16/04/2018

Learning Management System (LMS) - E4 Project

IMC

1 Contents

1. Introduction

Learning management System (LMS) is a software application for the administration, documentation, tracking, reporting and delivery of educational courses. It helps the instructor deliver course material to the students, administer tests and other assignments, track student progress, and manage record-keeping. Proposed LMS in this project is mainly focused on online tests delivery but support a range of other uses; it will act as a platform for fully online exercises evaluation system.

This project is expected to deliver a flexible, easy to use and secure online portal that will act as a dashboard for several types of users including students and teachers.

The new LMS will facilitate professors with dynamic graphical reports of student evaluation in form of charts, interactive illustrations and will allow students to attempt exercises through a user friendly platform and communicate with professors by easy to use messaging system which are lacking in the currently available systems.

2. Project Objectives and Description

The project objectives is to enable different types of users (students, teachers, administrators…) to view and improve their work using statistics and graphical tools. The main source of data for this project are the information recorded about students and their work. Students currently use two educational tools which are WIMS and PL (Premier Langue) to solve exercises and sheets of exercises, with each set of questions and sheets and exercises being tied to a certain class. With this information, we can enable students to view their related work, how they are improving their skills with respect to other students in the class. We can also enable teachers and other concerned parties to view many helpful statistical data about the students which will improve the way how they view their students' performance and how they manage their classes.

3. Technology and Framework

3.1 Django Framework

Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Some well-known sites that use Django include the Public Broadcasting Service, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor. It was used on Pinterest, but later the site moved to a framework built over Flask.

3.2 Python

Python is an interpreted high-level programming language for general-purpose programming. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. These features have made python a powerful scripting language and highly flexible; definitely what we were looking for our project. In this project, python is almost used exclusively to program all the parts in Django.

3.3 D3.js

D3.js (D3 for Data-Driven Documents) is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely implemented SVG, HTML5, and CSS standards. In contrast to many other libraries, D3.js allows great control over the final visual result. Its development was noted in 2011, as version 2.0.0 was released in August 2011.

D3.js is used on hundreds of thousands of websites. Some popular uses include creating interactive graphics for online news websites, information dashboards for viewing data, and producing maps from GIS map making data. In addition, the exportable nature of SVG enables graphics created by D3 to be used in print publications.

3.4 Overview of the Django Framework

Django relies on programmers to write code in Python to program:

1. Applications, those are the main parts that you work on in Django. They contain the HTML templates, CSS, models.py, urls.py, views.py among other files that help power that app.
2. Project settings.py file which contains the main configurations for the project. Application names that you wish to use in the project are specified inside as well as other parameters like the type of database used
3. Urls.py files that describe the URL link that is used to access the project or the existing applications in the project. What kind of URL patterns are possible and what are not.
4. Views.py files that process requests from users. Inside those Views files, models (tables in the database) can be queried to retrieve data, data manipulation (through Python, of course) can take place and then injected dynamically into the HTML code at run-time for the user to see.

5. Models.py files, which are basically the definitions of your tables inside the database. Fields are defined based on their datatypes, fields can also be referenced from other tables to define foreign key constraints.
6. HTML and CSS for the design of the actual page.

3.4.1 Django with Database

Django gives you the option of automatically creating the database tablespace using engines like SQLite3 and PostgreSQL. “models.py” is where the views (tables) are programmed and specified. **“makemigrations”** command is used to create migration files for each model file written/modified. After that **“migrate”** command is used to apply those migration files to the database.

4 Project Development Segmentation

4.1 Front-end Design

4.1.1 Work done

- Designing of the login page used by multiple user types to login in.
- Designing of the main HTML template that is used to display the pages in project.
- Extracting the ideas and requirements from the interviews and figuring out which kind of charts that best reflect those requirements.

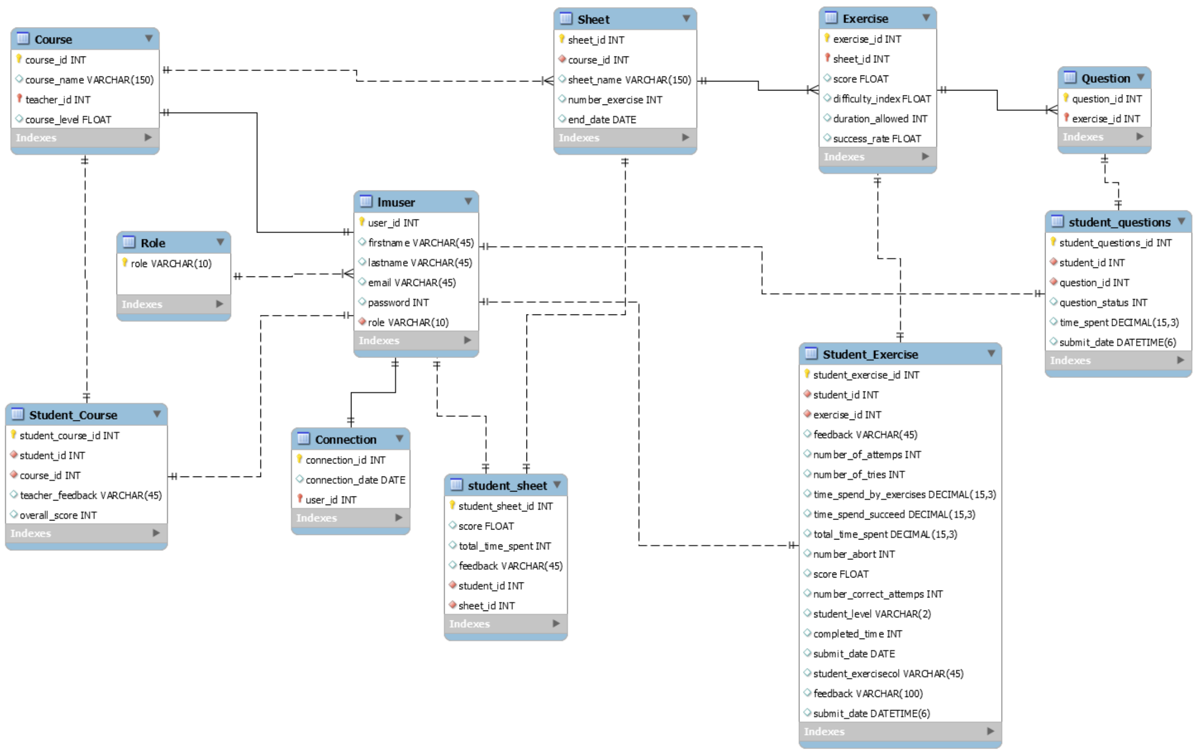
4.1.2 Work to be done

- Following teacher portal subpages are planned to be designed:
- Charts detail pages, consisting individual student exercise reports.
- Report page for running and completed tasks.
- Students’ messages page along with a messages popup form in dashboard.
- Detailed report pages for student statistics (number of enrolled students in each course; conducted tests and number of students took tests).
- Profile page.

4.2 Back-end

4.2.1 Work done

Originally, the idea was to figure out how WIMS is structured in terms of database tables and their relationships so we can replicate the same structure in our Django project but due to the fact that we were not able to get access to WIMS until later on, we had to create our own database model that tries to replicate that data types



used in WIMS based on the interviews provided.

The work was divided among the team members to read the interviews (6 interviews in total) and figure out the columns that we need for the database. After each interview was studied, we combined the individual work and designed the relationships between the tables and the foreign key constraints.

4.2.2 Work to be done

After installing WIMS, we can create GET requests from the server and read the structure of the JSON string returned by the server. After realizing the structure of the data, we can create a database model that resembles that of WIMS and perhaps propose possible changes for the database structure.

4.3 Proposed Features

Based on the idea of developing a user friendly system; the aim is to make the software easy to use and teacher/student focused.

The following features are proposed for future improvement of the software that can benefit both students and teachers to utilize the software in a more efficient way.

4.3.1 Interactive Charts

- Bar chart
- Line chart
- Area chart
- Column chart (Horizontal bar chart)
- Scatter chart
- Histogram
- Pie chart
- Donut chart
- Radar chart
- Polar area chart

Purposed chart libraries:

- Flot Chart
- Morris Chart
- Google Chart
- Chartist Charts
- Chartjs
- C3 Chart
- Sparkline Chart
- Jquery Knob

4.3.2 2. Dynamic filterable tables

4.3.3 Statistic Tiles

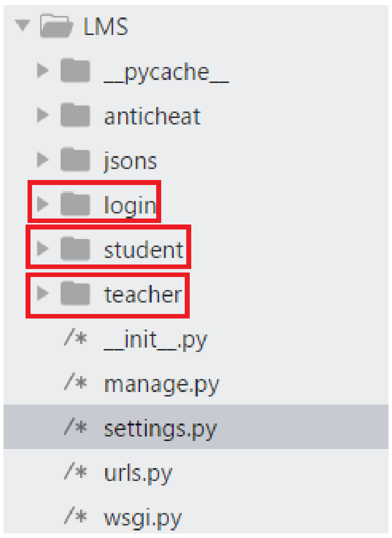
4.3.4 Student Task notifications feature

4.3.5 Student performance segmented report

4.4 Programming

4.4.1 Work done

So far, in Django, we have managed to handle to logging in of the user, creation of a student dashboard as well as a teacher dashboard. This required programming multiple python files in the project. Details about the coding will be found in the code documentation. In addition to programing the views and models for the applications, the D3.js visual library was used inside the project to display the graphs through SVG.



We created 3 applications that were used in the presentation. The “login” app which is responsible for handling the login requests. The “student” app, responsible for handling requests by users who are of type students. The “teacher” app, responsible for handling requests by users who are of type students.

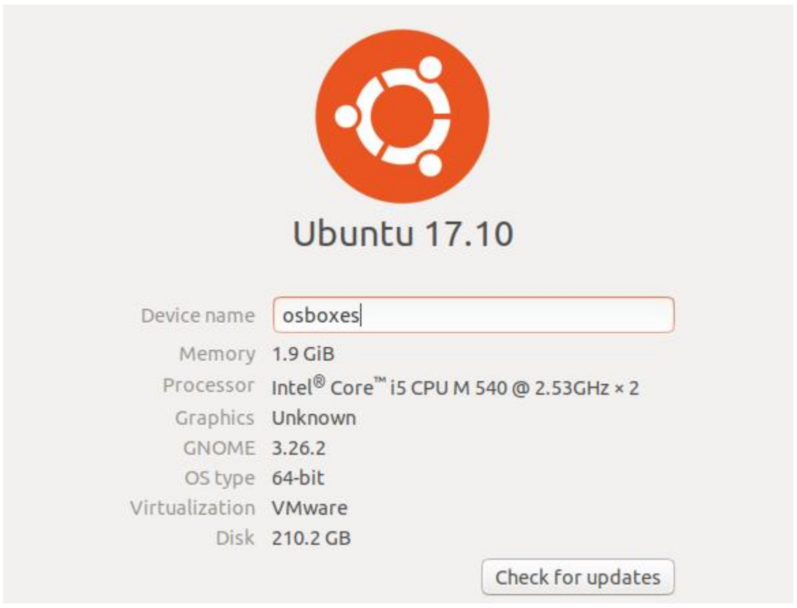
4.4.2 Work to be done

More users are to be implemented in the project. Missing users according to the project requirements are: “Creator”, “Didactician” and “Institutional Head”

5 WIMS Installation

5.1 Environment

For installing WIMS, we used virtual machine Ubuntu 17.10 on VMware.



Ubuntu 17.10

Device name	osboxes
Memory	1.9 GiB
Processor	Intel® Core™ i5 CPU M 540 @ 2.53GHz × 2
Graphics	Unknown
GNOME	3.26.2
OS type	64-bit
Virtualization	VMware
Disk	210.2 GB

5.2 Install using the commands

Before installing the WIMS you need run 2 commands

```
apt-get update

apt-get upgrade  --
show-upgrade
```

These 2 commands will update your operation system and initially prepare the machine.

After successfully running these commands, run the command written below.

```
apt-get install wims
```

This will install the latest wims package available on Debian: https://packages.debian.org/en/stretch/wims

Using this, WIMS will be installed here: **/ var / lib / wims**

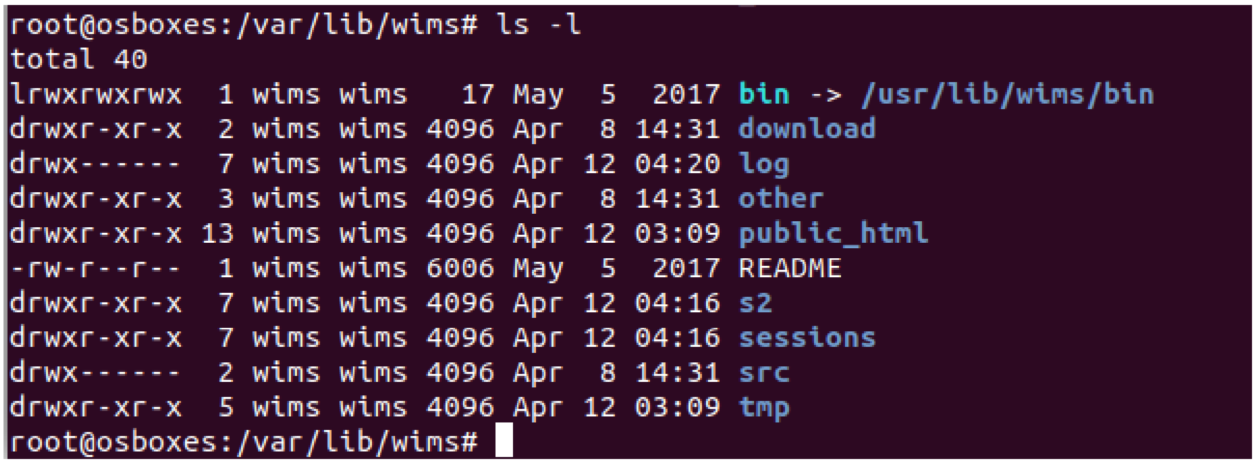
Note: before running this command, your machine need to have the apache2 installed using below command:

```
sudo apt-get install apache2
```

Whether you are using automatic or manual installation, we recommend you upgrade to WIMS Configuration.

5.3 WIMS configuration

By default, it would be installed in **/var/lib/wims.**

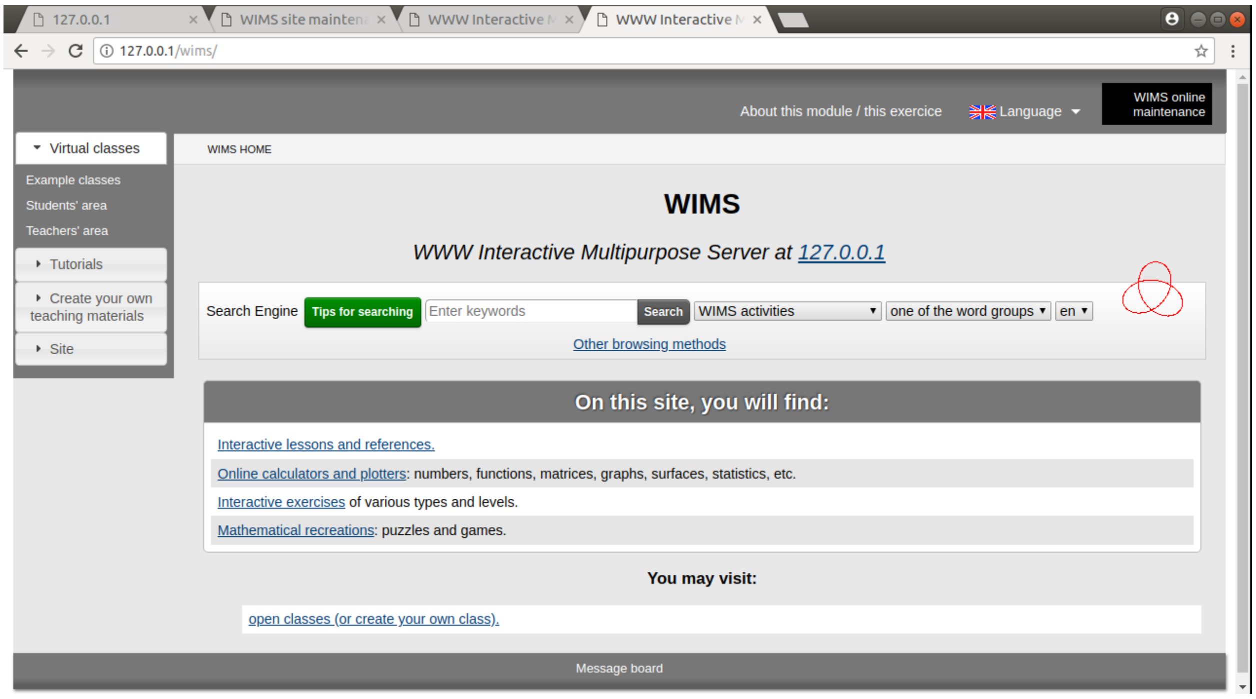


```
root@osboxes:/var/lib/wims# ls -l
total 40
lrwxrwxrwx  1 wims wims   17 May  5  2017 bin -> /usr/lib/wims/bin
drwxr-xr-x  2 wims wims 4096 Apr  8 14:31 download
drwx------  7 wims wims 4096 Apr 12 04:20 log
drwxr-xr-x  3 wims wims 4096 Apr  8 14:31 other
drwxr-xr-x 13 wims wims 4096 Apr 12 03:09 public_html
-rw-r--r--  1 wims wims 6006 May  5  2017 README
drwxr-xr-x  7 wims wims 4096 Apr 12 04:16 s2
drwxr-xr-x  7 wims wims 4096 Apr 12 04:16 sessions
drwx------  2 wims wims 4096 Apr  8 14:31 src
drwxr-xr-x  5 wims wims 4096 Apr 12 03:09 tmp
root@osboxes:/var/lib/wims#
```

Open a browser on the machine where you have installed WIMS and go this address

```
http://127.0.0.1/wims/
```

And you will see the webpage like the one below:



WIMS

WWW Interactive Multipurpose Server at 127.0.0.1

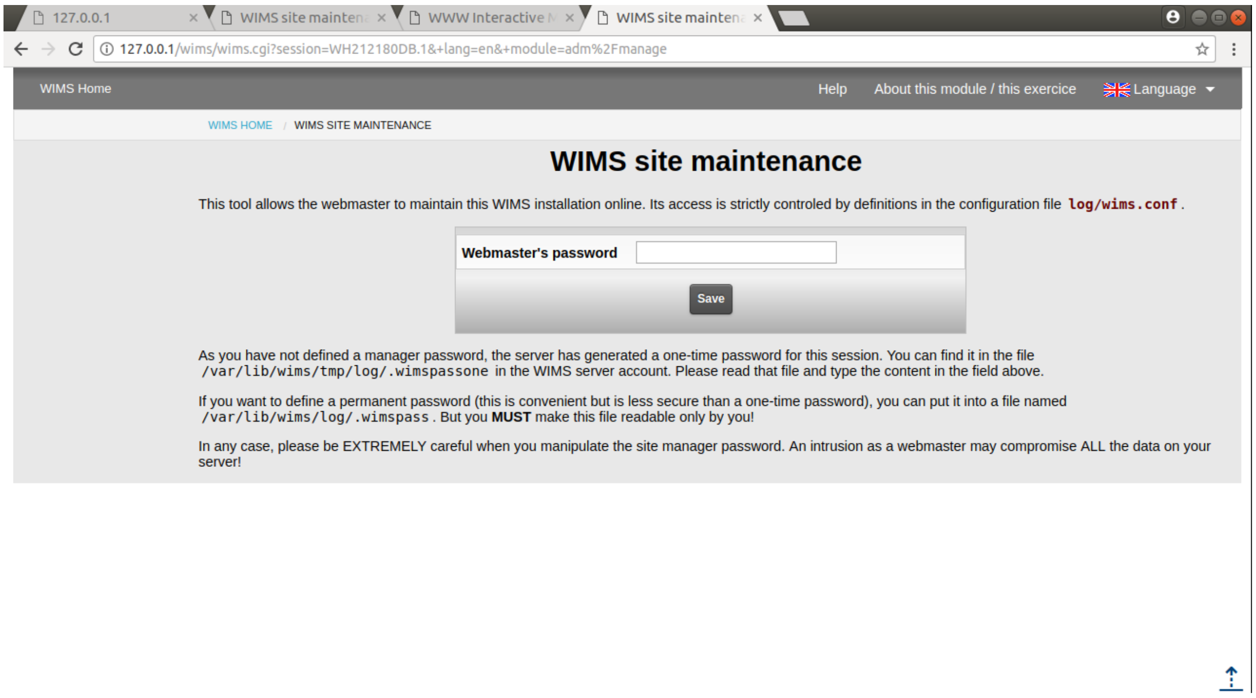
On this site, you will find:

- Interactive lessons and references.
- Online calculators and plotters: numbers, functions, matrices, graphs, surfaces, statistics, etc.
- Interactive exercises of various types and levels.
- Mathematical recreations: puzzles and games.

You may visit:

open classes (or create your own class).

If you want login with administrator, there is black button “WIMS online maintenance” in the top right corner and you will see below page.


127.0.0.1
WIMS site mainten
WWW Interactive M
WIMS site mainten
127.0.0.1/wims/wims.cgi?session=WH212180DB.1&+lang=en&+module=adm%2Fmanage
WIMS Home
Help
About this module / this exercice
Language
WIMS HOME / WIMS SITE MAINTENANCE

WIMS site maintenance

This tool allows the webmaster to maintain this WIMS installation online. Its access is strictly controled by definitions in the configuration file `log/wims.conf` .

Webmaster's password
Save

As you have not defined a manager password, the server has generated a one-time password for this session. You can find it in the file `/var/lib/wims/tmp/log/.wimspassone` in the WIMS server account. Please read that file and type the content in the field above.

If you want to define a permanent password (this is convenient but is less secure than a one-time password), you can put it into a file named `/var/lib/wims/log/.wimspass` . But you **MUST** make this file readable only by you!

In any case, please be EXTREMELY careful when you manipulate the site manager password. An intrusion as a webmaster may compromise ALL the data on your server!

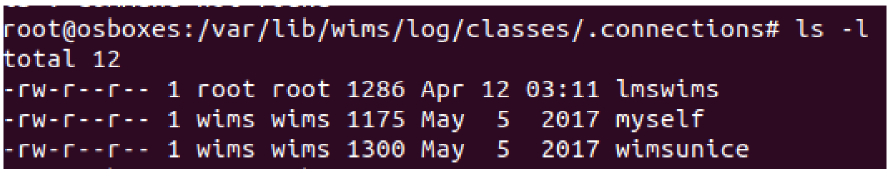
For login this page you can use temporarily password and permanent password.

Temporarily password located in **/var/lib/wims/tmp/log/.wimspassone**

If you want to define a permanent password (this is convenient but is less secure than a one-time password), you can put it into a file named /var/lib/wims/log/.wimspass. But you MUST make this file readable only by you!

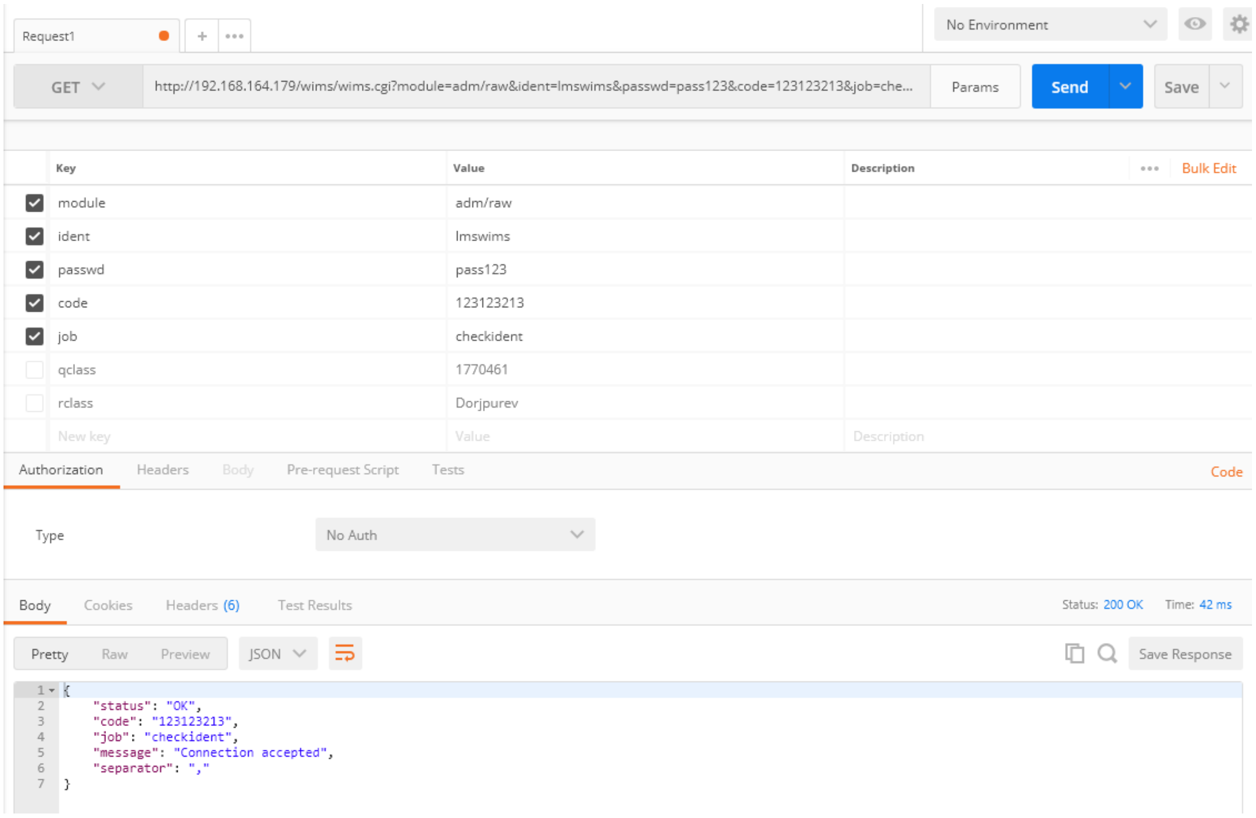
5.4 API service

To connect WIMS from another server or application, we used Protocol for WIMS direct connection. For any documentation you have to go http://127.0.0.1/wims/wims.cgi?module=adm/raw&job=help this link. As mentioned there, first, we need to declare the connection file under **/var/lib/wims/log/classes/.connections/** directory and create a new customized file for you.



```
root@osboxes:/var/lib/wims/log/classes/.connections# ls -l
total 12
-rw-r--r-- 1 root root 1286 Apr 12 03:11 lmswims
-rw-r--r-- 1 wims wims 1175 May  5  2017 myself
-rw-r--r-- 1 wims wims 1300 May  5  2017 wimsunice
```

And I attached **lmswims** connection file with this documentation.



```
http://192.168.164.179/wims/wims.cgi?module=adm/raw&ident=lmswims&passwd=pass123&code=123123213&job=che...
```

	Key	Value	Description
✓	module	adm/raw	
✓	ident	lmswims	
✓	passwd	pass123	
✓	code	123123213	
✓	job	checkident	
	qclass	1770461	
	rclass	Dorjpurev	

Type: No Auth

Status: 200 OK Time: 42 ms

```
{
    "status": "OK",
    "code": "123123213",
    "job": "checkident",
    "message": "Connection accepted",
    "separator": ","
}
```

As we see the above generated GET HTTP request, we succeed to connect our virtual WIMS and connection accepted.

6 Project Development, Distribution & Problems Faced

It is important to point out the duration of the Project was around two months given to the responsibility of 8 students. The project was first proposed to us on the 1st of February and was expected to be delivered on the 12th of April. During this time, we had to learn how Django works, explore WIMS and get response messages from the WIMS server in order to understand how the data is structured so that we can store them in Django's database. However, during this time we struggled to connect to the WIMS server due to the fact that it was restricted to GET any data from the server as there was no restriction on the access to the server, hence, it was not permissible to connect to the server. This caused a lot of delay for us. It also made us have to come with our own database structure that somehow replicates the data structure of WIMS. Another problem faced during this project was the lack of experience of the teammates for this project. This put pressure on the programmers as there were only 2 people in charge of programming which caused the project to move in a slow pace.

In a generalized overview of how the work load was distributed, I (Ahmad Nadar) and Ulziiburen Dorjpurev were both responsible for the programming part and the part of realizing the database structure, while the rest of the team was responsible for reading the interviews and figuring out what the tables and columns for the database are, as well as figuring out the graphs needed for the project based on the interviews.

7 Conclusion

In conclusion, the LMS project was expected to deliver a portal for various types of users (including teachers and students) where users are able to view and track information related to their classes and records. Due to problems faced, we were only able to create the portal for the students and teachers and not for other users. Of course, the project is a work in progress and needs more implementation and development, as well as expanding and optimization.