FINAL PROJECT
BYTE ACADEMY

PROJECT NAME : SIMPLE E-LEARN

AUTHOR : Ananth Adhikarla

DATE : March 5, 2018

TECHNOLOGY USED: DJANGO, PYTHON, DJANGO REST API, JQUERY,AJAX, SQLITE3

Project Description

This project based on creating a e-learning platform (CMS) that allows instructors to create courses and manage their contents and students to access and enroll into the courses.

The e-Learning platform will offer courses in various subjects. Each course will be divided into a number of modules based on the course and instructor, and each module will contain a number of contents. There will be contents of various types: text, file, image, or video. The following example shows what the data structure of our course will look like:

We have defined four different content models

- Text: To store text content.
- File: To store files, such as PDF, DOC, PPT.
- Image: To store image files.
- Video: To store videos. We use an URLField field to provide a video URL in order to embed it.

Subject - example : computer science

Course – example : python programming

Module 1 – example : intro to python basics

Content (text, file, image or video)

Module 2 – example : intro to data structures

Content (text, file, image or video)

Course – example : Git and github

Module 1 – example : what is git

Content (text, file, image or video)

Module 2 – example : what is github

Content (text, file, image or video)

Module 2 – example : how to use git and github

Content (text, file, image or video)

I used a generic relationship for the content types mentioned above. I added a limit_choices_to argument to limit the ContentType objects that can be used for the generic relationship. Then used the model__in field lookup to filter the query to the ContentType objects with a model attribute that is 'text', 'video', 'image' or, 'file'.

Therefore the models created are suitable to add diverse content to the course modules.

There is also a custom field model which

- Automatically assign an order value when no specific order is provided. When no order is provided while storing an object, our field should automatically assign the next order based on the last existing ordered object. If there are two objects with order 1 and 2 respectively, when saving a third object, we should automatically assign the order 3 to it if no specific order is given.

- Order objects with respect to other fields. Course modules will be ordered with respect to the course they belong to and module contents with respect to the module they belong to.

Content Management System (CMS)

The following features provided are

- Registration for students and instructors
- Login/Logout to the CMS.
- List the courses created by the instructor.
- Create, edit, and delete courses.
- Add modules to a course and re-order them.
- Add different types of content to each module and re-order contents.
- Allow students to enroll into the course and access contents
- Python shell embedded, Python interpreter embedded and NodeJS interpreter
- Instructor and student permissions