**CHAPTER – 1**

**INTRODUCTION**

**1. Introduction**

This project report will introduce you about the project ‘**Engineering Notes**’. How every single part of Engineering Notes is built using the Django framework which is a web framework of python.

The project titled ‘**Engineering Notes**’ is a web application that can serve as a central repository for all the engineering students to access notes of their subjects from anywhere. The students can remotely access the notes of their subject either on the mobile, computers or other digital display devices like Ipods etc. This project will let you to view your notes online and also you can download it so you can access it when there is no internet connection. The project title suggest that we can search for only notes here but it is not like that, we can also search for syllabus and schemes on this web portal. Every single activity related to your exams like preparing the syllabus and get good notes for studying and scoring good marks in the exam can be done by this site.

**1.1 Objective**

The main objectives of this application are :

* Create a repository for engineering students to score good marks in exam.
* Remotely access the notes from anywhere.
* Remotely access the schemes from anywhere.
* Remotely access the syllabus from anywhere.
* Get a notification for every update of notes on website.
* Upload notes from your side if there are.
* Register a complaint if there are mistakes in notes.

When at the last moment of the exam student do not find their notes to study so this portal will be helpful for them to get notes.

It will give you handwritten as well as the printed notes for your study.

**1.2 Scope**

The major scope of making notes is that research shows that taking notes by hand is more effective for remembering conceptual information over the long term than taking notes on a computer or laptop. So our site also provides you the hand written notes.

I recently read two pieces about research conducted by Mueller and Oppenheimer demonstrating that students who write out their notes on paper learn more than students who type notes on laptops. The first was in [Scientific American](http://www.scientificamerican.com/article/a-learning-secret-don-t-take-notes-with-a-laptop/) and the second was in [KQED Public Radio News](http://ww2.kqed.org/mindshift/2015/08/18/taking-notes-is-the-pen-still-mightier-than-the-keyboard/).

Because students can type significantly faster than they can write, those who use laptops in the classroom tend to take more notes than those who write them out. At first, this may seem like an advantage. However, when students take notes using laptops they tend to take notes verbatim, writing down every last word uttered by the presenter. This requires a different, less challenging type of cognitive processing than taking notes by hand. Because paper note takers can’t write everything down, they have to listen, digest, and paraphrase succinctly to capture information in notes. This “forces” their brains to be more active which in the end fosters stronger comprehension and long-term memory.

Mueller and Oppenheimer’s studies revealed that while both types of note-takers performed equally well answering questions that involved recalling facts, laptop note-takers performed significantly worse on conceptual questions.  The same results occurred even when the researchers told the students to avoid taking verbatim notes on the laptop, suggesting that the urge to do so when typing is hard to overcome. Basically, the shallow transcription that happens with typing notes fails to promote meaningful understanding or application of the information.

**1.3 Problem Identification**

Student Often find problems because books and notes are limited and only few students can have the books and notes . Student need to fall in line and then ask for the related notes and references .Through this, some students are not able to follow their lesion or notes.

The answer for this problem in the enhancement of technology by developing the Engineering Notes .The efficiency of this Enotes are arranged and how easily one can get the notes of their choice.

## Believe it or not, the syllabus for each class you take at the University is your passport for success. It is a contract of sorts and is filled with valuable information. By staying enrolled in a class, you are indicating that you have read the syllabus, understand what you need to do to be successful in that class, and accept the terms of this agreement. But like reading any other document, it helps to have guidelines to understand the information you find in a syllabus.

A scheme of work is a guideline that defines the structure and content of an academic course. It maps out clearly how resources (e.g. books, equipment, time) and class activities (e.g. teacher-talk, groupwork, practicals, discussions) and assessment strategies (e.g. tests, quizzes, Q&A, homework) will be used to ensure that the learning aims and objectives of the course are met successfully. It will normally include times and dates. The scheme of work is usually an interpretation of a specification or syllabus and can be used as a guide throughout the course to monitor progress against the original plan. Schemes of work can be shared with students so that they have an overview of their course.

It is a brief record of writing something to assist the memory or for future important references. Notes are usually being taken to record the speeches or dictations after listening to it or after reading any materials like a book, magazine or an article. It is initiated to refer whenever needed and may be reproduced in the desired manner. This is article is on note making examples.The process of grasping knowledge is vast and unlimited. Our memory is not able to store so many information altogether, hence notes making is quite handy and helpful. With the assistance of note making, we may recall the entire information of past events. For the students, it makes their preparation less strenuous.

**1.4 Description**

The project is built using the Django framework which is a framework for web development in python. It is a framework that is based on the MVC pattern i.e. the Model View and Controller Pattern. In the later section we have discussed about the MVC pattern in detail.

Our project have various features like:

* Download Notes
* Download Schemes
* Download Syllabus
* Search for notes, scheme and syllabus
* Uploading of notes
* Subscribe to our website
* Get notification for all the updates on the website.
* Admin of this website can add new faculty members who can upload notes.

**CHAPTER – 2**

**BACKGROUND AND LITERATURE SURVEY**

**2. Background And Literature Survey**

**2.1 Existing System**

* The Admin has to perform Update , Delete , Rename etc. manually using django Framework.
* The User has to perform Subscribe option in Engineering Notes Web application.
* Update facility is not available for User or Students.

**2.2 Requirement Specification**

* HTML – HTML refers to hyper text markup language in this project we use for web page layout.HTML is document-layout and hyperlink - specification language used to design the layout of a document and to specify hyperlink.
* CSS is a language that describes the style of an HTML document. CSS describe be how HTML elements should be displayed. CSS stands for cascading style sheet.
* Bootstrap ,the world’s most popular framework for building responsive mobile-first sites. We use bootstrap in our project ,also it is free and open source front end web frameworks, itconcerns itself with front end development only.
* Javascript – Javascript is the programming language of HTML and the web. Javascript has many functions to develop a website. Javascript supporting object-oriented, imperative and declarative styles.
* Django framework- Django is an opensource python web framework used for rapid development, pragmatic ,clean design and secure website. Django has its own naming system for all function. It is fast and simple.
* Python - Python is a high level , general purpose programming language. Python has a design philosophy that emphasize code reliability. We use python because it can be used to create web application.

**2.3 Feasibility Specifications**

* Technical Feasibilities – Technical Feasibility centers around the existing Computer System (Hardware and Software etc)and to what extends it support the proposed addition. The tools, operating system and programming language used in this localization process is compatible with the existing one.
* Economical Feasibilities – The cost to conduct a full system Investigation , The benefits in the form of reduced cost or fewer costly errors.
* Operational Feasibilities – it is the measures of how well a proposed system solves the problems and it takes advantages of the opportunities identified during the scope definition and problem analysis phases.
* Legal Feasibilities – Since the proposed system is going to be implemented in institute, it may be subject to rule and regulation imposed and even funding constrains.

**2.4 Innovativeness And Usefulness**

* It Increases efficiency
* It Reduces the cost of managing a library
* It saves Time
* It increases the productivity
* It enhance the presentation of Notes (1)Accurate Information around the clock
* Information about the library database is accurate at any given point of time since its all automatic and updated after every transaction.

**2.5 Modification and improvement over the existing implementation**

* Accurate Information around the clock - Information about the library database is accurate at any given point of time since its all automatic and updated after every transaction.
* Ease of access - Users can find items, reserve book, change profile details, etc., from their homes as the library’s transactions online.
* Maintenance is ease - Managing the database is easy and so is its maintenance. It does not require much of care and goes on until it isn’t affected due to heavy trafficking.
* Sorting of data - Data is stored in different databases and sorted into different tables to differentiate content and it can easily converted into reports for presentations.

**CHAPTER – 3**

**PROCESS MODEL**

**3. Process Model**

**3.1 Software process model used**

A software process model is a simplified representation of a software process in our Engineering notes project. This model represents a process from a specific perspective.

We’re going to take a quick glance about very general process models. These generic models are abstractions of the process that can be used to explain different approaches to the project. They can be adapted and extended to create more specific processes.

**3.2 Proposed Project Model**

Proposed System provides with following solutions:

* It provides “better and efficient service to users or students.
* Reduce the workload of Student and teachers .
* Faster retrieval of information about the desired Subject.
* Provide facility for proper monitoring reduce paper work and provide data security.
* All details will be available on a click.

**3.3 Market Potential and Competitive Advantage**

Our findings indicate that formal strategic planning does indeed evolve along similar lines in different institutes, albeit at varying rates of progress. This progression can be segmented into sequential phases, each marked by clear advances over its predecessor in terms of explicit formulation of issues and alternatives, quality of preparatory , readiness of top management to participate in and guide the strategic process, and effectiveness of implementation.

**3.4 Project Estimation**

Estimation of various project parameters is a basic project planning activity. The important project parameters that are estimated include: project size, effort required to develop the software, project duration, and cost. These estimates not only help in quoting the project cost to the customer, but are also useful in resource planning and scheduling. There are three broad categories of estimation techniques:

(1) Empirical estimation techniques - Empirical estimation techniques are based on making an educated guess of the project parameters. While using this technique, prior experience with development of similar products is helpful.

(2) Heuristic techniques - Heuristic techniques assume that the relationships among the different project parameters can be modeled using suitable mathematical expressions. Once the basic (independent) parameters are known, the other (dependent) parameters can be easily determined by substituting the value of the basic parameters in the mathematical expression. Different heuristic estimation models can be divided into the following two classes: single variable model and the multi variable model.

(3) Analytical estimation techniques - Analytical estimation techniques derive the required results starting with basic assumptions regarding the project. Thus, unlike empirical and heuristic techniques, analytical techniques do have scientific basis. Halstead’s software science is an example of an analytical technique. Halstead’s software science can be used to derive some interesting results starting with a few simple assumptions. Halstead’s software science is especially useful for estimating software maintenance efforts. In fact, it outperforms both empirical and heuristic techniques when used for predicting software maintenance efforts.

**3.5 Project Schedule**

Online Engineering notes is an application which refers to enotes systems which are generally small or medium in size. It is used by Students And Teachers manage the notes using a computerized system where he/she can record various transactions like Download the notes, view the notes, addition of new books, addition of new students and Addition of new admin etc.

Notes, Scheme and Syllabus maintenance modules are also included in this system which would keep track of the students using the enotes and also a detailed description about the notes a software contains. With this computerized system there will be no loss of Notes record or member record which generally happens when a non computerized system is used.

**CHAPTER – 4**

**DESIGN**

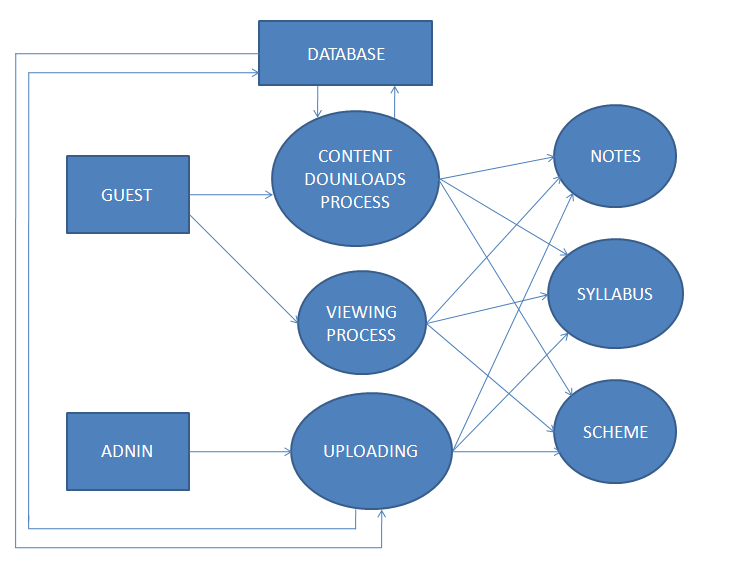
**4. Design**

**4.1 Data Flow Diagram**

DFD goes deeper into parts of Engineering Notes Project. In this Data flow diagram contains more details of Student(Guest), Admin, Book Scheme Notes Syllabus etc.

Low level functionalities of Engineering Notes (e.Notes) System

* Admin login to the system and manage all the functionalities of Engineering Notes (e.Notes) System.
* Admin can edit, upload, view, delete, the record.
* Admin can manage all the details of Notes, Syllabus, Scheme.

****

**Figure No. 4.1**

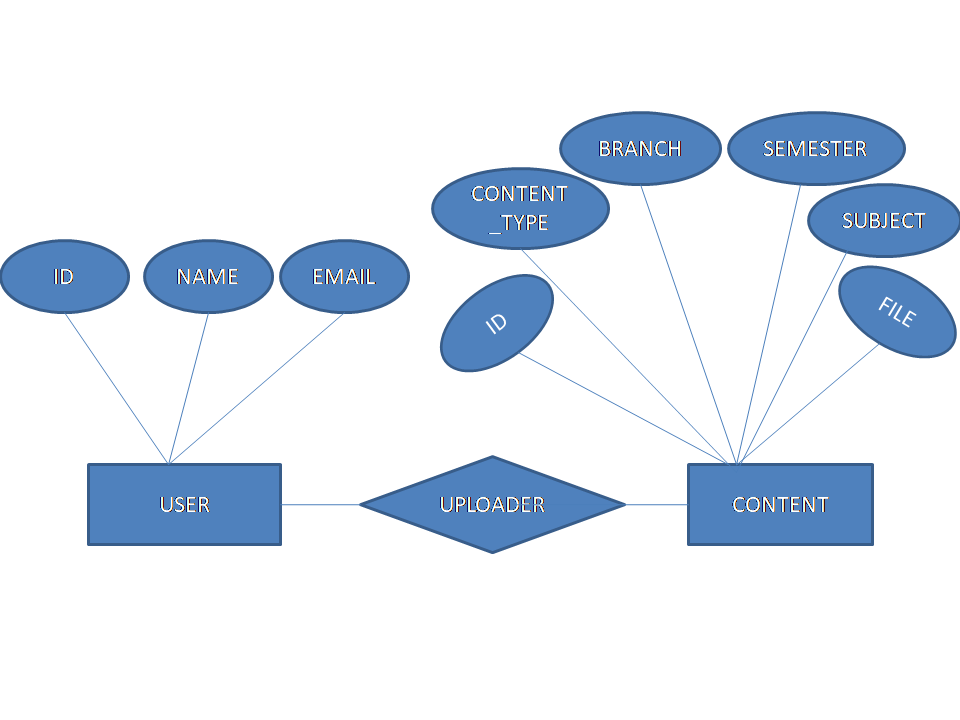
It does not show information about process timing or whether processes will operate in sequence or parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.

**4.2 ER diagram**

This ER (Entity Relationship) Diagram represent the model of Engineering Notes. The entity relationship diagram of Engineering Notes System shows all the visual instrument of database tables and the relations between books, Student, Admin etc. It used structure data and to define the relationship between structured data groups of Notes functionalities. The main entities of the Engineering Notes project are Student, Admin, Notes, Syllabus, Subscribe option etc.

Engineering Notes System entities and their attributes:

* User Entity : Attribute of user entity are User\_ID, User\_Name, User\_Email.
* Content : Attributes of content entity are ID, Content\_type, Branch, Semester, Subject, File.

****

**Figure No. 4.2**

**Description of Engineering Notes System Database**

1. The details of Notes is store into the django database tables respective with all tables.
2. Each entity (Subscribe, Content, Syllabus, Scheme, update etc.) contains keys.
3. The entity issues, member has binded with user, Notes entities with django database keys.
4. There is one-to-one and one-to-many relationships available between Member, admin, users.
5. There are NoSql database use.

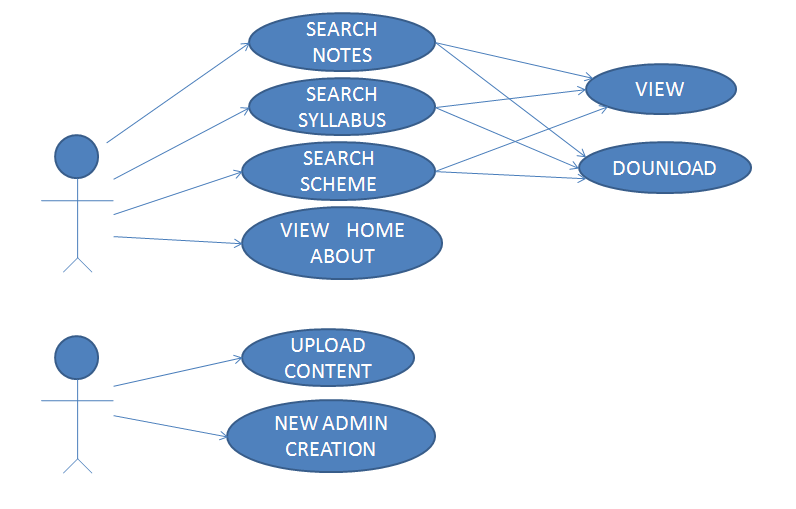
**4.3 Flow Chart**

The system flow diagram is a visual representation of all processed in sequential order.  
The System flow chart diagram is a graphical representation of the relation between all the major parts or step of the Engineering e.Notes system. Flow chart diagram can not include minor parts of the Engineering e.Notes system.

The activity diagram used to describe flow of activity through a series of actions. Activity diagram is a important diagram to describe the system. The activity described as a action or operation of the system.

A system flow diagram is a way to show relationships between a business and its components, such as customers (according to IT Toolbox.) System flow diagrams, also known as process flow diagrams or data flow diagrams, are cousins to common flow charts.

**4.4 Use Case Diagram**

****

**Figure No. 4.3**

A use case diagram is the primary form of system/software requirements for a new software program of our engineering notes project under developed. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

A use case diagram is usually simple. It does not show the detail of the use cases in this project :

(1) It only summarizes some of the relationships between use cases, admin, and systems.

(2) It does not show the order in which steps are performed to achieve the goals of each use case for our Engineering Notes Project.

**4.5 Sequence Diagram**

Sequence diagram descries interaction among classes in terms of an exchange of message over time. sequence diagram demonstrate the behavior of objects in a use case by describing the object and messages they pass. A sequence diagram depicts the sequence of actions that occurs in system. the invocation of methods in each object and the order in which they captured in a sequence diagram. this makes sequence diagram very useful.

Whenever a Student or user requires for a notes, the django database checks the availability of Notes, if the Notes is available then the rack number of that subject will be returned to User end. Admin then checks the validity of user by subscribing the Engineering Notes Project site. if member is valid then the number of Notes download to him is less then maximum allowed and transaction is created. Admin then update the number of status of syllabus.

**4.6 Activity Diagram**

An activity diagram for Engineering Notes online project . Online user can browser search items, view specific item, add it to our machine, view and download the notes related his subject, scheme and syllabus checkout. User can view engineering Notes online website at any time. Checkout is assumed to include Admin login. This site is not using partitions, most of the actions are assumed to be fulfilled by online.

**4.7 Class Diagram**

In class diagram describes the structure of system by showing the system classes, attributes, methods, and associations between classes.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The Engineering E-notes System class diagram consist of 8 classes:

* Admin Login
* Scheme
* Syllabus
* Notes
* User
* View
* Download
* Subscribe

**4.8 Algorithm**

**Main Function**

**Steps**

* Start With Welcome Screen
* Go to user panel
* Go to Admin Section or panel
* Check Admin password. Is password correct

Yes:-goto step 5

No:-goto step 3or2

* Display Admin menu as below

1 add notes

2 delete notes

3 update notes

4 edit notes

5 view notes

6 close application

* Get choice from user

Choice:-1 Select Scheme

Choice:-2 Select Syllabus

Choice:-3 View Notes

Choice:-4 Download Notes

Choice:-5 Goto step 7

* Stop

**CHAPTER – 5**

**TOOLS AND TECHNOLOGY USED**

**5. Technology And Tools used**

**5.1 Front-end Tools**

1. HTML: - Html is a Hyper text markup language used for creating web pages.

Uses of Html:

1. Web Document Creation

2. Internet Navigation

3. Cutting edge feature

4. Responsive image on web pages

5. Client-side storage

6. Offline capabilities usage

2. CSS : - CSS stands for Casecading Style Sheet. CSS describes how Html elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

External stylesheets are stored in CSS files.

3. Java Script : - JavaScript is most commonly used as a client side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it.

Used in Web pages, JavaScript is a "client-side" programming language. This means JavaScript scripts are read, interpreted and executed in the client, which is your Web browser. By comparison, "server-side" programming languages run on a remote computer, such as a server hosting a website.

**5.2 Back-end Tools**

1. Django framework :

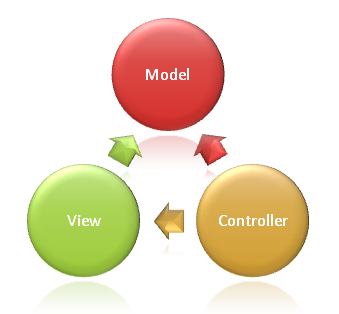
Django is an open source web application frame work written in Python. The primary goal of Django is to make the development of complex, data-based websites easier. Thus Django emphasizes the reusability and plug ability of components to ensure rapid developments.

Django consists of three major parts: model, view and template

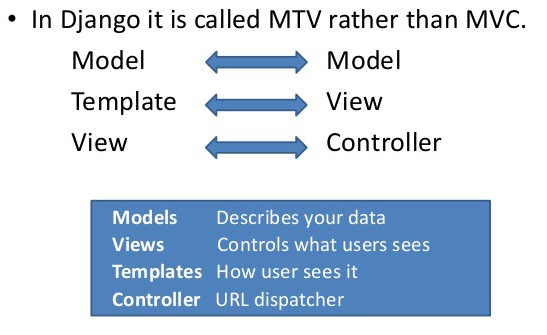
**Model**: This handles your data representation, it serves as an interface to the data stored in the database itself, and also allows you to interact with your data without having to get perturbed with all the complexities of the underlying database.

**View**: As the name implies, it represents what you see while on your browser for a web application or In the UI for a desktop application.

**Controller**: provides the logic to either handle presentation flow in the view or update the model’s data i.e it uses programmed logic to figure out what is pulled from the database through the model and passed to the view,also gets information from the user through the view and implements the given logic by either changing the view or updating the data via the model , To make it more simpler, see it as the engine room.

****

**Figure No. 5.1**

****

**Figure No.5.2**

**Model**: Just like the Model explanation in the MVC pattern , this also takes the same position as the interface or relationship between the data and contains everything related to data access and validation.

**Template**: This relates to the View in the MVC pattern as it is the presentation layer that handles the presentation logic in the framework and basically controls what should be displayed and how it should be displayed to the user.

**View**: This part relates to the Controller in the MVC pattern and handles all the business logic that throws down back to the respective templates. It serves as the bridge between the model and the template.

**5.3 Operating System**

Any Operating System can be used to run the project. Examples of various operating systems are

* Windows 7, 8, 8.1, 10 and upcoming versions.
* IOS 10, 11, 12 and upcoming versions.
* Linux
* And many more.

**5.4 Additional Software Requirements**

This project runs on web browser and require a terminal to turn its server on and also django framework to built it.

**CHAPTER – 6**

**HARDWARE REQUIREMENT**

**6. Hardware Requirement**

|  |  |
| --- | --- |
|  | **6.1 System Configuration (CPU, RAM, HDD, Network used)** |

A normal computer system is required to run this project.

RAM – 1- 4 GB

HDD – 512MB – 2 TB

**CHAPTER – 7**

**CODING**

**7. Coding**

**7.1 models.py**

from django.db import models

class Subscriber(models.Model):

name = models.CharField(max\_length=20)

email = models.EmailField(max\_length=30)

def \_\_str\_\_(self):

return self.name

class Content(models.Model):

content\_type = models.CharField(max\_length=20)

branch = models.CharField(max\_length=20)

semester = models.IntegerField()

subject = models.CharField(max\_length=40)

file = models.FileField()

def \_\_str\_\_(self):

return self.subject

class User(models.Model):

username = models.CharField(max\_length=30)

password = models.CharField(max\_length=30)

def \_\_str\_\_(self):

return self.username

**7.2 views.py**

from django.shortcuts import render, redirect

from .models import Subscriber, Content, User

from django.core.mail import EmailMessage

def index(request):

return render(request,'Index.html')

def about(request):

return render(request,'About.html')

def subscribe(request):

name = request.POST['fullname']

email = request.POST['email']

data = Subscriber(name=name,email=email)

data.save()

mail = EmailMessage('Welcome','Welcome to Oriental Notes. You have successfully registered to this website. You will be notified through a mail for all the updates on this website.',to = [email])

mail.send()

return redirect('/')

def notes(request):

return render(request,'Notes.html')

def search(request):

content\_type = request.POST['type']

branch = request.POST['branch']

semester = request.POST['semester']

data = Content.objects.all().filter(content\_type=content\_type,branch=branch,semester=semester)

return render(request,'NotesSearch.html',{'data':data})

def adminlogin(request):

return render(request,'AdminLogin.html')

def checklogin(request):

username = request.POST['username']

password = request.POST['password']

login = False

data = User.objects.all().filter(username=username)

for i in data:

if i.password==password:

login = True

break

if login:

return render(request,'Upload.html')

else:

return redirect('/AdminLogin')

def upload(request):

content\_type = request.POST['type']

branch = request.POST['branch']

semester = request.POST['semester']

subject = request.POST['subject']

file = request.FILES['file']

data = Content(content\_type=content\_type,branch=branch,semester=semester,subject=subject,file=file)

data.save()

data = Subscriber.objects.all()

reciever = []

for i in data:

reciever.append(i.email)

email = EmailMessage('Content Uploaded','We have uploaded '+content\_type+' for '+branch+' branch and '+semester+' semester for the subject '+subject,to=reciever)

email.send()

return render(request,'Upload.html')

**7.3 urls.py**

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$',views.index,name='Index'),

url(r'^About/',views.about,name='About'),

url(r'^Subscribe/',views.subscribe,name='Subscribe'),

url(r'^Notes/',views.notes,name='Notes'),

url(r'^Search/',views.search,name='Search'),

url(r'^AdminLogin/',views.adminlogin,name='AdminLogin'),

url(r'^CheckLogin/',views.checklogin,name='CheckLogin'),

url(r'^Upload/',views.upload,name='Upload'),

]

**CHAPTER – 8**

**TESTING AND MODIFICATION**

**8. Testing and Modification**

**8.1 Testing Methods Used**

Two types of testing have been used by us in this application

* **Unit Testing -** Unit testing is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. We have created each and every module one by one and after the successful testing only we have gone for development of next module.
* **Beta Testing -** Beta Testing is one of the Acceptance Testing types, which adds value to the product as the end-user (intended real user) validates the product for functionality, usability, reliability, and compatibility. Inputs provided by the end-users helps in enhancing the quality of the product further and leads to its success. This also helps in decision making to invest further in the future products or the same product for improvisation. We have tested all the modules with the end user’s do the website is accepted after beta testing.

**8.2 Test Case With Results**

The test cases we done with the

* Uploading the content.
* Downloading the content.
* Subscribing the website.
* Creation of new administrator.
* Creatiion of a new faculty (Uploader).

**8.3 Result Analysis**

The result after all the testing and test cases was positive. No further changes in the website were required. All the test cases passed successfully.

When the user subscribes to website he is notified.

Uploaded content can be viewed.

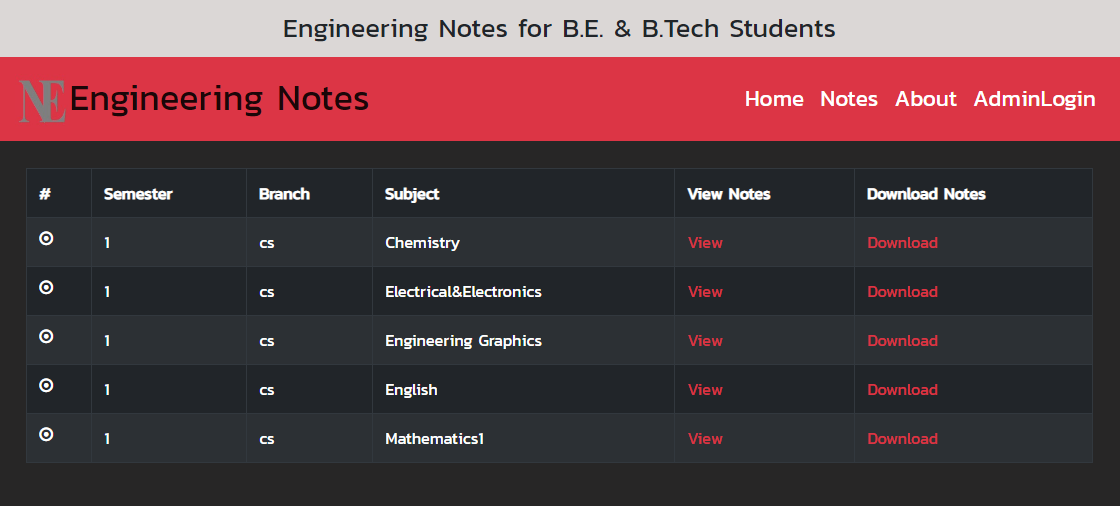
Downloaded content is well stored.

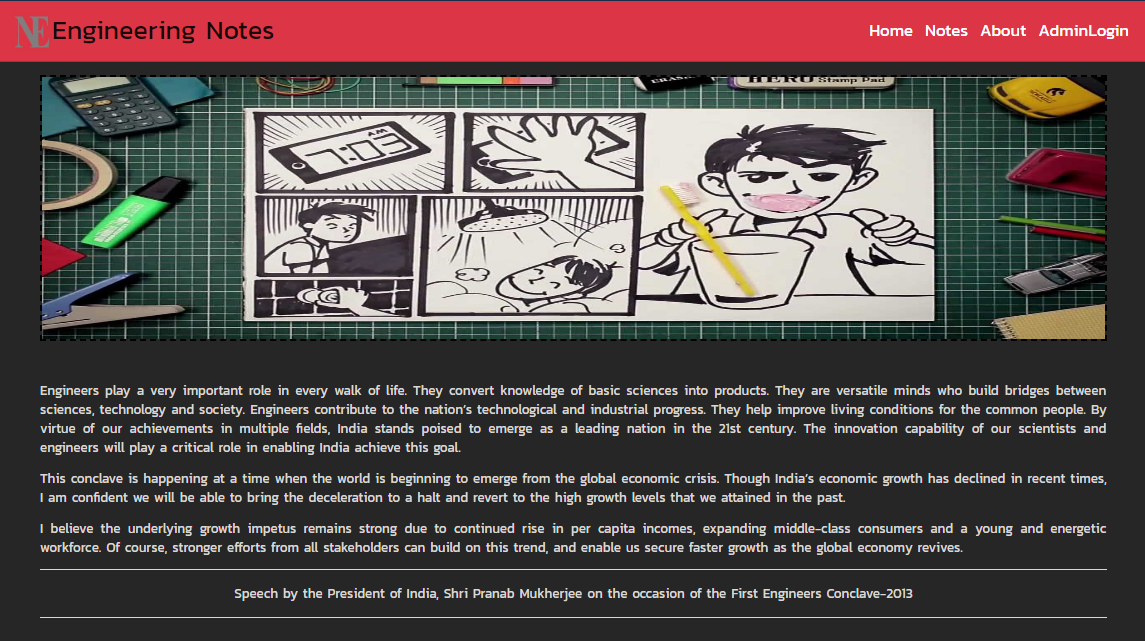
Integrity of the content is maintained.

**CHAPTER – 9**

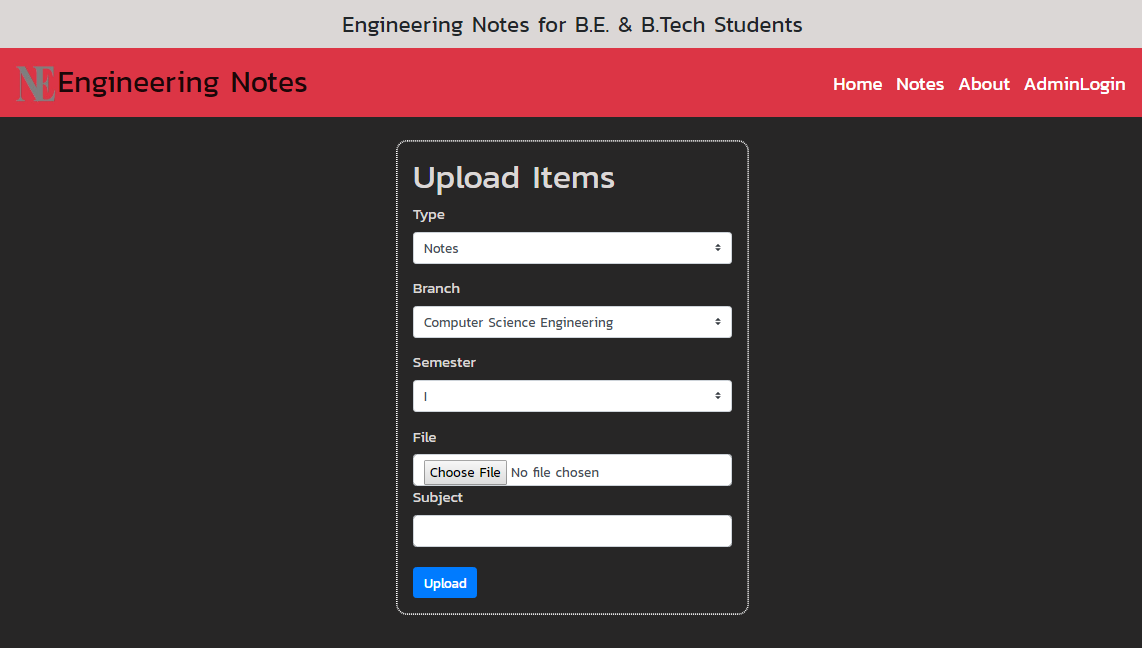
**SCREEN LAYOUT**

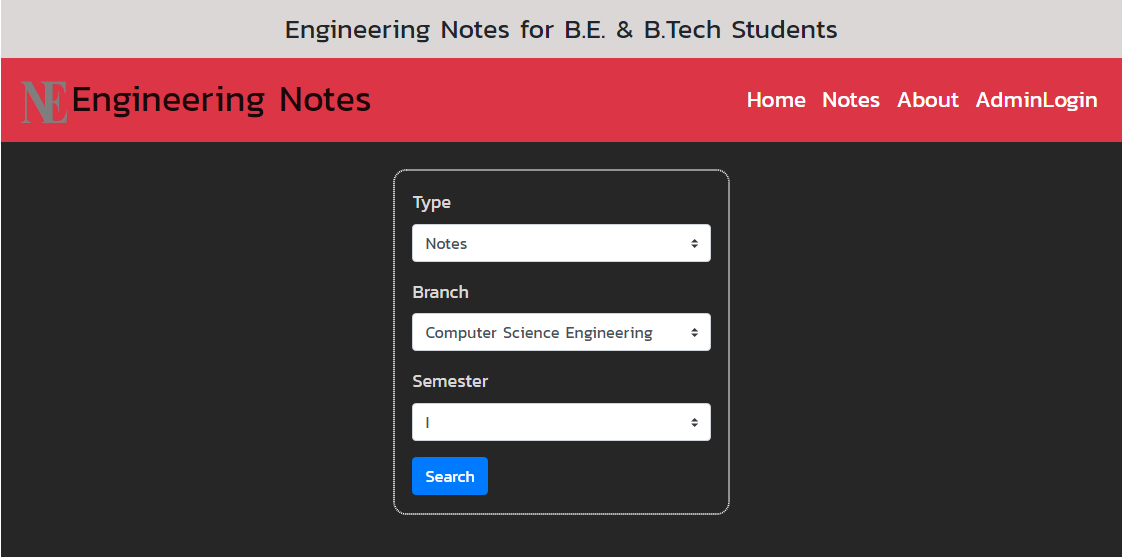
**9. Screen Layout**

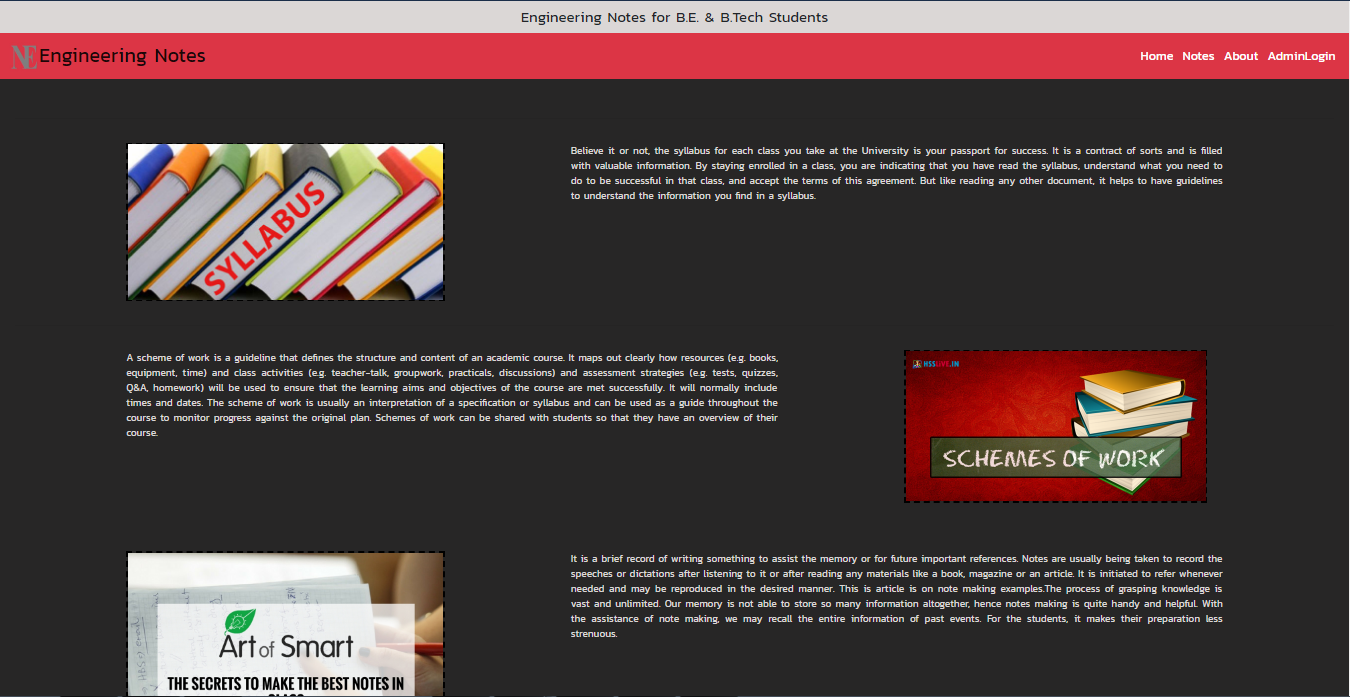
****

****

****

****

****

****

**CHAPTER – 10**

**FUTURE ENHANCEMENT**

**10. Future Enhancement**

These following modifications are to be done in the future version of this website

* Self Registration of faculty members to upload notes.
* Feature to update the existing notes.
* Register a complaint on any integrity defect in notes.
* Review and feedback system for students.
* Student can send us notes to upload on website.

**CHAPTER – 11**

**APPENDICES**

**11. APPENDICES**

The project is all about the study material or content which can be used as a serving hand in various engineering exams.

The various models used in the project are

* User Model : It contains fields

1. Id
2. Username
3. Password

* Content Model : It Contains fields

1. Id
2. Content\_type
3. Branch
4. Semester
5. Subject
6. File

* Subscriber : It contains fields

1. Id
2. Name
3. Email-ID

The project aims to redirect the examination goals of the students.

**CHAPTER – 11**

**REFERENCES**

**12. References**

* Two scoops of django – Daniel Roy
* HTML & CSS – John Ducket
* JavaScript and jQuery – John Ducket
* Online sources like youtube channels
* Official Documentation – www.python.org
* Faculty Members – Prof. Swati Pandey
* HOD Sir – Pratik Buchke