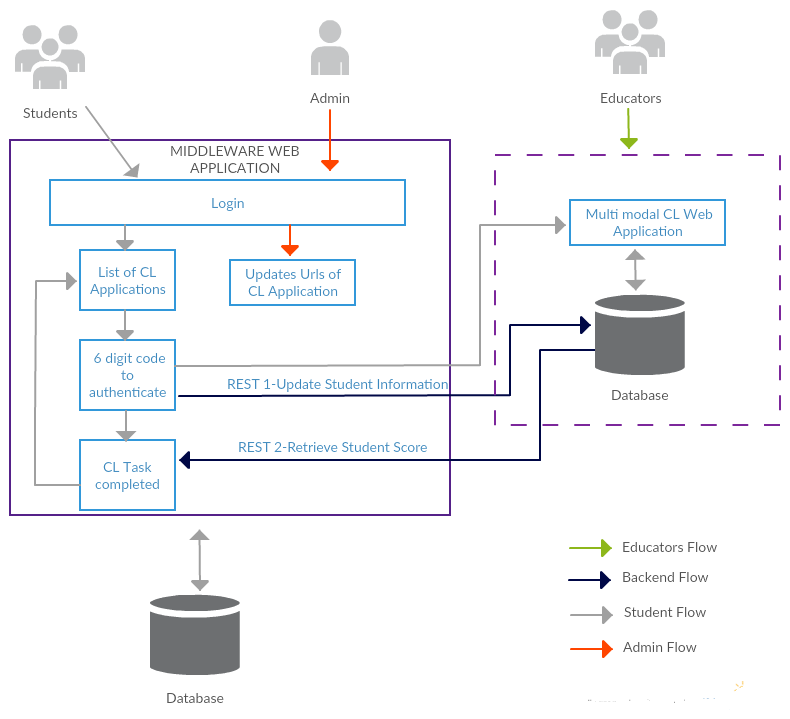
return generatedToken.toString();

<http://stackoverflow.com/questions/30953662/how-to-generate-otp-number-with-6-digits>

<https://www.securecoding.cert.org/confluence/display/java/MSC02-J.+Generate+strong+random+numbers>

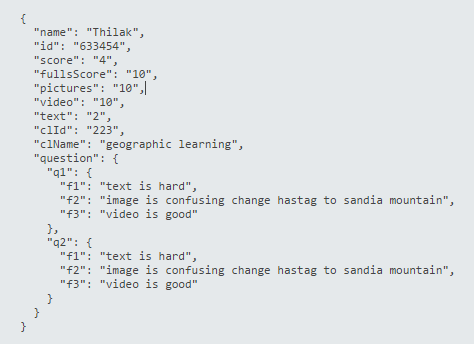
<http://resources.infosecinstitute.com/random-number-generation-java/>

Users once redirected to the CL application is required to enter the generated OTP and perform the task

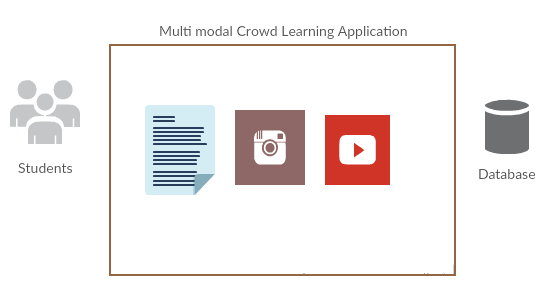


Once the task is completed User is redirected back to the middleware to see the list of Completed Task and On click of the completed Task user is able to view his score. (Backend REST call to the respective CL application to retrieve the Meta data of the user of a particular application)

Meta Data Structure:



Multimodal CL Application:



Quiz on Volcanoes, mountains

Provide Clues as Instagram Images and YouTube videos

Collect feedback

Track User behaviour

JavaMVC Application with HTML5 front end and MYSQL back end

Quiz Database

User(userid,taskid,password)

Task (taskid,taskname,description)

Quest(taskid,questionid)

Questions(questionid,question)

QuestionInstaClues(questionId,InstaId)

InstagramHastag(Instaid,hashtag)

QuestionVideoClues(questionId,videoId)

VideoURL (VideoId,URL)

UserScore(UserId,TaskId,Score)

UserBehavior(UserId,QuestionId,IsText,IsInsta,IsYoutube,IsCorrect)

UserFeedback(UserId,QuestionId,TextFeedback,InstaFeedback,yFeedack)

Instagram Images are retrieved using Instafeed.js

<http://instafeedjs.com/>

Youtube URL s are mebedded into HTML using HTML5 Embedding.

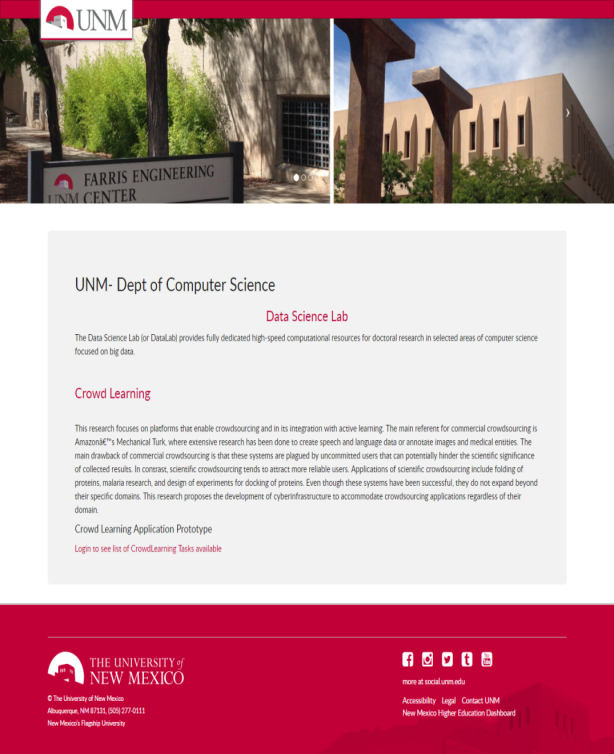
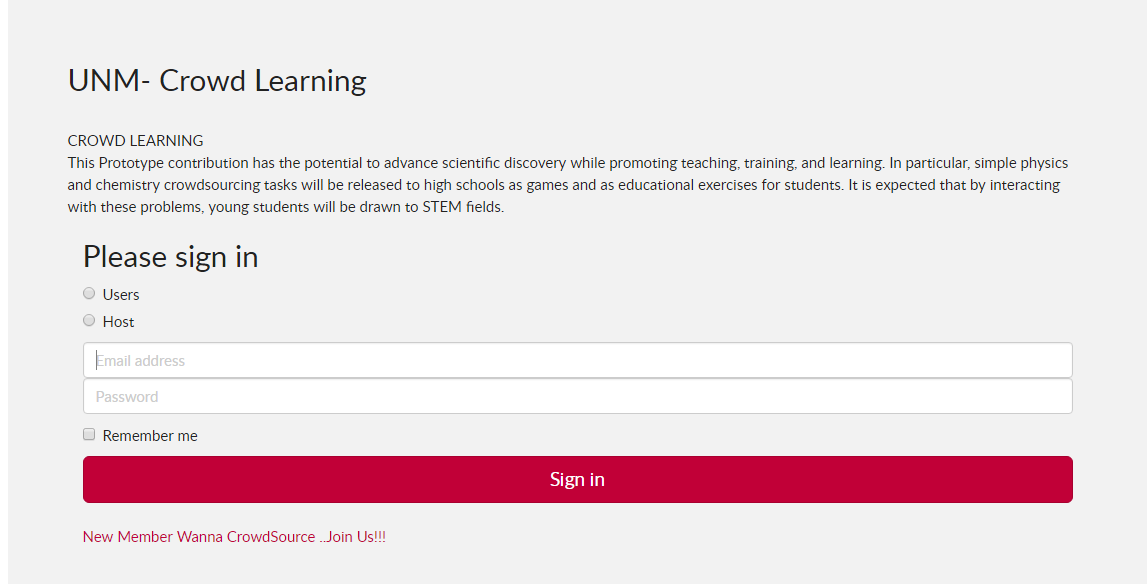
Huge meta data is collected to make sure if the user is trustable or not.

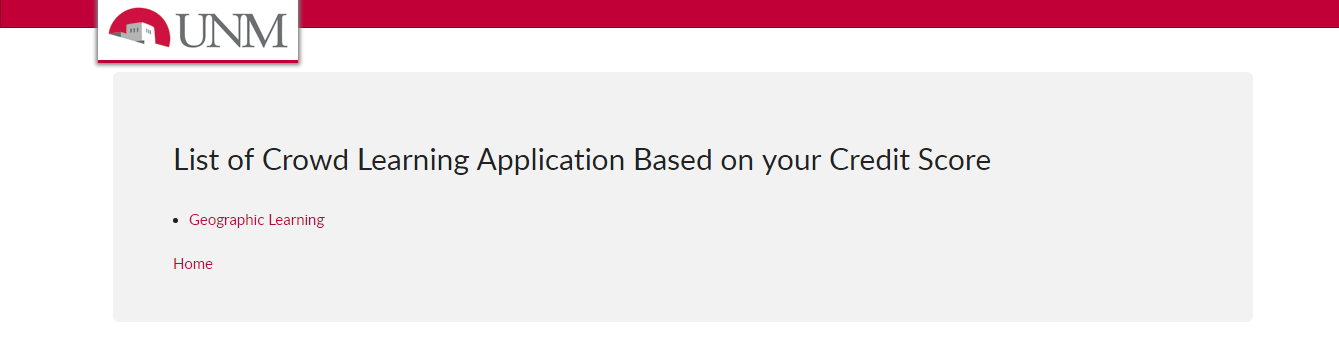
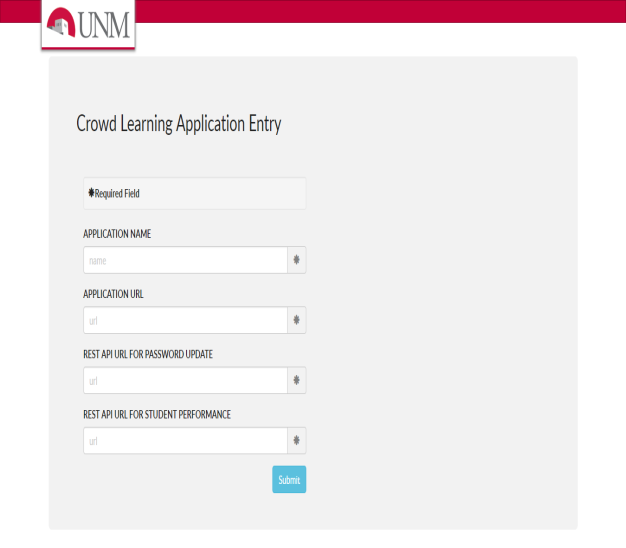
Based on the scores the user gets in application a credit score is generated-Logic:75% of score on each task is given with 1 point. As the number of tasks performed increases the score increases).

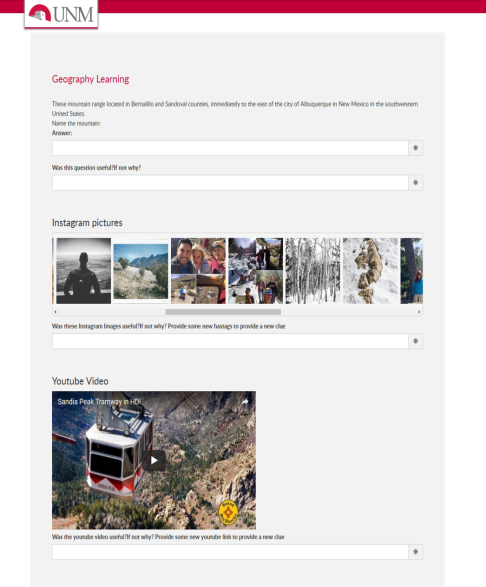
Currently working recommendation system to recommend tasks on user and Oauth 2.0

secure authentication.

Application flow screen shot



**Technical Aspects:**

Regarding Images:I am just holding the answer for the question as Hashtag of the image and storing it in table as

QuestionInstaClues(questionId,InstaId)

InstagramHastag(Instaid,hashtag)

This Hashtags are retrieved for a particular question and I use Instafeed.js,  a javascript code to retrieve images for a particular hashtag in front end.

Also I am holding some images as Blob in database and convert the Blob into jpeg image

QuestionImage(questionId,ImgId)

Image(ImgId,Image)

So for a particular question i retrieve the Blob content and  covert it to jpeg image by a java code

import java.awt.image.BufferedImage;

import java.awt.image.RenderedImage;

and display it as a image in jsp page.

For videos i use only the URLs of a particular video and store it in my database.

QuestionVideoClues(questionId,videoId)

VideoURL (VideoId,URL)

On front end of a particular question i retrieve the URL and embed the URL on using HTML5 tag called iframe.

For Java Flash Player during initial development,i was storing it as LONGBLOB and retriving it back but then i stopped it as i was only using Youtube videos..:(

All the text content is a also stored in Database for a particular question and i retrieve it.

Since the i used MYSQL and Java, I used JDBC Connectivity to call the database and retrieve a particular data.

Also followed MVC pattern with getters and setters for the Business Model -  my front end being-JSP,Backend is MySQL, and My Code-The Business logic is Java.

The random 6Digit code i generate for authentication is also secure as i use

Secure random method

import java.security.SecureRandom;

UI i used BootsrapCSS which is very easy to use and basic JavaScript functionalities like validation and required fields.

**for every resource (like for every video) can you keep userid, video link, description, and tags? so that we could find videos per tag search?**

Yes i can keep that professor as i have seperate tables as userInfo,Video Info and i link the question with a particular video as a seperate table for Normalization

QuestionVideoClues(questionId,videoId)

I also created a new table linking the UserId with Video Id