IITB Summer Internship 2014

Design and build a portal for contributing teaching content for school education

Submitted in fulfillment of internship project

By

Aakash School Education Team

Principal Investigator
Prof. D.B. Phatak

Project In-Charge
Kiran Khosla

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Team Members :

S. No	Name	College	E-mail
1	Priyanja Singh	MNNIT, Allahabad	priyanjasingh15@gmail.com
2	Khushbu Agarwal	NIT, Durgapur	khushbu.ag23@gmail.com
3	Priyamvada Chundawat	Techno India NJR Institute of Technology, Udaipur	pri.chundawat@gmail.com
4	Marati Sukanya	Balaji Institute of Technology and Science, Narsampet	maratisukanya@gmail.com
5	Janaki Veluri	GVIT, Bhimavaram	janaki2594@gmail.com
6	Jeet Banerjee	Vellore Institute of Technology	jeet.23nz@gmail.com
7	Sudheer Macherla	RGUKT IIIT, Basar	sudhirmacherla1993@gmail.com

Team Mentors :

Pradip Tivhale
Sachin Patil

Summer Internship 2014
Project Approval Certificate

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

The project entitled “Design and build a portal for contributing teaching content for school education” submitted by Mr. Jeet Banerjee, Mr. Sudheer Macherla, Ms. Priyanja Singh, Ms. Khushbu Agarwal, Ms. Priyamvada Chundawat, Ms. Janaki Veluri, and Ms. Marati Sukanya is approved for Summer Internship 2014 programme from 10th May 2014 to 6th July 2014, at Department of Computer Science and Engineering, IIT Bombay.

Prof. Deepak B. Phatak	Mrs. Kiran Khosla
Dept of CSE, IITB	Project In-charge
Principal Investigator	Dept of CSE, IITB

Place: IIT Bombay, Mumbai
Date: June 30, 2014

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 adequately cited and referenced the original sources. I also declare that 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 or from whom proper permission has not been taken when needed.

Jeet Banerjee

Sudheer Macherla

Priyamvada Chundawat

Marati Sukanya

Janaki Veluri

Khushbu Agarwal

Priyanja Singh

Acknowledgement

We, the summer interns of Aakash School Education, consider it our privilege to undertake the project of “Design and build a portal for contributing teaching content for school education “. The success and final outcome of this project required the assistance of many people without whom we could not have materialized our vision for the application. We are deeply indebted to **Dr. D.B.Phatak** for giving us the opportunity to be a part of this esteemed project, for giving us inspiration to pursue the project and guiding us in this endeavor. He has been a constant source of motivation and encouragement for us. Our profound sense of gratitude is due to our Project Manager Mrs. Kiran Khosla for constant encouragement and valuable guidance. We thank her for all the initiative and zeal she has filled us with throughout the course of this project. Furthermore, we heartily thank our mentors Mr. Pradip Tivhale and Mr. Sachin Patil with whom the exchange of knowledge and skills helped in making this an enriching experience for us all. We extend a warm thank you to Mr. Rahul Kharat for making our stay here as summer interns comfortable and for all their administrative help. Finally, we would also like to thank the support staff at the institute who have directly or indirectly helped us in the completion of the project.

Abstract

Aakash School Education is committed to offer the students free access to its many academic resources. We seek to capitalize on the potential of the internet to eliminate borders and geographic distance as obstacles to the instantaneous exchange of knowledge and new ideas. Unlike distance learning programs that charge tuition, provide formal instruction and limit participation, we offer all course materials free to everyone with online access. People from around the world may upload their content and students can make use of our extensive library of content from any computer with access to the web. ASE does not require any registration and is not a degree-granting or certificate-granting activity rather it is instead an effort to share knowledge and make sure the students have access to world class education content making full use of the Internet's potential.

webportal Documentation

Release 0.9

interns2014

July 01, 2014

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 Introduction :

Aakash Web Portal is a content management site aimed at providing the students high quality educational content which include interactive animations, videos and reading material about a specific topic. It makes effective use of Information and Communication Technologies to remove barriers in knowledge sharing, presently catering to the 7th, 8th, 9th and 10th classes it has the potential to extend and cater to the needs of all the age groups.

1.1.1 Purpose:

The purpose of this document is to present a detailed description of Aakash Web Portal. This will explain in detail the purpose and features of the system, the interfaces of the portal, what the portal will do, the constraints under which it must operate and how the portal was developed. This document is intended for both the stakeholders and the developers of this system. This document discusses the functional requirements in a software development perspective and it also covers the non-functional requirements like performance, reliability, availability, security, maintainability etc.

1.1.2 Scope:

Aakash Webportal is a free content providing website. The main goal of this application is to bridge the distance between world class education and under previleged students who face difficulty in accessing the best educational content, due to scarcity of trained teachers. The content of our website which will be uploaded by renowed teachers/faculties will be freely available to the students without any charge or registration.

1.1.3 Intended Audience:

Our portal's target audience include:

- Children
- School Teachers
- University Professors
- Anyone with good teaching content.

1.1.4 Definitions, Acronyms and Abbreviations:

- **Portal :** A web portal/portal is most often one specially-designed Web page which brings information together from diverse sources in a uniform way.
- **Python :** Python is a widely used general-purpose, high-level programming language.
- **Django :** Django is a high-level Python Web framework.
- **Sqlite3 :** A standalone program that can be used to create a database.
- **Sphinx :** Sphinx is a tool that makes it easy to create intelligent and beautiful documentation.

1.1.5 Document Conventions:

In general this document aims and prioritizes in writing the requirements for the webportal and analyzing in details the functionalities being provided to its users. Every requirement is having its own priority that are none conflicting. In addition to this quite a few figures are also being provided to make the reader understand the requirements more clearly.

CHAPTER

TWO

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Overall Description:

2.1.1 Product Perspective:

The Aakash Webportal app is supposed to be an open source, under the GNU general Public License, the source code is reposited on Github. It is a web based app implementing the MVC model. The Aakash Web Portal provides simple mechanism for users to share educational content online and gain knowledge.

The main features of the Portal are as follows :

- **Cross platform support :** Offers operating support for most of the known and commercial operating systems.
- **User account :** The system allows the user to create their accounts in the system and provide many features to update and maintain it.
- **Upload :** The system allows a contributor to upload quality content performing checks if it is a proper file with the correct format.
- **Review :** The system allows atleast 3 reviewers to check and review the content before it can be viewed on the site.
- **Downloadable materials :** Allows a user to view and download study material of topics without any registration.
- **Search :** Search is simply local search based on the topic name and subject whatever the user wishes to use.
- **Feedback and comments :** A reviewer can comment and give feedback on a particular topic which the contributor can see.

2.1.2 Product Function's:

The Web portal's main functionality is to provide the user downloadable good quality educational content that has been uploaded by reputed teachers and reviewed by a panel of expert reviewers, in short the end-output must be a *content management system*.

2.1.3 User Characteristics :

The intended users will be children, teachers and Professors. Users are not expected to have a very high level of technical expertise to use the webportal. But the user is expected to do have a basic knowledge of operating the internet so as to have access to the contents of the portal. The contributors and reviewers are expected to be Internet literate and are expected to have a valid e-mail ids.

2.1.4 Operating Environment :

The Aakash webportal will be developed to run on any desktop, laptop, mobile or tablet PC provided it has a browser and good internet connectivity like LAN or Wi-Fi.

2.1.5 Design and Implementation constraints :

This portal is provisioned to be built on Django - a high-level Python Web framework that encourages rapid development and clean, pragmatic design. The portal is required to run on Desktop, Mobile, Tablet-PC and Laptops so the portal has to be responsive.

2.1.6 Dependencies :

- The portal will depend on the type and version of browser being used in the system i.e. browser version should be used which is JavaScript enabled and also adobe flash player must be present to see the uploaded flash and other animations.
- The portal also depends on the speed and bandwidth of Internet present for video content buffering.

CHAPTER

THREE

SOFTWARE DESIGN DESCRIPTION

3.1 External Interface Requirements :

3.1.1 User Interface :

The interface has to be user friendly so that a student/user even with a little internet knowledge can move through the links and find the content of his choice. The user interface must also be responsive enough to run on any device.

- **Inputs :** The user selects from the number of available links present in the homepage.
- **Outputs :** The portal responds to the input by displaying the request in the screen.

3.1.2 Hardware Interface :

Desktop or the device using the portal should be enabled with Internet facilities like LAN or Wi-Fi. Storing devices (USB flash, optical disks etc.) for the user to see a video, pdf or animation in offline mode.

3.1.3 Software Interface :

- **Django 1.5** framework using Python 2.7 is used to develop the modules of the portal
- **Sqlite3** has been used as a database for local server on desktop, **MySql** will be used for web hosting.

3.2 Functional Requirements :

3.2.1 Normal User :

- **User registration :** No user registration is required for a user who just want to download the contents or study material present in the portal.
- **Content :** A user/student can browse through the links on the homepage to the content page to download or view the topics.
- **Search** A search bar where the student/user can type and search for the topic he wants, the search is flexible and supports search based on both topic and subject name.
- **Contact Us** A contact us page where the user can contact the admistrators in case of a problem when using the portal.

3.2.2 Contributor :

- **Registration** A user wishing to be a contributor to the portal must 1st register with the portal by entering his details and uploading a document to support his credentials.
- **Login** A contributor should be logged in before he can upload any content to the portal.
- **Profile** A profile page of the contributor with upload more, edit profile must be there and this page should show all the previous uploads from the contributor clicking on which should show the comments and feedback given by reviewers.
- **Upload more** A contributor should be able to upload content to the site from this page and login is a requisite for this.

3.2.3 Reviewer :

- **Registration** A user wishing to be a reviewer for the portal contents must 1st register with the portal by entering his details and the admin will later give him permission for login and review,
- **Login** A reviewer should be logged in before he can review any content on the portal.
- **Profile** A profile page of the reviewer should has all the latest uploads to the site with a approve button which the reviewer clicks after checking wheather the content is worthy enough to be on the portal.
- **Comments** A comment section which opens on clicking the topic should be there so that a reviewer can give his comments on the topic for the contributor to see.

3.3 Non-Functional Requirements :

3.3.1 Performance Requirements:

The portal should be easy to use should have a responsive interface. There should be minimal delay in retrieving any content or associated detail.

- **Maintainability:** Portal needs to be upgraded if required in future according to the needs and feedback.

3.3.2 Reliability:

As the web portal provides the educational material such as pdf files, videos and animations editing course materials, It must be made sure that the system is reliable in its operations and uploaded data should be clean and free from any virus.

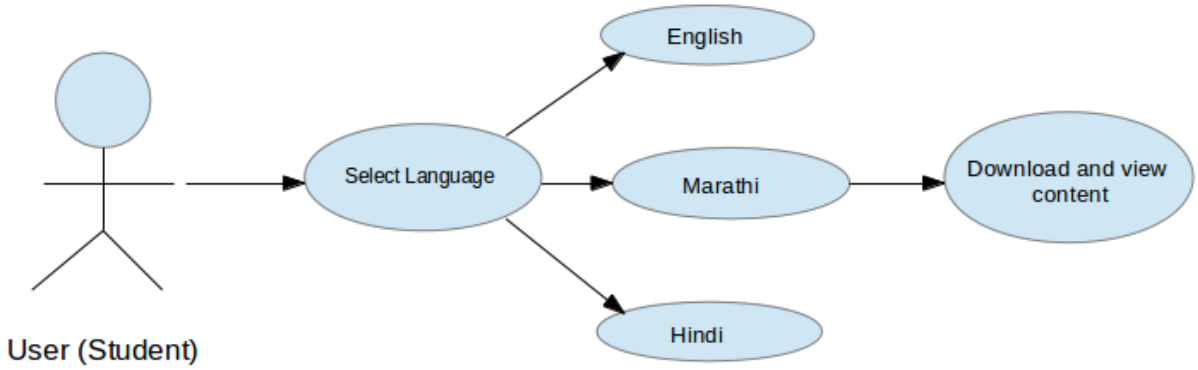
3.3.3 Security Requirements:

The portal must be secure enough to sensitive details like login and personal details.

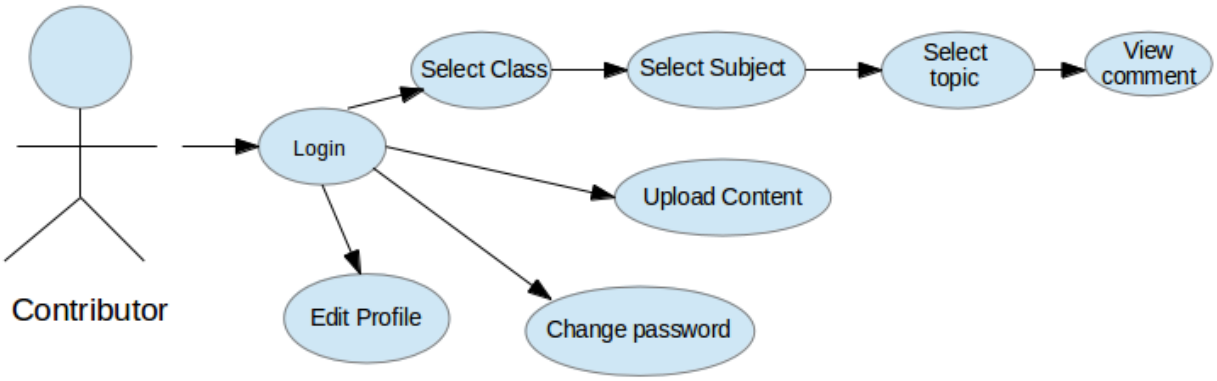
3.4 Behavioral Requirements:

3.4.1 Use Case Diagrams:

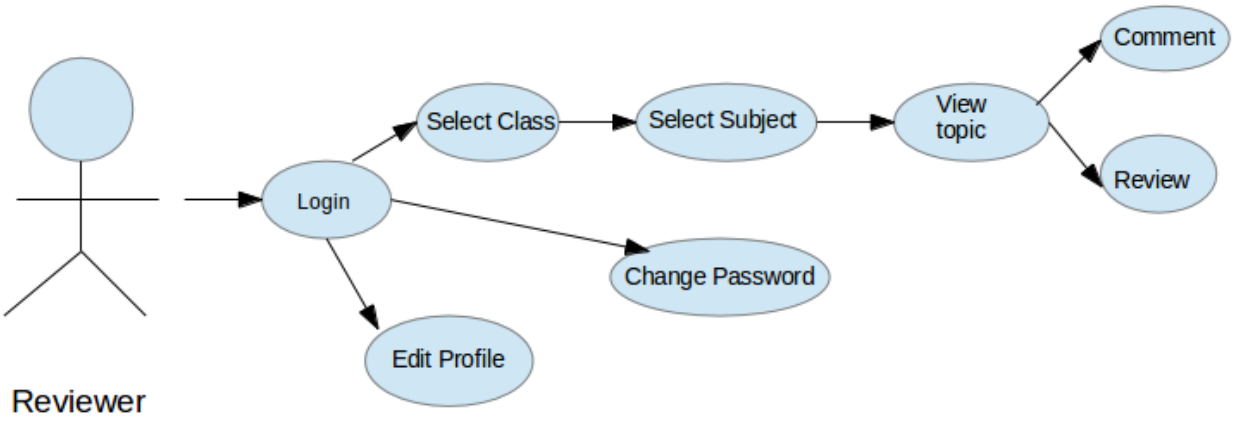
Use Case Diagram for a Normal user



Use Case Diagram for a Contributor



Use Case Diagram for a Reviewer



3.4.2 Class Diagram :

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations(or)methods and the relationships between the classes.

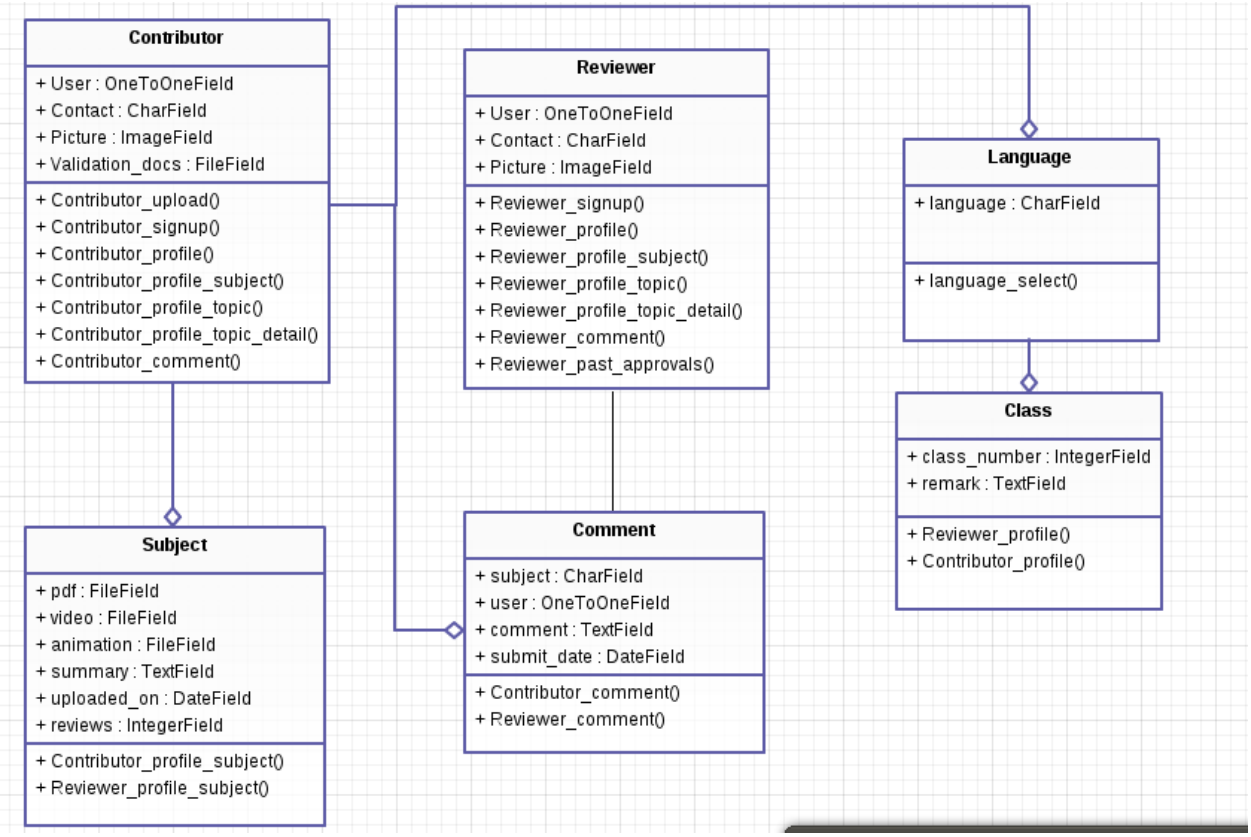


Figure 3.1: Class Diagram for the portal

3.4.3 E-R Diagram :

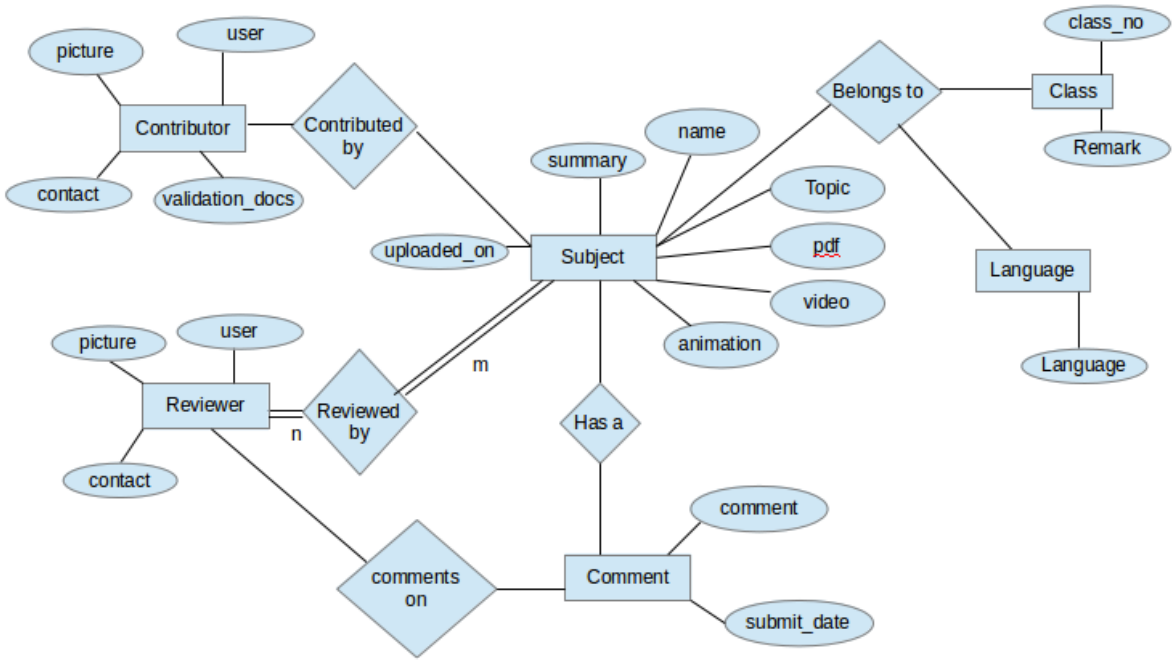


Figure 3.2: ER Diagram for the portal

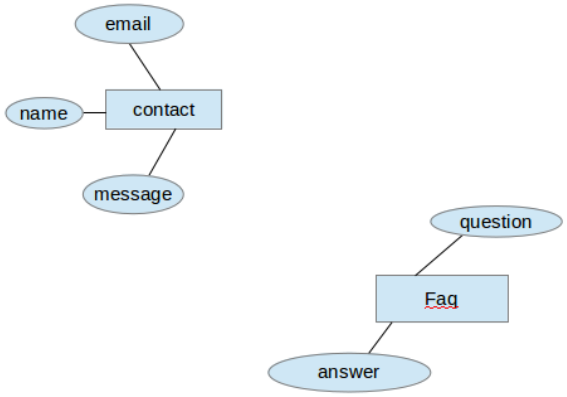


Figure 3.3: ER Diagram for FAQ and Comments

3.4.4 Activity Diagram :

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

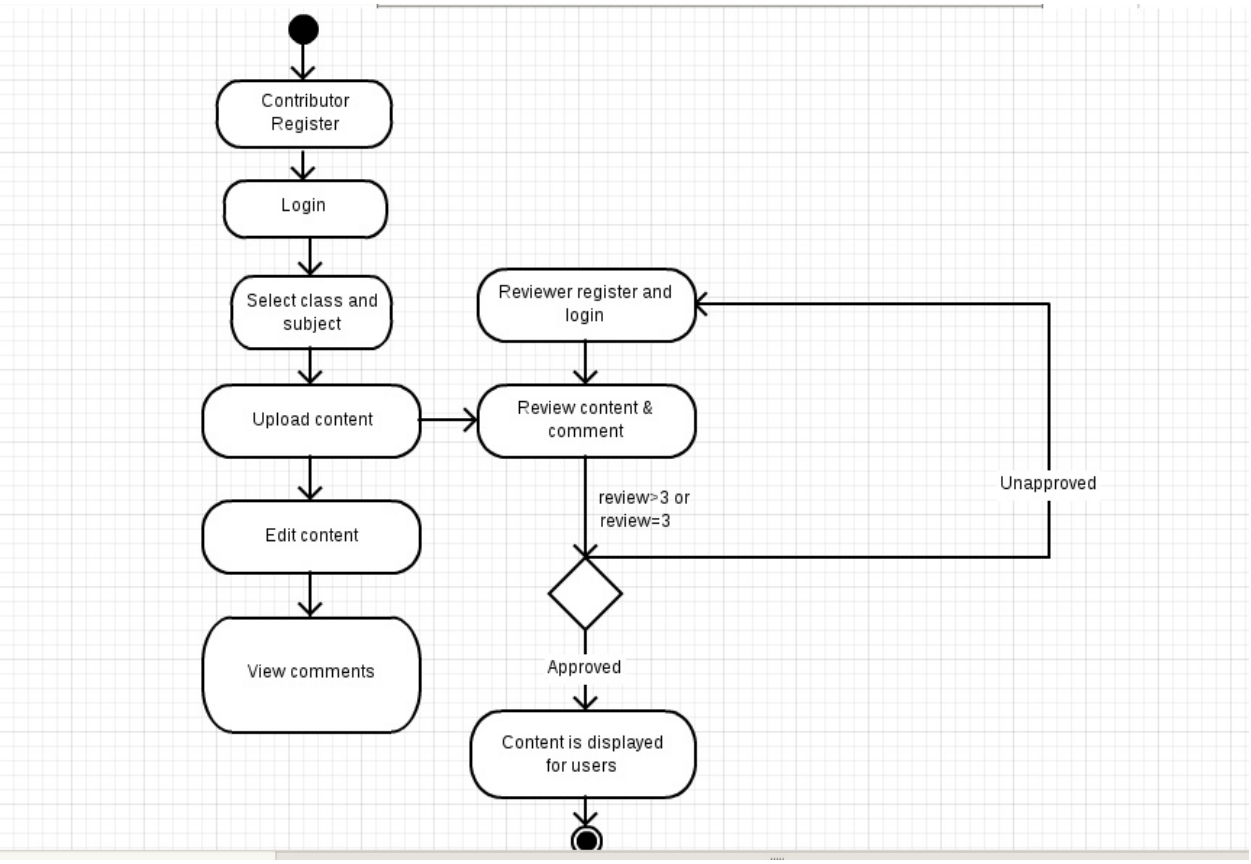


Figure 3.4: Activity Diagram for the portal

3.4.5 Sequence Diagram :

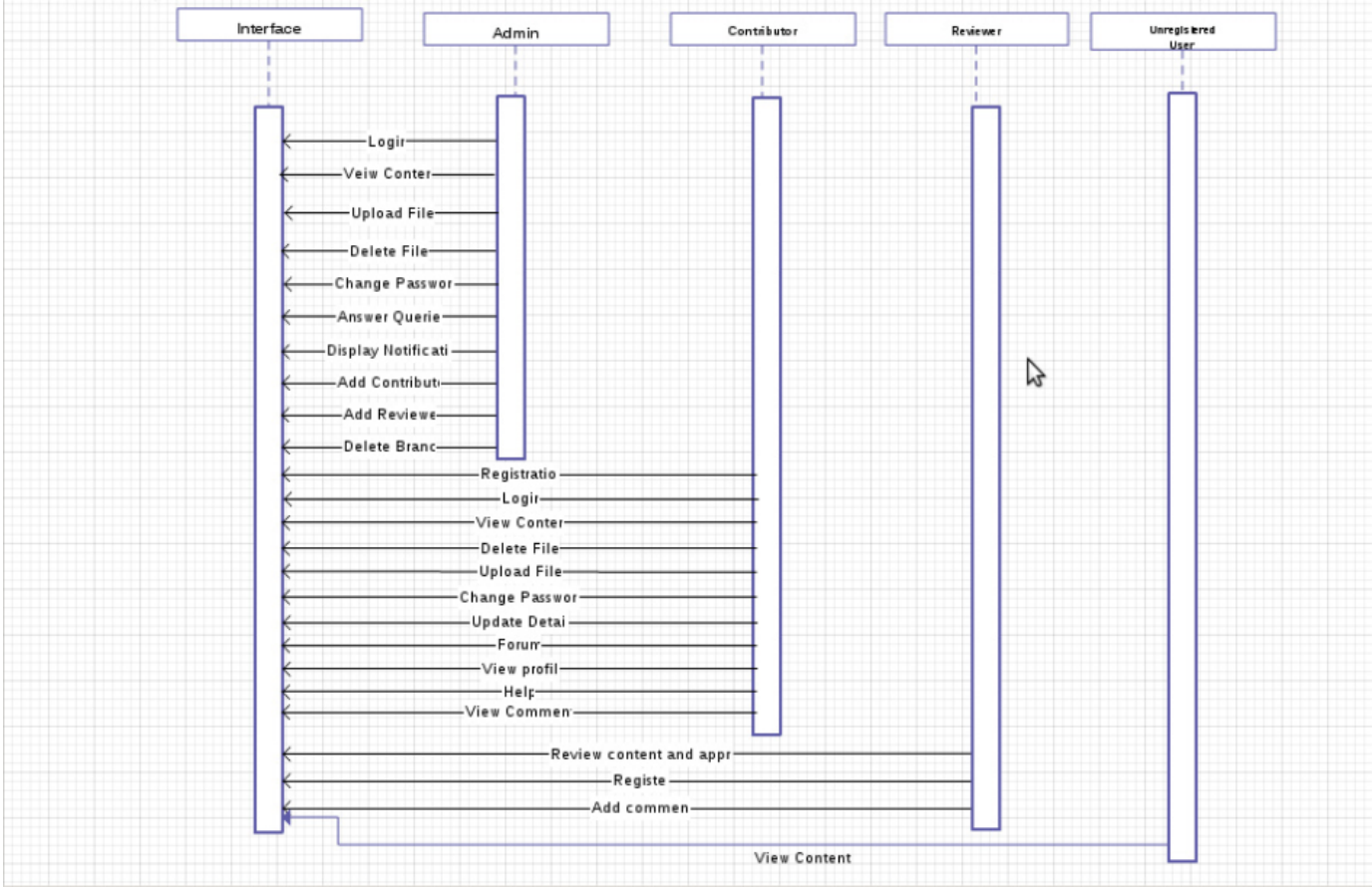


Figure 3.5: Sequence Diagram for the portal

CHAPTER

FOUR

USER'S GUIDE

4.1 Introduction

Aakash School Education is committed to offer the students free access to its many academic resources. We seek to capitalize on the potential of the internet to eliminate borders and geographic distance as obstacles to the instantaneous exchange of knowledge and new ideas. Unlike distance learning programs that charge tuition, provide formal instruction and limit participation, we offer all course materials free to everyone with online access. People from around the world may upload their content and students can make use of our extensive library of content from any computer with access to the web. ASE does not require any registration and is not a degree-granting or certificate-granting activity rather it is instead an effort to share knowledge and make sure the students have access to world class education content making full use of the Internet's potential.

- Why Python-Django
  - Python is a very-high-level dynamic object-oriented programming language.It's designed to be easy to learn, read and program.
  - ***Python is open*: Python is open source software, distributed** under a liberal license, and can thus be distributed freely.
  - ***Python is general purpose*: Python is a general purpose language,** which means that many things are made easy. Examples are string processing, reading/writing files, sockets, websites, databases, GUI's.
  - *Python is dynamic*: Being a dynamic language means that new code can be executed during runtime without limitations.
  - Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
  - Django was designed to handle two challenges: the intensive deadlines of a newsroom and the stringent requirements of the experienced Web developers who wrote it.
  - Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development.
  - Django follows the model–view–controller architectural pattern.MVC Design Pattern, the first thing to note about this is that Django has four Python files (`models.py`, `views.py`, `urls.py`) and an HTML template (`latest_uploads.html`):
  - `models.py` (the database tables)

```
from django.db import models

class Subject(models.Model):
    name = models.CharField(max_length=50)
    upload_date = models.DateField()
```

- `views.py` (the business logic)

```
from django.shortcuts import render
from models import Subject

def latest_uploads(request):
    uploads_list = Subject.objects.order_by('-upload_date')[:10]
    return render(request, 'latest_uploads.html', {'uploads_list': uploads_list})
```

- `urls.py` (the URL configuration)

```
from django.conf.urls.defaults import *
import views

urlpatterns = patterns('',
    (r'^latest/$', views.latest_uploads),
)
```

- `latest_uploads.html` (the template)

```
<html><head><title>Uploads</title></head>
<body>
<h1>Uploads</h1>
<ul>
{% for uploads in uploads_list %}
<li>{{ uploads.name }}</li>
{% endfor %}
</ul>
</body></html>
```

Advantage of using python over other programming languages

- Python programs run slower than the Java codes, but python saves much time and space. Python programs are 3-5 times smaller than java programs.
- Python is dynamic typed language. Python programmers don't need to waste time in declaring variable types as in java.
- The program length in python is 5-10 times shorter than that in C++.
- Unlike other programming languages python code has to be strictly indented but the indentation helps in much cleaner code(readable).
- High level data structures are list, directory are well suited in python.
- The elegant design and syntax rules of this programming language make it quite readable even among the multi programmer development teams.

GIT

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes.
- GitHub is now the largest online storage space of collaborative works that exists in the world.
- If you and a co-worker are both updating code on the same website. You make your changes, save them, and upload them back to the website. So far, so good. The problem comes when your co-worker is working on the same code as you at the same time. One of you is about to have your work overwritten and erased. A version control application like Git keeps that from happening.

- You and your co-worker can each upload your revisions to the same code, and Git will save two copies. Later, you can merge your changes together without losing any work along the way.

GNU/Linux

- GNU is a Unix-like computer operating system developed by the GNU Project. It is composed wholly of free software. It is based on the GNU Hurd kernel and is intended to be a "complete Unix-compatible software system".
- The system's basic components include the GNU Compiler Collection (GCC), the GNU C library (glibc), and GNU Core Utilities (coreutils), but also the GNU Debugger (GDB), GNU Binary Utilities (binutils),the bash shell and the GNOME desktop environment.
- The GNU Project has endorsed variants using the Linux kernel, such as gNewSense, Trisquel and Parabola GNU/Linux-libre.
- LINUX is also a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open source software development and distribution.
- The underlying source code may be used, modified, and distributed—commercially or non-commercially—by anyone under licenses such as the GNU General Public License.
- Typically, Linux is packaged in a format known as a Linux distribution for desktop and server use. Some popular mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora, Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server.
- Linux distributions include the Linux kernel, supporting utilities and libraries and usually a large amount of application software to fulfill the distribution's intended use.
- The open-source software movement arose to clarify the environment that the new copyright, licensing, domain, and consumer issues created.
- Open-source code is typically a collaborative effort where programmers improve upon the source code and share the changes within the community so that other members can help improve it further.
- Distros designed to replace your existing operating system some other linux distros are Ubuntu,OpenSUSE,Fedora etc.
- Ubuntu is a Debian-based Linux operating system, with Unity as its default desktop environment (GNOME was the previous desktop environment).Ubuntu is the most popular desktop Linux distribution.
- The Ubuntu project is publicly committed to the principles of open source development; people are encouraged to use free software, study how it works, improve upon it, and distribute it.

4.2 Description

4.2.1 Workflow

- A model is the single, definitive source of information about the data. It contains the essential fields and behaviors of the data that is stored. Generally, each model maps to a single database table. Each attribute of the model represents a database field.

  In this website the various models used :

- User: This model stores the details of all the users, that is contributors and reviewers of the website. The details include their username, first name, last name, password, email.
- Contributor: A contributor is an user of the website who can sign up by filling their user details, contact, picture and validation-docs in the registration form. The administrator will be notified about the new request from a user

via a email and will verify the details of the contributor and if the access permissions are granted then he/she can upload study materials for the website.

- Reviewer: A reviewer is an user of the website who can sign up by filling their user details, contact and picture in the registration form. The administrator will be notified about the new request from a user via a email and will verify the details of the details of reviewer and if the access permissions are granted then they can review and give their comment on the unreviewed contents.
- Class: A class describes which class the study material belong to. Based on the class, the student accesses the study materials. Administrator stores number of classes in the database so that the contributor can contribute or the reviewer can review the content in a particular class.
- Subject: A contributor can upload content for the website by filling a form that includes subject name, topic, class-name and files that can be in pdf, video, animation formats. Rating can be given by administrator based on the quality of the contents.
- Comment: The comments are given by the reviewer on a topic of the subject. The date on which the comments are submitted are also noted along with the reviewer details.
  - Every user who logs in must be either a contributor or reviewer.
  - If the person who logs in, is a reviewer he/she can just see the materials but cannot modify the data rather he can comment on it.
  - If the person who logs in, is a contributor he will be given permission to upload contents for the website. The contributor can see the comments given by the reviewers.
  - Reviewer doesn't have the permission to upload any files.
  - An upload will be available to the non-user if and only if it is approved by atleast 3 reviewers.
  - A user can search the study materials based on the subject name or topic.
  - A user may view or download the files which is pdf, video or animation files.
- Contact: Users can give their suggestions on the different contents or request for new contents. He/she can send their name, email id and message in order to contact the adminstrator.
- Faq: Faq consists of fields like questions and answers. It describes the frequently asked questions and answers by users.
- Language: User can view the content of different languages i.e English, Hindi and Marathi as present in the model.

4.2.2 Interface

User Interface

User Interface for the "WebPortal" application begins with the homepage of the portal, showing the recent uploads, number of classes,subjects and uploads. It also has links to the Contact Us, Content, Register(drop-down of contributor/reviewer), Login and more pages which includes a drop-down of the Docs, About us and details of the users.

Once a person is logged in, the Login button changes to his username and a drop-down comes on clicking his username, which takes him to his profile or enables him to logout.

Contact Us

Clicking on this link redirects a user to a new page with a contact us form using which the user can contact the site administrators.

`Example:`

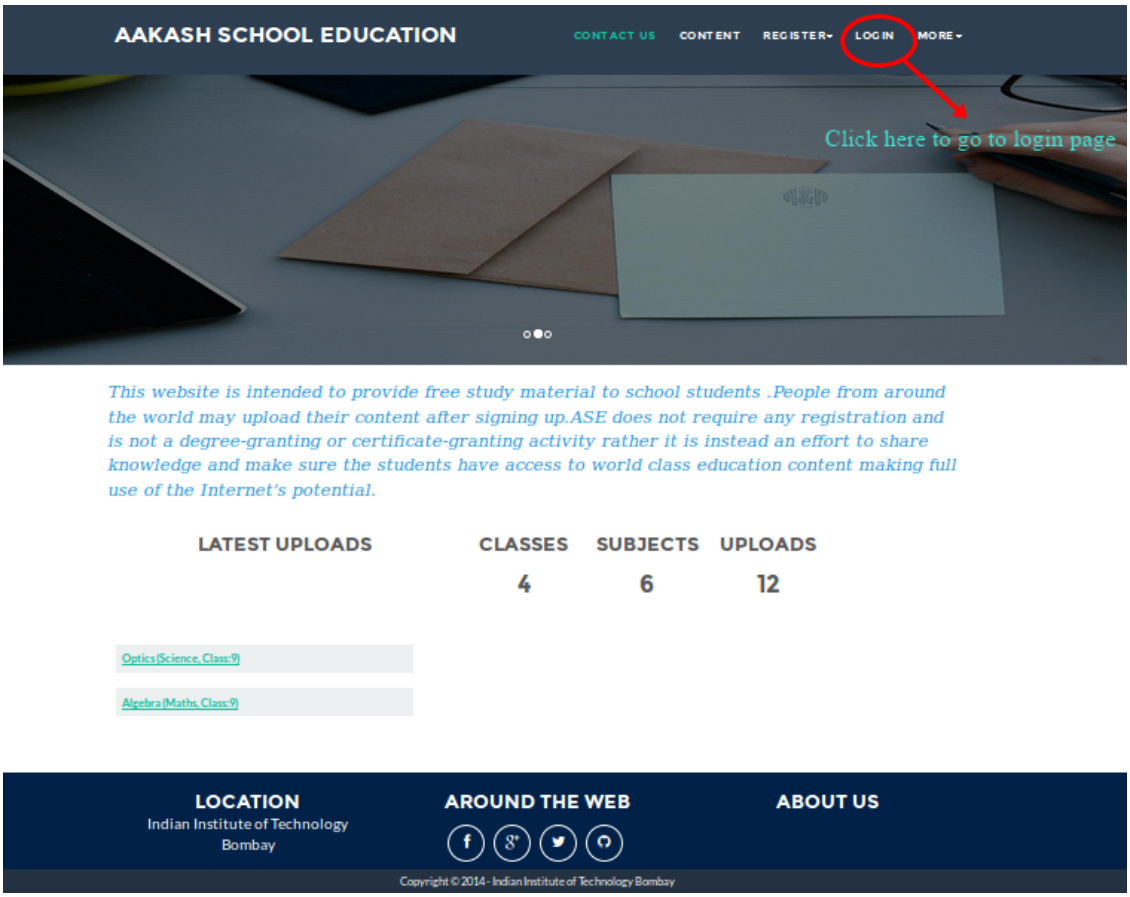


Figure 4.1: Home page (without logging in)

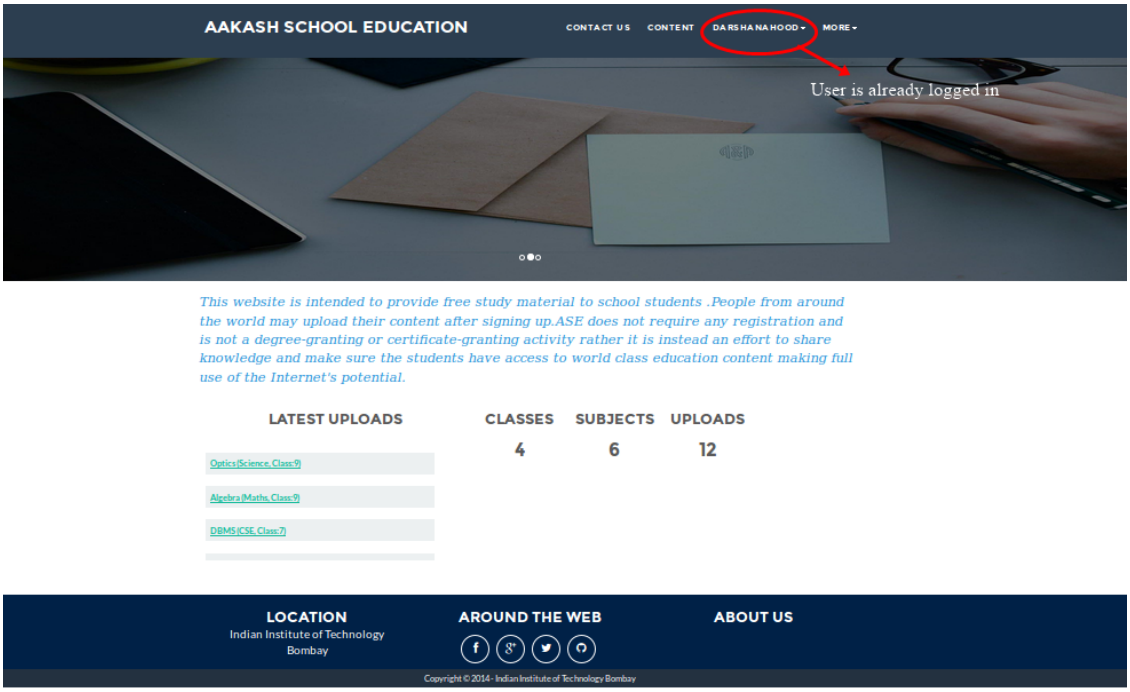


Figure 4.2: Home page (after logging in)

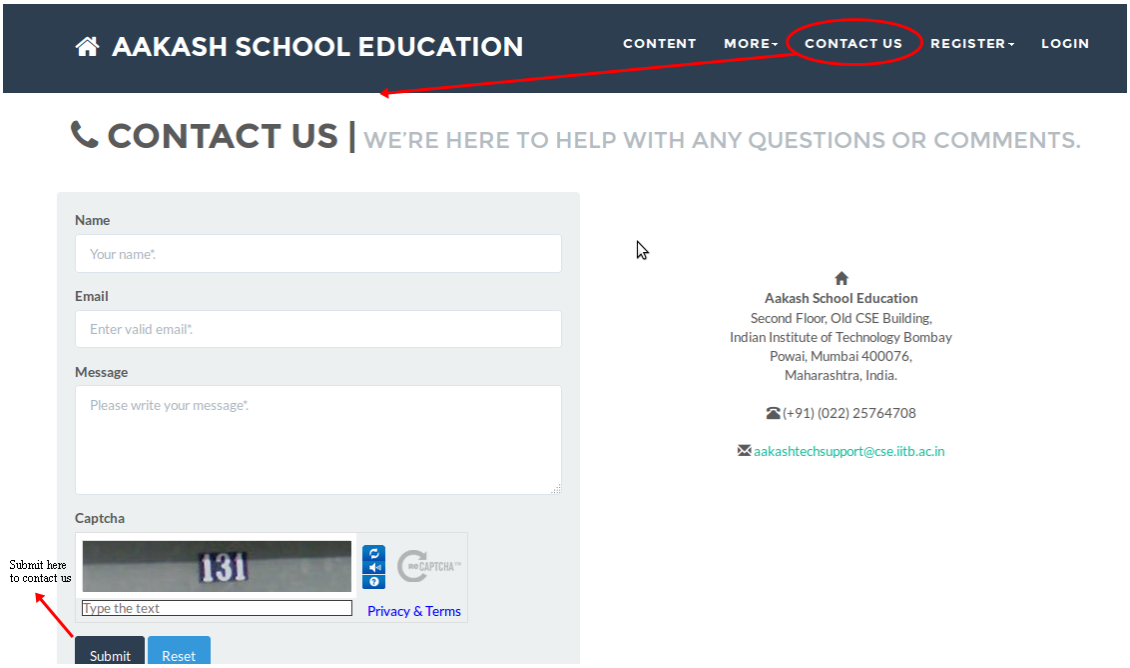


Figure 4.3: Contact Us

About us

Clicking on this link will give an overview of our website, regarding the main motive of this website and how will it help the students, and its relation to the Aakash School Education.

Content

This section opens on clicking on the content link present in the homepage. This section is for showing the entire contents which is present in the website. Initially one has to Select a language in which we want to see the content.

Contents corresponding to that language

After selecting the language, the contents corresponding to that language will get displayed. It gets displayed in the form of a table with its fields as Class, Subject, Topic, Summary, PDF, Video and Animations present.

Search bar

There is also an option to search in the contents page. The search box provides us an option to enter either the subject or the topic of a subject, to search for. On clicking the search icon, the given string is matched with the available contents and wherever there is a match, the corresponding topics are displayed on the next page. Also there is a button to Go Back to the content's page.

Register

If a person wants to register in the website, he can do it here. There are 2 options for registering, i.e. As a Contributor or as a Reviewer.

Register as a Contributor This takes a user to register in the website as a Contributor i.e. the person who is going to upload the documents of various subjects and topics. He has to fill the form displayed in the page, the fields are username, firstname, lastname, email, password, profile picture, contact and the validation files(which checks if the contributor has the required qualification or not). Then he has to click the register button to get himself registered.

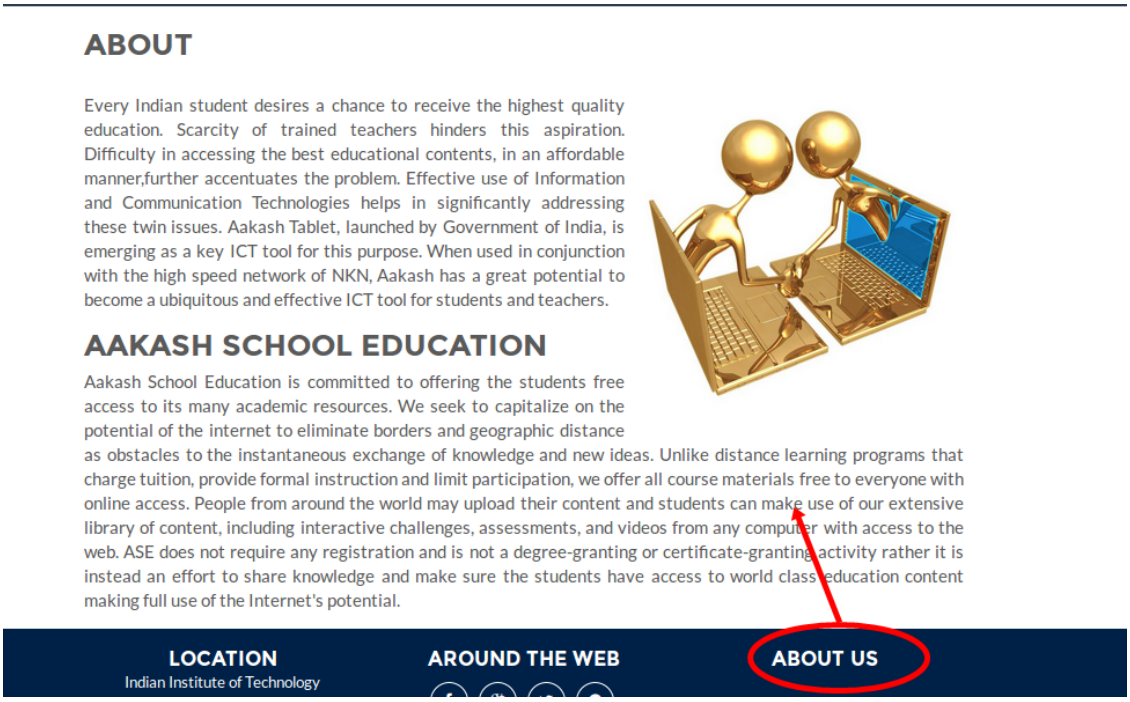


Figure 4.4: About Us

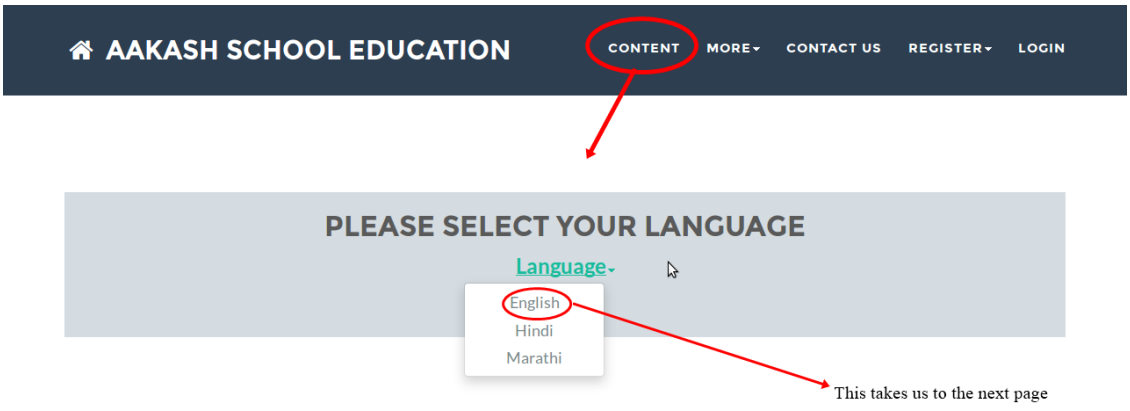


Figure 4.5: Select a language

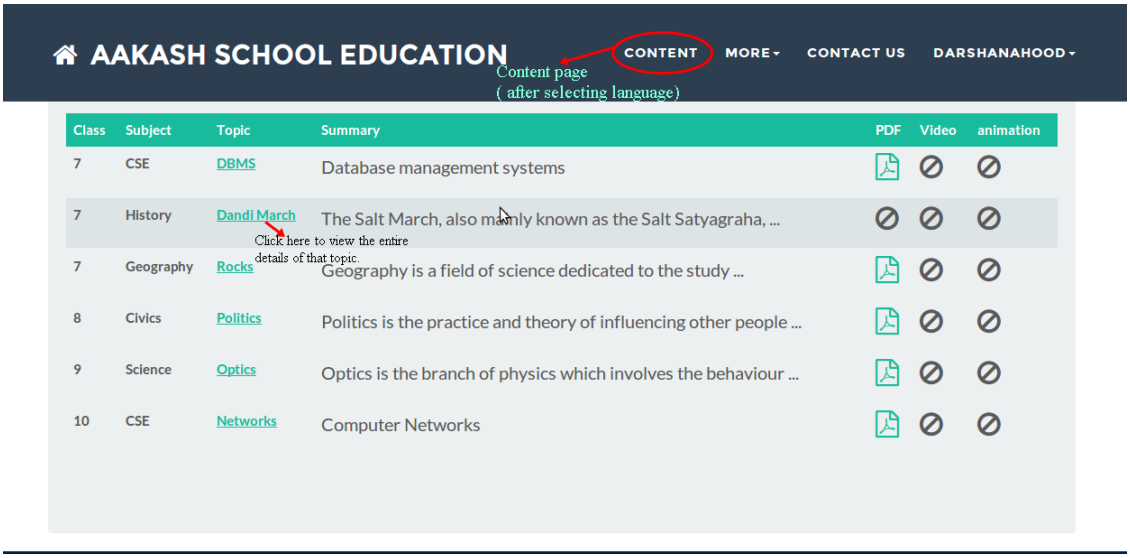


Figure 4.6: Contents corresponding to that language

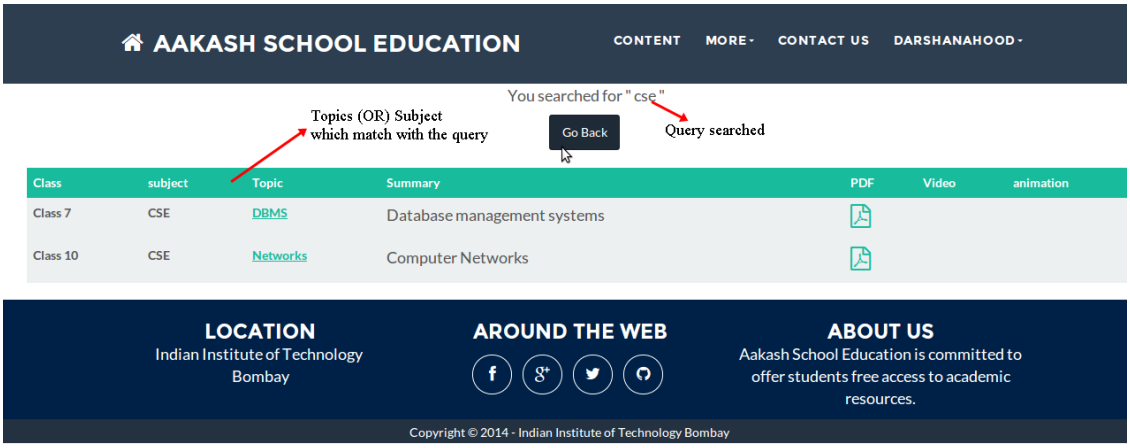


Figure 4.7: Search

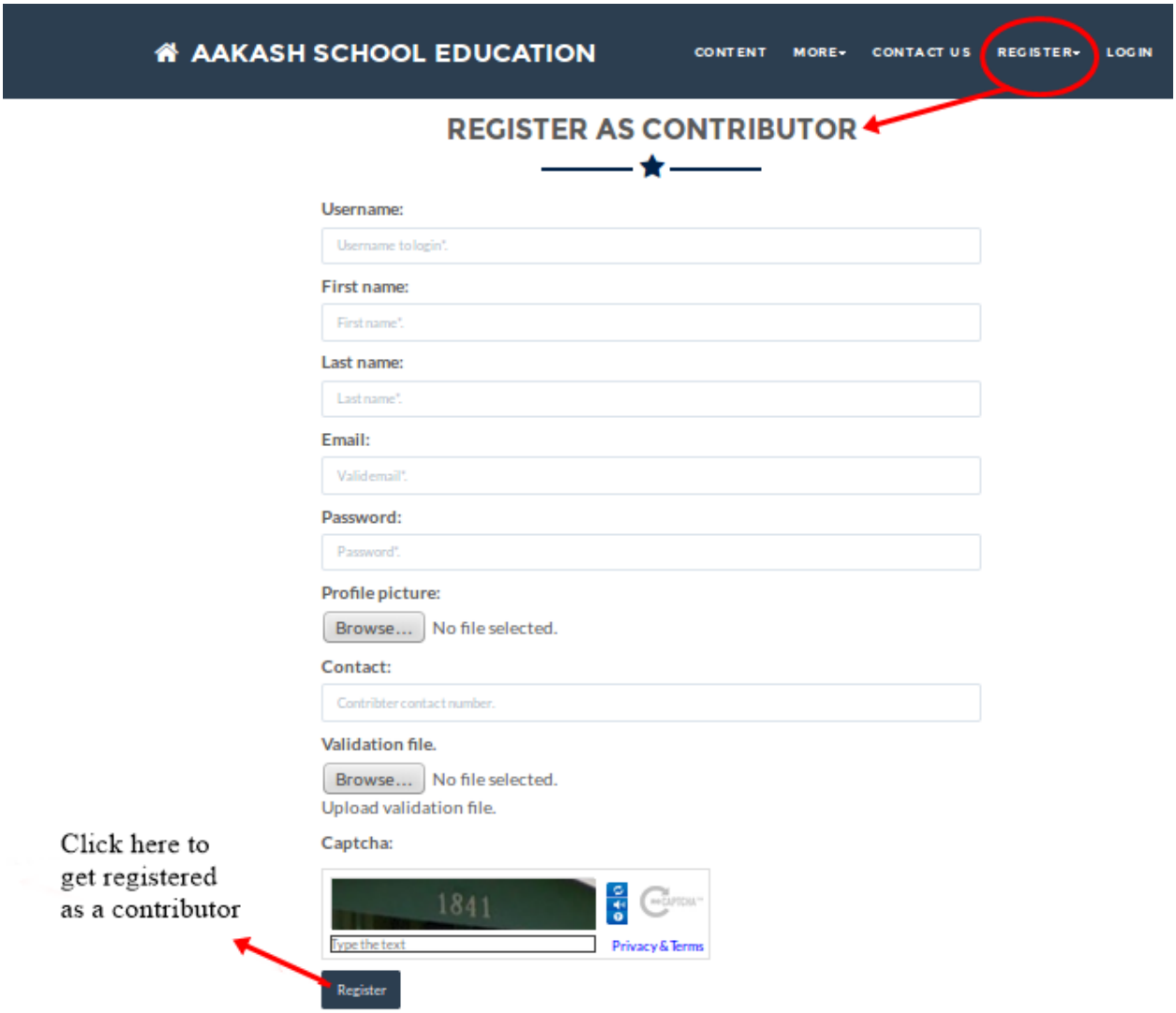


Figure 4.8: Register as a contributor

Register as a Reviewer This takes a user to register in the website as a reviewer i.e. the person who is going to review the uploaded documents. He has to fill the form displayed in the page, the fields are username, firstname, lastname, email, password, profile picture and contact. Then he has to click the register button to get himself registered.

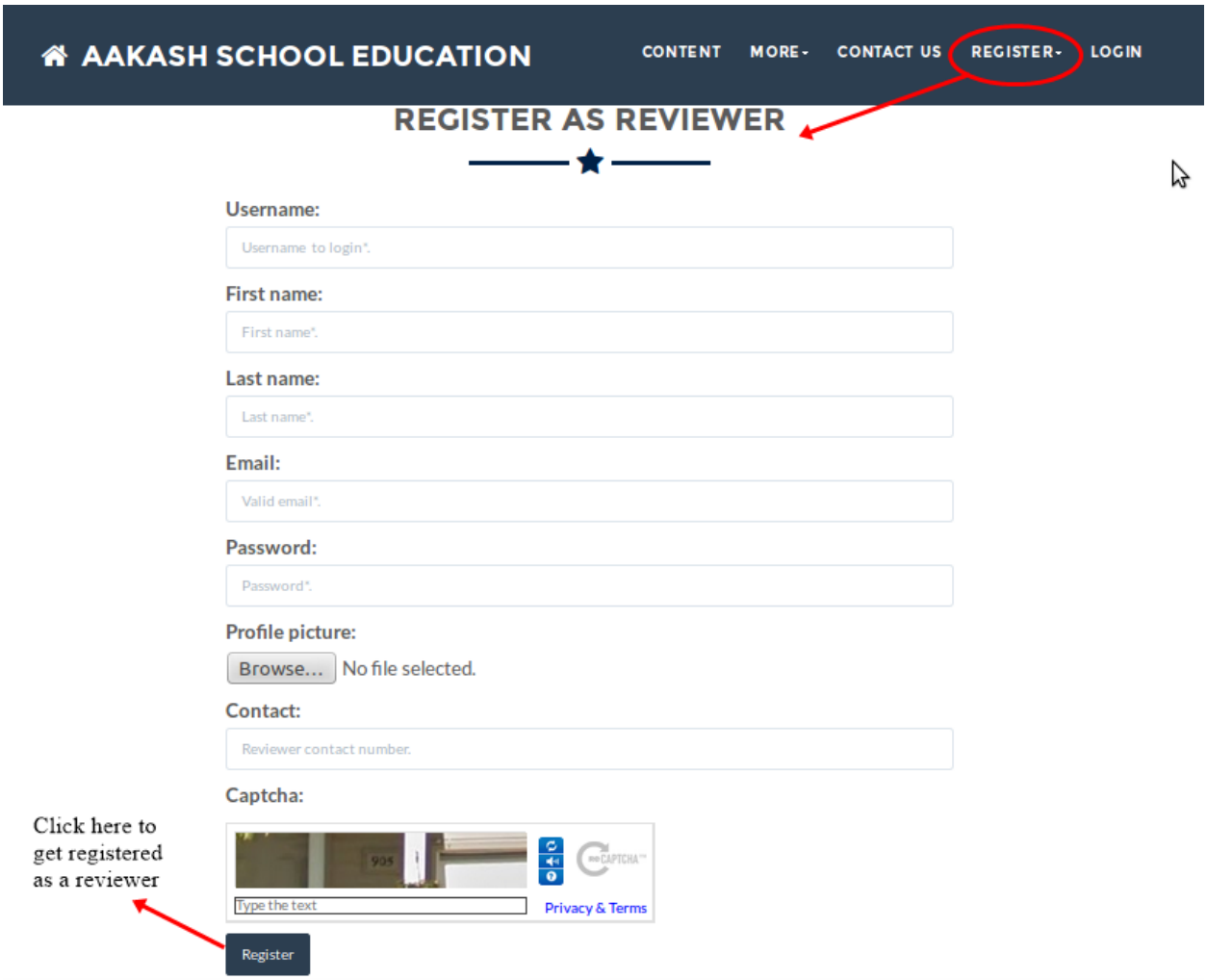


Figure 4.9: Register as a reviewer

Login This is used by both the contributor and reviewer to Login. The user has to enter his username and password and the click on Sign In to login to his profile. In case his username and password do not match due to wrong credentials, he will get an error message saying Bad Login.

Forgot Password This is an option to let the user to retrieve his password in case he forgets it. He has to enter his email through which he registered in the website, and a mail will be sent which would contain his old password. He can then later login and change his password (discussed later)

Contributor's Profile After a contributor logs in, it takes him to his profile, where he sees an "Upload more" button which when clicked takes him to the Upload Section. There are two more buttons i.e. Edit Profile and Change Password. These 3 buttons always remain fixed for the entire section when the contributor is logged in.

The first page he sees after logging in is the List of classes in which documents are uploaded.

The second page contains the list of subjects under a particular class.

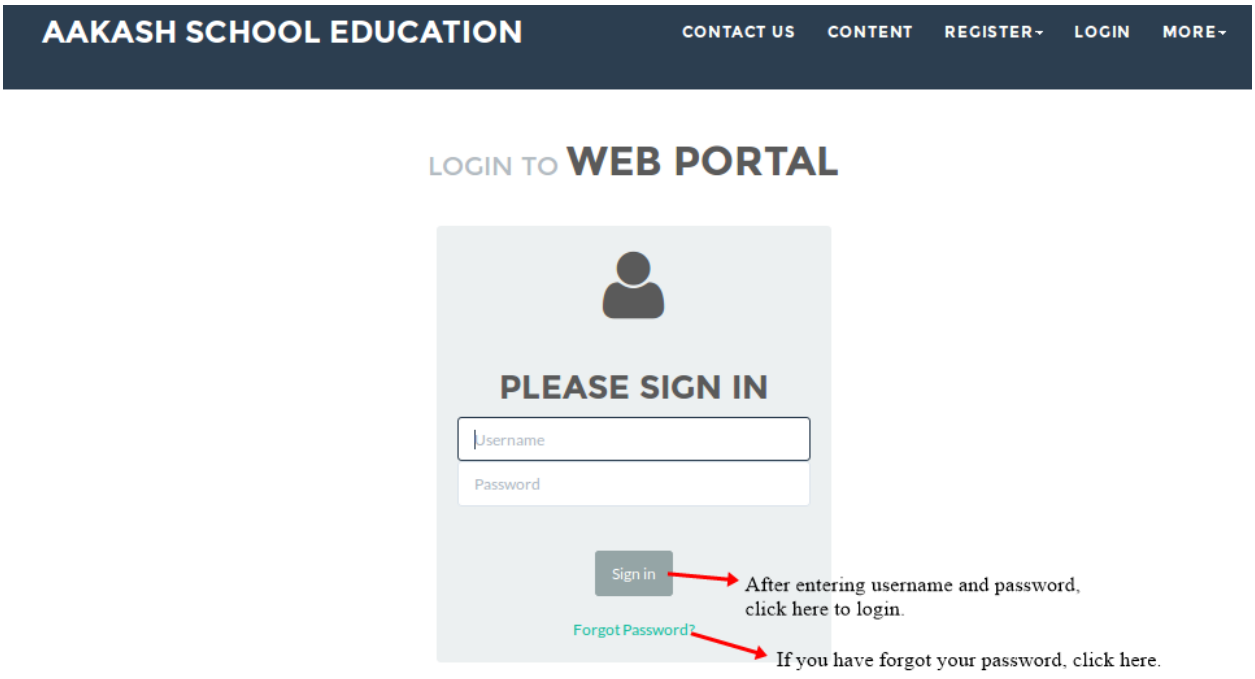


Figure 4.10: Login

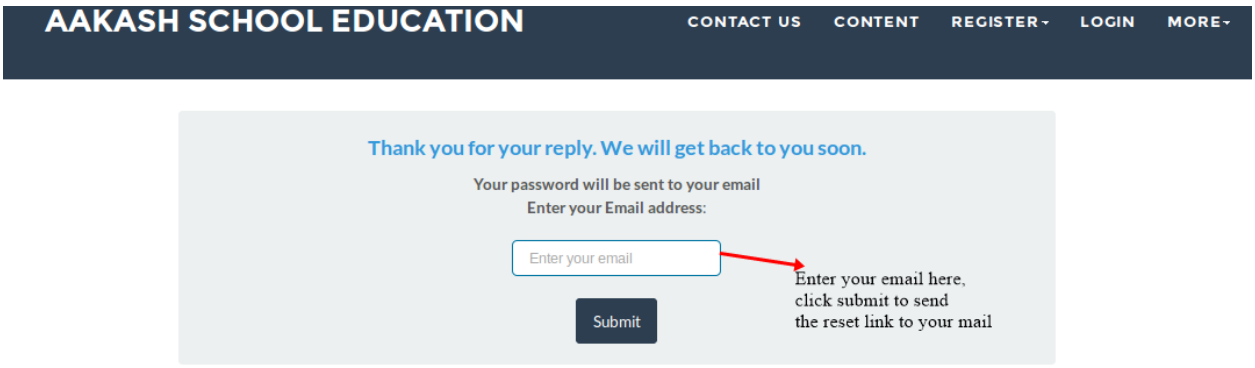


Figure 4.11: Forgot Password

The third page contains the list of topics under a particular subject and its details, such as Summary, PDF, Video and animation.

The fourth page contains the list of comments under a particular topic.

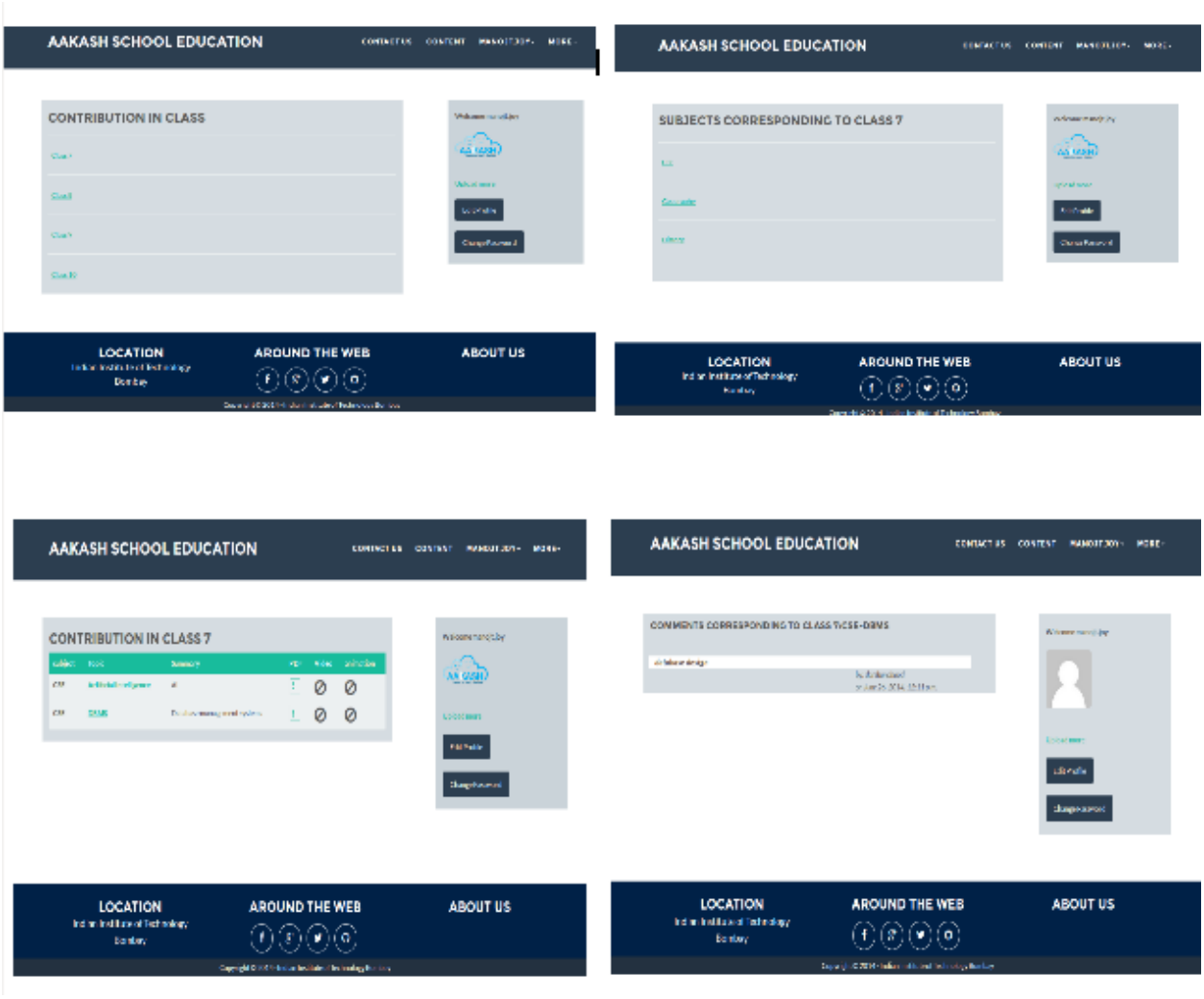


Figure 4.12: Contributor Profile

Upload More When a contributor clicks on the upload more button, it takes him to the upload more form which contains various fields such as language, class, Subject name, topic, pdf, video, animation and summary. Once he fills the entire form, he clicks on upload more which uploads the content. If any required field is missing or it is not a valid file, it raises error. If none of PDF, video or animation is present, it raises an error. Also, there is a limitation of file size of pdf, upon exceeding it raises an error.

Reviewer's Profile After a reviewer logs in, it takes him to his profile. Also he sees an "Past Approvals" button which when clicked takes him to his recent past approvals. There are two more buttons i.e. Edit Profile and Change Password.

These 3 buttons always remain fixed for the entire time when the reviewer is logged in.

The first page he sees after logging in is the list of classes in which documents are uploaded.

The second page contains the list of subjects under a particular class.

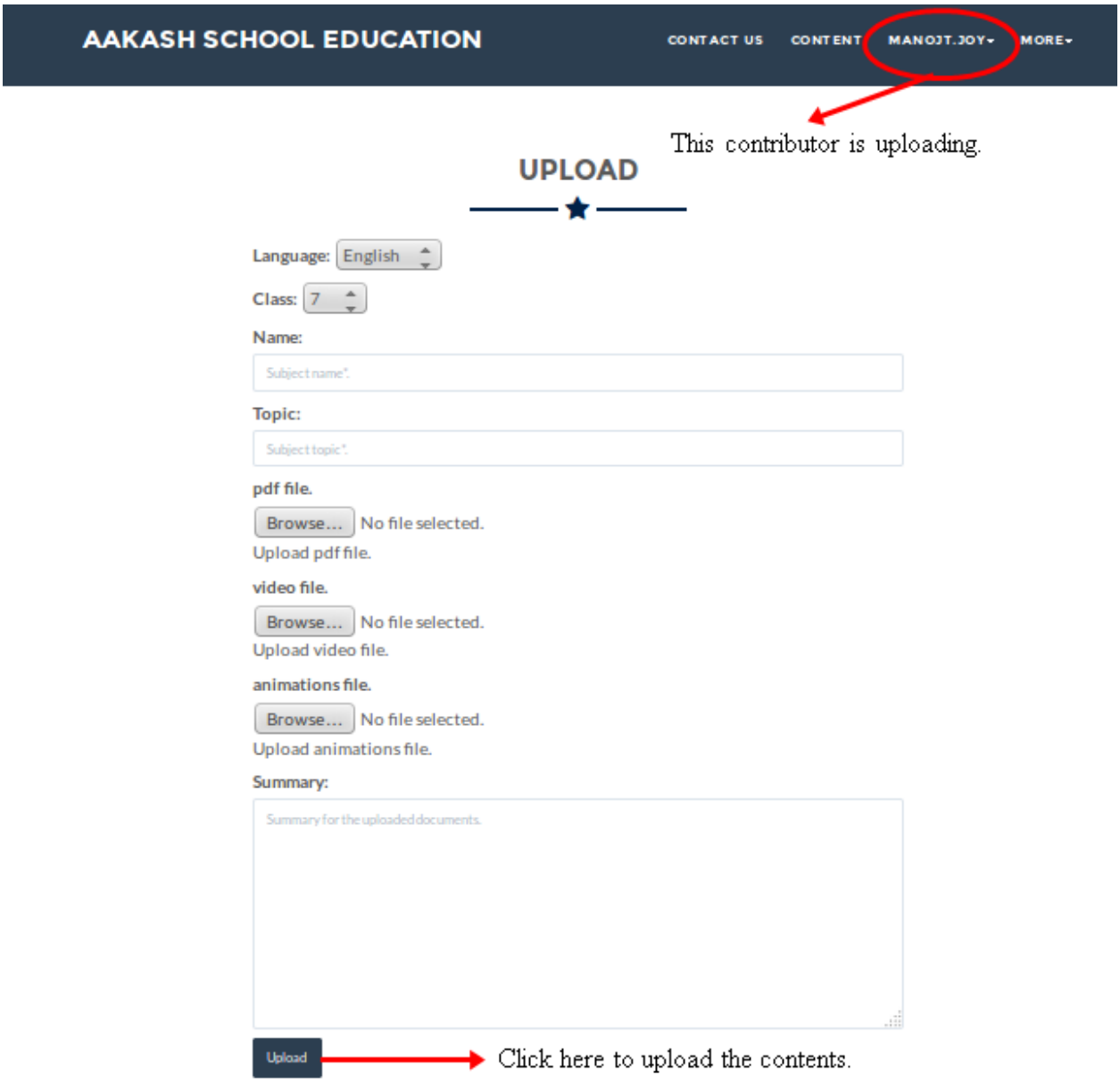


Figure 4.13: Upload more

The third page contains the list of topics under a particular subject and its details, such as Summary, PDF, Video and animation. Also, there is an approve button which the reviewer has to click if he feels that the uploaded documents are suitable for the topic and can be published. If a topic is approved by 3 or more reviewers, it is deemed accepted and published in the Content page.

The fourth page contains the list of comments under a particular topic. Also, since the user is a reviewer, he has the freedom to add any number of comments, and view all the previous comments on the topic. Upon commenting, the date, time and username of the reviewer comes beneath the comment. These comments are viewed by the the contributor so that he can improve his uploaded content.

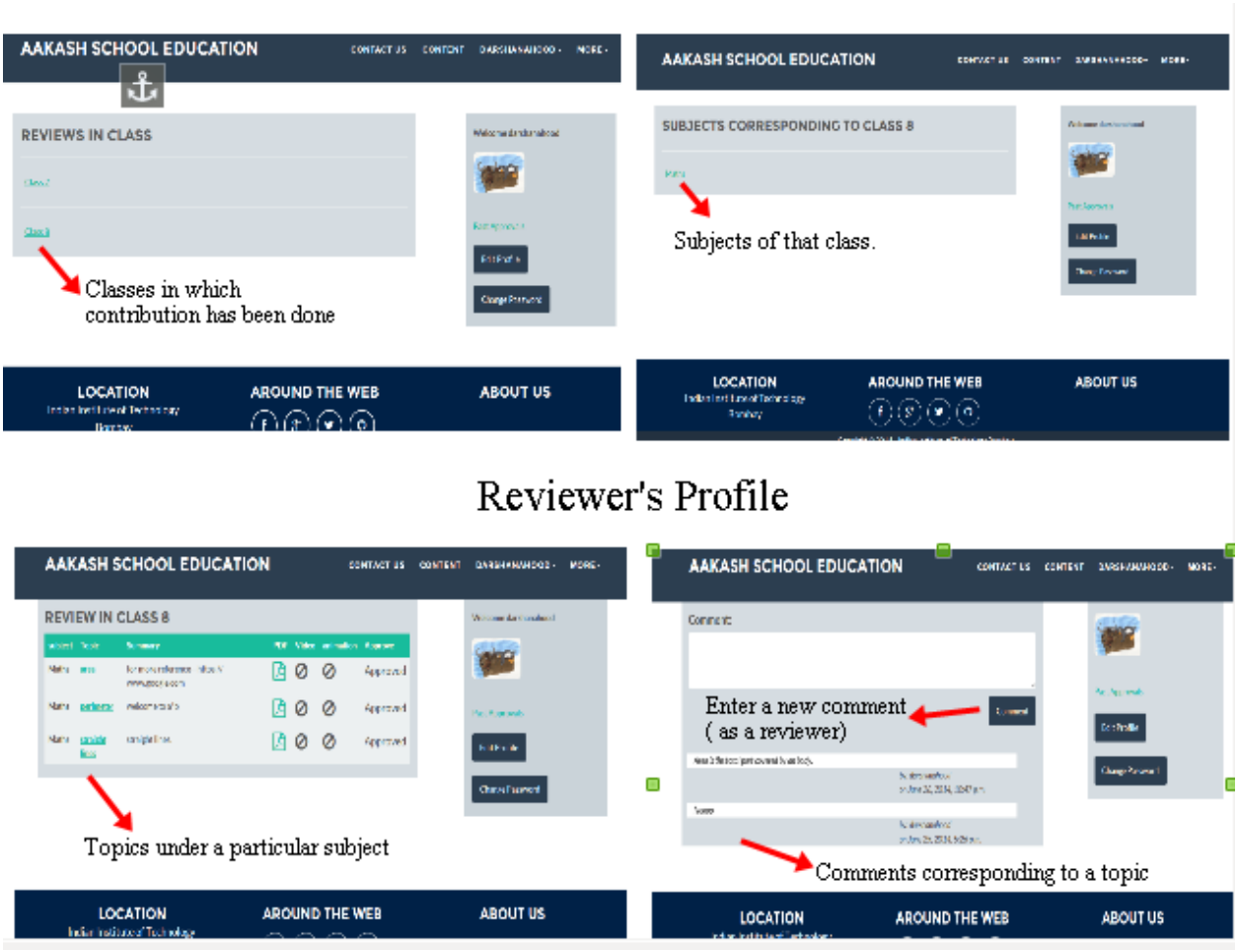


Figure 4.14: Reviewer Profile

Past Approvals When a reviewer clicks on the "Past Approval" link, it takes him to the past approval table which contains various fields such as class, Subject name, topic, pdf, video, animation, summary and the Approved status. The content already approved by the reviewer previously in his profile will be displayed here.

- The next two sections are common both for contributor and reviewer

Edit Profile Upon clicking the edit profile button, the user (contributor or reviewer) can edit his profile. The user form and the contributor/reviewer form is displayed, with an instance of the contributor/reviewer present. So if a user wants to edit anything, he can see his previous information and feed in the new information i.e he can make new changes to his previous details. The password has to be filled again and then he has to click Save Changes to save the changes.

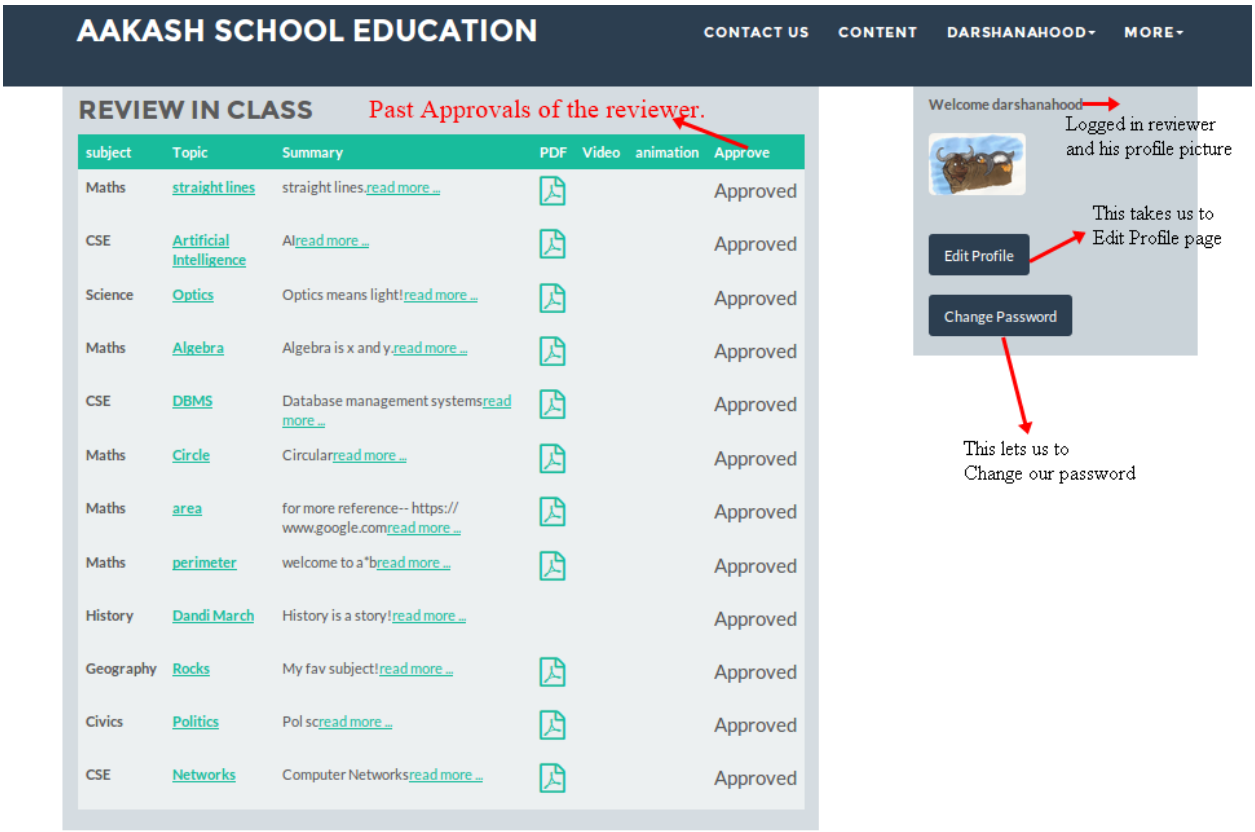


Figure 4.15: Past Approvals

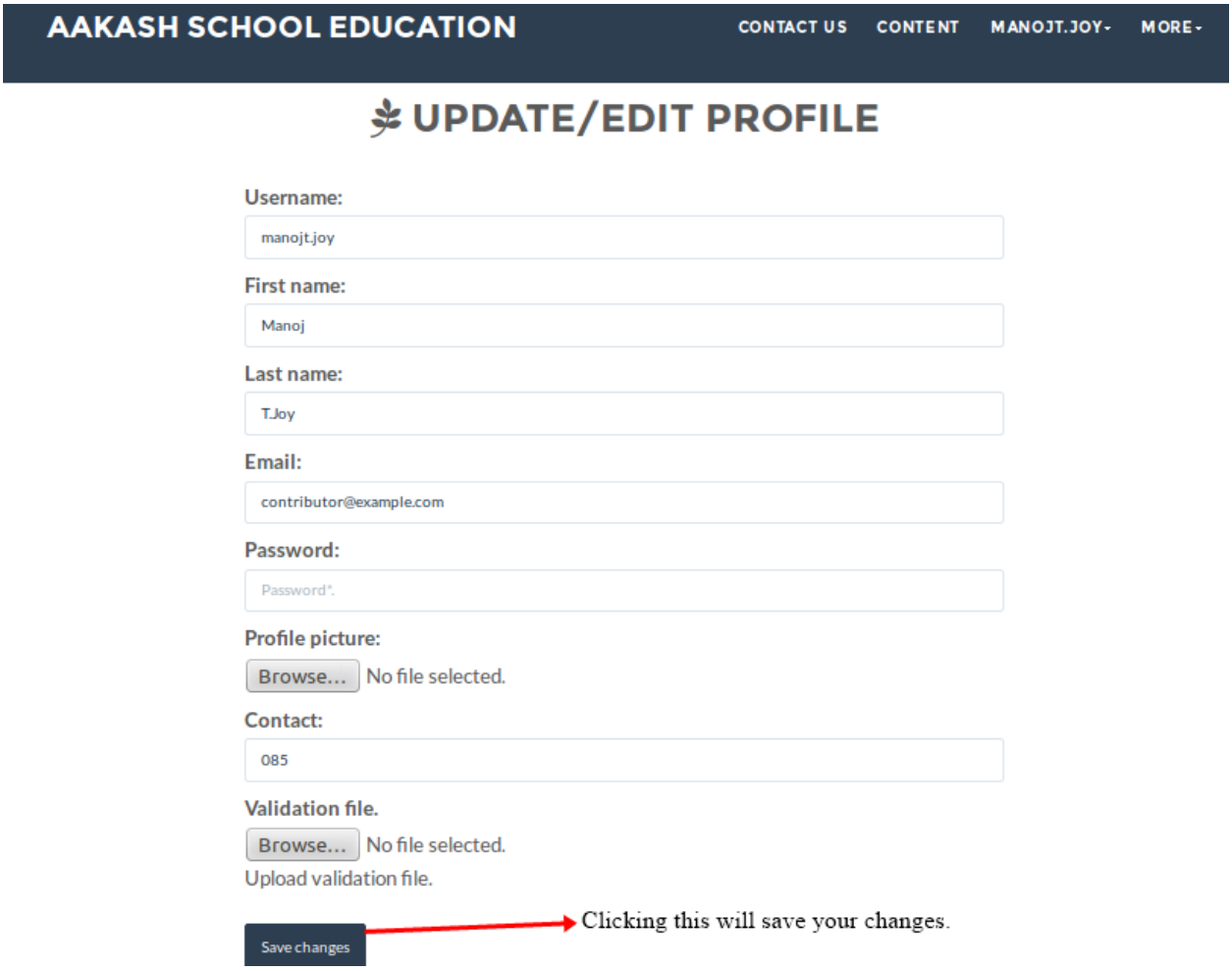


Figure 4.16: Contributor Edit Profile

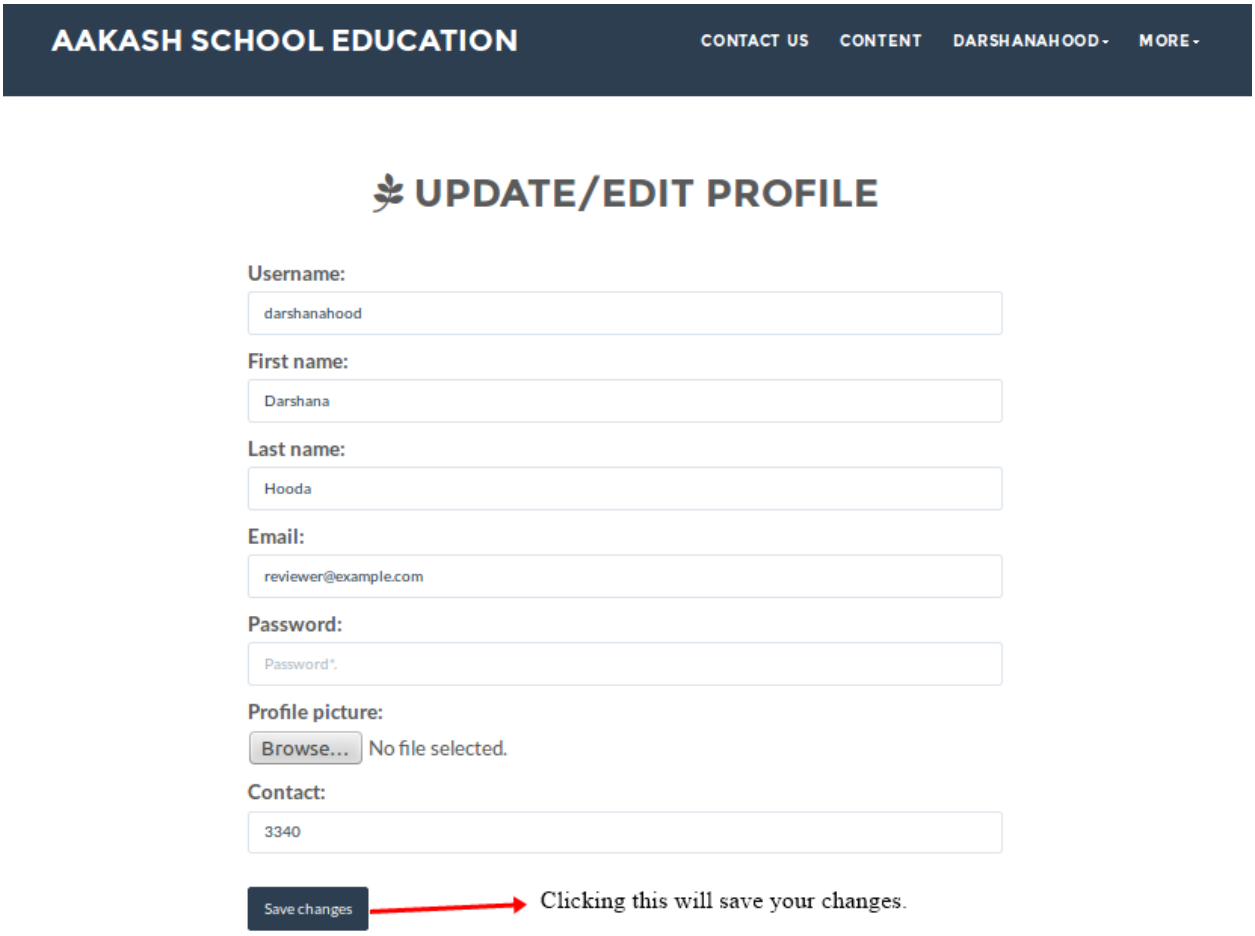


Figure 4.17: Reviewer Edit Profile

Change Password Upon clicking this a new page opens up. The user has to enter his old password and his new password twice for confirmation.If the old password is correct and both the entered new passwords match, his password is changed and the success message is displayed, else an error message pops up.

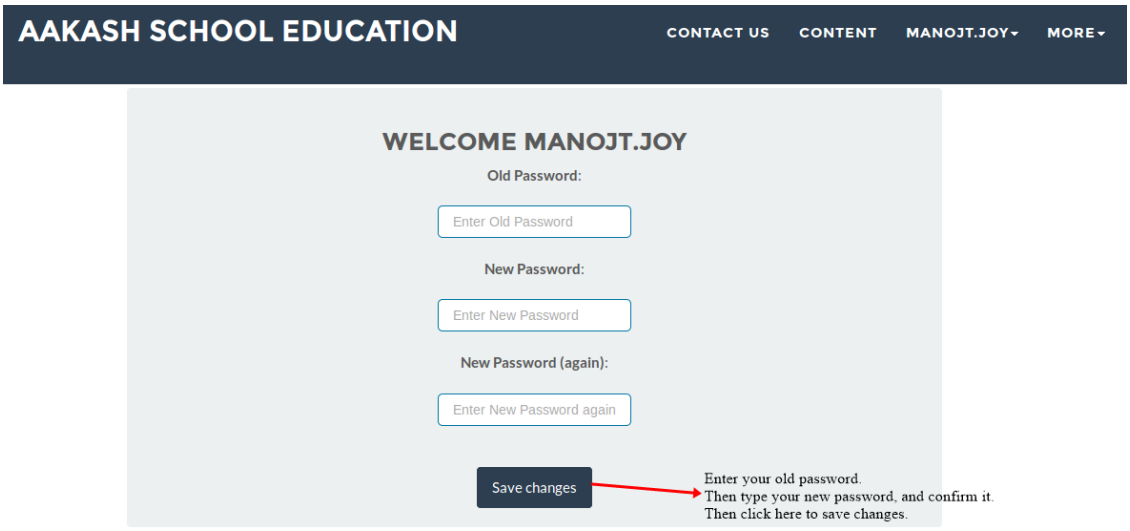


Figure 4.18: Password Change

CHAPTER

FIVE

CONCLUSION

With the advent of time and continued evolution of technology, ICT enabled education is a must for the betterment of the school education. The Aakash webportal is a step towards achieving this goal, where user can freely gain access to the educational contents of the highest quality which will facilitate them in the learning process. The lack of trained and quality teachers hinders the process of learning for school students so this is where the Aakash webportal comes into picture and solves the problem very precisely and dedicatedly, it provides a platform for teachers, faculties and professers to contribute their contents so as to be freely available to the user. The design of this portal is very simple and user-friendly too, which make it more efficient. Thus we can say that Aakash webportal has a great scope in future for the students and users of Aakash tablet and developers too who want to enhance the functionalities further because a well documented developer guide is given and it is also reposited on Github.

CHAPTER

SIX

REFERENCE

- https://www.djangoproject.com/
- http://www.tangowithdjango.com/
- http://www.startbootstrap.com/
- http://www.stackoverflow.com/
- http://www.aakashlabs.org/
- http://sphinx-doc.org

CHAPTER

SEVEN

DEVELOPER'S GUIDE

7.1 Views

webapp.views.**about**(*request*)
About page.

Argument:

request: Request from client This function takes the request of the client and direct it to the page

consisting of the description about the site.

webapp.views.**commentpost**(*request*)
Argument:

request: This redirects the comments to comments.html page.

webapp.views.**contact**(*request*)
Contact us page.

Arguments: *request*: Request from the client. This function takes the request of the client and direct it to the contact

us page.

webapp.views.**content**(*request*, *lang*)
Argument:

request: This requests the particular content.

lang: This indicates the language of the contents.

webapp.views.**contributor_profile**(*request*)
Arguments:

request: Request from user

This function takes the request of user and direct it to profile page.

webapp.views.**contributor_profile_comment**(*request*, *class_num*, *sub*, *topics*, *id*)

Arguments:

request: Request from user.

class_num: Class in which the logged in contributor has contributed.

sub: Subject in which the logged in contributor has contributed.

topics: Subject topic in which the logged in contributor has contributed.

This function takes the request of user and direct it to profile page

which consists of his comments of reviewer on a specific subject of a specific class.

webapp.views.**contributor_profile_edit**(*request*, **args*, ***kwargs*)

Argument:

request: Request form contributor to edit his profile.

Edit user's/Coordinators profile.

webapp.views.**contributor_profile_subject**(*request*, *class_num*)

Arguments:

request: Request from user

class_num: Class in which the logged in contributor has contributed

This function takes the request of user and direct it to profile page

which consists of his contributions in a specific class.

webapp.views.**contributor_profile_topic**(*request*, *class_num*, *sub*)

Arguments:

request: Request from user

class_num: Class in which the logged in contributor has contributed

sub: Subject in which the logged in contributor has contributed

This function takes the request of user and direct it to profile page

which consists of his contributions in a specific subject of a specific class.

webapp.views.**contributor_profile_topic_detail**(*request*, *class_num*, *sub*, *topics*, *id*)

Arguments:

request: Request from user.

class_num: Class in which the logged in contributor has contributed.

sub: Subject in which the logged in contributor has contributed.

topics: Subject topic in which the logged in contributor has

contributed.

id: Id of the subject in which the logged in contributor has

contributed.

This function takes the request of user and direct it to profile page which

consists of his comments of reviewer on a specified topic of a subject of a specific class.

webapp.views.**contributor_signup**(*request*)

Argument:

request: Request from contributor to sign up.

This function is used for a new contributor to sign up.

Usage:

```
# Create an instance for UserForm() and ContributotForm()
user_form = UserForm(data=request.POST)
contributor_form = ContributorForm(data=request.POST)
if user_form.is_valid() and contributor_form.is_valid():
```

```
        user = user_form.save()
        # do stuff
    else:
        # do stuff
```

webapp.views.**contributor_upload**(*request*)
Argument:

request: Request from contributor for a new upload.

This function is used to upload a new file by contributor.

webapp.views.**detail_user**(*request*)
Argument:

request: This function redirects to the page having detailed information of contributor and reviewer

webapp.views.**index**(*request*)
Argument:

request: Request from client.

This function takes the request of client and direct it to home page.

webapp.views.**language_select**(*request*)
Argument:

request: This requests the particular content.

webapp.views.**reviewer_past_approvals**(*request*)

Argument:

request: Request from contributor to sign up.

This function takes the request of user and directs it to the profile page

which consists of the reviewer's past approvals.

webapp.views.**reviewer_profile**(*request*, **args*, ***kwargs*)
Arguments:

request: Request from user.

This function takes the request of user and directs it to the profile page.

webapp.views.**reviewer_profile_comment**(*request*, *class_num*, *sub*, *topics*, *id*)

Arguments:

request: Request from user.

class_num: Class in which the contributor has contributed.

sub: Subject in which the contributor has contributed.

topics: Topic on which reviewer commented.

id: Id of the reviewer.

This function takes the request of user and directs it to the profile page

which consists of the contributor's contributions in a specific subject of a specific class.

webapp.views.**reviewer_profile_edit**(*request*, **args*, ***kwargs*)
Argument:

request: Request from reviewer to edit his profile.

Edit user's/Reviewer's profile.

webapp.views.**reviewer_profile_subject**(*request*, *class_num*)

Arguments:

request: Request from user.

class_num : Class in which the contributor has contributed.

This function takes the request of user and direct it to the profile page which consists of the contributor's contributions in a specific class.

webapp.views.**reviewer_profile_topic**(*request*, *class_num*, *sub*)

Arguments:

request: Request from user.

class_num : Class in which the contributor has contributed.

sub: subject in which the contributor has contributed.

This function takes the request of user and directs it to the profile page which consists of the contributor's contributions in a specific subject of a specific class.

webapp.views.**reviewer_profile_topic_detail**(*request*, *class_num*, *sub*, *topics*, *id*)

Arguments:

request: Request from user.

class_num: Class in which the logged in contributor has contributed.

sub: Subject in which the logged in contributor has contributed.

topics: Subject topic in which the logged in contributor has contributed.

id: Id of the subject in which the logged in contributor has contributed.

This function takes the request of user and direct it to profile page which consists of his comments of reviewer on a specified topic of a subject of a specific class.

webapp.views.**reviewer_signup**(*request*)

Argument:

request: Request from reviewer to sign up.

This function is used for a new revieweer to sign up.

webapp.views.**search**(*request*, *lang*)

Argument:

request: This requests the searched content.

lang: This indicates the language of the searched contents.

webapp.views.**user_logout**(*request*)

Argument:

request: Request from user to log out

This function is used for logging out.

webapp.views.**userlogin**(*request*)

Argument:

request: Request from the user to login This function takes the request of the user and direct it to the login

page.

7.2 Models

class webapp.models.**Class**(**args*, ***kwargs*)

Fields are:

class_number: This will be class number like first, second .. eight ..

tenth.

remark: This is the remark given by the teacher.

class webapp.models.**Comment**(**args*, ***kwargs*)

Fields are:

subject: This field indicates the subject on which comments are arrived.

This field is the foreign key to the Subject class.

user: This field indicates the reviewer who has commented on the uploaded

files of the contributor. This is a foreign key to the Reviewer class.

comment: This describes the actual comments of the reviewer.

submit_date: Submit_date field tells us when the comment was submitted,

at which time and on which date. This uses DateTime field.

class webapp.models.**Contact**(**args*, ***kwargs*)

Fields are:

name: This field indicates the name of the non-user who wants to suggest

any modifications.

email: This fied indicates the mail-id of the non-user.

message: This field describes the modifications suggested by the non-user,

if any.

class webapp.models.**Contributor**(**args*, ***kwargs*)

Fields are:

user: This is the Default django user object

contact: This is the contact number of the user. It must be an integer.

picture: The profile picture of the contributor. may be jpg or jpeg or png

or bmp.

validation_docs: The Valid certificate of the user stating his educational

qualifications.

class webapp.models.**Faq**(**args*, ***kwargs*)

Fields are:

question: This field descibes the frequently asked questions.

answer: This field describes the answers for the frequently asked questions.

class webapp.models.**Language**(**args*, ***kwargs*)

Language(id, language)

class webapp.models.**Reviewer**(**args*, ***kwargs*)

Fields are:

user: This is the Default django user object.

contact: This is the contact number of the user. It must be an integer.

picture: The profile picture of the contributor. may be jpg or jpeg or png or bmp.

class webapp.models.**Subject**(**args*, ***kwargs*)

Fields are:

contributor: This field is a foreign key to the Contributor class. This is used to refer to the contributors who are already signed up.

name: This field is the name of the subject,the contributor is specialized in.

topic: The subject of the topic the contibutor is going to contribute.

class_number: This may be the class number like first, second .. eight .. tenth for which the contributor is uploading his files.

pdf: The contributor has to upload the files.This field describes that the uploaded file must be a pdf file.

video: The contributor has to upload the files.This field describes that the uploaded file must be a video. May be mp4.

animation: The contributor has to upload the files.This field describes that the uploaded file must be an animation file. This may be gif.

pdf_url: If the size of the uploaded file is high contributor can just mention the url of the file. This field is used to mention the url of the file.

video_URL: If the size of the uploading video is greater than the limited size, contributor can just mention the url of the video. This field is used to mention the url of the video.

animation_url: If the size of the uploaded Animation is greater than the limited size, contributor can just mention the url of the animation. This field is used to mention the url of the animation.

uploaded_on: This field is used to obtain the date on which the file is uploaded by the contributor. This must be the date field.

summary: This is the Summary given by the contributor about the specified topic of the specified subject including when to use and how to use.

rating: The field indicates the rating given by the reviewer for his uploaded files. This must be an integer and default value is 0.

review: This field indicate the number of reviews made by the reviewers. This This must be an integer and default value is 0.

language: This field is a foreign key to the Language class. This is used to refer to the language in which the contributor is contributing.

7.3 Forms

class webapp.forms.**CommentForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Field is:

comment: This field describes the comment form which takes maximum length of 1000 characters. The comment is made by the reviewer.

class webapp.forms.**ContactForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Fields are:

name: This field tells the name of the student contacting us.

email: This field tells the email of the student contacting us.

message: This field tells the message of the student contacting us

class webapp.forms.**ContributorForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Fields are:

contact: This is the contact number of the user. It must be an integer.

validation_docs: The Valid certificate of the user stating his educational qualifications.

picture: The profile picture of the contributor. may be jpg or jpeg or png or bmp.

clean_validtion_docs_file()

Limit doc_file upload size.

class webapp.forms.**ContributorUploadForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Fields are:

class_number: This will be class number like first, second .. eight ..

tenth.

name: This field is the name of the subject,the contributor is specialized

in.

topic: The subject of the topic the contibutor is going to contribute.

pdf: The contributor has to upload the files.This field describes that the

uploaded file must be a pdf file.

video: The contributor has to upload the files.This field describes that

the uploaded file must be a video. May be mp4.

animation: The contributor has to upload the files.This field describes

that the uploaded file must be an animation file. This may be gif.

summary: This is the summary given by the contributor about the specified

topic of the specified subject including when to use and how to use.

clean_animation()
Limit doc_file upload size.

clean_pdf()
Upload a valid.

clean_video()
Limit doc_file upload size.

class webapp.forms.**ReviewerForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Fields are:

picture: The profile picture of the Reviewer. may be jpg or jpeg or png

or bmp.

contact: This is the contact number of the user. It must be an integer.

class webapp.forms.**UserForm**(*data=None*, *files=None*, *auto_id=u'id_%s'*, *prefix=None*, *initial=None*, *error_class=<class 'django.forms.util.ErrorList'>*, *label_suffix=u':'*, *empty_permitted=False*, *instance=None*)

Fields are:

username: This field tells the user who sign in.

first_name: This field tells the First name of the user.

last_name: This field tells the Last name of the user.

email: Email field tells the mail id of the user who sign in.

password: This tells the user to set his own password.

W