16/04/2018

IMC

# 

Learning Management System (LMS) - E4 Project

# 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.

# Technology and Framework

## 2.1 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.

## 2.2 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.

## 2.3 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.

## 2.4 Overview of the Project Framework

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

1. URL files that dictate how the “website” is accessed. What kind of URLs are possible and what are not.
2. Views 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.
3. Model 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.
4. HTML and CSS for the design of the actual page.

# 3. Project development segmentation

**3.1. Front-end Design**

**3.1.1. 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.

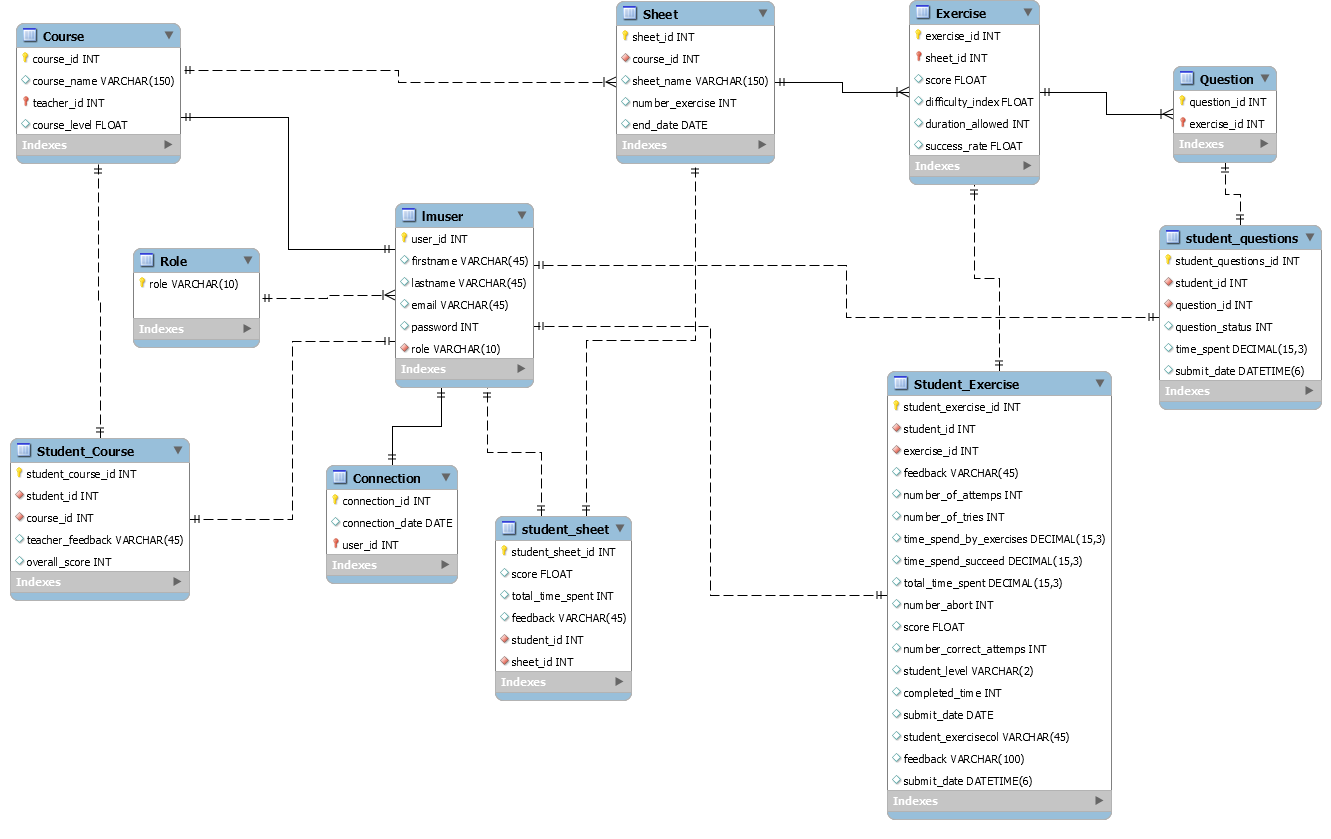
**3.1.2. Work to be done**

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

**3.3. Back-end**

**3.3.1. 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.

**3.3.2. 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.

**3.2. Programming**

**3.2.1. Work done**

**3.2.2. Work to be done**

**3.4. Documentation**

**3.4.1. Work done**

**3.4.2. Work to be done**

**4. Proposed Features**

**5. Conclusion**