School of Computing Science and Engineering
Greater Noida, Uttar Pradesh
Fall 2021 - 2022

A Django Based Educational Resource Sharing Website

S.No	Enrollment Number	Admission Number	Student Name	Degree / Branch	Sem
1	19021011677	19SCSE1010506	Aditya Raj	B.Tech / CSE	5
2	19021011877	19SCSE1010732	Lav Patel	B.Tech / CSE	5

Under the Supervision of

Mr. Samson Ebenezar U

TABLE OF CONTENTS

Abstract

Technological Implementations in the field of Academics has assisted Students with welling Professionals in vital ways. The availability of all educational resources helps the students a ton in their educational life. Our Project will show a website model with the assistance of which Students can have the option to get to class notes, earlier year question papers, syllabus, and can sell their old books from a similar digital platform also. The paper Will likewise portrays the job of software engineering in project development. The task will be created on Django Framework; the back-end development will be in Python, Jinja2 and SQLite. The frontend comprises of HTML, CSS and JavaScript. Suitable SDLC Model and Testing procedures will be utilized in the development process. Each progression of the SDLC Model (Iterative Model) will portrayed completely and particular ER Diagrams and Flow Charts will be show. The task created is profoundly effective, easy to use and straightforward.

INTRODUNCTION

In this era of digitalization, the availability of different resources on different digital devices is making our lives easy and convenient. In the field of academics too, there are many educational websites that share educational resources like short notes, video lectures, presentations and books etc. But, none of them provide class notes or question papers related to courses taught in a particular college or university; and most of the students due to lack of communication from batch mates or seniors, often face the problem of not getting class notes, if were absent or previous year question papers from seniors. Taking these problems under consideration, the idea of an online platform for resource sharing purposes is developed named Shrewish (*Sh*aring *Re*sources *I*n *C*ampus). The idea is to provide all class notes and previous year question papers of different courses of a college/university at a single platform.
Also, in colleges after completion of the course, students either sell their books to junkyards or few of them donate it to their juniors. Only a few of them keep those books with themselves. So, the website also works like an e-commerce website where old books can be sold or donated as per the wish of the seller.The project is developed in the Django framework. The back-end consists of Python, Django, SQLite and Jinja2. Cloudinary, an online cloud service has been used for storing data. Finally, black box and white box testing were done in order to test the functional, structural and logical features of the website.

The improvement interaction followed the Iterative Model of Software Development. The thought was to add usefulness and afterward to plan it, test it and execute it. Albeit this methodology takes more resources however with every iteration the following iteration sets aside less effort

to be created and with the assistance of this methodology, mistakes were handily found and amended simultaneously.

The plausibility concentrate on assisted us with enrolling the primary goals viz.

1.Different Educational Categories (viz. Entrance, Recruitment, Academics and Entertainment).

2.Different Subcategories inside each Educational Category.

3.Each Subcategory will contain particular Old Books, Question Papers and Class Notes.

4.Each User can add Class Notes, Old Books or Question Papers which once confirmed by Admin Users will be added on the website.

5.Old Books can be bought through Cash on Delivery Method.

6.A Chat-Box for correspondence among Customers and Sellers.

7.The User will see resources applicable to its own University as it were.

The testing or we can say that checking of the project was done in two ways viz. Discovery Testing which was finished by Users. The Users were approached to run the project and actually look at the highlights in general. The criticism was recorded and changes were made as required; and White Box Testing in which diverse experiments were made for every unit of source code and were tried. For each experiment, the ideal result was normal. At the point when the ideal result was not experienced it prompted a bug. Every mistake was taken out from the source code and all units were coordinated finally.

The project is created under the Django Python Web Framework. It energizes quick turn of events and perfect, commonsense plan. Worked by experienced designers, it deals with a large part of the problem of Web improvement, so we can zero in on composing our application without expecting to waste time. The front finish of the project has been made easy to use with effective utilization of HTML, CSS and JavaScript. Also one of the best thing is that The back end has been coded in Python Programming Language. We effectively accomplish more capacities with less lines of code utilizing Python. Web advancement with Python is extremely well known in light of its lucidness and proficiency.

Python is utilized for the advancement cycle taking the security issues in concern. Python is safer than several broadly utilized programming dialects. Django further assists undertakings with upgrading the security of their websites and web applications by forestalling an assortment of safety assaults — cross-web page prearranging (XSS), cross-website demand phony (CSRF), SQL infusion, and click jacking. The web application is made to trade information with the web server safely by sending the web application behind HTTPS. The security issues are settled to forestall any undesirable assault on the information base just as the server.

The project later advancement was sent on a public server with the domain www.shreic.com. The website is exceptionally easy to use and simple to operate. This project will help understudies a great deal in their instructive life.

Literature Survey

Various websites and research papers have been developed keeping the idea of educational resource sharing and its importance in mind. Few of them are listed here:

1. Used Books Factory:
It is an online platform to sell old books of different categories.

2. Vioric-Torri, C. & Alexandrache, C. (2012):
The study reflects how educational technology influences the learning styles of students and how to form and develop the competences of learning in the new generations.

3. TutorialsPoint:
The website provides tutorials on different topics related to computer science and technology. Provides pdf notes for the same and also provides guidance for competitive exams.

4. The Physics Classroom:
For PDF notes and tutorials related to the various fields of Physics.

5.Kelly, L., & Breault, K. (2006):
The objective of the research project was to provide the Australian Museum with guidance on how to best develop a website that meets the needs of students and teachers in the primary and secondary levels across a range of curriculum areas. General objectives were to gain insights into how students and teachers are using the internet and what they are looking for when they access websites

Solution Designed

The development process will follow the Iterative Model of Software Development. The idea is to add functionality and then to design it, test it and implement it. Although this approach takes more resources but with each iteration the next iteration will take less time to be developed and with the help of this approach, errors will easily find and will rectify at the same time.

Required Hardware and Software

The following specifications
are required for the project to run on any device.

1) System Specifications

Processor: Intel(R) Core (TM) ie-5005U CPU @ 2.00GHz

RAM: 2 GB

System Type: 32-bit/64-bit operating system, x32 or x64 based processor

Operating System: Windows 7/8/10.

2) Software Interface

Front End: HTML, CSS, Bootstrap, JQuery

back-end: Django

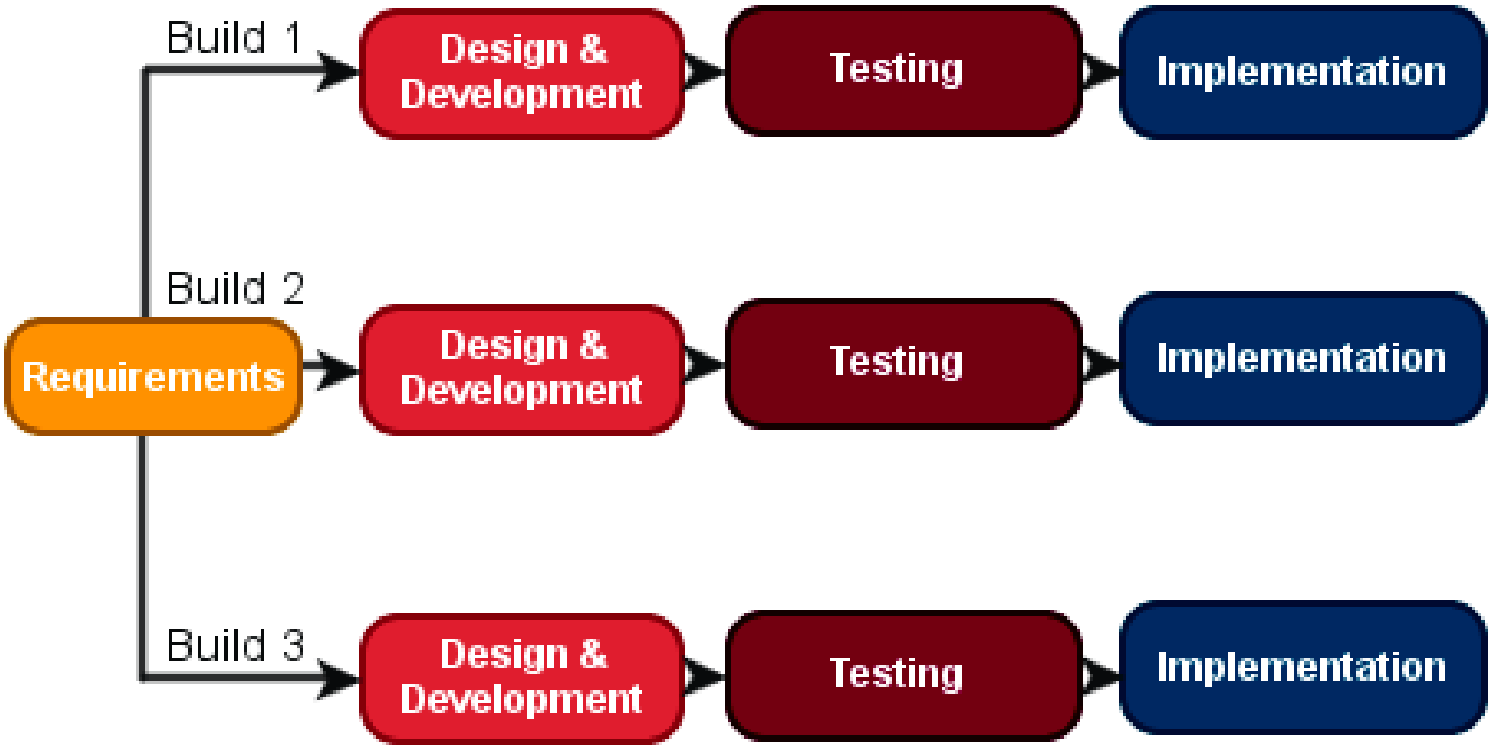
Local Access Link: localhost:8000

Global Access Link: https://www.xyz.com/home/

In this life cycle model, a Project Control List (PCL) based on current realized prerequisites is created. A PCL is a rundown containing the series of errands/functionalities that are to be available in the given framework. On the off chance that at a specific period of improvement, we run over any new necessity, we add it to our Project Control List.
For fostering the website, an undertaking is browsed the given
PCL and Planning, Analysis, Designing, Testing and Evaluation is proceeded as displayed in Figure 1. At the point when the particular usefulness is added we eliminate it from the Project Control List. Additionally, each errand in turn from PCL is picked, executed and afterward eliminated from PCL. This cycle iterates until the ideal prerequisites of the item are not met.Later every iteration, the supervisory crew can take care of business on hazard the board and plan for the following iteration. Since a cycle incorporates a little piece of the entire software process, it is simpler to deal with the improvement interaction.

In the Iterative model, the more current iterations are steadily further developed variants of previous iterations. Besides, if another iteration in a general sense breaks a framework in a cataclysmic way, a previous iteration can rapidly and effectively be carried out or "moved back," with insignificant misfortunes, which is an aid for post-discharge upkeep.

In the Iterative Model, the underlying go through of all stages might take some time, however each ensuing iteration will be progressively fast, permitting the existence pattern of each new iteration to be managed down to only days or even hours now and again.



Feasibility Analysis

A feasibility analysis is used to determine the viability of an idea, like ensuring a project is legally and technically feasible as well as economically justifiable.

During the development of this project, the feasibility study Will done as follows:

1. Different Educational Categories (viz. Entrance, Recruitment, Academics and Entertainment).

2. Different Subcategories within each Educational Category.

3. Each Subcategory will contain respective Old Books, Question Papers and Class Notes.

4. Each User can add Class Notes, Old Books or Question Papers which once verified by Admin Users will be added on the website.

5. Old Books can be purchased via Cash on Delivery Method.

6. A Chat-Box for communication between Customers and Sellers.

7. The User will see resources relevant to its own University only.

Work Plan layout

First Phase of Work:

The project will develope in the Django framework. The back-end consists of Python, Django, SQLite and Jinja2. Cloud nary, an online cloud service will be used for storing data.

Second Phase of Work:

Finally, black box and white box testing will be done in order to test the functional, structural and logical features of the website.

E-R Diagram

The project consists of many relational models. Every model consists of different attributes. The Primary Key is shown with *Institute of Science, BHU Varanasi, India* an underline under the name. Each Relation is characterized by Structural Constraint where the first number denotes Participation Constraint (1 for Total Participation and 0 for Partial Participation) and the second number denotes the Cardinality Ratio (1:1 or 1: N or M: N).

1) Participation Constraint:
Total Participation: If each entity of an Entity Type has a relationship instance in Relationship Set then the participation is total. Partial Participation: If few entities of an Entity Type have a relationship instance in Relationship Set then the participation is partial.

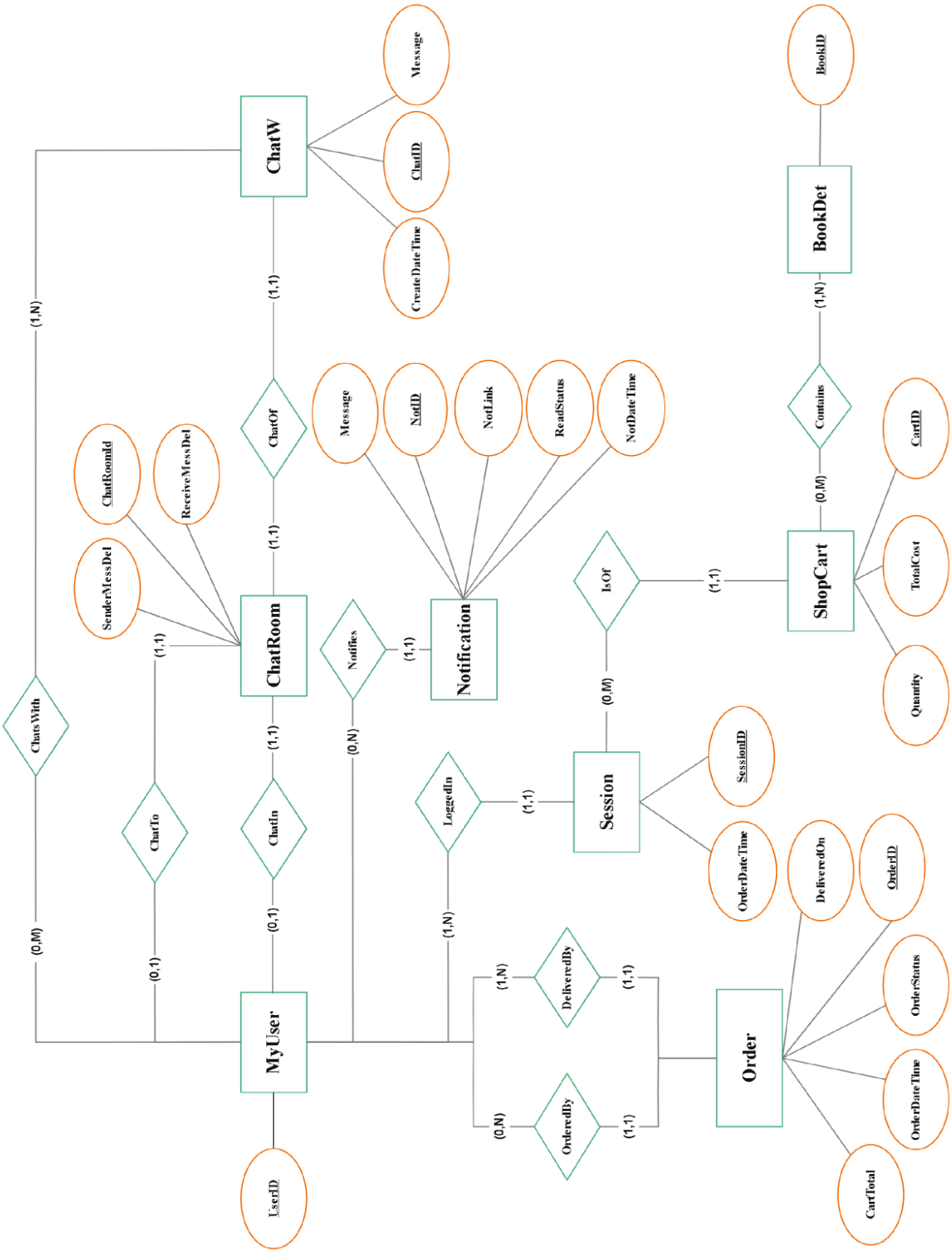
2) Cardinality Ratio:

1:1: Only one entity of an Entity Set can be related to anyone entity of the other Entity Set.

M: N: Many entities of an Entity Set can be related to many entities of the other Entity Set.

1: N: One entity of an Entity Set can be related to any number of entities of the other Entity Set. There are 3 ER Diagrams that are used in the development process. The ER Diagram in Figure 2 illustrates the relation of MyUser Model with different Models related to user details. The Model MyUser is connected to Models Gender, College, City,
State, Country and User. The Model Country is connected to Model State and Model State is connected to Model City.

The ER Diagram shown in Figure 3 describes the relation of MyUser Model with different Models related to book and note
details. The Model MyUser can add BookDet to the database. MyUser can request for book which will be recorded in
BookReq Model. MyUser can add Notes to NotesDet Model. Book will be of a Category and corresponding to it there will be Subcategory I and Subcategory II. Notes also have Category, Subcategory I and Subcategory II. And, the last one as shown in Figure 4, illustrates the relation of MyUser Model with different Models related to chats, notification and orders. The Model MyUser can chat with another MyUser in Chatroom. ChatW will create a ChatRoom for MyUser if is not created yet. All notifications that notifies MyUser is in Notification. MyUser can add products to ShoppingCart in the Session logged in. MyUser can order an Order which will be delivered by another instance of MyUser.


ChatW
Message
ChatID
CreateDateTime
ChatsWith
(1,N)
(0,M)
ChatOf
(1,1)
(1,1)
ChatRoom
ChatRoomId
ReceiveMessDel
SenderMessDel
(1,1)
ChatTo
(0,1)
ChatIn
(1,1)
(0,1)
MyUser
UserID
Notifies
(0,N)
(1,1)
Notification
Message
NotID
NotLink
ReadStatus
NotDateTime
LoggedIn
(1,N)
(1,1)
Session
SessionID
OrderDateTime
IsOf
(0,M)
(1,1)
ShopCart
Contains
(0,M)
(1,N)
BookDet
BookID
CartID
TotalCost
Quantity
(1,N)
DeliveredBy
(1,1)
(0,N)
OrderedBy
(1,1)
Order
DeliveredOn
OrderID
OrderStatus
OrderDateTime
CartTotal

IMPLEMENTATION

Technologies Used Various front-end and back-end technologies are available in this era of digitalization. The technologies used in this project are discussed briefly in the following sections.

1) Front End Technologies

a) HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display (Musciano & Kennedy,1996). Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

b) CSS

CSS (Powell, 2010) stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on the screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web

pages all at once.

C) JavaScript/JQuery

JavaScript (JS) is a high level, interpreted programming language. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and firstclass functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web (Flanagan, 2006). JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it. JavaScript supports event-driven, functional and imperative (including object-oriented and prototype-based) styles. It is important to validate the form submitted by the user because it can have inappropriate values. So, validation is must to authenticate the user. JavaScript provides the facility to validate the form on the client-side so data processing will be faster than server-side validation.

c) BootStrap Bootstrap (Shenoy & Sossou, 2014) is a free and opensource CSS framework directed at responsive, mobilefirst front-end web development. It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. To use bootstrap, we are required to either install in our system or use CDN. CDN is short for content delivery network. A CDN is a system of distributes servers that deliver pages and other web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.

3) Back End Technologies

a) Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991 (Kuhlman, 2011), Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects. In this website, python is used as backend language to code database part and all functionalities that the website can perform. The version of Python used in this development is Python 3.6.

b) Django

Django (Holovaty & Kaplan-Moss, 2008) is a highlevel Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so we can focus on writing our app without needing to reinvent the wheel. It's free and open source. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code; low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin

models. The version of Django used during development is Django 2.1.5.

c) SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. By default, Django used SQLite3 as its default database. Django provides a specific way to define our database using the python programming language.

d) Jinja2

Jinja2 (Lokhande et al, 2015) is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools.

CODING-PART

```
                                                # Aditya Raj

from django.shortcuts import render
from BRMapp.forms import NewBookForm,SearchForm
from BRMapp import models
from django.http import HttpResponse,HttpResponseRedirect
from django.contrib.auth import authenticate,login,logout
from django.contrib.auth.decorators import login_required
```

```
# def userLogin(request):
#     data={}
#     if request.method=="POST":
#         username=request.POST['username']
#         password=request.POST['password']
#         user=authenticate(request,username=username,password=password)
#         if user:
#             login(request,user)
#             return HttpResponseRedirect('/BRMapp/view-books/')
#         else:
#             data['error']="Username or Password is incorrect"
#             res=render(request,"BRMapp/user_login.html",data)
#             return res
#     else:
#         return render(request,'BRMapp/user_login.html',data)
# def userLogout(request):
#     logout(request)
#     return HttpResponseRedirect('BRMapp/login')
# @login_required(login_url='/BRMapp/login/')
def searchBook(request):
    form=SearchForm()
    res=render(request,'BRMapp/search_book.html',{'form':form})
```

```
    return res
def search(request):
    form=SearchForm(request.POST)
    books=models.Book.objects.filter(title=form.data['title'])
    res=render(request,'BRMapp/search_book.html',{'form':form,'books':books})
    return res

                                        # ADITYA RAJ AND LAV PATEL

def deleteBook(request):
    bookid=request.GET['bookid']
    book=models.Book.objects.filter(id=bookid)
    book.delete()
    return HttpResponseRedirect('BRMapp/view-books')
def editBook(request):
    book=models.Book.objects.get(id=request.GET['bookid'])
    fields={'title':book.title,'price':book.price,'author':book.author,'publisher':book.publisher}
    form=NewBookForm(initial=fields)
    res=render(request,'BRMapp/edit_book.html',{'form':form,'book':book})
    return res
def edit(request):
    if request.method=='POST':
     form=NewBookForm(request.POST)
     book=models.Book()
     book.id=request.POST['bookid']
     book.title=form.data['title']
     book.price=form.data['price']
     book.author=form.data['author']
     book.publisher=form.data['publisher']
     book.save()
    return HttpResponseRedirect('BRMapp/view-books')
# @login_required(login_url='/BRMapp/login/')
def viewBooks(request):
    books=models.Book.objects.all()
    res=render(request,'BRMapp/view_book.html',{'books':books})
    return res
# @login_required(login_url='/BRMapp/login/')
def newBook(request):
    form=NewBookForm()
    res=render(request,'BRMapp/new_book.html',{'form':form})
    return res
def add(request):
    if request.method=='POST':
        form=NewBookForm(request.POST)
        book=models.Book()
        book.title=form.data['title']
        book.price=form.data['price']
        book.author=form.data['author']
        book.publisher=form.data['publisher']
        book.save()
    s="Record Stored<br><a href='/BRMapp/view-books'>View all Books</a>"
    return HttpResponse(s)
```

```python
from django import forms

class NewBookForm(forms.Form):
    title=forms.CharField(label='Title',max_length=100)
    price=forms.FloatField(label='Price')
    author=forms.CharField(label='Author',max_length=100)
    publisher=forms.CharField(label='Publisher')
```

```python
class SearchForm(forms.Form):
    title=forms.CharField(label='Title',max_length=100)
```

```python
class admin(forms.Form):
    username=forms.CharField(label='username',max_length=100)
```

```python
from django.db import models
from django.contrib.auth.models import User
# Create your models here.
class Book(models.Model):
    title=models.CharField(max_length=100)
    price=models.FloatField()
    author=models.CharField(max_length=100)
    publisher=models.CharField(max_length=100)

class BRMuser(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    nickname=models.CharField(max_length=20,null=False)
```

```python
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'firstproject.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

```python
if __name__ == '__main__':
    main()
```

```python
from django.shortcuts import render
from .models import Product
from math import ceil

# Create your views here.
from django.http import HttpResponse
```

```
def index(request):
    # products = Product.objects.all()
    # print(products)
    # n = len(products)
    # nSlides = n//4 + ceil((n/4)-(n//4))
```

```
    allProds = []
    catprods = Product.objects.values('category', 'id')
    cats = {item['category'] for item in catprods}
    for cat in cats:
        prod = Product.objects.filter(category=cat)
        n = len(prod)
        nSlides = n // 4 + ceil((n / 4) - (n // 4))
        allProds.append([prod, range(1, nSlides), nSlides])
```

```
    # params = {'no_of_slides':nSlides, 'range': range(1,nSlides),'product': products}
    # allProds = [[products, range(1, nSlides), nSlides],
    #             [products, range(1, nSlides), nSlides]]
    params = {'allProds':allProds}
    return render(request, 'shop/index.html', params)
```

```
def about(request):
    return render(request, 'shop/about.html')
```

```
def contact(request):
    return render(request, 'shop/contact.html')
```

```
def tracker(request):
    return render(request, 'shop/tracker.html')
```

```
def search(request):
    return render(request, 'shop/search.html')
```

```
def productView(request):
    return render(request, 'shop/prodView.html')
```

```
def checkout(request):
    return render(request, 'shop/checkout.html')
```

```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Edit Book Record</title>
    <link rel='stylesheet' href="{% static 'css/BRMstyle.css' %}"/>
  </head>
  <body>
    <div id='header'>
      <h1>Book Record Management</h1>
      <div id='nav'>
        <a href='/BRMapp/view-books'>View All</a>
        <a href='/BRMapp/new-book'>Add New</a>
        <a href='/BRMapp/search-book'>Search</a>
      </div>
    </div>
    <h2 id="edit"> Edit Book Record</h2>
    <div id='main'>
      <form action="/BRMapp/edit" method='post'>
        {% csrf_token %}
        <input type="hidden" name="bookid" value="{{book.id}}">
        {{form}}
        <input type='submit' value='Save'>
      </form>
    </div>
  </body>
```

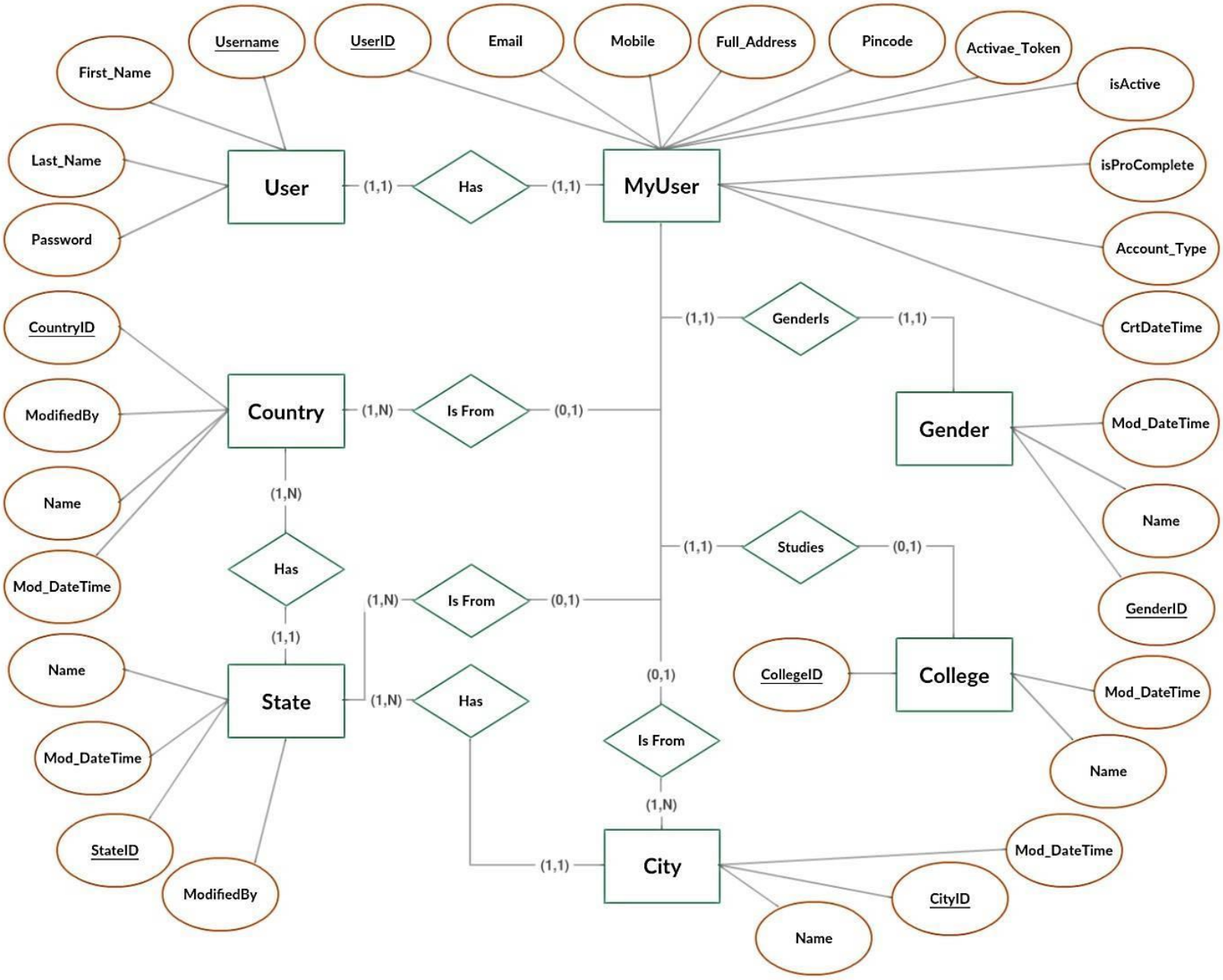
```
</html>
```

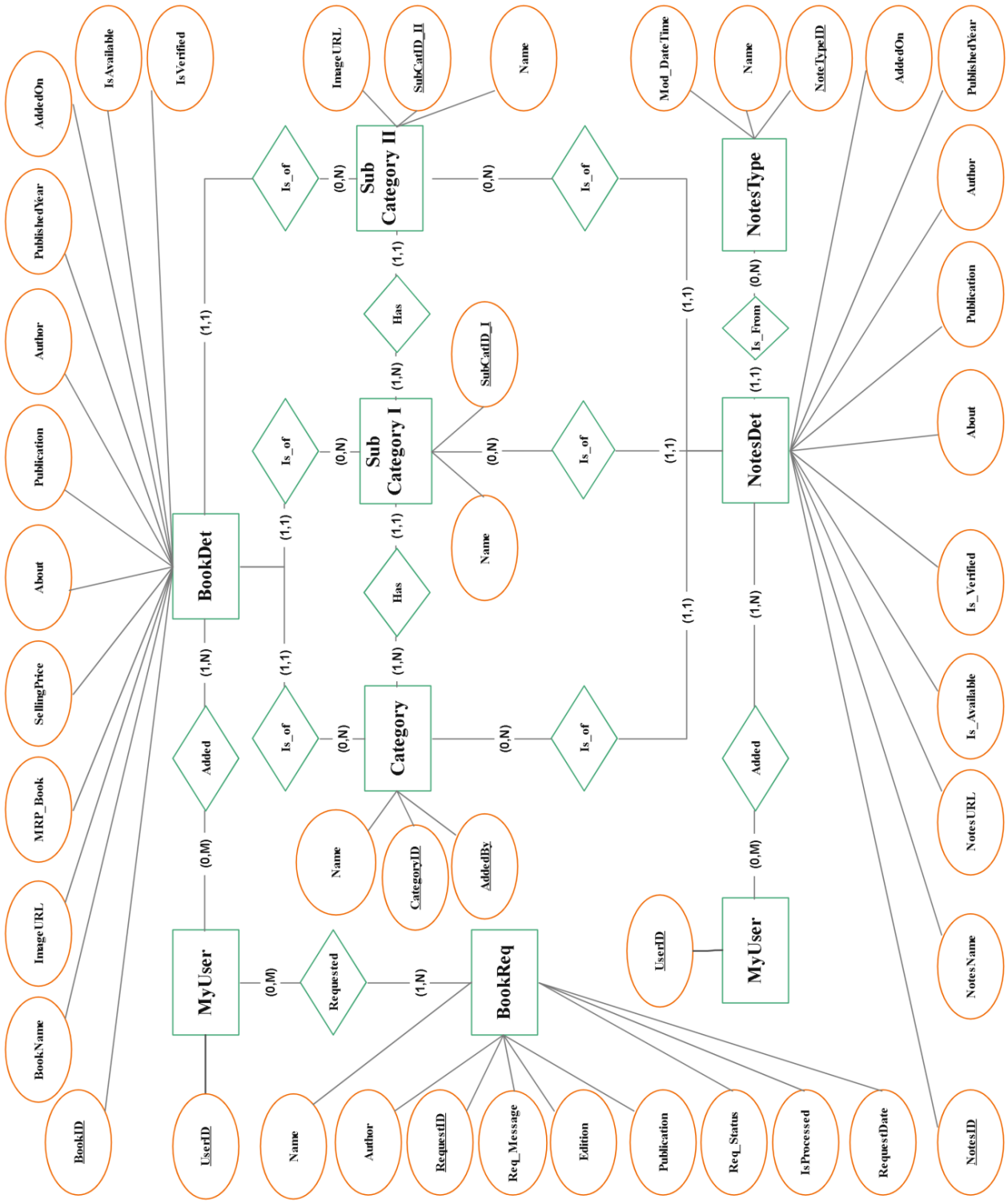
```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>AddBook</title>
    <link rel='stylesheet' href="{% static 'css/BRMstyle.css' %}"/>
  </head>
  <body>
    <div id='header'>
      <h1 style="letter-spacing: 4px">Famous Books And Authors List</h1>
      <div id='nav'>
        <a href='/BRMapp/view-books'>View All</a>
        <a href='/BRMapp/new-book'>Add New</a>
        <a href='/BRMapp/search-book'>Search</a>
      </div>
    </div>
    <h2 id="new"> Add New Book Record</h2>
    <div id='main'>
      <form action="/BRMapp/add" method='post'>
        {% csrf_token %}
        {{form}}
        <input type='submit' value='Add'>
      </form>
    </div>
  </body>
</html>
```

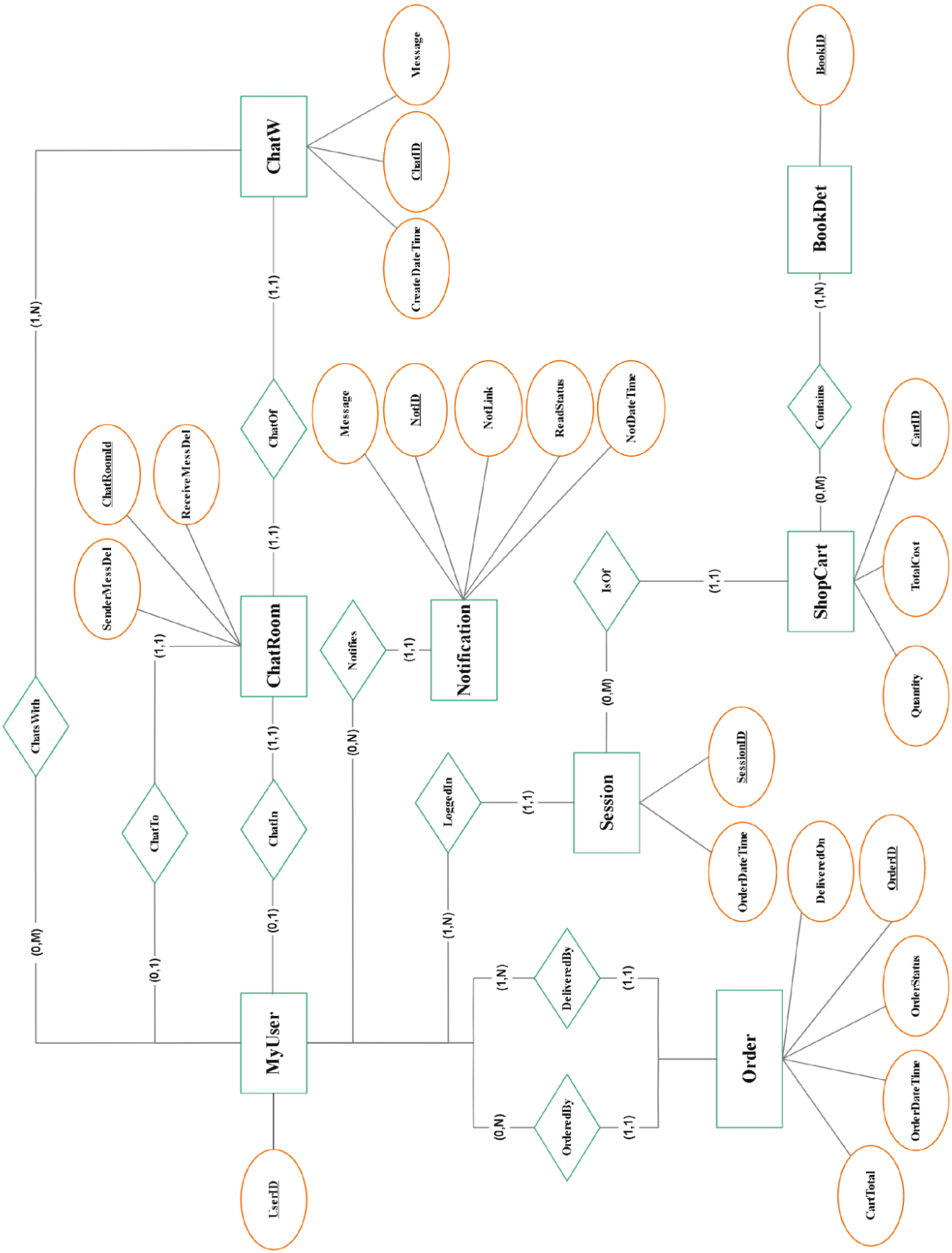
```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Search Book</title>
    <link rel='stylesheet' href="{% static 'css/BRMstyle.css' %}"/>
  </head>
  <body>
    <div id='header'>
      <h1 style="letter-spacing: 4px">Famous Books And Authors List</h1>
      <div id='nav'>
        <a href='/BRMapp/view-books'>View All</a>
        <a href='/BRMapp/new-book'>Add New</a>
        <a href='/BRMapp/search-book'>Search</a>
      </div>
    </div>
    <h2 id="search"> Search Book Record</h2>
    <div id='main'>
      <form action="/BRMapp/search" method='post'>
        {% csrf_token %}
        {{form}}
        <input type='submit' value='Search'>
      </form>
      {% if books %}
       <table>
         <tr>
           <th>'Book Title'</th>
           <th>'Price'</th>
           <th>'Author'</th>
           <th>'Publisher'</th>
         </tr>
      {% endif %}
      {% for i in books %}
      <tr>
          <td>{{i.title}}</td>
          <td>{{i.price}}</td>
```

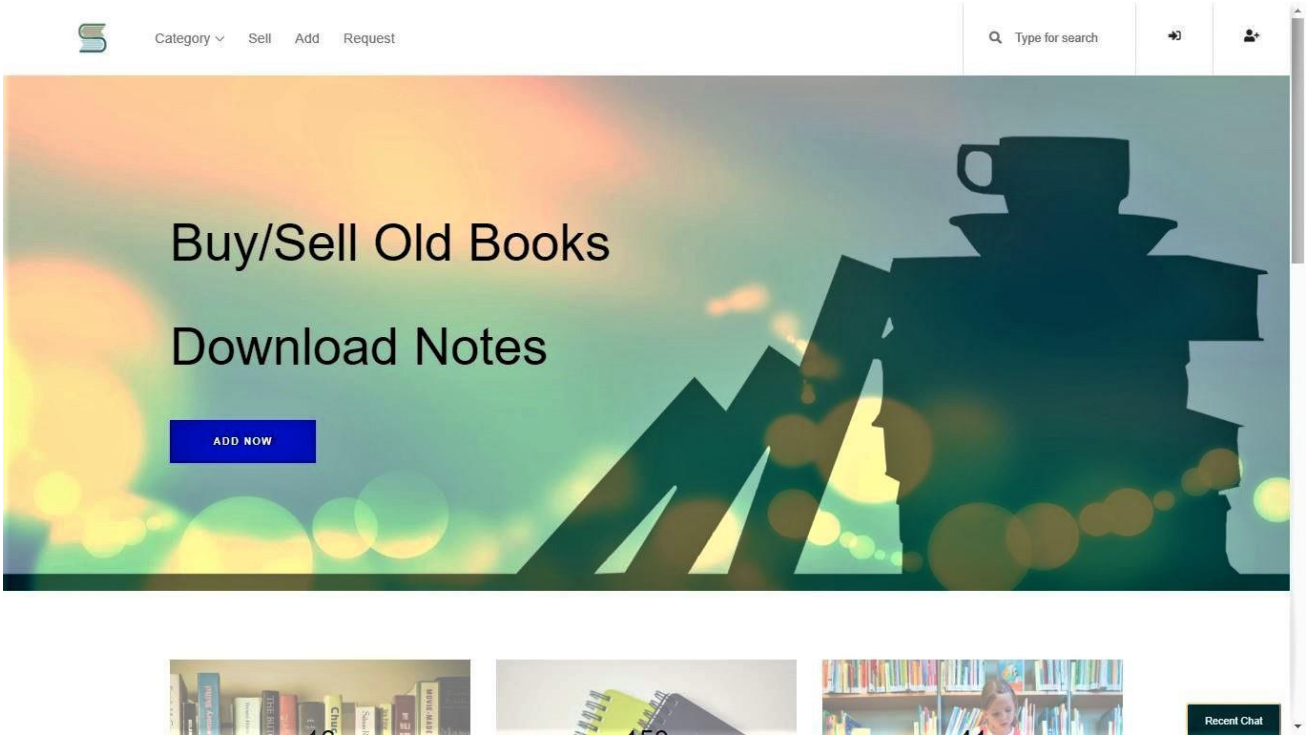
```
          <td>{{i.author}}</td>
          <td>{{i.publisher}}</td>
      </tr>
      {% endfor %}
      </table>
    </div>
  </body>
</html>
```

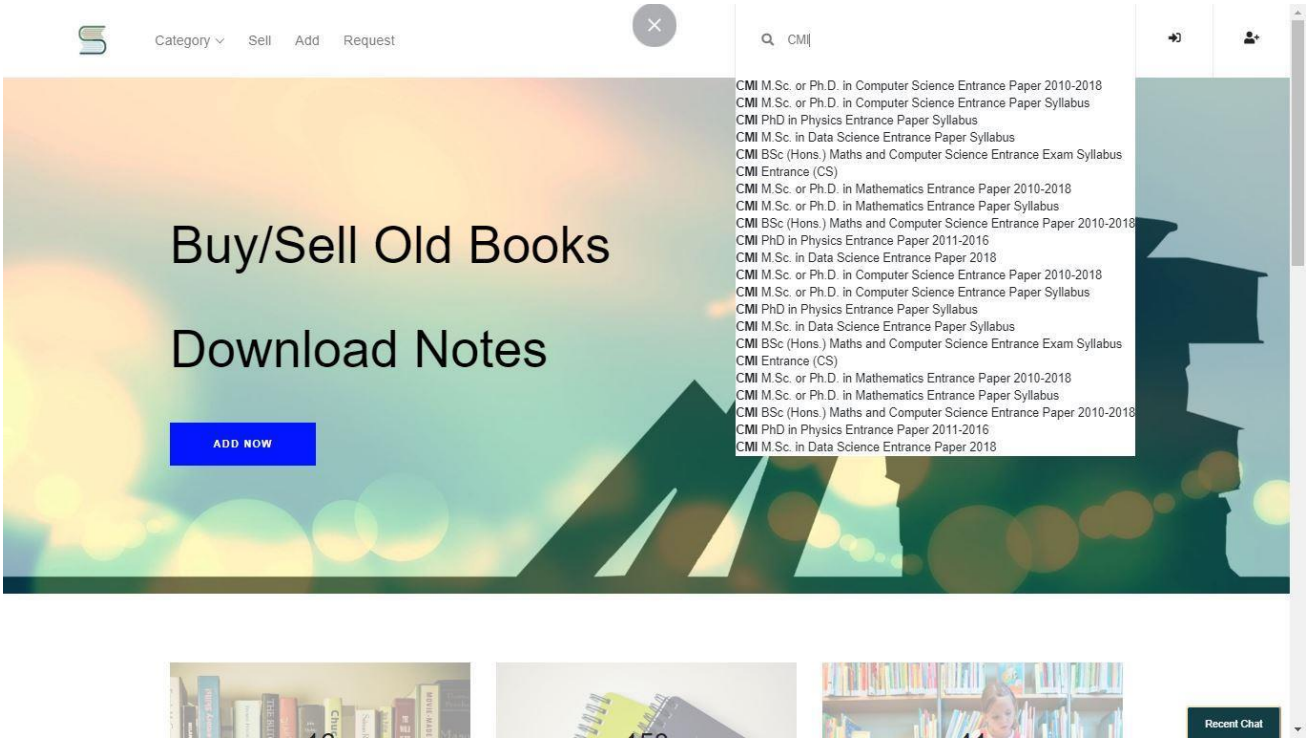
```
<!DOCTYPE html>
{% load static %}
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Login BRMapp</title>
    <link rel='stylesheet' href="{% static 'css/BRMstyle.css' %}"/>
  </head>
  <body>
    <div id='header'>
      <h1>Book Record Management</h1>
      <div id='nav'>
        <a href='/BRMapp/view-books'>View All</a>
        <a href='/BRMapp/new-book'>Add New</a>
        <a href='/BRMapp/search-book'>Search</a>
      </div>
    </div>
    <div id="main">
      <form method="post">
        {% csrf_token %}
        <table>
          <tr>
            <td><label for="username">Username</label></td>
            <td><input type="text" name="username" id="username"></td>
          </tr>
          <tr>
            <td><label for="password">Password</label></td>
            <td><input type="password" name="password" id="password"></td></tr>
          <tr>
             <td></td>
             <td><input type="submit" value="Login"></td>
          </tr>
        </table>
      </form><br>
      <p>{{error}}</p>
  </body>
</html>
```


Username
UserID
Email
Mobile
Full_Address
Pincode
Activae_Token
First_Name
isActive
Last_Name
User
(1,1)
Has
(1,1)
MyUser
isProComplete
Password
Account_Type
CrtDateTime
(1,1)
GenderIs
(1,1)
CountryID
Country
(1,N)
Is From
(0,1)
Gender
Mod_DateTime
ModifiedBy
Name
(1,N)
(1,1)
Studies
(0,1)
Name
Has
Mod_DateTime
(1,N)
Is From
(0,1)
GenderID
(1,1)
Name
State
(1,N)
Has
(0,1)
CollegeID
College
Mod_DateTime
Is From
Name
Mod_DateTime
(1,N)
StateID
(1,1)
City
Mod_DateTime
ModifiedBy
CityID
Name


BookDet
MyUser
Added
Is_of
Category
Sub Category I
Sub Category II
Has
Requested
BookReq
NotesDet
NotesType
Is_From
Name
CategoryID
AddedBy
SubCatID_I
SubCatID_II
ImageURL
BookID
BookName
Author
Publication
PublishedYear
MRP_Book
SellingPrice
About
AddedOn
IsAvailable
IsVerified
UserID
RequestID
Req_Message
Edition
Req_Status
IsProcessed
RequestDate
Mod_DateTime
NoteTypeID
Is_Verified
Is_Available
NotesURL
NotesName
NotesID
(1,1)
(1,N)
(0,M)
(0,N)


ChatW
Message
ChatID
CreateDateTime
ChatOf
(1,1)
(1,1)
ChatRoom
ChatRoomId
ReceiveMessDel
SenderMessDel
(1,1)
(1,1)
ChatIn
(0,1)
MyUser
UserID
ChatsWith
(1,N)
(0,M)
ChatTo
(0,1)
Notifies
(0,N)
Notification
Message
NotID
NotLink
ReadStatus
NotDateTime
(1,1)
LoggedIn
(1,N)
Session
SessionID
OrderDateTime
(1,1)
IsOf
(0,M)
(1,1)
ShopCart
CartID
TotalCost
Quantity
(0,M)
Contains
(1,N)
BookDet
BookID
(1,N)
DeliveredBy
(1,1)
(0,N)
OrderedBy
(1,1)
Order
DeliveredOn
OrderID
OrderStatus
OrderDateTime
CartTotal


Category
Sell
Add
Request
Type for search
Buy/Sell Old Books
Download Notes
ADD NOW
Recent Chat


Category
Sell
Add
Request
CMI
CMI M.Sc. or Ph.D. in Computer Science Entrance Paper 2010-2018
CMI M.Sc. or Ph.D. in Computer Science Entrance Paper Syllabus
CMI PhD in Physics Entrance Paper Syllabus
CMI M.Sc. in Data Science Entrance Paper Syllabus
CMI BSc (Hons.) Maths and Computer Science Entrance Exam Syllabus
CMI Entrance (CS)
CMI M.Sc. or Ph.D. in Mathematics Entrance Paper 2010-2018
CMI M.Sc. or Ph.D. in Mathematics Entrance Paper Syllabus
CMI BSc (Hons.) Maths and Computer Science Entrance Paper 2010-2018
CMI PhD in Physics Entrance Paper 2011-2016
CMI M.Sc. in Data Science Entrance Paper 2018
CMI M.Sc. or Ph.D. in Computer Science Entrance Paper 2010-2018
CMI M.Sc. or Ph.D. in Computer Science Entrance Paper Syllabus
CMI PhD in Physics Entrance Paper Syllabus
CMI M.Sc. in Data Science Entrance Paper Syllabus
CMI BSc (Hons.) Maths and Computer Science Entrance Exam Syllabus
CMI Entrance (CS)
CMI M.Sc. or Ph.D. in Mathematics Entrance Paper 2010-2018
CMI M.Sc. or Ph.D. in Mathematics Entrance Paper Syllabus
CMI BSc (Hons.) Maths and Computer Science Entrance Paper 2010-2018
CMI PhD in Physics Entrance Paper 2011-2016
CMI M.Sc. in Data Science Entrance Paper 2018
Buy/Sell Old Books
Download Notes
ADD NOW
Recent Chat

Working

The Project is developed via multiple steps. The major steps are enlisted here:

1. Installing Python and adding it to the windows path.

2. Creation of Virtual Environment (Following commands are written in Command Prompt)

- pip install virtualenvwrapper-win

- mkvirtualenvenvironmentname (any name can be given)

- Workonenvironmentname

3. Installing Django
- pip install Django

4. Go to Destination Place where you want the project to be kept, using cd command.

5. Create Project as follows

- django-admin startproject someprojectname

- cd someprojectname

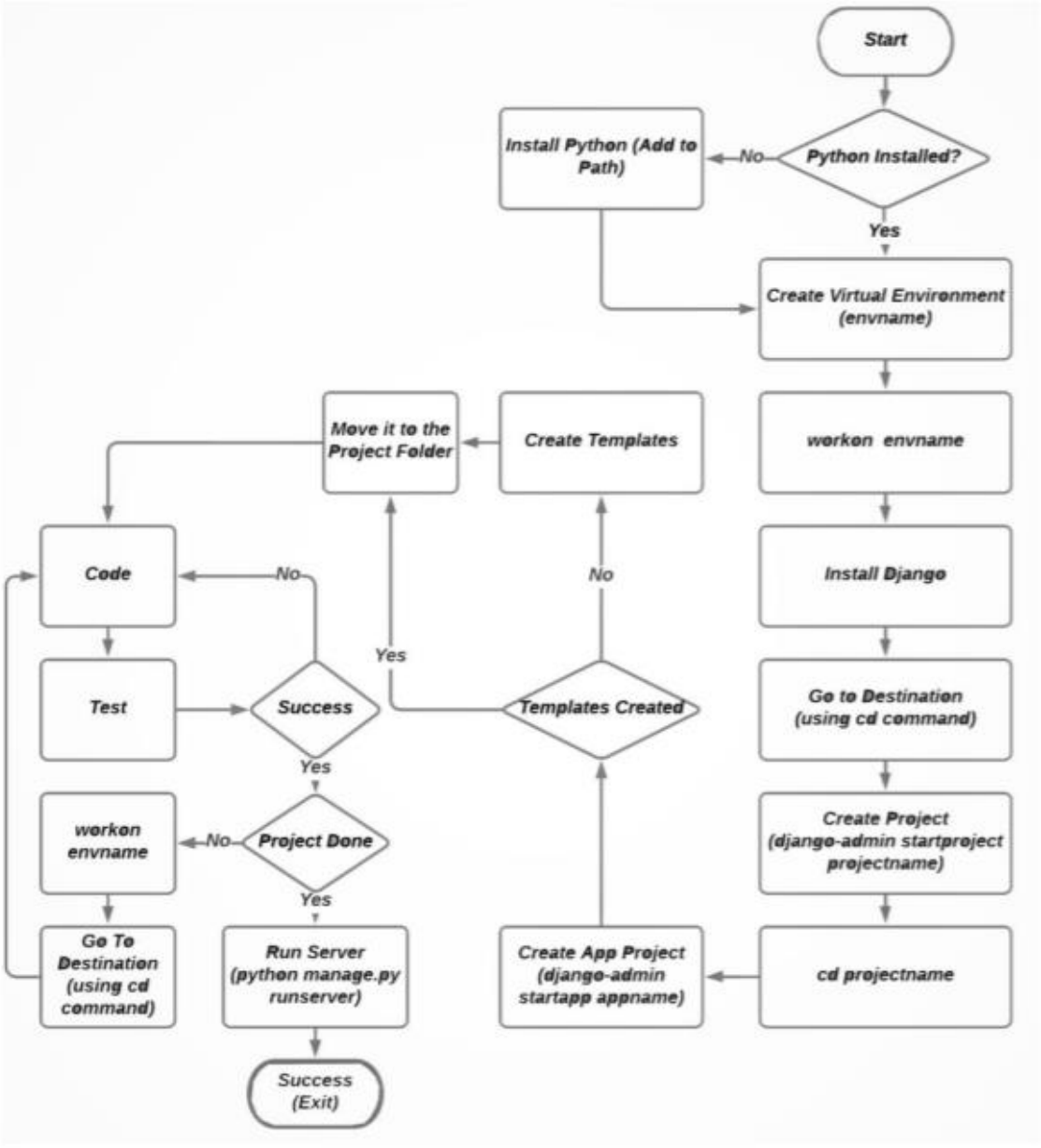
6. Create App of the project as

- django-admin startapp appname
- python manage.py make migrations
- python manage.py migrate

7 Run Server (localhost:8000)

- python manage.py run server

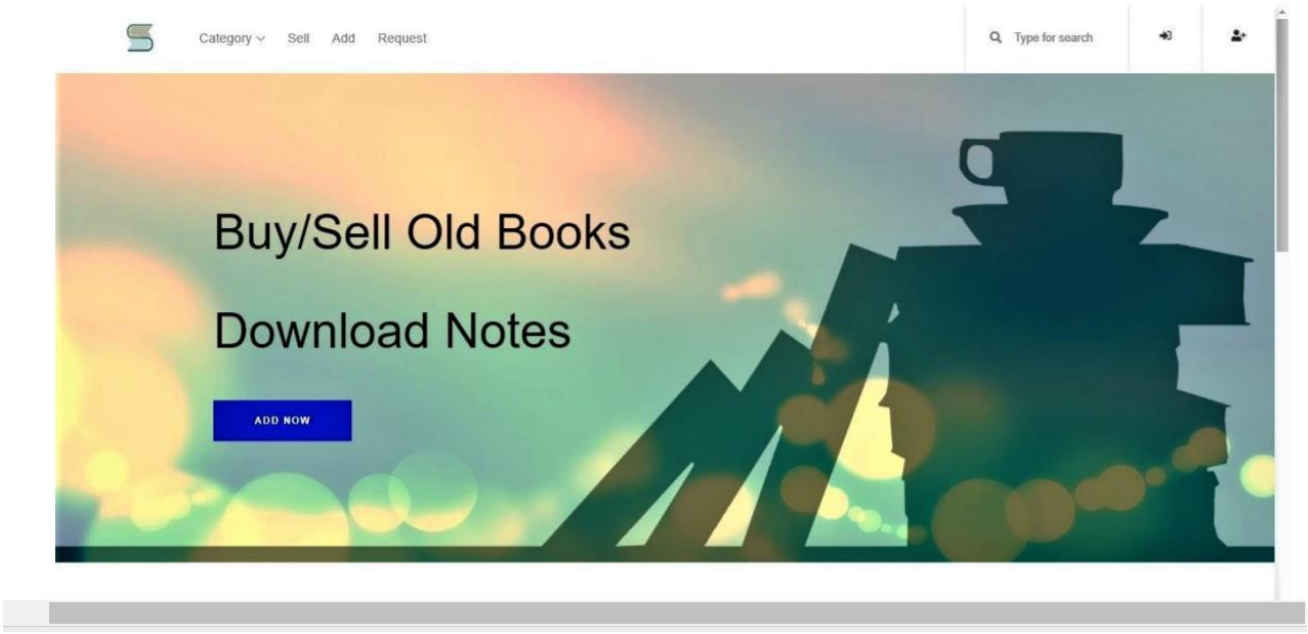
Modules Description via Diagram



Flow chart of the methodology

RESULT

Few snaps of the website are shown here showing the main functionalities of the project. The Home Page (Figure 6) has the top bar consisting of the website logo, the drop-down menu of categories, sell books button, add notes button, request an educational resource button, a dynamic search bar and the signup/login option. Below the top bar is the banner representing the main motive of the website. After that there is stats of the total number of books, notes and users of the website. On the bottom right corner is the recent chat button which shows the recent chats the user had done.



CONCLUSION

After analyzing the results obtained, the project developed can be considered satisfiable. It can be concluded that the website will be very helpful to students in their educational life as it provides all educational resources required in a college or school life. As the project works as an Educational cum E-Commerce Website and thus students can donate or sell their old books too.

To conclude, the project is developed using the proper Software Engineering process, following the Iterative Model of SDLC. A Project Control List was created after doing the feasibility study for functionalities as well as non-functional requirements. Then proper schema and tables that were supposed to be required in the development process were made and relationships between each table were drawn. For this ER Diagram was made which has been illustrated in the paper. Also, the flow chart was created so that each process can be done sequentially. After that, each task from the project control list was coded, tested using White Box Testing and implemented separately as per the Iterative Model. At last every unit was integrated and users were selected for Black Box Testing. Each user was asked to

run the project and test each functionality of the project. After the testing, feedback and suggestions were recorded and accordingly the amendments were made. Security issues were resolved with the help of CSRF tags given by the Django Framework and by deploying the Web Application behind HTTPS. The approach used in the System Development Model can act as a road-map for the development of similar kinds of Web Applications efficiently.

REFERENCES

Sachan, N. (2019, February 20). Welcome to BHU Student Club, BHU Student Club. Retrieved from *http://bhustudentclub.in/*.

Jalote, P. (2003). *An Integrated Approach towards Software Engineering*. Narosa Publishing House.

Musciano, C., & Kennedy, B. (1996). *HTML, The Definitive Guide*. O'Reilly & Associates.

Powell, T. A. (2010). *HTML & CSS: The Complete Reference*. The McGraw-Hill Companies.

Flanagan, D. (2006). *JavaScript: The Definitive Guide*. O'Reilly Media, Inc.

Shenoy, A., & Sossou, U. (2014). *Learning Bootstrap*. Packt Publishing Ltd.

Kuhlman, D. (2011). *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. Platypus Global Media.

Holovaty, A., & Kaplan-Moss, J. (2008). *The Definitive Guide to Django: Web Development done right*. Apress.

Lokhande, P. S., Aslam, F., Hawa, N., Munir, J., & Gulamgaus, M. (2015). Efficient way of Web Development using Python and Flask. *International Journal of Advanced Research in Computer Science*, 54-57.

Thakur, M. S. (2017). Review on Structural Software Testing Coverage Approaches. *International Journal of Advance Research, Ideas and Innovations in Technology*, 281-286.