**MOOC Prototype Course**

**Summer Internship 2013**

**Submitted in fulfillment of internship project**

**By**

**MOOC Team**

Under the Guidance of

**Prof. D. B. Phatak**

****

Department of Computer Science and Engineering

Indian Institute of Technology Bombay

Powai, Maharashtra, INDIA

July 2013

Summer Internship 2013

Project Approval Certificate

Department of Computer Science and Engineering

Indian Institute of Technology Bombay

The project entitled “MOOC Prototype Course” submitted by Mr. Abhilash Kadhane, Mr. Abhinav Puri, Ms. Anushree Jangid, Mr. Dhruva Bhaswar, Ms. Divya Pitta, Mr. Kushal Yarlagadda, Mr. Mundru Mahendra and Mr. Roshan Piyush is approved for Summer Internship 2013 Programme from 9th May 2013 to 6th July 2013, at Department of Computer Science and Engineering, IIT Bombay

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Deepak B. Phatak Mr. Parag Tiwari

Dept of CSE, IITB Dept of CSE, IITB

Principal Investigator Project In-charge

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Prakash L. Vaidya

Dept of CSE, IITB

External Examiner

Place: IIT Bombay, Mumbai

Date : 2nd July 2013

**Acknowledgement**

We, the summer interns of MOOC prototype, whole heartedly thank Prof. D. B. Phatak for providing us with this wonderful opportunity to work under his esteemed project. We would like to express our sincere gratitude to our Project Manager Technical Mr. Parag Tiwari for his kind co-operation. We are indebted to our mentors- Mr. Ajay Babar, Mr. Vivek Nadar, Mr. Arun Nair and Mr. Praveen Pal for their valuable suggestions, guidance and encouragement. We would also like to thank Mr. Rahul Kharat and Mr. Dilip Sable for making our stay in IIT Bombay comfortable and for all the administrative help. Finally, we would like to thank all our friends in Summer Internship for helping us out whenever required.

**Team Members**

**Interns:**

|  |  |  |
| --- | --- | --- |
| **Name** | **College** | **E-mail** |
| Abhilash Kadhane | VNIT, Nagpur | abhilashkadhane@yahoo.com |
| Abhinav Puri | IIT Ropar | pabhinav@iitrpr.ac.in |
| Anushree Jangid | BIT, Jaipur | anushreejangid@gmail.com |
| Dhruva Bhaswar | BIT, Mesra | dhruvabhaswar@gmail.com |
| Divya Pitta | IIT Jodhpur | pittadivya@iitj.ac.in |
| Kushal Yarlagadda | IIT Kanpur | kushaly@iitk.ac.in |
| Mundru Mahendra | RGU-IIIT RK Valley | mahendra.mundru.cse@gmail.com |
| Roshan Piyush | NIT, Calicut | piyushroshan@gmail.com |

**Project Manager (Technical):** Parag Tiwari

**Project Mentors:** Ajay Babar, Vivek Nadar, Praveen Pal and Arun Nair

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited from whom proper permission has not been taken when needed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Abhilash Kadhane, VNIT Nagpur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Abhinav Puri, IIT Ropar

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Anushree Jangid, BIT, Jaipur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Dhruva Bhaswar, BIT, Mesra

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ms. Divya Pitta, IIT Jodhpur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Kushal Yarlagadda, IIT Kanpur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Mundru Mahendra, RGU-IIIT RKValley

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Roshan Piyush, NIT, Calicut

**ABSTRACT**

**IITBX** is a Massive Online Open Course (MOOC) being built on the base of edX-platform. As its name suggests, the two main features of IITBX are open access and ability to support large scale enrollments. Its aim is to provide good quality online education to one and all. It consists of two parts: Learning Management System (LMS) and Content Management System (CMS).

A student who wants to register on IITBX and enroll for a course will be looking at the LMS part of it. Every course consists of several video lectures (with subtitles and transcripts), quizzes, assignments, student’s progress report, discussion forums and wiki. An instructor sees and works with the CMS part of it. He/she makes use of the several features available to make the course as interactive and engrossing as possible.

Table of Contents

[1 Introduction 11](#_Toc360445547)

[1.1 Purpose 11](#_Toc360445548)

[1.2 Project Scope 11](#_Toc360445549)

[1.3 Intended Audience 11](#_Toc360445550)

[1.4 Definitions, Acronyms and Abbreviations 12](#_Toc360445551)

[1.5 References 12](#_Toc360445552)

[1.6 Overview 12](#_Toc360445553)

[2 Overall Description 12](#_Toc360445554)

[2.1 Product Perspective 12](#_Toc360445555)

[2.2 Product Functionality 13](#_Toc360445556)

[2.3 User Characteristics 14](#_Toc360445557)

[2.4 Operating Environment 14](#_Toc360445558)

[2.5 Design and Implementation Constraints 14](#_Toc360445559)

[3 Specific Requirements 15](#_Toc360445560)

[3.1 External Interface Requirements 15](#_Toc360445561)

[3.2 Functional Requirements 19](#_Toc360445562)

[3.2.1 Sign up 19](#_Toc360445563)

[3.2.2 Course enrollment 19](#_Toc360445564)

[3.2.3 View courseware 20](#_Toc360445565)

[3.2.4 Downloadable videos 20](#_Toc360445566)

[3.2.5 Take a quiz 20](#_Toc360445567)

[3.2.6 Discussion forum 20](#_Toc360445568)

[3.2.7 View grades 20](#_Toc360445569)

[3.2.8 Wiki-based collaborative learning 20](#_Toc360445570)

[3.2.9 Calendar based scheduling 20](#_Toc360445571)

[3.2.10 Find Courses 20](#_Toc360445572)

[3.2.11 Progress report 20](#_Toc360445573)

[3.2.12 Online test and assignments 21](#_Toc360445574)

[3.2.13 Video transcripts 21](#_Toc360445575)

[3.2.14 Merge wiki-edit versions 21](#_Toc360445576)

[3.2.15 Sign up and login (instructor) 21](#_Toc360445577)

[3.2.16 New course 21](#_Toc360445578)

[3.2.17 Schedule course 21](#_Toc360445579)

[3.2.18 Course overview 21](#_Toc360445580)

[3.2.19 Grading 21](#_Toc360445581)

[3.2.20 Create assignment 21](#_Toc360445582)

[3.2.21 Add/delete instructors 22](#_Toc360445583)

[3.2.22 Course updates 22](#_Toc360445584)

[3.2.23 Add static pages 22](#_Toc360445585)

[3.2.24 Import course 22](#_Toc360445586)

[3.2.25 Export a course 22](#_Toc360445587)

[3.3 Behavior requirements 23](#_Toc360445588)

[3.3.1 Use case view 23](#_Toc360445589)

[3.3.2 Activity diagrams 25](#_Toc360445590)

[4 Other non-functional requirements 28](#_Toc360445591)

[4.1 Performance requirements 28](#_Toc360445592)

[4.2 Safety and Security Requirements 28](#_Toc360445593)

[4.3 Software quality attributes 29](#_Toc360445594)

[5 User Manual 30](#_Toc360445595)

[5.1 CMS user manual 30](#_Toc360445596)

[5.1.1 Link for CMS documentation 30](#_Toc360445597)

[5.1.2 CodeJail Configuration 30](#_Toc360445598)

[5.1.3 Email Integration 32](#_Toc360445599)

[5.1.4 Discussion Forum 33](#_Toc360445600)

[5.2 LMS user manual 34](#_Toc360445601)

[5.2.1 Introduction 34](#_Toc360445602)

[5.2.2 Purpose 34](#_Toc360445603)

[5.2.3 How to use Learning Management System of IITBX website 34](#_Toc360445604)

[5.2.4 Registration for IITBX account 34](#_Toc360445605)

[5.2.5 Registration Page 36](#_Toc360445606)

[5.2.6 Instructions 37](#_Toc360445607)

[5.2.7 Finding Course 37](#_Toc360445608)

[5.2.8 Registering a Course 38](#_Toc360445609)

[5.2.9 Accessing Course Contents 39](#_Toc360445610)

[5.2.10 Multiple Choice Questions 40](#_Toc360445611)

[5.2.11 Usage of Buttons on the bar 41](#_Toc360445612)

[5.2.12 Multiple components in the same unit 41](#_Toc360445613)

[5.2.13 To go to profile 42](#_Toc360445614)

[5.2.14 To change password and full name and Email 42](#_Toc360445615)

[5.2.15 Logout 43](#_Toc360445616)

[5.2.16 Help, Contact, Team 43](#_Toc360445617)

[5.2.17 FAQ 43](#_Toc360445618)

[5.2.18 About page 44](#_Toc360445619)

[5.2.19 Contact Page 44](#_Toc360445620)

[6 Bibliography 45](#_Toc360445621)

# Introduction

IITBX is a MOOC (Massive Online Open Course) platform created by IITB on the base of edX Studio (edX’s **C**ontent **M**anagement **S**ystem) and edX’s **L**earning **M**anagement **S**ystem.

## Purpose

The purpose of this document is to provide a detailed description of the MOOC platform. It will explain the purpose and features of the platform, the interfaces of the platform, the constraints under which it will have to operate and the technologies used to develop this platform. It describes all the functional as well as non-functional requirements to develop, use or work on this platform.

## Project Scope

This platform has been designed around the key principles of diversity, openness, interactivity and autonomy. It enables access to world-class faculty and an unconventional curriculum. Not only does a course contain video lectures and quizzes but also discussion forums for better learning through communication. The quizzes are analyzed either by the in-built technologies or through peer-to-peer grading which in turn helps the students in acquiring extra skills apart from the course curriculum. Though MOOCs are still in experimental stages, participation of many motivated students could really bring about a change in our conventional education system.

## Intended Audience

This document is intended for the developers, testers, project managers and most importantly the users of this platform.

The developer will use it to stick to the platform’s actual purpose and not make the system inconsistent with its requirements before starting any work on developing the platform. The user will use this document to check if he/she has all the required software and specifications before installation and use.

## Definitions, Acronyms and Abbreviations

* CMS: Content Management System
* LMS: Learning Management System
* SQL: Structured Query Language
* GUI: Graphical User Interface
* HTTP: Hyper Text Transfer Protocol
* HTTPS: HTTP utilizing TCP
* RAM: Random Access Memory

## References

[1] http://files.edx.org/Getting\_Started\_with\_Studio.pdf

[2] https://people.csail.mit.edu/ichuang/edx/

## Overview

The SRS will mainly comprise two sections, namely:

**Overall Description:** This section will describe major components of the system, interconnections, and external interfaces.

**Specific Requirements:** This section will describe the functions of actors, their roles in the system and the constraints faced by the system.

Databases (Mongo + SQL)

# Overall Description

Django Server

## Product Perspective

Client

Data and Views

Templates and other non-static files

Web Server

Static files

Courses are uploaded and edited on CMS by the respective instructor. She/he communicates with the server to access and update information including course statistics and students’ performances. Students are able to enroll and see lectures on the LMS. They access the data, finish their lectures and take the quizzes from the LMS. The data associated with each user is updated respectively in the database for the changes to be available to any client i.e. both instructor and student in the future.

## Product Functionality

The major features that are included in this platform are:

* Open-source code
* Multi-platform support (preferably Ubuntu 12.04 onwards and Mac OS X)
* Open access to all free of cost
* Scalability: This platform is designed for a large number of students.
* Account for each user and login necessary to enroll in a course. Separate data maintained for each user to allow him to start from where he/she left the last time every time he/she logs in.
* IITBX Studio/CMS allows instructor to edit each and every part of the course and its structure along with an option to import/export courses.
* Courseware which includes all the course material including videos, quizzes, exams, etc.
* Video lectures can be uploaded on Youtube with various speeds and the links can then be used in CMS to embed these videos in a course.
* Video lectures along with subtitles and transcript whose speed can be adjusted depending on the student.
* Quizzes and assignments with deadlines. Other than this, course material and few books are provided depending on the course.
* News and updates from the instructor regarding the course.
* Calendar that shows the course structure and important dates.
* Discussion forums for each topic to encourage students to take part in active discussions and clarify their doubts.
* Wiki: A collaborative space for students to share their knowledge and skills.
* Search option: Search course by name or search for all the courses provided by a certain university.
* Notifications through email
* Progress Report updated along with the progress of the user in the course.

## User Characteristics

This platform has been designed to be used by an instructor and a large number of students. An instructor uses this platform to add, delete or edit courses using all the features that are available via Studio (CMS). He/she needs to have an idea about the basic features of CMS or else one can learn it easily with the user manual as it has a very interactive UI. [1] Students have access to the content via LMS and it has been designed such that a large number of students must be able to access the same data simultaneously if required. A student must have basic knowledge with surfing the web to be able to do this without much trouble.

## Operating Environment

A client may use this platform easily on the following operating systems:

* Ubuntu 12.04 or higher
* Apple Mac OS X

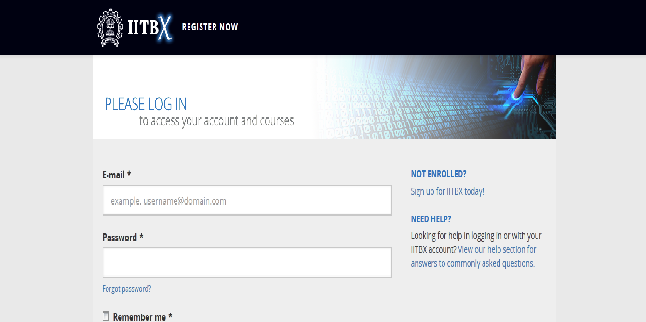
## Design and Implementation Constraints

* The GUI is available only in English.
* Only registered users can access the course content.
* This is currently available on a single server.
* It is limited to HTTP/HTTPS.
* Currently there is no support for installing this platform on Fedora.
* For installation on Windows, a virtual environment is required. [2]
* This is meant for a large number of users and deals with very large data, hence an appropriate server and DBMS is required for smooth running. On the client side, one requires some space on the hard disk and a good enough RAM which is now available on almost all the systems.

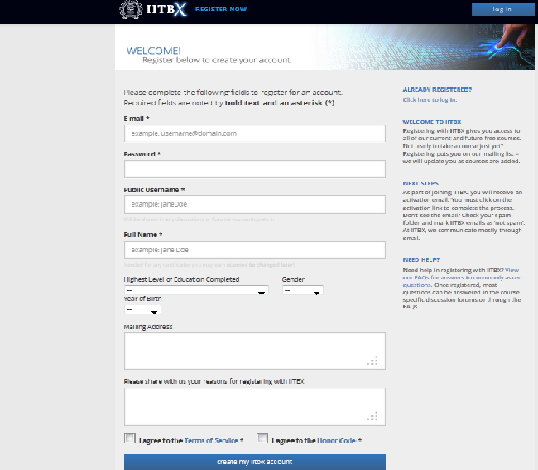
# Specific Requirements

## External Interface Requirements

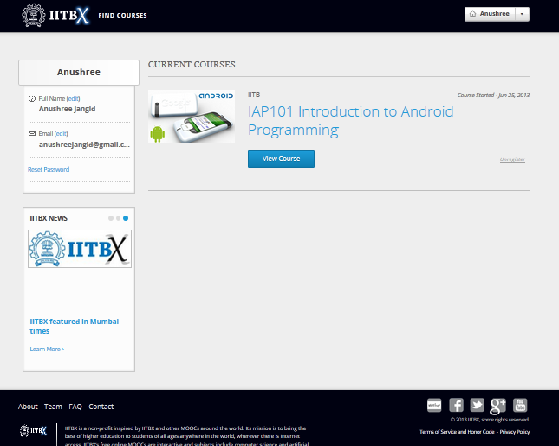
### Login interface



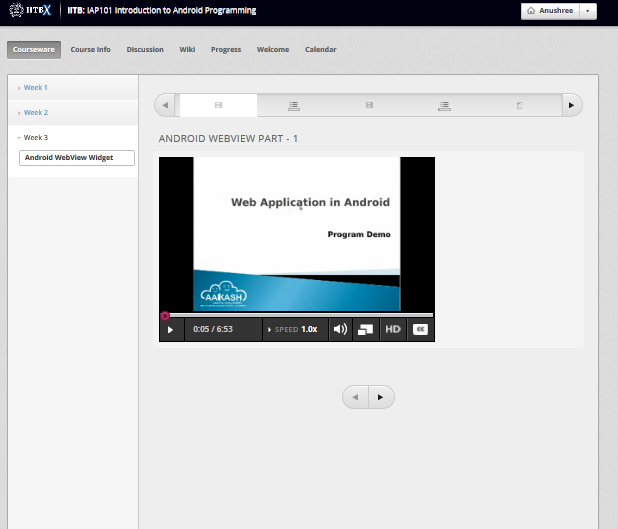
### Registration interface



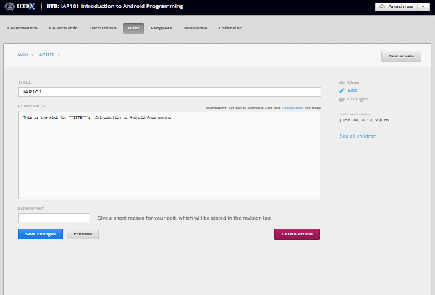
### User Dashboard



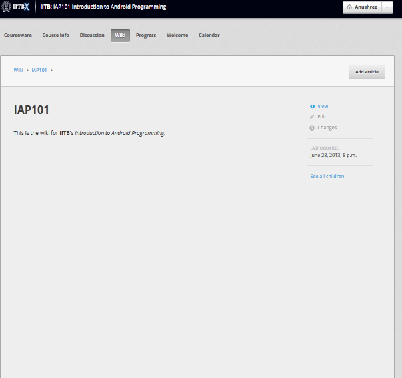
### Courseware interface



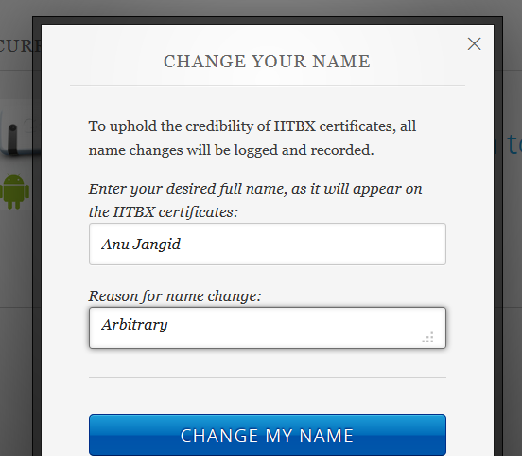
### Wiki-edit interface



#### Wiki-view interface



### Edit information interface



## Functional Requirements

The functional requirements of LMS are as follows:

### Sign up

The user can sign up on the website to enroll into a course.

### Course enrollment

The user can register for any course of interest which is currently going on or is scheduled for future. The user may re-take IITBX courses as often as he/she wish. Each offering of a course is assessed independently. The user may unregister from an IITBX course at any time, there are absolutely no penalties associated with incomplete IITBX studies, and may register for the same course (provided we it is still offered) at a later time.

### View courseware

After enrolling into a course, user can see the ongoing activities in the course. The registered user can view lecture videos available for the course, solve problem sets, assignments etc.

### Downloadable videos

The registered user can download the videos in many formats as available for the course.

### Take a quiz

The registered user can take up the quiz as available for the course.

### Discussion forum

The registered user can use this forum to connect with fellow students, professors and teaching assistants. Also he/she can search for a particular post, edit previous posts, like the comments/ posts by fellow members etc.

### View grades

Once the course is finished and grades are released the registered use can view his/her grades.

### Wiki-based collaborative learning

The registered user can wiki edit some of the contents (on the basis of the rights provided by the author), add articles

### Calendar based scheduling

Each course is scheduled i.e. it has a start and finish date. Few assignments have deadlines. All this is managed by the course instructor.

### Find Courses

The user can search for the courses according to the university or simply local search engine by using keywords.

### Progress report

The registered user can check his/her progress report on the basis of test assessments and assignment evaluation on the basis of homework, lab assignments, midterm &final exams with the help of graphs and numerical scores.

### Online test and assignments

The registered user can take tests and online assessments for the course for which he/she has registered.

### Video transcripts

The registered user can use the facility of video transcripts to skip parts of videos which he/she doesn’t want. He/she can also download the transcripts.

### Merge wiki-edit versions

The registered user can use this feature to merge concurrent wiki-edits.

The functional requirements of CMS are the following:

### Sign up and login (instructor)

This feature is the same as that in LMS feature 3.2.1

### New course

The registered instructor can create a new course in which he/she can specify the course name, course organization and course number. In the course he/she can create new sections and subsections, set the course release date or view the course live.

### Schedule course

The registered instructor can schedule the course by specifying the course start and end date, time and the enrollment start and end date time.

### Course overview

The registered instructor can provide the course overview of the course created by him in which he/she can specify course description, prerequisites, course-staff and other information.

### Grading

The registered instructor can define the grading rules and policies for the course authored by him/her.

### Create assignment

The registered instructor can define the type of assignment for e.g. homework, midterm exams etc., its weightage and the number of assignments.

### Add/delete instructors

The registered instructor can add/delete users to manage the course team.

### Course updates

The registered instructor can make announcements or notifications that he/she wants to share with the class. Other course authors can them for important exam/date reminders, change in schedules, and to call out any important steps students need to be aware of.

### Add static pages

The registered instructor can add static pages. Static Pages are additional pages that supplement courseware. Other course authors can use them to share a syllabus, calendar, handouts, and more.

### Import course

The registered instructor can import a course in gzipped format (tar.gz) and must contain a minimum of course.xml file.

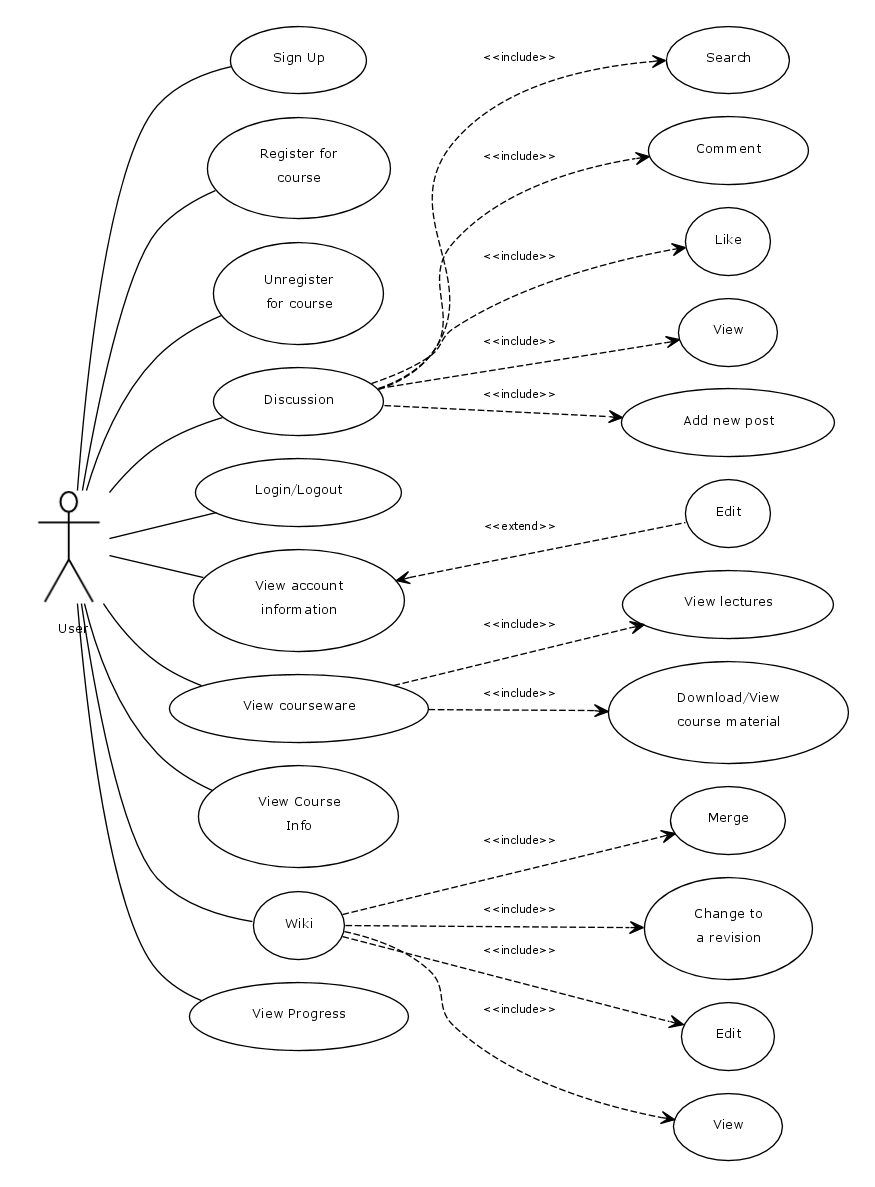
### Export a course

The registered instructor can export the course designed by him/her in gzipped format.

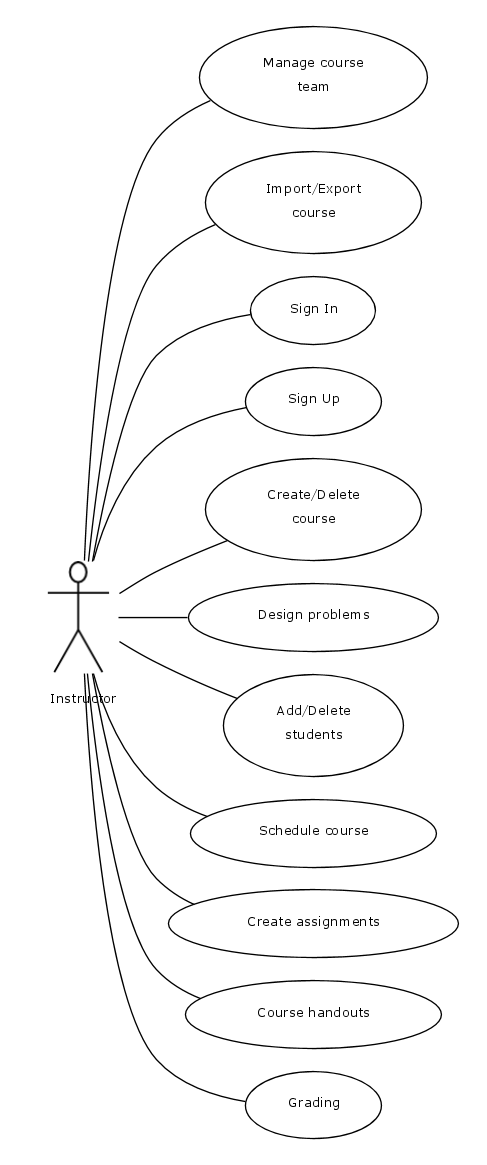
## Behavior requirements

### Use case view

#### User



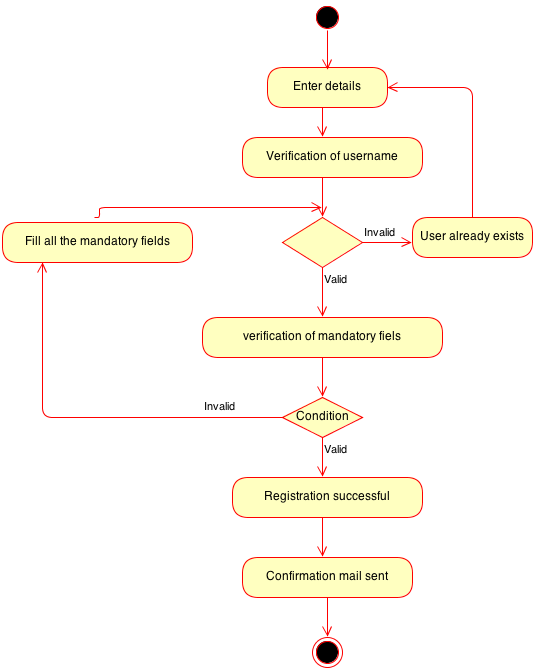
#### Instructor use case



### Activity diagrams

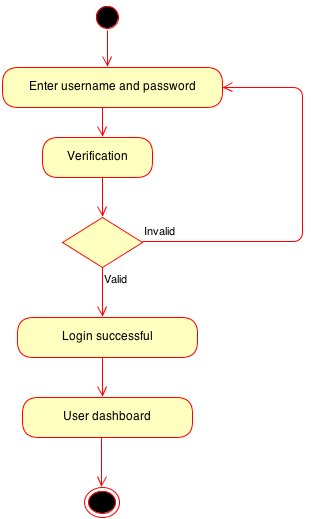
#### User registration

Initially user is made to fill all mandatory fields filled in registration form. Once the user clicks create an IITBX account, the username is verified. If the username is already present, then the user is again taken back, so that he can change the username. If the username is not present then it checks for password and remaining mandatory fields. If any of the mandatory field is left empty or filled incorrect, then the user is informed to enter the correct values. Once all these verifications are succeeded, then the registration is done and a confirmation mail is sent.

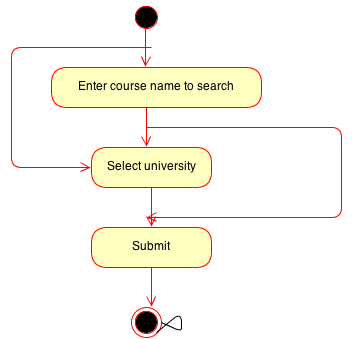


#### User login activity

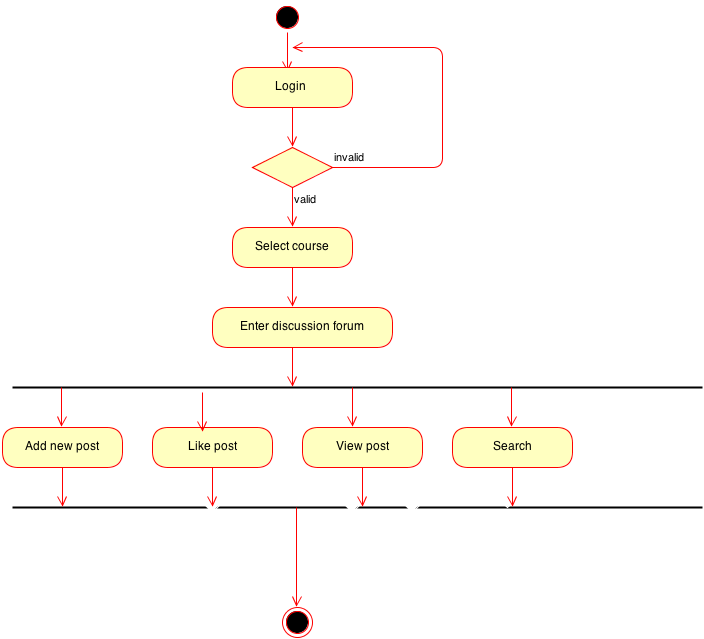
User is made to enter the username and password, and then entered values are verified. If it is a valid username and password, then the user is logged in, or else he/she is asked to re-enter the values.



#### Find courses



#### Discussion forum activity



#### Check progress activity



# Other non-functional requirements

## Performance requirements

Since it is an online platform, hence the delay involved in system interaction must be less else the user will feel that the system is down. In case of JavaScript interactions the delay involved is less than 3 seconds. The sorting and searching of course takes place in less than 1 second.

## Safety and Security Requirements

The form submissions are validated by csrf tokens, hence protecting it from spammers. User authentication is done by encrypting the session login variable. Users are required to confirm their identity by responding to a confirmation mail. Each service integration is protected by secret key. Thus, there is no possibility of intrusion.

## Software quality attributes

* **Availability**

Availability means more than just being up and running 24/7/365, availability also means that the web software must be available when accessed by diverse browsers. Hence, the application must be accessible in its original form across multiple browsers of all platforms.

* **Reliability**

If web software is unreliable, websites that depend on the software will lose users. Grading of students is crucial data and the software must be reliable to keep them intact. User submissions are like transactions as they affect the grades of the students, hence it should have quality testing measures.

* **Scalability**

The system should be highly scalable and modular in design to alter features as required

# User Manual

## CMS user manual

### Link for CMS documentation

[http://help.](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[edge](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[.](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[edx](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[.org/kb/beta-users/](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[edx](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)**[-studio-documentation](http://help.edge.edx.org/kb/beta-users/edx-studio-documentation)

### CodeJail Configuration

* Unsafe code cannot be allowed to run on a server otherwise the hackers may run malicious codes to cripple the server. Codejail basically provides the facility of running unsafe code in secured sandboxes.
* By default the Codejail is configured for Python code execution but with little modification it can be configured to execute code in any language.
* The security is imposed with the aid of AppArmor but if the Operating System does not support AppArmor it won't impose any security measures.

Configuration:

* Create a new virtual environment alongside the main project’s virtual environment but the name of the virtual environment being appended with “-sandbiox”. $ sudo virtualenv <SANDENV> If virtual environment wrapper is installed use

$ mkvirtualenv <SANDENV>

e.g. mkvirtualenv mooc-edx

* If some specific packages need to be made available to the sandbox install them in the virtual environment.

$ source <SANDENV>/bin/activate

$ sudo pip install -r sandbox-requirements.txt

If wrapper is installed then

$ workon <SANDENV>

$ sudo pip install -r sandbox-requirements.txt

* Add the user sandbox- the under priviledged user who actually executes the code in the sandbox.

$ sudo addgroup sandbox

$ sudo adduser --disabled-login sandbox --ingroup sandbox

* Let the web server run the sandboxed Python as Sandbox. Create the file /etc/sudoers.d/01-sandbox using

$ visudo -f /etc/sudoers.d/01-sandbox

<WWWUSER> ALL=(sandbox) NOPASSWD:<SANDENV>/bin/python  
 <WWWUSER> ALL=(ALL) NOPASSWD:/usr/bin/pkill

Note: <WWWUSER> is the user running the LMS

* Edit an AppArmor profile. This is a text file specifying the limits on the sandboxed Python executable. The file must be in /etc/apparmor.d and must be named based on the executable, with slashes replaced by dots. For example, if your sandboxed Python is at /home/el3ktr0n/.virtualenvs/mooc-sandbox/bin/python, then your AppArmor profile must be /etc/apparmor.d/home.el3ktr0n..virtualenvs.mooc-sandbox.bin.python:

$ sudo vim /etc/apparmor.d/home.el3ktr0n..virtualenvs.mooc-sandbox.bin.python  
  
#include <tunables/global>  
  
<SANDENV>/bin/python {  
   #include <abstractions/base>  
   #include <abstractions/python>  
  
   <SANDENV>/\*\* mr,  
   # If you have code that the sandbox must be able to access, add lines  
   # pointing to those directories:  
   /the/path/to/your/sandbox-packages/\*\* r,  
  
   /tmp/codejail-\*/ rix,  
   /tmp/codejail-\*/\*\* rix,  
}

* Parse the profile

$ sudo apparmor\_parser <APPARMOR\_FILE>

* Reactivate the main project’s virtual environment.

Integration into Edx:

* Open the file ~/edx\_all/edx-platform/lms/envs/common.py
* Set 'ENABLE\_DEBUG\_RUN\_PYTHON' to True
* Find the “Python Sandbox” section and make the following changes:
  + - Provide the path to the python library of <SANDENV> in python\_bin(e.g. python\_bin: ‘/home/el3ktr0n/.virtualenvs/mooc-sandbox/bin/python’)
    - Provide the course ids as the regular expressions for which you want to allow unsafe code execution.

### Email Integration

The IITBX platform has to send e-mails to users for registration-activation, for changing user’s password and to notify about various course deadlines updates etc.

For the platform, we use the mail server of IIT Bombay. The settings for the same can be found in *common.py* files in the ‘*envs*’ folder for the respective runtime *(cms* or *lms*).

This document will guide you through the process of configuring the mailing settings.

**To change the IITBX studio mailing  settings:**

* + Change your directory to    ~*/edx\_all/edx-platform/cms/envs*
  + Open the file *common.py*.
  + Jump to the part which looks like this.
  + Change the above env variables to the values you wish.
  + Note the  EMAIL\_HOST, EMAIL\_PORT and  EMAIL\_USE\_TLS  should remain the same unless you want to change the smtp server.These are settings for the smtp server.Once the site is live , the smtp server you change to is working , users will not receive any mail .
  + If the you want the emails to appear on the console , then change the EMAIL\_BACKEND variable to *django.core.mail.backends.console.EmailBackend .*With this setting ,mails will not be sent to the user ,instead will appear on the console ,on which the server is running.
  + Change the DEFAULT\_FROM\_EMAIL to whatever from address you want the user to see.It need not be a valid email address.But if it is not ,better let the user know that is a no-reply address.(Let the name be something like [no-reply@iitbx.org](mailto:no-reply@iitbx.org)).There is another downside if you use a unregistered domain name like *iitbx.org* ,then most likely the mails will go into the users spam mail.
  + SERVER\_EMAIL - This is used as the from address for mails to the admin when the server crashes.
  + The admins variable will be used for mailing all the admins at one go. List email addresses of all the admins in here.
  + See that all mails which you expect the user to reply back to should be valid email addresses.

**To change the IITBX LMS  mailing  settings:**

* + Change your directory to    ~*/edx\_all/edx-platform/lms/envs*
  + Open the file *common.py*.
  + Follow the steps we did for the studio mailing settings.

**To change the content of the emails sent to user :**

For LMS:

* + - Change your directory to  *~/edx\_all/edx-platform/lms/templates/emails*
    - Use a text-editor to change the concerned email text.

For IITBX studio:

* + - Change your directory to  *~/edx\_all/edx-platform/cms/templates/emails*
    - Use a text-editor to change the concerned email text.

### Discussion Forum

● Ruby Installation

* http://blog.coolaj86.com/articles/installing­ruby­on­ubuntu­12­04.html
* Visit the above link to install ruby­rvm
* CS\_COMMENT\_SERVICE
* Install cs\_comment\_service
* Requirements: Successfully installed ruby-rvm & ruby1.9.3.Fully functional

edx-platform.

* Install Elastic Search for debugging

● wget --http-user=$http\_user --http-password=$http\_password

https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-0.

90.1.deb

Replace $http\_user with your ldapid and $http\_password with your ldap password

● sudo dpkg –i elasticsearch-0.90.1.deb

● sudo service elasticsearch start

● workon edx-platform

● cd $HOME/edx\_all/

● [git clone https://github.com/edx/cs\_comments\_service.git](git%20clone%20https:/github.com/edx/cs_comments_service.git)

● cdcs\_comments\_service/

● rvmuse1.9.3@cs\_comments\_service--create

● bundle install

● bundle exec rakedb:init

● bundle exec rakedb:seed

Then enter your API key in config/application.ymlfile (A random key say

“0u3ru93u398rq3083u0u83”).Enter the same key in edx-platform/lms/envs/dev.py

to the variable COMMENTS\_SERVICE\_KEY="PUT\_YOUR\_API\_KEY\_HERE"

To run the discussion forum

● ruby app.rb

## LMS user manual

### Introduction

Learning Management System is a part of IITBX platform. IITBX is a massive open on-line course platform created by modifying the existing edX platform.

### Purpose

The main purpose of this document is “how to use IITBX MOOC website LMS to utilize world class video lectures, reading materials and quizzes.”

### How to use Learning Management System of IITBX website

#### Requirements

* He need to have any of the System (PC, Laptop etc.)
* Internet Facility needed.
* IITBX account is necessary to register courses.
* Registration of course is necessary to use course resources.

### Registration for IITBX account

* Open [www.iitbx.org](http://www.iitbx.org/) website to create an account.
* When you open the website it looks as below except red mark.
* Click the REGISTER NOW button to create an account at IITBx website to get access to IITBX resources.

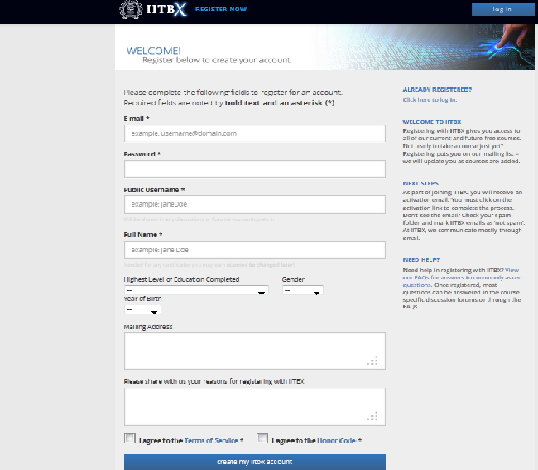
click to register



* When you click on the REGISTER NOW button, you will get a page like shown in

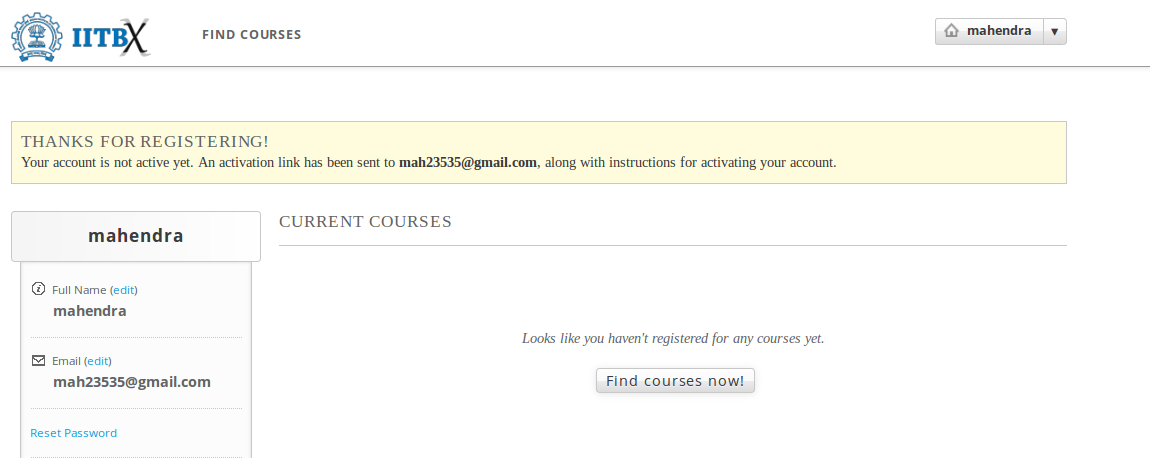
### Registration Page

* Registration page to create an account.



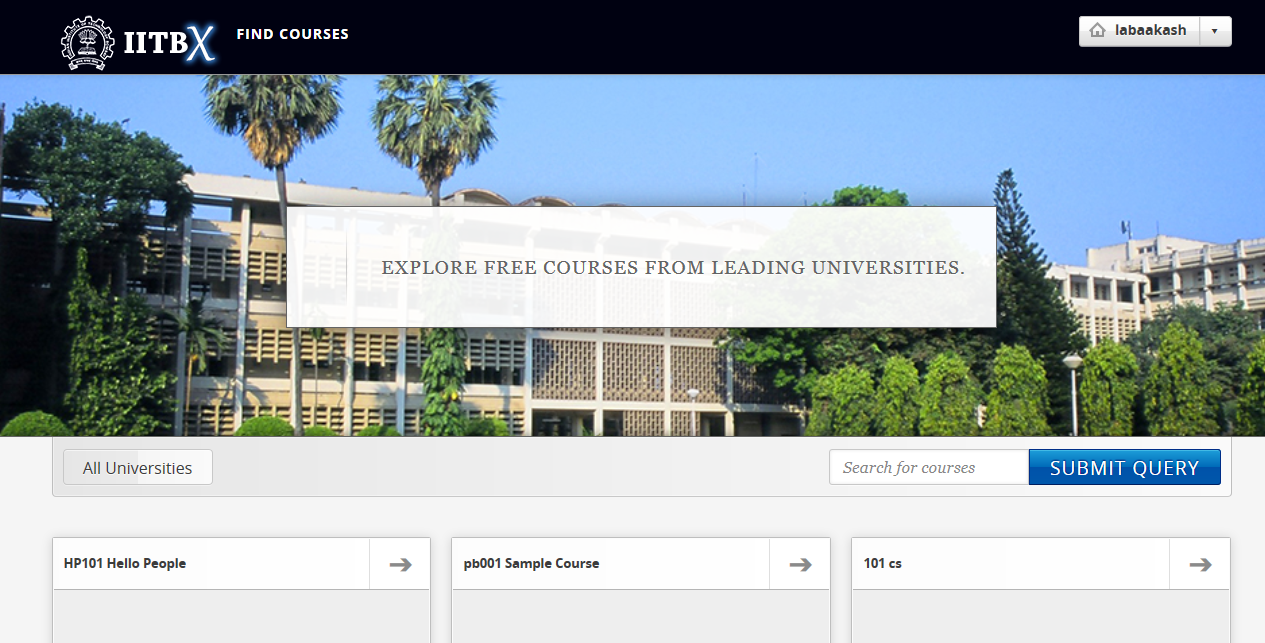
### Instructions

* Enter a valid email address because it will send an activation email to your email account.
* Enter correct details and check the I agree to the Terms of Service and I agree to the Honor Code to continue.
* Click on create my iitbx account so you get an activation link to your entered email
* Then you will get a message saying thanks for registering
* The new page displayed like this



### Finding Course

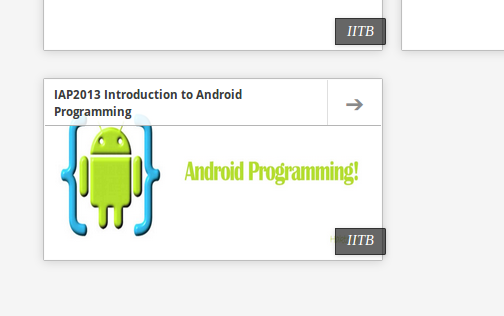
* To see what are the course available click on Find Courses now.
* Then new will be displayed like this.



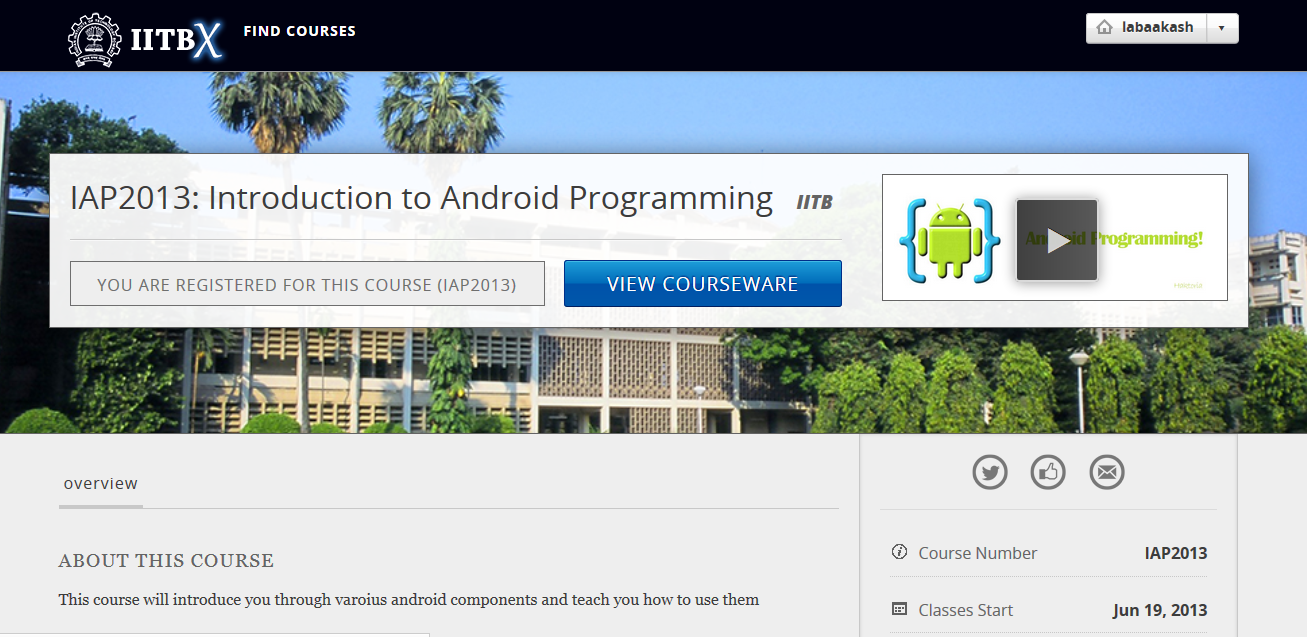
* You can search course in the search box and then click submit Query.
* So it displays all related courses.

### Registering a Course

* Click on the course you are interested.



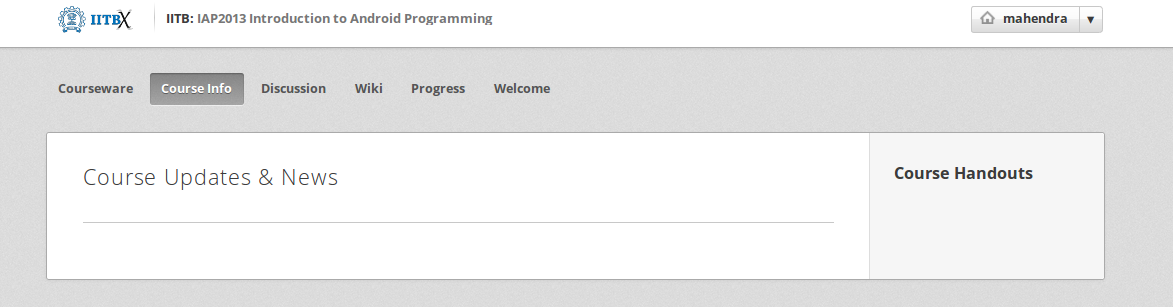
* After clicking on the course you will get a page asking to register that course.



* Click on REGISTER FOR IAP2013 (here )
* Then you are registered the course.
* To view the course Just Click on View Course.



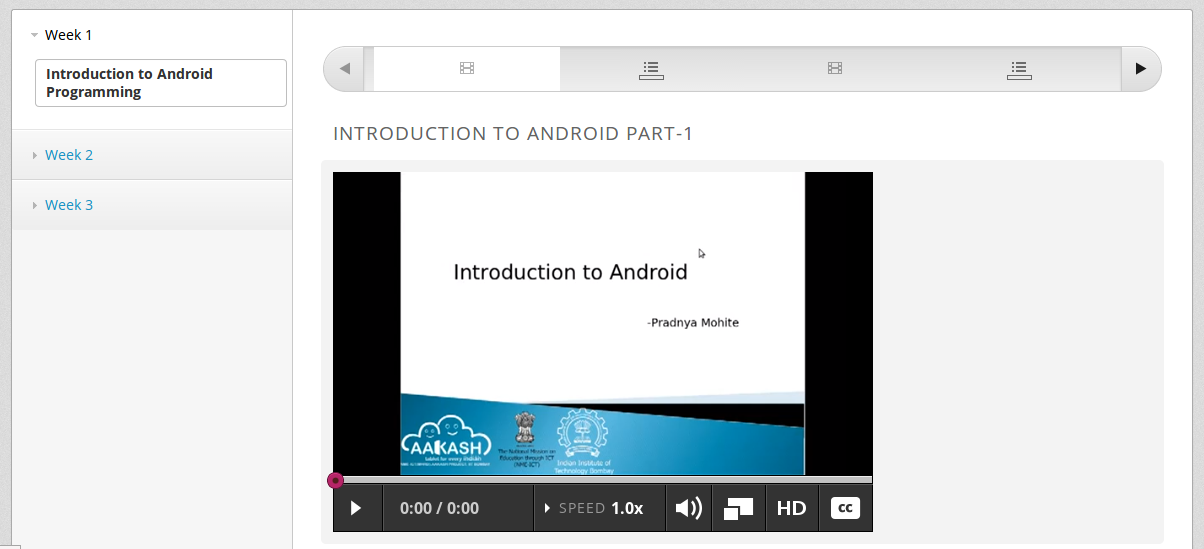
* When you Click on view course to view videos, reading materials and quizsetc
* The page looks like this

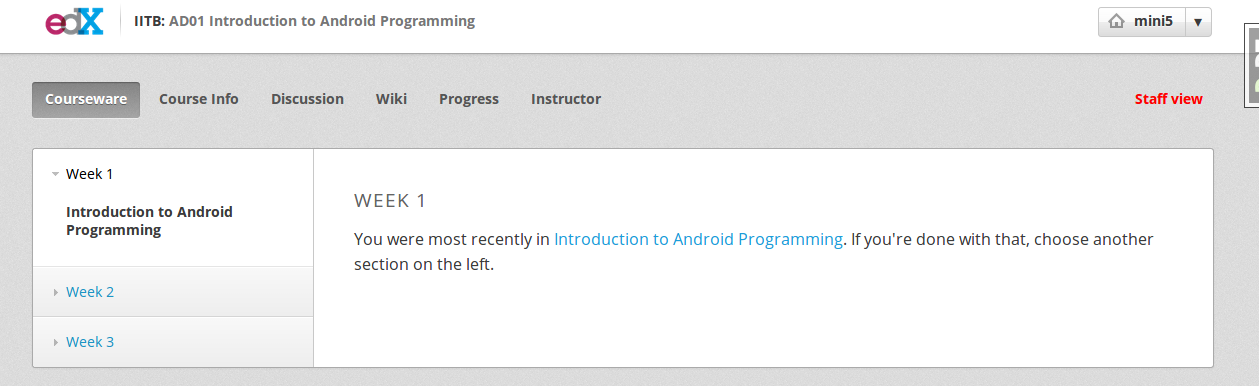


### Accessing Course Contents

* Here click on Course info or It is default which shows any update news and information about the course
* Click on Courseware to view all videos, reading materials etc
* When we click on Courseware it shows the sections of course as week1, week2 and so on.
* And The one that you recently used.
* Click on week1 to get week1 material.
* To get the corresponding things click on its subsections

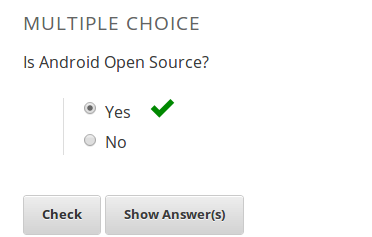
e.g. Introduction to Android Programming here

* Then it shows the units it have e.g. here video



### Multiple Choice Questions

* Click on the option to give your answer
* Then click on check to check your answer



* Click on Show Answer to see the explanation of the answer

### Usage of Buttons on the bar

* When we click on below things the first red mark it will give the video.
* The second marked one gives the Discussion forum.

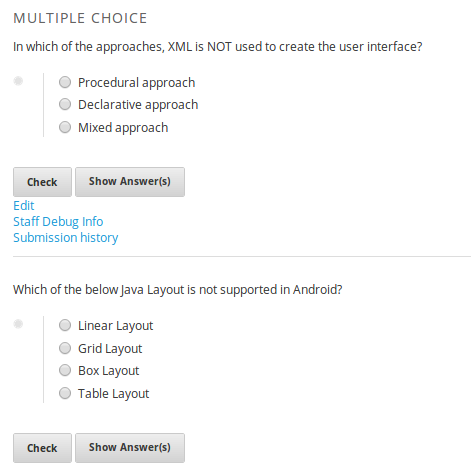
forum

video



### Multiple components in the same unit

* For example, a unit can have one or more components i.e. quiz, video and reading material components can be displayed in same unit.



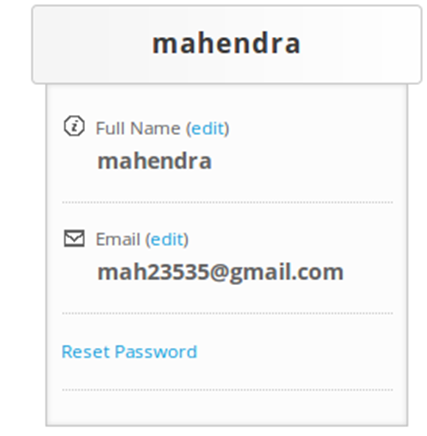
### To go to profile

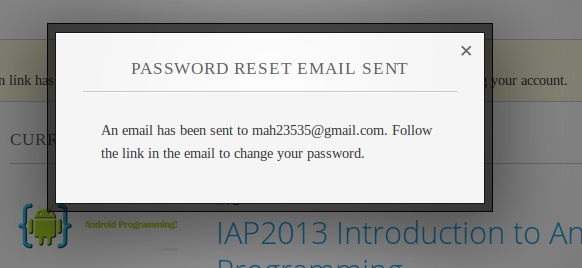
* + First login to your account
  + click on this on top-left corner of page (home button)
  + To go to your profile just click the button which looks like this



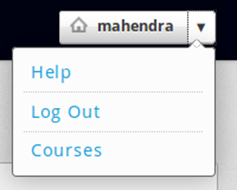
### To change password and full name and Email

* click on reset password (for password change)
* click on edit for changing name or email
* When you click on Reset password a new pop up kind of a thing appears showing some information
* Then check your email and follow the steps in the email



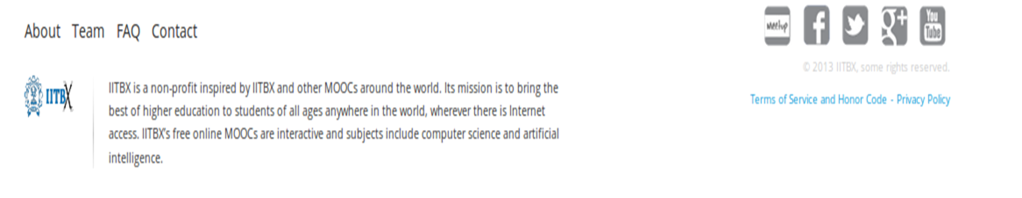


### Logout

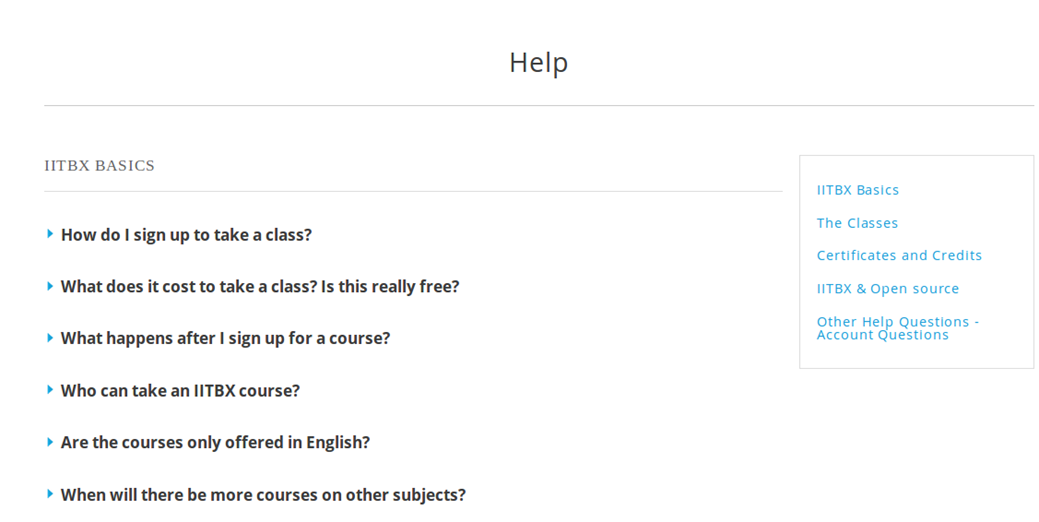


### Help, Contact, Team

* See the bottom of the page which looks like this

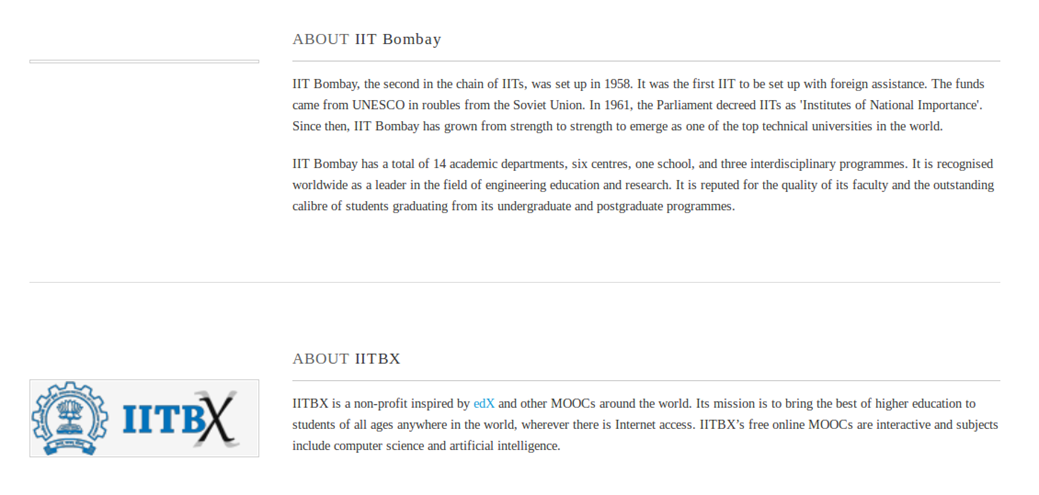


### FAQ



### About page

* When you click on about it will display a page like this
* It gives information about the IITBX



### Contact Page

* When you click on the contact page you will get this page
* You can use the emails to contact respective person



# Bibliography

[1] <https://github.com/edx>

[2] [https://docs.djangoproject.com](https://docs.djangoproject.com/)

[3] <http://docs.python.org>

[4] <http://docs.mongodb.org/manual/>