

Summer Internship 2013

Name: Roshan Piyush

Project: MOOC Prototype Course

Objective: To create a MOOC(Massive Open Online Course) Prototype to host the course:”Introduction to Android Programming”.

Week 1

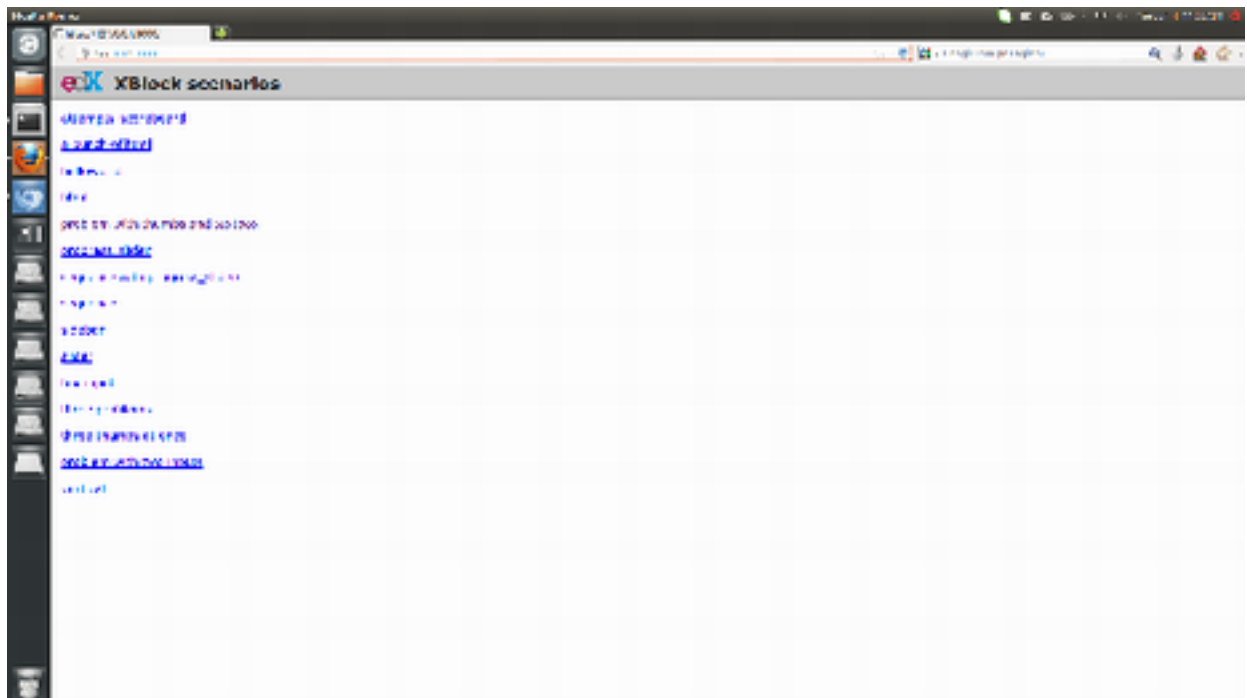
Installed and explored **xblock**.

Set up virtual environment.

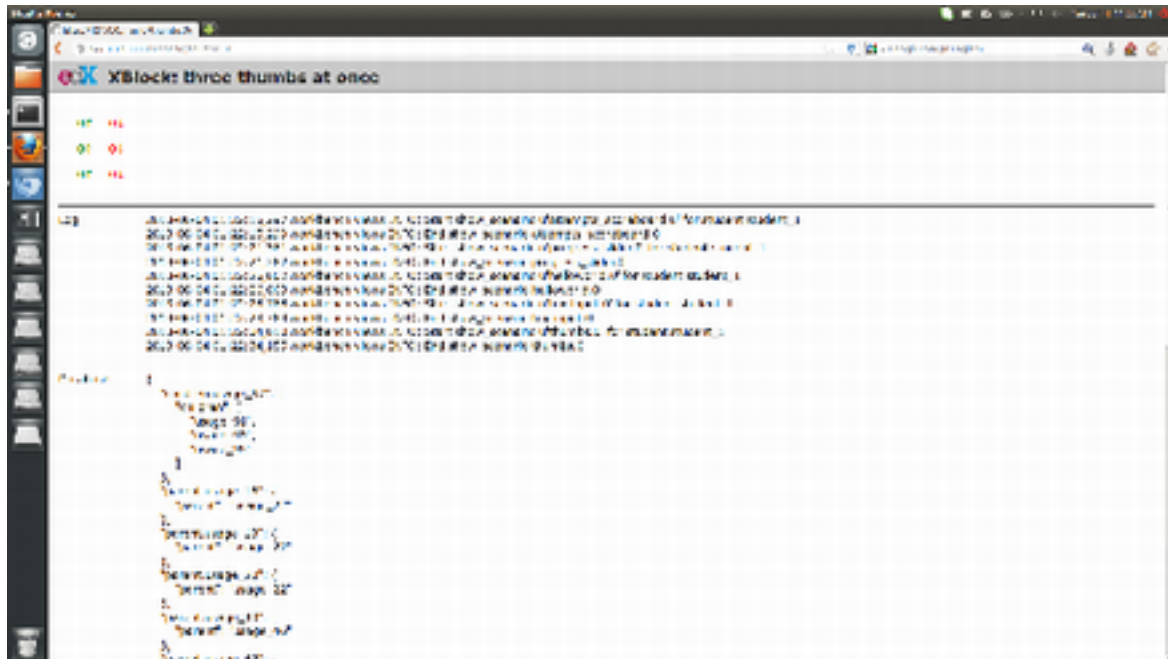
Explored django features.

Explored edx features.

XBlock Screenshot



XBlock Demo



Virtual Environment

To Install

[sudo] pip install virtualenv

Create Virtual Environment

virtualenv ENV

or

virtualenv --distribute ENV

#this will install setup tools

or

virtualenv --system-site-packages ENV

#this will inherit system packages from /usr/lib/python*

##replace ENV with your environment name

To activate ENV

source bin/activate

Now you can install your packages in the virtual environment.

For more references visit <https://pypi.python.org/pypi/virtualenv>

Installed and explored **Class2Go** Mooc platform

Learned youtube api. Created Developer Account for google.

Week2

Took Python basics and Django session on how to create a blog app.

Django Project StartUp Guide

1. Add a new project
`django-admin.py startproject mooc`
2. Create a database mysql. Say “**moocdb**” with say user “**moocuser**” and password “**moocpass**”
3. Add db details to **mooc/settings.py** in mooc project. Uncomment admin and admin docs in Installed apps.

Uncomment the next line to enable the admin:

'django.contrib.admin',

Uncomment the next line to enable admin documentation:

'django.contrib.admin docs'

Add static and media url and files location.

Absolute filesystem path to the directory that will hold user-uploaded files.

Example: "/home/media/media.lawrence.com/media/"

MEDIA_ROOT = '/home/roshan/IITB/mooc/media/'

URL that handles the media served from MEDIA_ROOT. Make sure to use a

trailing slash.

Examples: "http://media.lawrence.com/media/", "http://example.com/media/"

MEDIA_URL = '/media/'

Absolute path to the directory static files should be collected to.

Don't put anything in this directory yourself; store your static files

in apps' "static/" subdirectories and in STATICFILES_DIRS.

Example: "/home/media/media.lawrence.com/static/"

STATIC_ROOT = '/home/roshan/IITB/mooc/static/'

URL prefix for static files.

Example: "http://media.lawrence.com/static/"

STATIC_URL = '/static/'

4. Run **`python manage.py syncdb`** to sync the database models.
Uncomment admin instructions from views.py in mooc directory to enable admin.

from django.contrib import admin

admin.autodiscover()

Uncomment the admin/doc line below to enable admin documentation:

url(r'^admin/doc/', include('django.contrib.admin docs.urls')),

Uncomment the next line to enable the admin:

url(r'^admin/', include(admin.site.urls)),

5. **`python manage.py runserver`** or
`python manage.py runserver ipaddress:portno`

Tutorials

<https://code.djangoproject.com/wiki/Tutorials>

The simplest way to look at Django is to break it down into its component parts.

models.py file which defines your data model by extrapolating your single lines of code into full database tables and adding a pre-built (totally optional) administration section to manage content.

urls.py file which uses regular expressions to capture URL patterns for processing.

views.py in which the actual processing happens in your views which, if you haven't seen the pattern yet. This is really the meat of Django, since views are where you grab the data you're presenting to the visitor.

Here's what happens when a visitor lands on your Django page:

1. First, Django consults the various URL patterns you've created and uses the information to retrieve a view.
2. The view then processes the request, querying your database if necessary.
3. The view passes the requested information on to your template.
4. The template then renders the data in a layout you've created and displays the page.

Database requirements

Models for database of Courses.

```
from django.db import models
```

```
# Create your models here.
```

```
class Department(models.Model):
    deptCode = models.CharField(max_length = 10)
    deptName = models.CharField(max_length = 200)
```

```
class Course(models.Model):
    department = models.ForeignKey(Department)
    name = models.CharField(max_length = 200)
    code = models.CharField(max_length = 20)
    startDate = models.DateField()
    duration = models.PositiveIntegerField()
    university = models.CharField(max_length = 200)
    instructor = models.CharField(max_lenth = 200)
```

```
class Week(models.Model):
    course = models.ForeignKey(Course)
    weekNo = models.SmallIntegerField()
    readingMaterial = models.FileField(upload_to = "/uploads/file", max_length=200)
    problemSet = models.FileField(upload_to = "/uploads", max_length=200)
```

```
class Lecture(models.Model):y
    week = models.ForeignKey(Week)
    name = models.CharField(max_length= 200)
    noOfQ = models.SmallIntegerField()
    videold = models.CharField(max_length = 20)

class Quiz(models.Model):
    lecture = models.ForeignKey(Lecture)
    title = models.CharField(max_length = 200)
    noOfQ = models.SmallIntegerField()

class Question(models.Model):
    quiz = models.ForeignKey(Quiz)
    ques = models.TextField()
    flagImg = models.BooleanField()
    img = models.ImageField(upload_to="/uploads/image")
    noOfOptions = models.SmallIntegerField()
    ans = models.SmallIntegerField()

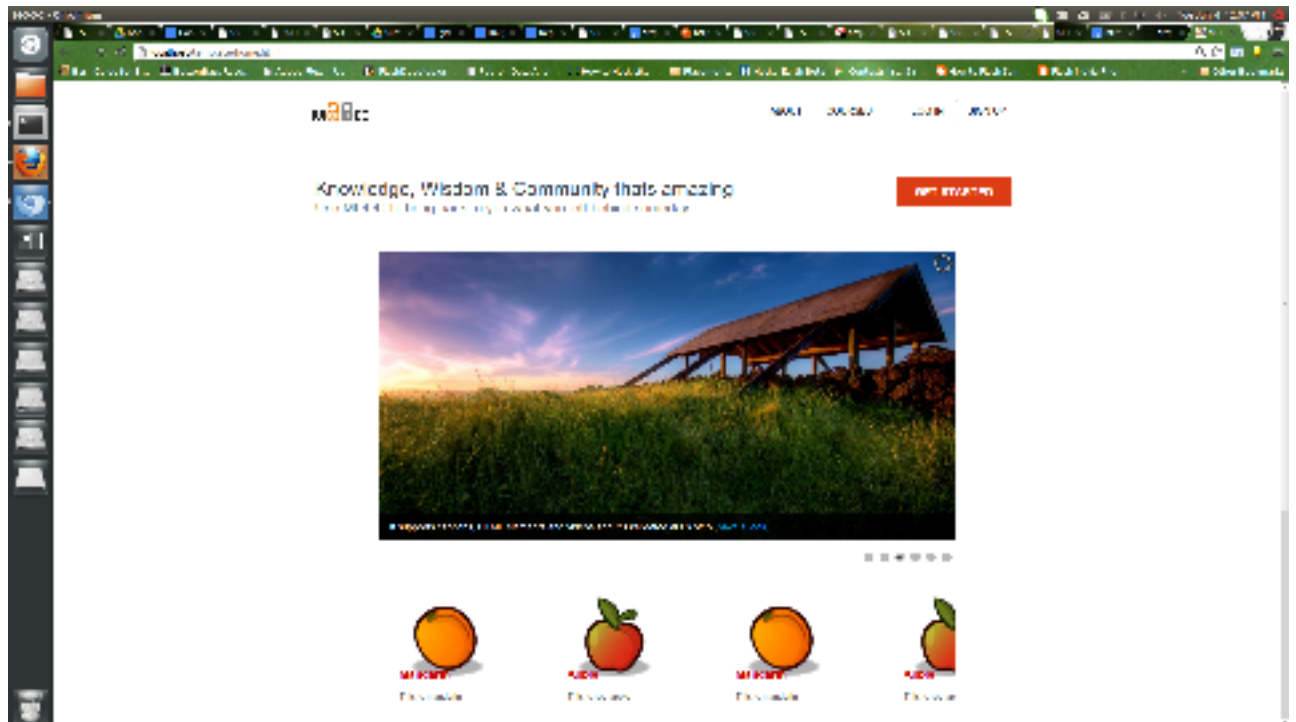
class Options(models.Model):
    ques = models.ForeignKey(Question)
    text = models.TextField()
```

Week3

Went through bootstrap api.

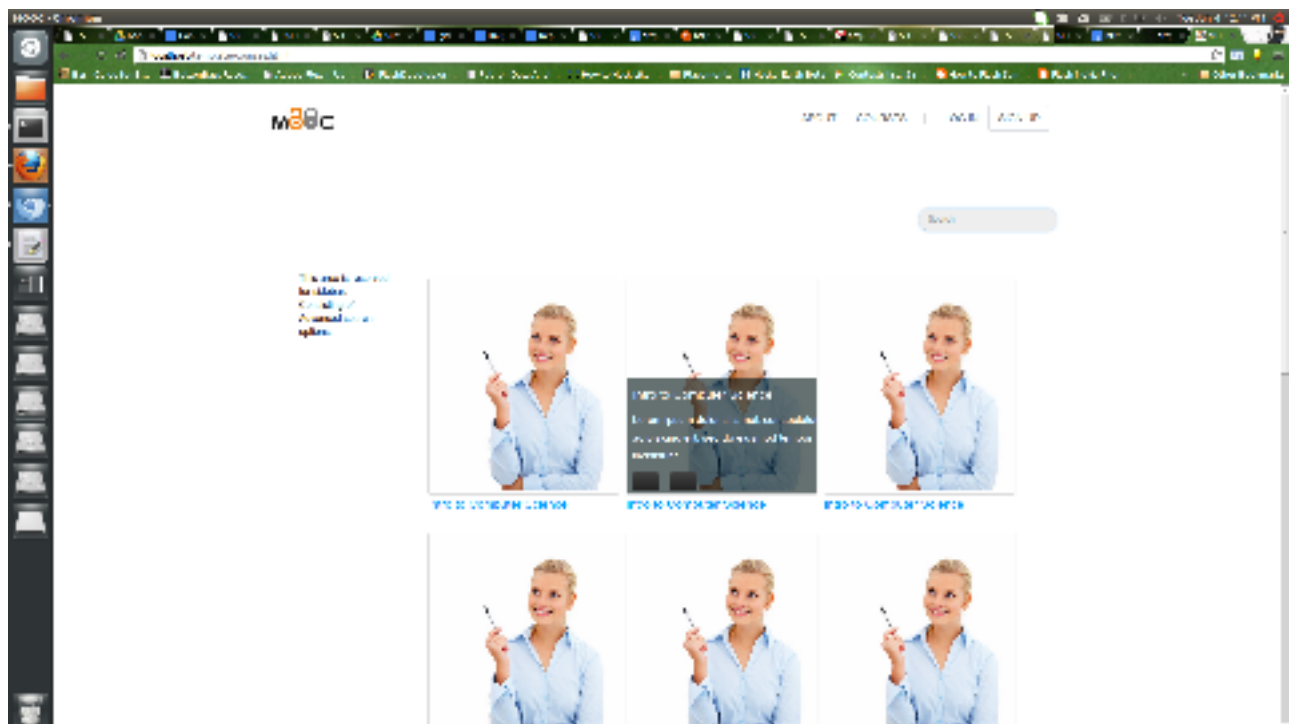
Designing templates i.e. header footer and homepage with Anushree.

Home



Designed courses, courseware and introduction page.

Courses



Github repo

<https://github.com/anushreejangid/mooctemplate>

Week 4

Setting up edx-studio platform.

Trying to understand the structure of edx-studio.

Fixing scripts for latest version of ubuntu.