LAMAD Students Information V 0.24

Team member: Alfredo de Cea

Piotr Bartczak

Zhentian Wan

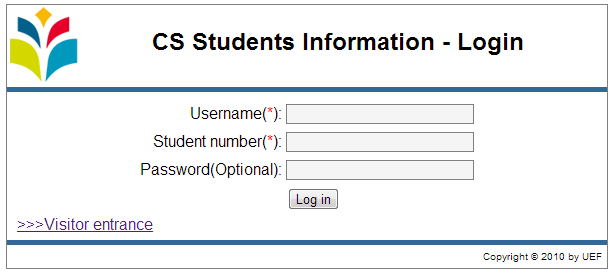
**Outline**

1. Introduction
2. Existing features
3. System Interface
   1. Login
   2. Welcome
   3. Student Name Search
   4. Country and Program and Year Search
   5. Information Update
   6. Chat interface and the related table in database
   7. Adding a new student
4. Code structure
5. Main logic and functions for each implemented part
6. Queries and data parsing
7. Apache server installation (for website - data base communication from student PC)
8. Ideas for future improvement
9. **Introduction**

This document describes the LAMAD course project implemented as a website. The goal is to display the information of international students (in university of Eastern Finland) including name, program, supervisor, start and graduation time and also country. The home countries of students are displayed on Google map provided on the website where the related markers on the map are clustered in case of overlap. The information of new students is added to the database manually. Different search options according to country name, student name, start and graduate year or according to program have been implemented. Everyone can see the results but students can login by their passwords and change the profile photo or their passwords using the provided interface. A chat interface has been also considered on the website so that a student can communicate with other students.

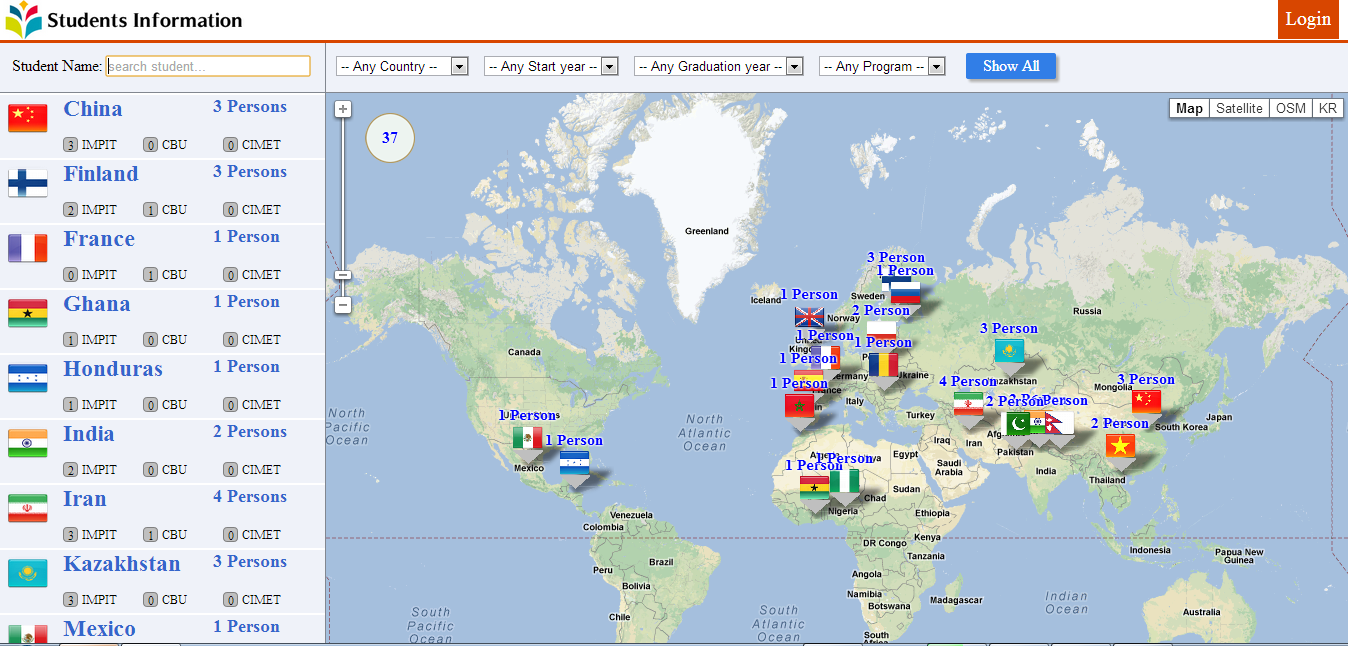
This website is a location-based application where distribution of students in Eastern Finland university all over the world can be viewed on the map.

1. **Existing features**
   1. Country List
   2. Student List
   3. Student Search
   4. Country and Program and Year Search
   5. Student Login and Logout
   6. Student Chatting: Server side, interface and database implementation
   7. Adding student to the database
2. **System Interface**
   1. **Login**



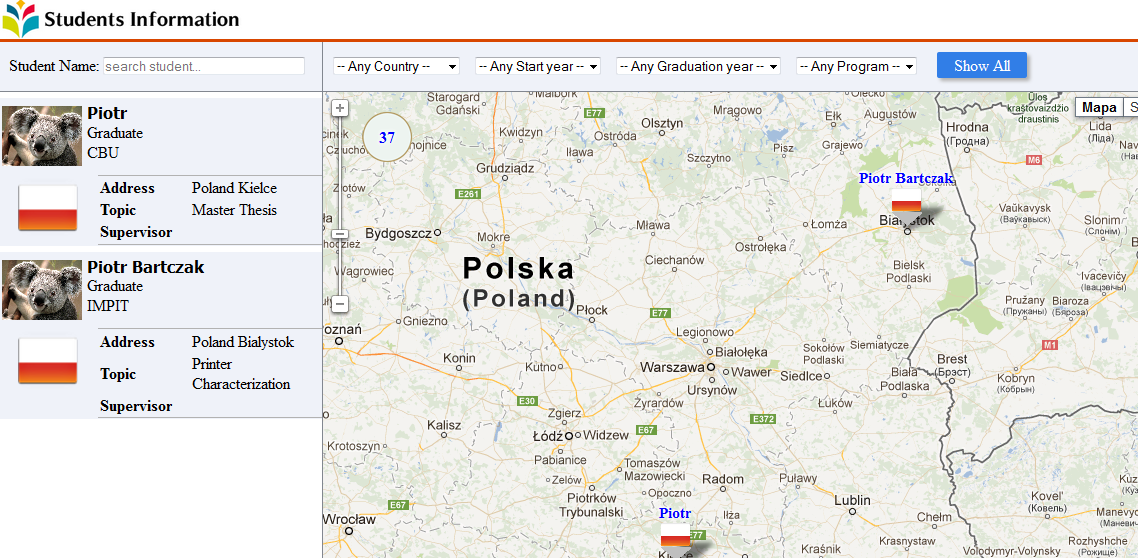
**Future task to do**

1. When you log in current location of the user should be updated on the server. Maybe pop up message “do you want update your current location” or just automatic updating which was agreed by user in his first login.
2. Automatically created Mopsi account which will link user and show his current location.
   1. **Welcome**



**Future task to do:**

1. When you click on Student Information logo on the right top it should move to the first web site as it’s done by “Show All” button
2. When you click on cluster or flag it zooms in to this region. Then when you click on marker of your interest nothing changes. It would be good to refresh left part with student info and display only info of the user corresponding to the marker (or at least show a message on the map with the basic information of the student).

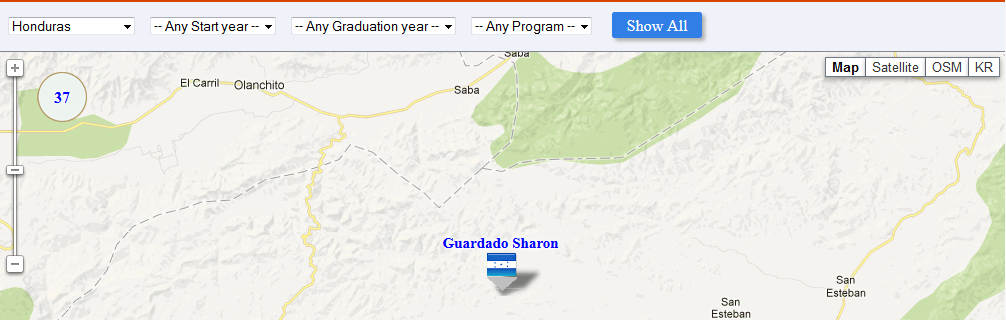


1. Maybe some info like “Your current location is …..” or “Your last log in was ….” or how many users are online etc. This information could be located on the top after log in. Other info about events like “This student graduated” or “This student updated his current location” etc. or ESCE summer school could be announced also.
   1. **Student Name Search**



**Future task to do:**

1. Maybe it is good idea to put Program in the brackets after name and surname
2. Some surnames and names are messed up (some users are like “Name Surname” and other like “Surname Name”). We should fix it too. Everything has to be checked and correct at the data base part because order of name and surname are not correct.
   1. **Country and Program and Year Search**

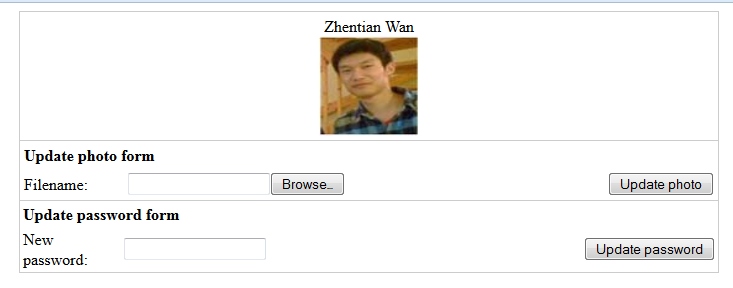


Under this link ([http://cs.uef.fi/pages/franti/lamad/LAMAD-API.pdf](https://owa.uef.fi/owa/redir.aspx?C=6d8ASFUDP0qji5pRcYi6gx80Ywu8KNBIuN_2NVG-W9Hy1wqq9uAnyTeP_HC2XiFFLC_549Ixeqw.&URL=http%3a%2f%2fcs.uef.fi%2fpages%2ffranti%2flamad%2fLAMAD-API.pdf)), you can find PDF file where all possible queries can be found. In basic words we can fetch all student data into one structure (to see example please check further *part 3. Data parsing and queries*). You can send request to the API and get list of students suiting your query. For example if you want student from Poland, then send request:

[http://cs.joensuu.fi/lamad/server.php?param={%22request\_type%22:%22get\_students\_with\_params%22,%22country%22:%22Poland%22}](http://cs.joensuu.fi/lamad/server.php?param=%7b%22request_type%22:%22get_students_with_params%22,%22country%22:%22Poland%22%7d)

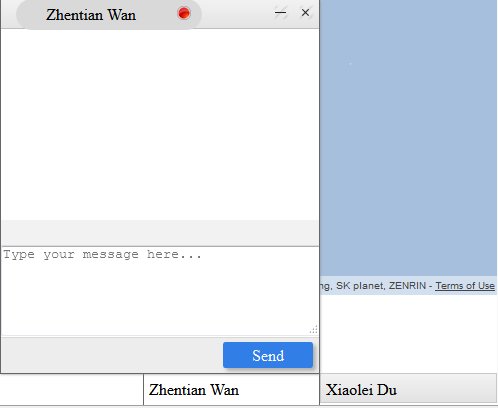
**Future task to do:**

1. Update API for CIMET students and include them in the query (several students each semester from all over the world, all are well documented and we have good contact with them, therefore a lot of possible new users can be found here)
2. Update recent years
   1. **Information Update**

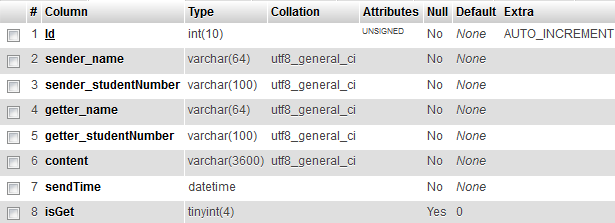


**Future task to do:**

1. Additional option for photo uploading as cropping the image, or image properties changes like from color to W&B, sepia…decreasing image size, quality etc.
2. Maybe it’s good idea to write your old password just to make it more secured. Then if old password is fine new password can be updated.
   1. **Chat Interface**

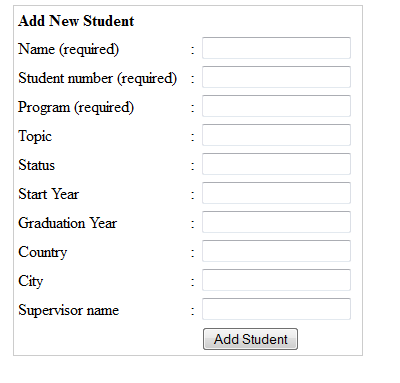


**Chat Database**



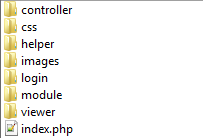
**Future task to do:**

* 1. You need set loggedin user as sender and getter. As a sender, he can sender message to other students. As a getter, he will get message from others. (0.5 hour)
  2. Think about how to display information when you receive a message or more message. (2 h)
  3. Add function "Get Chat History". (1 h)
  4. Delete Chat message.(0.5 h)
  5. Empty chat history.(0.5 h)
  6. Server maintains: Delete the message which are 3 months ago.(2 h)
  7. **Adding student to the data base:**

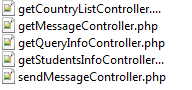


**Future task to do:**

1. Now each student has to be added manually but maybe there is possibility to add a file like excel file and list of student will be added automatically.
2. Updating student info for supervisor purpose should be implemented
3. After adding student we should display full display with his info just for error checking
4. **Code structure**



**controller folder**



**css** folder:

Contains all the css files which needed.

**helper** folder:

Contains javascript and php tools code.

**images** folder:

Contains images.

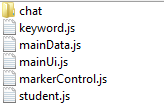
**login** folder:

All the login functions and interface.

**module** folder:

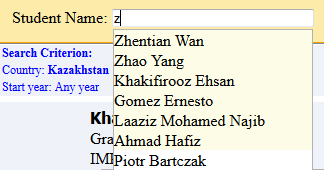


**viewer** folder:



chat : contains the interface related code.

keyword.js: related to search students by name.



mainData.js: contains AJAX code. Get data from server and send data to interface.

mainUi.js: The interface code for the website.

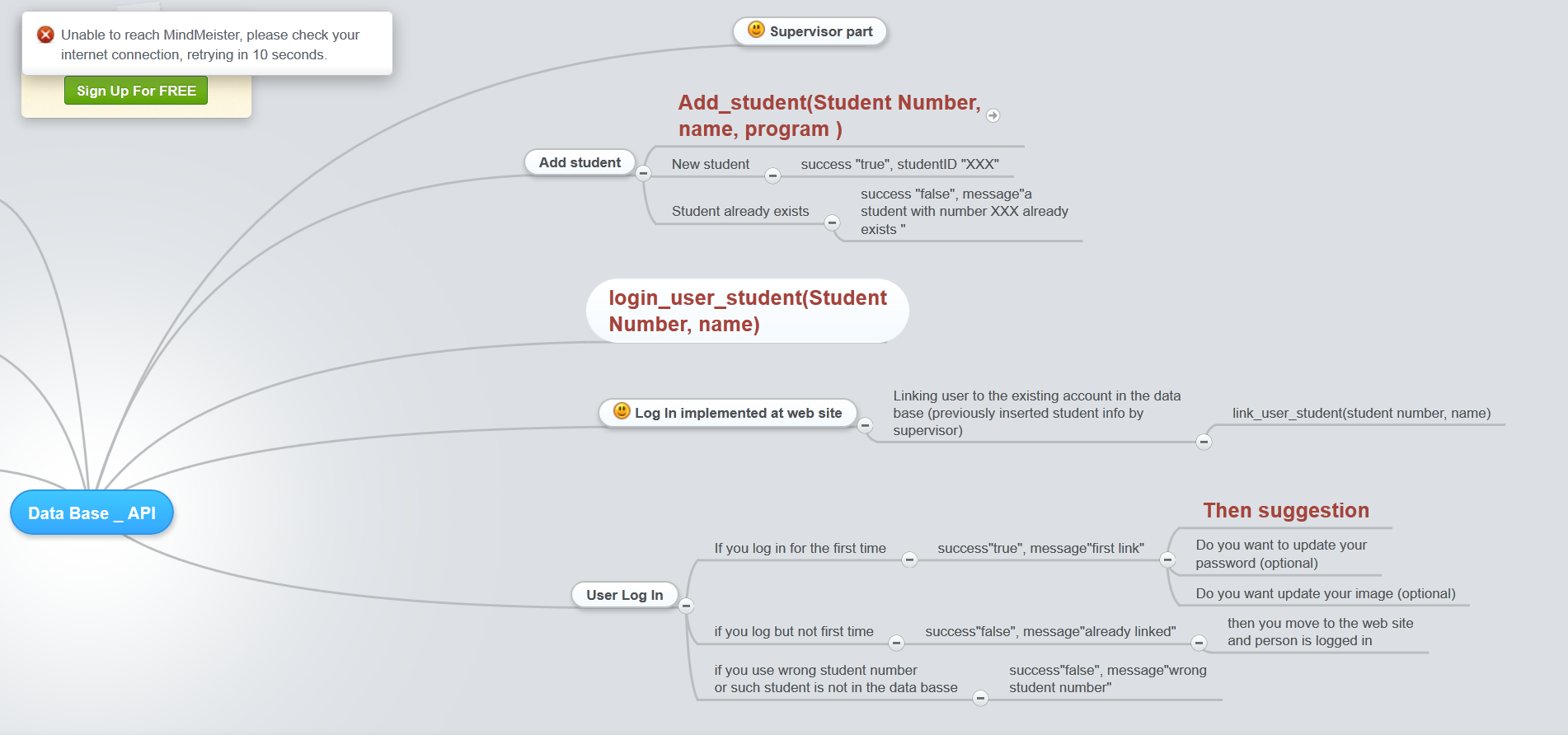
markerControl.js: Google map related.

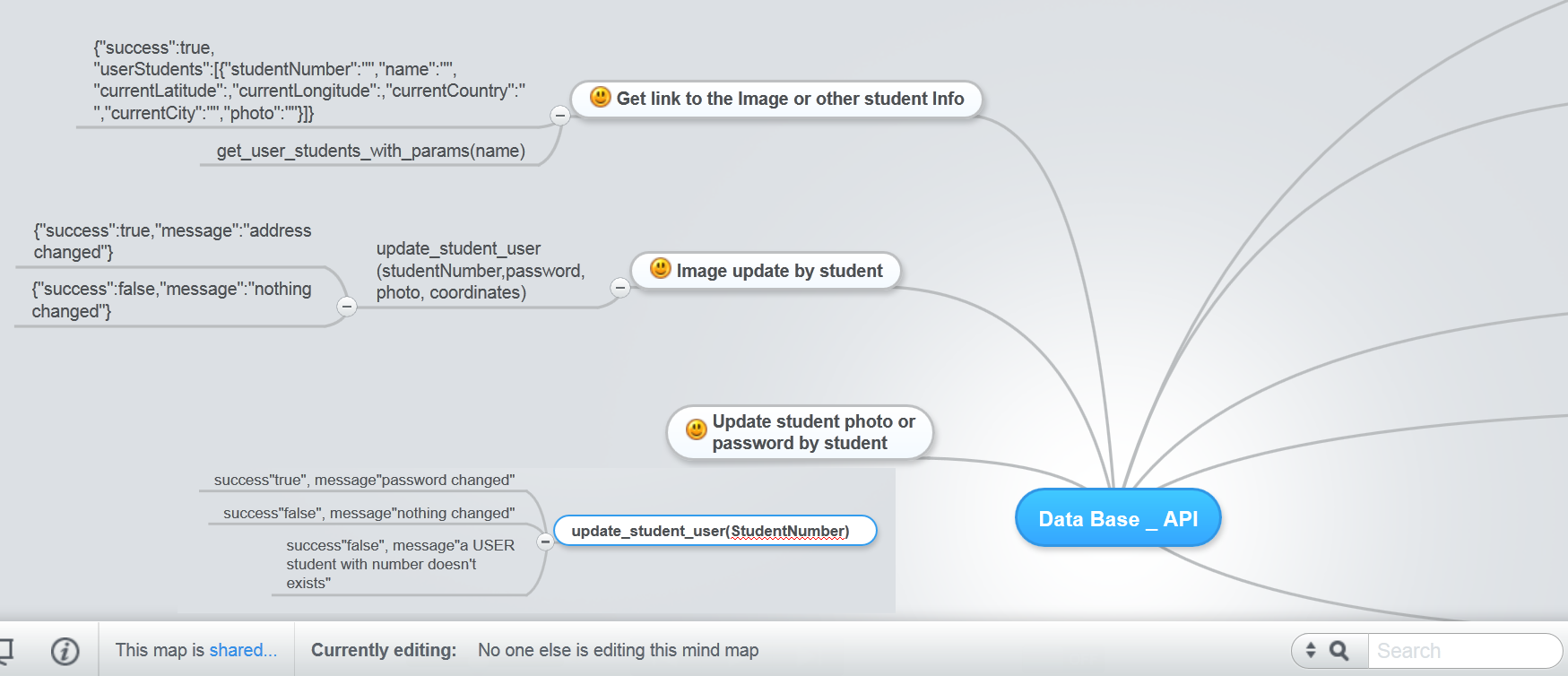
student.js: Interface logic control.

1. **Main logic and functions for each implemented part**

At first the student has to be added to the database by a supervisor. The STUDENT ADD (sample code included) function was implemented for easy and manual student adding. Then not linked student account is available at data base and is waiting for first linking.

UEF student knows his name surname and student number, therefore on the LOG IN web site he is asked to put both name surname, and password. If all are correct then student is linked with his account at DATA BASE. Then if the student is logging FIRST time there is a suggestion to UPDATE PASSWORD and IMAGE, both are optional. User has to be logged in order to change those parameters.

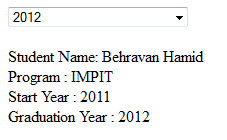




1. **Queries and data parsing**

Parsing Json

|  |
| --- |
| <html>  <head></head>  <body>  <?php  if (isset($\_GET['GraduationYear'])){  $year = $\_GET['GraduationYear'];  $jsonString = file\_get\_contents('http://cs.joensuu.fi/lamad/server.php?param={'  . '"request\_type":"get\_students\_with\_params",'  . '"graduationYear":"' . $year . '"}');  }  $json = json\_decode($jsonString,true);  echo '<pre>';  print\_r($json);  exit; |
|  |
| <html>  <head></head>  <body>  <?php  if (isset($\_GET['GraduationYear'])) {  $year = $\_GET['GraduationYear'];  $jsonString = file\_get\_contents('http://cs.joensuu.fi/lamad/server.php?param={'  . '"request\_type":"get\_students\_with\_params",'  . '"graduationYear":"' . $year . '"}');  }  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Parsing data  $json = json\_decode($jsonString, true);  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* See structure of received String  //echo '<pre>';  //print\_r($json);  //exit;  // \*\*\* if empty cell in success array then error, else parse and display Info  $successInfo = $json['success'];  if (empty($successInfo)) {  echo 'No Such Student In Our Data Base';  } else {  // display Name Program Start Year Graduation Year  foreach ($json['students'] as $StudentInfo) { {  echo 'Student Name: ' . $StudentInfo['name'] . '<br />';  echo 'Program : ' . $StudentInfo['program'] . '<br />';  echo 'Start Year : ' . $StudentInfo['startYear'] . '<br />';  echo 'Graduation Year : ' . $StudentInfo['graduationYear'] . '<br />';  }  }  }  ?>  </body>  </html> |



## Login system

When you log in the system the first time you only have to put your name and student number. If they match then you enter the system and a cookie and a session are created. When you want o access to any website aimed only for logged in users the first thing that the page will do is to check if you are logged in. For this it checks if there’s a session. If there’s no session but there’s a cookie it restores the session. The duration of the cookie can be modified. When you logout the session and the cookie are destroyed. The session and the cookie store the name and the student number of the user while they are active.

Code to put in the very beginning of every page aimed only to logged in users:

<?php

session\_start();

include("classLogin.php");

$login = new login();

$login->start();

?>

This code will redirect you to the main page of you are not logged in.

1. **Apache installation**

The Apache HTTP Server is a web server software program notable for playing a key role in the initial growth of the World Wide Web. It is necessary if you want to execute php code on your PC.

|  |
| --- |
| <http://httpd.apache.org/download.cgi>  Win32 Binary without crypto (no mod\_ssl) (MSI Installer): [httpd-2.0.64-win32-x86-no\_ssl.msi](http://www.nic.funet.fi/pub/mirrors/apache.org/httpd/binaries/win32/httpd-2.0.64-win32-x86-no_ssl.msi)  Then:  <http://windows.php.net/download/> VC6 x86 Thread Safe (2011-Mar-22 13:29:30)  * [Zip](http://windows.php.net/downloads/releases/php-5.2.17-Win32-VC6-x86.zip) [10.06MB] sha1: 23e1cf2f6e1bf64585ae921462340e5748fcc939   Then:  <http://www.thesitewizard.com/php/install-php-5-apache-windows.shtml>  Some files have to be changed (follow steps in the instruction) ready files are attached:  httpd.conf  php  index |

1. **Ideas for future improvement**
2. Finish the function logic of student chatting.
3. Graphical interface for supervisor for adding new students and updating their info.

Maybe also special log in for supervisor at web site

1. Update student home country address.
2. Apply Clustering to country markers.

d.1. Using MOPSI project component to apply grid-based clustering to the country markers on the map.

d.2. After applying clustering, some existing functions might be affected, need to modify them to make system works

1. Browser Compatibility, Current system works for FireFox, Chrome, Opera, Safari. But not for IE.
2. Apply CSS style to the page you see when you first log in (when it suggests you to update your information). Improve the visual aspect of the interface to make it more visual attractive.