

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

TUTOR'S CORNER

A

Project Work

Submitted as Major Project in Partial fulfilment for the award of Graduate Degree
in Bachelor of Technology in Computer Science & Engineering.

Submitted to

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

Submitted By

Ashis J	(20bcs022)
Ashwani Kumar	(20bcs023)
Hemang Ranga	(20bcs057)
Rakshita Y	(20bcs107)
Veekshith Reddy	(20bcs111)
Sunil Kumar Patidar	(20bcs131)
Yugal deep Singh	(20bcs140)

Under the Guidance of

Dr Shankar Biradar

Dr Uma Sheshadri

(Department of Computer Science & Engineering)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD



INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled “TUTOR’S CORNER” being submitted by students of IIIrd Semester, Degree in Computer Science & Engineering have done their work as MAJOR PROJECT for Partial fulfilment of the degree from IIT Dharwad (Karnataka) is a record of bonafide work carried out by them under our supervision.

Dr. Shankar Biradar
Guide
Department of
Computer Science & Engineering

Dr. Uma S
Head
Department of
Computer Science & Engineering

ACKNOWLEDGEMENT

I take the opportunity to express my cordial gratitude and a deep sense of indebtedness to my guide **Dr Shankar Biradar** for the valuable guidance and inspiration throughout the project duration. I feel thankful to him for his innovative ideas, which led to the successful submission of this major project work. I feel proud and fortunate to work under such an outstanding mentor in the field of Tutor's Corner. He has always welcomed my problem and helped me to clear my doubt. I will always be grateful to him for providing me with moral support and sufficient time.

At the same time, I would like to thank **Dr Uma S (HOD, CSE)** and all other faculty members and all non-teaching staff of the Department of Computer Science & Engineering for their valuable cooperation.

I would also thank my Institution, faculty members and mentor without whom this project would have been a distant reality. I also extend my heartfelt thanks to my family and well-wishers.

Ashis J	(20bcs022)
Ashwani Kumar	(20bcs023)
Hemang Ranga	(20bcs057)
Rakshita Y	(20bcs107)
Veekshith Reddy	(20bcs111)
Sunil Kumar Patidar	(20bcs131)
Yashu Mittal	(20bcs139)
Yugal deep Singh	(20bcs140)

APPROVAL CERTIFICATE

This is to certify that the project entitled Tutor's Corner being submitted by **Ashis J (20bcs022), Ashwani Kumar (20bcs023), Hemang Ranga (20bcs057), Rakshita Y (20bcs107), Veekshith Reddy (20bcs111), Sunil Kumar Patidar (20bcs131), Yashu Mittal (20bcs139), Yugal deep Singh (20bcs140)** students of Third Semester, Bachelor of Technology in Computer Science and Engineering have done their work as Major Project for Partial fulfilment of the degree from IIIT Dharwad, Dharwad (Karnataka).

Guide Name : Dr. Shankar Biradar

Date:

Signature:

ABSTRACT

Tutor's Corner is a web-based study material sharing and management system which helps students and teachers to share their study material online effectively. It reduces the wasting of time in manually distributing notes and books to each individual. It greatly overcomes the lack of availability and converts the manual old school paperwork to a fully automated and managed online system.

Tutor's Corner allows its users to securely register and log in to their accounts and create, read, update, delete notes, books and other study material according to their needs. It provides study material to everyone in a very secure manner. Multiple users can work in this system at the same time under centralized supervision by the administrator. It is a very useful study material management system for Colleges, Schools and other Institutes to manage and share their notes in a secure, efficient and effective manner.

TABLE OF CONTENTS

ABSTRACT

CHAPTER 1:- INTRODUCTION

- 1.1 Overview
- 1.2 Project Objective & Scope
- 1.3 Organization of Report

CHAPTER 2:- BACKGROUND AND LITERATURE SURVEY

- 2.1 Literature Review
- 2.2 Feasibility Report
 - 2.2.1 Innovativeness and Usefulness
 - 2.2.2 Market Potential and Competitive advantages
- 2.3 Proposed Project Model

CHAPTER 3:- PROCESS MODEL

- 3.1 Proposed Process Model
- 3.2 Project Plan
- 3.3 Project Scheduling
 - 3.3.1 GANTT CHART

CHAPTER-4 :- DESIGN

- 4.1 Use Case Diagram
- 4.2 Sequence Diagram
- 4.3 Activity Diagram
- 4.4 Class Diagram
- 4.5 ER Diagram
- 4.6 DFD Diagram
- 4.7 Agile Model

CHAPTER 5:- Software and Hardware Requirements

5.1 Software Specification

5.2 Hardware Specification

CHAPTER 6:- Implementation & Coding

CHAPTER 7:- Testing

CHAPTER 8:- Conclusion

CHAPTER 9:- References

Chapter-1

Introduction

1.1 .OVERVIEW

The website is user friendly and it doesn't require any training or extra effort to understand it properly. It provides several facilities with a single click of the mouse. This system is aimed to ease the work of a student pursuing engineering. The purpose of the website is to provide a common platform where needed study material can be provided and all the needed activities can be done.

Major features of the website include the following:

- It provides e-books of various subjects of all the semesters and all the technical topics which are needed to study.
- It provides campus papers of many premier companies.
- It provides a common forum for all students, teachers, mentors, etc. where they can discuss and give views on various topics and can also query their problems.

EXISTING SYSTEM

For the current system when someone needs a certain study material, he/she has to work hard to search for the material on the internet or even has to go to different shops in search of the material. She/he has to ask different sources for the study material.

Disadvantage

The major issue with the existing system is that it requires manual labour and a lot of time is wasted in the search and leads to total disappointment if the required data is not found. Additionally, sometimes you don't know where to search for the data.

1.2 . PROJECT OBJECTIVE & SCOPE

The problem faced by engineering students (especially of private colleges) is that there is no such organized portal where they can meet all the requirements, there is no such a common platform :

- Where they get all the course content and required material while pursuing engineering.
- Where they can get guidelines and papers related to campus interviews of various companies.

Where they are provided with the setup of all the software which may be required in the 4 years period of their engineering.

1.3 . ORGANIZATION OF REPORT

An organizational assessment is a process for obtaining accurate and concise information about the performance of a business and the factors that affect an organization's productivity. The report identifies areas of competence, room for improvement, and risks to modify decisions and support investment.

Chapter- 2

Background And Literature Survey

2.1 LITERATURE REVIEW

A literature review is a text which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic.

Currently, there are some similar applications and websites which provide somewhat similar services out of which some of them are listed below:

1. Scribd: It is a digital library, ebook and audiobook subscription service. Scribd began as a site to host and share documents. The founder was inspired to start Scribd after learning about the lengthy process required for publishing academic papers. Right now Scribd provides a 30days free subscription after that they start charging you money.

2. Evernote: Evernote is an app designed for note-taking. Users can create groups and can also share the study material among group members.

3. Glose: Glose is a social reading platform where you can read books and engage with other readers through shared annotations. But you need to buy books before you read them.

All the applications listed above provide only soft copies of the material and if you want hard copies of them, then again that will cost you money as well as you can use their services for free only for a short period of or up to a certain limit. The system that we are proposing is free for use.

2.2 FEASIBILITY REPORT

A feasibility report is a document that assesses potential solutions to the business problem or opportunity and determines which of these are viable for further analysis.

2.2.1 INNOVATIVENESS AND USEFULNESS:

It determines that work for the project is done with the present equipment and existing software technology, such as:

- Easy to access, doesn't require technical skills.
- It is supposed to perform all the activities needed by the user.
- It covers mainly two aspects. It determines how the proposed system will fit in the current operation and what if job retraining and restructuring may be needed at the end of the implementation of the system.
- It is operationally feasible also because it does not affect the other applications running on the system nor does it affect the network traffic in any way. Besides being a website it requires only limited resources as it is license-free so it does not affect the working of other software on the system.

2.2.2 MARKET POTENTIAL AND COMPETITIVE ADVANTAGES:

It looks at the financial aspects of the project. Economic feasibility is concerned with the returns of the investments in the project.

- It does not require any capital investment and is free to use.

Chapter - 3

Process Model

3.1 PROPOSED PROCESS MODEL

The project is designed to reduce the labour and the time wasted while searching for the study material in the existing system. This proposed system is a website that provides an interface for the person searching for particular study material. This system maintains a database of lists of users and users who have a particular kind and type of study material. The person just needs to login into the website and search for the keyword and the system will fetch the list of users and people who are having a particular kind of data.

- **Advantages of the Proposed System**

There are several advantages of the proposed system over the existing system. Firstly, it will directly provide the list of people who have a particular kind of data and are willing to lend it to the person in need. This system eliminates the manual searching process. It also helps to reduce the complexity and time wasted in the search. To start with some advantages of this system, first, it is a website so anyone having a phone can use this website. Secondly, it helps a lot to the person who is searching for a particular kind of data because he can search the whole community just by sitting on his comfortable chair or while laying down on his bed.

3.2 PROJECT PLAN

The V-model is a type of SDLC model where the process executes sequentially in V Shape. It is also known as the Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. The development of each step is directly associated with the testing phase. The next phase starts only after the completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

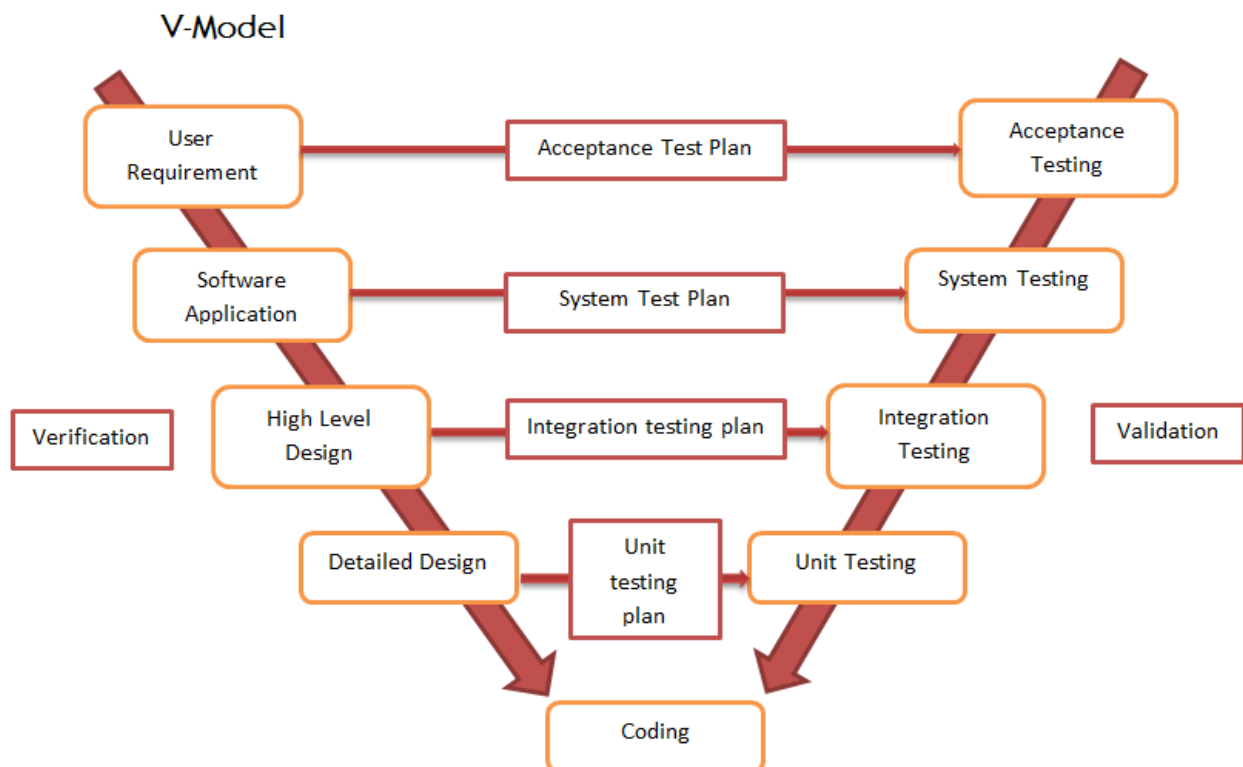


Fig.1: V-Model

So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phases in V-shape. Thus it is called V-Model.

3.3 PROJECT SCHEDULING

Project scheduling is a mechanism to communicate what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. A project schedule is a document collecting all the work needed to deliver the project on time.

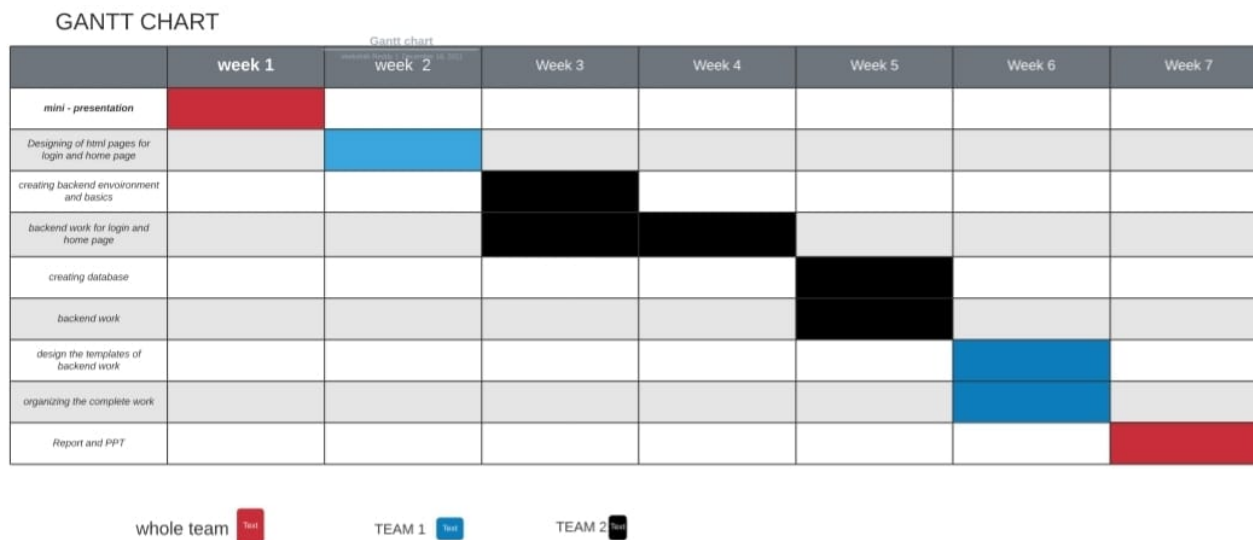


Fig. GANTT Chart

Chapter - 4

Design

The design module of the knowledge knots application can be well explained through tables, E-R Diagram, Relational Schema, which can be seen as shown below as follows:

- **ARCHITECTURAL DESIGN**

Architectural design values make up an important part of what influences architects and designers when they make their design decisions. For the project's architectural design refer to the diagram.

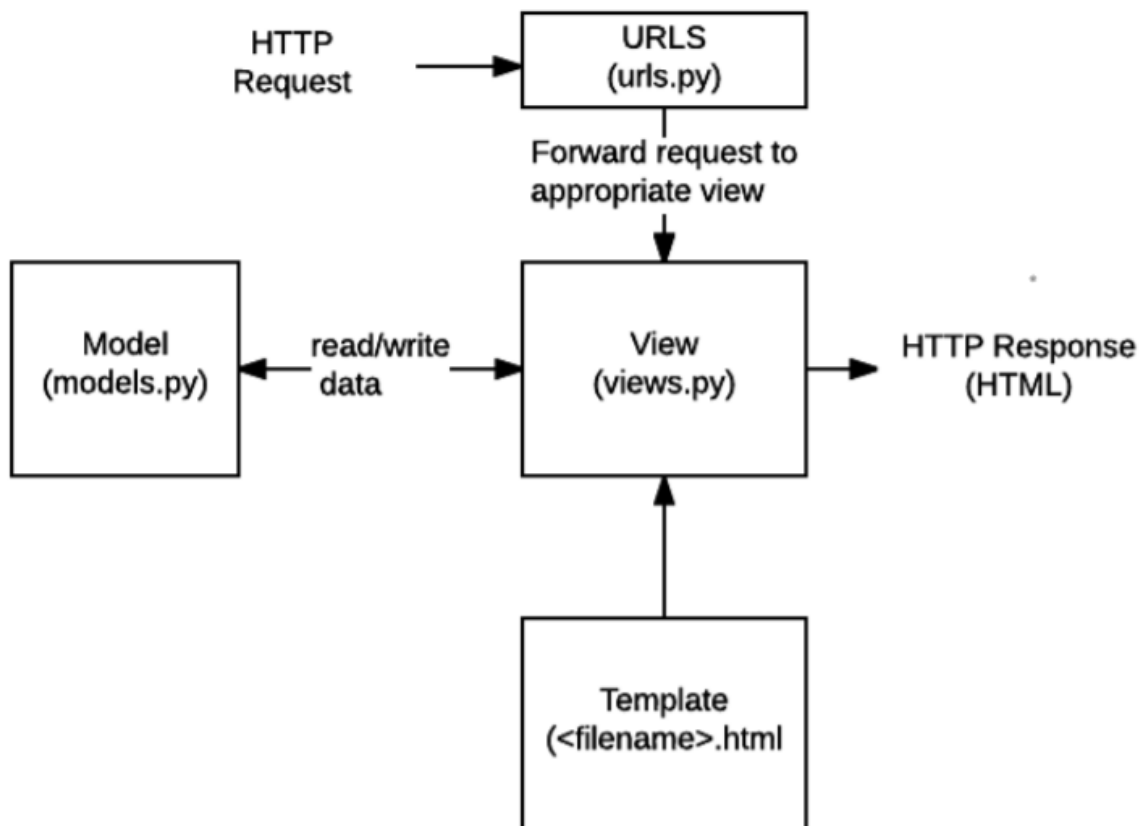


Fig: Architectural Design

❖ **PRESENTATION LAYER**

This is the first layer that defines the user interface and the presentation logic for the application in the following project we are using the ANDROID OS for the user interface and the presentation logic different components like ListView, Card View, Scroll Views, Web View, Text View, Buttons, Image Views, Edit Text are used.

❖ **BUSINESS LAYER**

As the name suggests, the layer focuses on the business front. In simple language it focuses on the way business will be presented in front of the end-users. Here we are using a website at the user end with which he/she interacts. Different Components of the business layer are

- **Components:** This defines the different components
- **Entities:** This defines all the entities used in the architectural design.

❖ **DATA LAYER**

At this third stage data, related factors are kept in mind. This includes Data access components, data helpers/utilities, and service agents.

❖ **COMMON LAYER**

This includes common configuration, security and communication agents. For user login security we are using the Firebase Authentication Services.

• **RELATIONAL SCHEMA**

Before you can define the database tables using a specific database management software and DDL (Data Definition Language), you must write a relation schema for each table. The set of relation schemas from the relational database schema is used as input when defining the database.

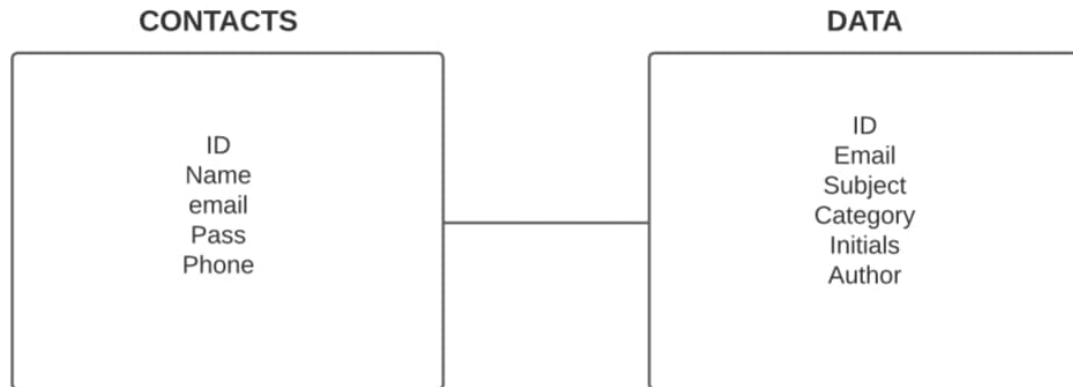


Fig. Relational Schema

4.1. USE CASE DIAGRAM

A Use Case Diagram is a type of behavioural diagram defined by and created from a Use-case analysis. It is used to identify the primary elements and processes that form the system. The primary elements are termed as "actors" and the processes are called "use cases." The Use case diagram shows which actors interact with each use case. The main purpose of a use case diagram is to show what system functions are performed for which actors.

The elements used for this representation are:

- **Actors**
- **Use Cases**
- **System Boundary**
- **Associations**

A use case can be related to another use case in three ways:

- **Include** - allows factoring common elements to two or more use cases
- **Generalization** - describes variation in normal behaviour
- **Extend** - similar to generalization but the variation is more formal as it is tied to the extension points

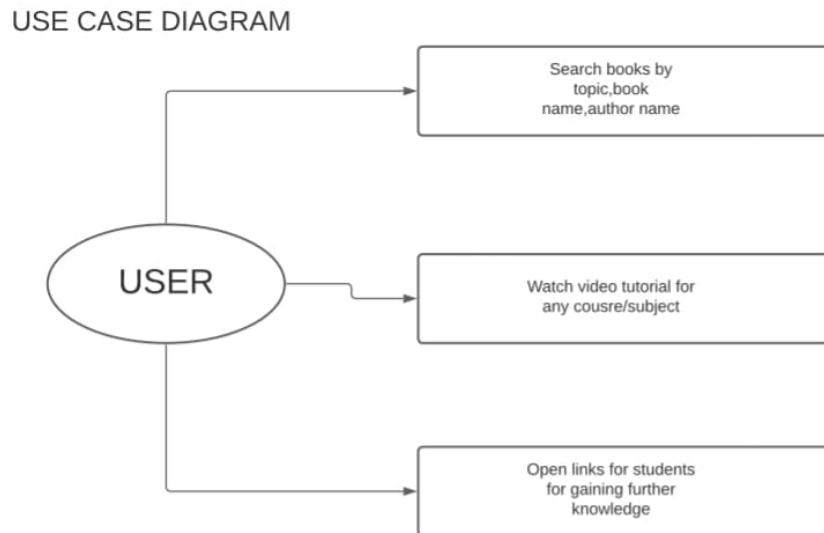


Fig: Use Case diagram

4.2 SEQUENCE DIAGRAM

It defines the interaction diagram that shows how processes operate with one another and in what order. It is a construction of a message sequence chart.

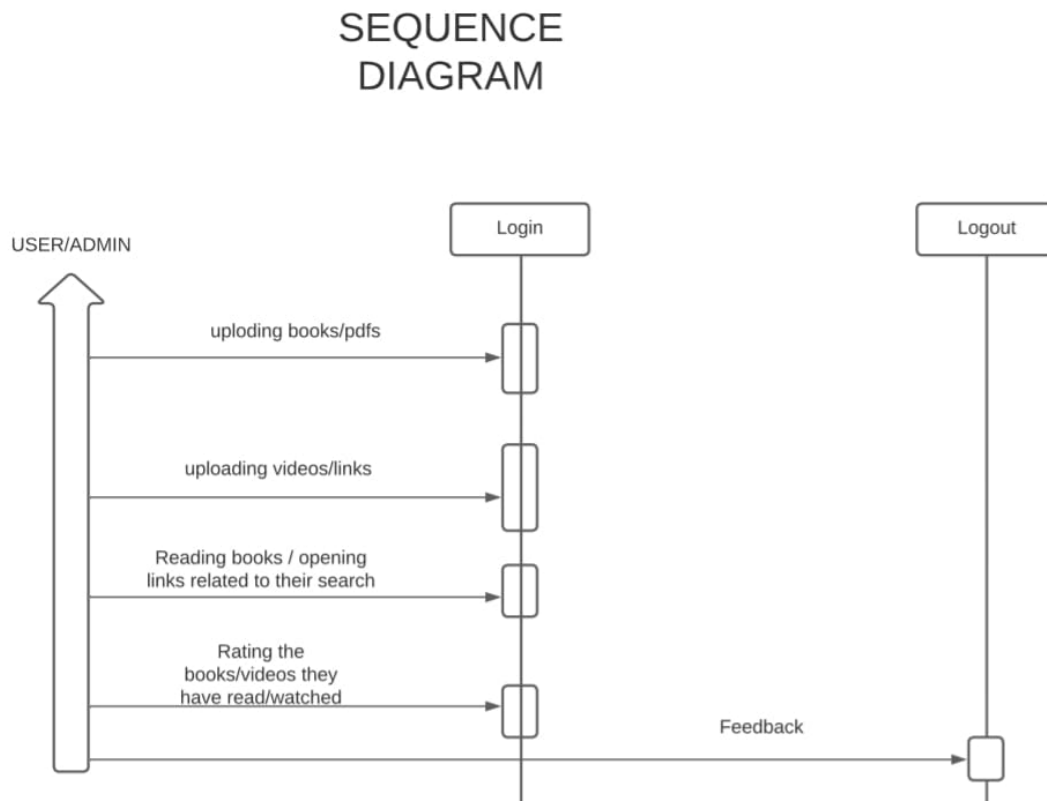


Fig: Sequence diagram

4.3 ACTIVITY DIAGRAM

An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine. Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in the state of the system or the return of a value.

Activity diagrams commonly contain:

- Activity states and action states
- Transitions
- Objects

ACTIVITY DIAGRAM

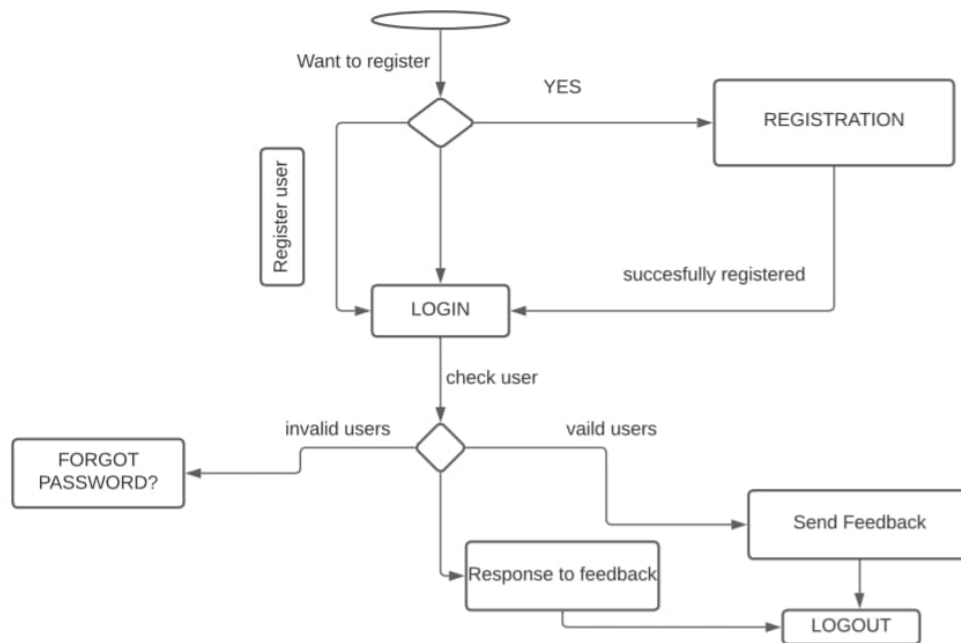


Fig : Activity diagram

4.4 CLASS DIAGRAM

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation.

class diagram

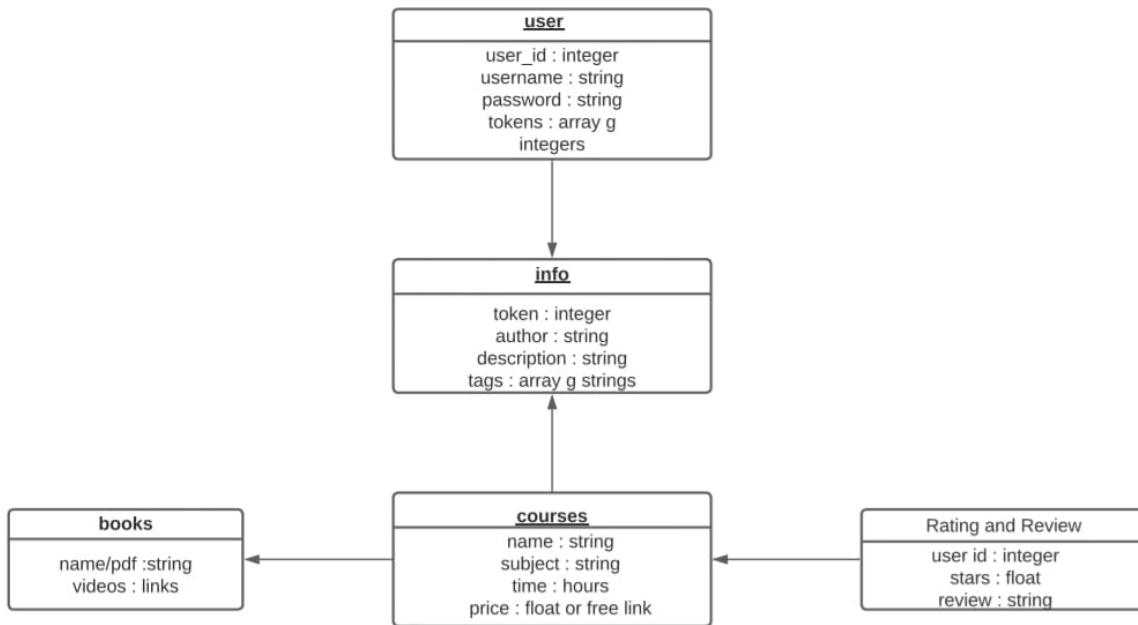


Fig: Class diagram

4.5 ER-DIAGRAM

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of the database. In the given fig and contacts module which are explained in detail as below:

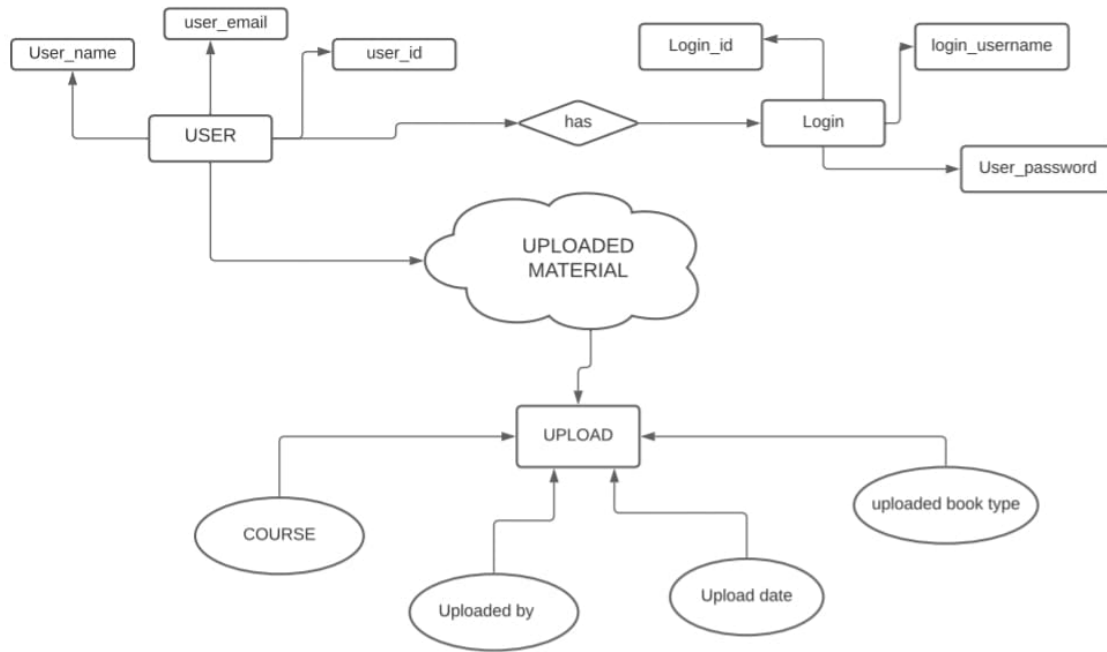


Fig: ER- Diagram

1. Contacts: It is a strong entity that is used to keep track of the user details that have been registered. The details mainly include the name, email, login password and phone number. The email is the primary key for the contacts table.

2. Data: It is a strong entity that comprises a list of users having a particular set of documents. Each event is uniquely identified by the id number. The details may include email, subject (in this case branch names), author name, initials of the document, category of the document (books, videos).

4.6 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

DATA FLOW DIAGRAM FOR USER

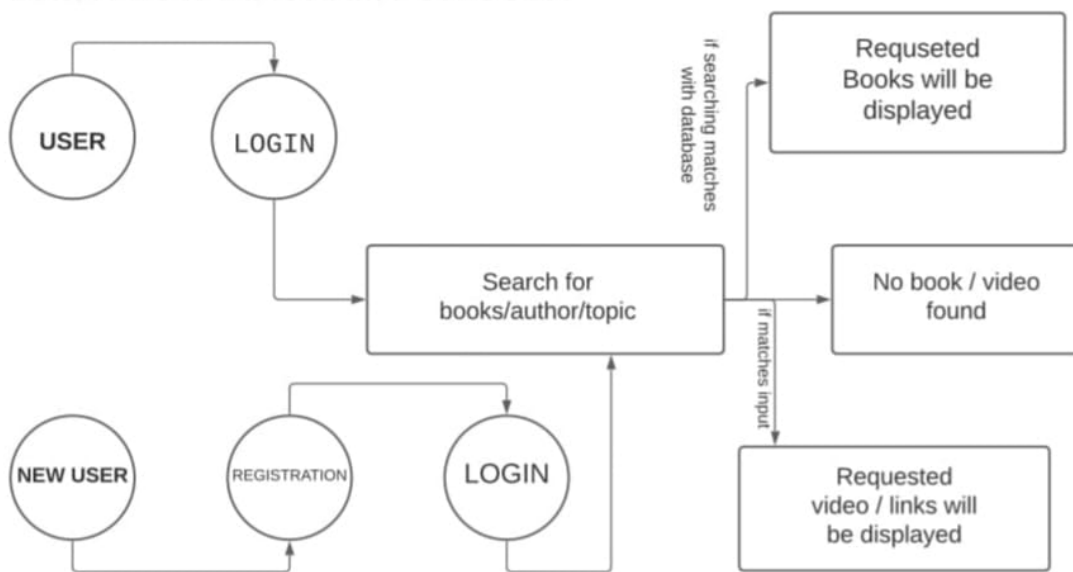


Fig: Data Flow Diagram

4.7 Agile model

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible responses to change. It is a conceptual framework that promotes foreseen tight iterations throughout the development cycle.



Fig: Agile Model

Chapter - 5

Software and Hardware Requirements

Hardware Requirement

- CORE 2 DUO OR HIGHER
- 20 GB Hard Disk,
- 1GB MB RAM or Above
- 10 Mbps Network Interface Card.
- Modem With Internet Connection

Software Requirement

For Development

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- Web Browser
- Text Editor (Vs code / Sublime Text)
- Postgresql
- Django
- psycopg2 python module (for communication between Django & PostgreSQL)

For Users

- Any Desktop / Phone with a good internet connection

Chapter - 6

Implementation And Coding

• SNIPPETS OF CODE

Urls.py:

```
urlpatterns =  
    path('register', views.register, name='register'),  
    path('login', views.login, name='login'),  
    path('logout', views.logout, name='logout'),  
    path('', views.home, name='home'),  
    path('addData/', views.addData, name='addData'),  
    path('disData/<int:id>', views.disData, name='disData'),  
    path('editData/<int:id>', views.editData, name='editData'),  
    path('deleteData/<int:id>', views.deleteData,  
        name='deleteData'),  
    path('dashboard/', views.dashboard, name='dashboard'),  
    path('addDatatoBase', views.addDatatoBase,  
        name='addDatatoBase'),  
    path('EditDatatoBase/<int:id>', views.EditDatatoBase,  
        name='EditDatatoBase'),
```

```
    path('search/', views.search, name='search')
```

```
]
```

Navbar:

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-9aIt2nRpC12Uk9gS9baD1411NQApFmC26EwAOH8WgZl5MYYYxFfc
+NcPb1dKGj7Sk" crossorigin="anonymous">
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand" href="/">TUTOR'S CORNER</a>
  <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarSupportedContent"
  aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse"
id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="/">Home <span
class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">About Us</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#" tabindex="-1"
aria-disabled="true">Contact</a>
      </li>
      {% if user.is_authenticated%}
      <li class="nav-item">
```

```

        <a class="nav-link" href="{% url 'dashboard' %}"
tabindex="-1" aria-disabled="true">Dashboard</a>
    </li>
    {%endif%}
    <li class="nav-item dropdown">
        {% if user.is_authenticated %}
            <a class="nav-link" href="{% url 'logout' %}"
tabindex="-1" aria-disabled="true">Logout</a>
        {%else%}
            <a class="nav-link" href="{% url 'login' %}"
tabindex="-1" aria-disabled="true">Login</a>
        {%endif%}
    </li>
</ul>
<form class="form-inline my-2 my-lg-0" action="search"
method="post">
    {% csrf_token %}
    <input class="form-control mr-sm-2" type="search"
placeholder="Search" name="search_key" aria-label="Search">
    <button class="btn btn-outline-success my-2 my-sm-0"
type="submit">Search</button>
</form>
</div>
</nav>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVY
Jew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.mi
n.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3Uks
dQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.
min.js"

```

```
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp  
75j7Bh/kR0JKI"  
crossorigin="anonymous"></script>
```

Base.html:

```
<!DOCTYPE html>  
<html lang="en">  
{%load static%}  
<head>  
    <meta charset="UTF-8">  
    <title>Tutor's Corner</title>  
    <link  
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;7  
00;900&display=swap" rel="stylesheet">  
    <link rel="stylesheet" href="{% static 'CSS/final1.css' %}">  
</head>  
  
<body>  
    <header>  
        <div class="wrapper">  
  
            {% include 'navbar.html' %}  
            {%block content%}  
            {%endblock%}  
  
        </div>  
  
    </header>  
  
</body>  
  
</html>
```

Home.html

```
{% extends 'base.html' %}
{% block content %}

    <div class="welcome-text">
```

```
        <h1>
            We are the <span>Tutors</span></h1>
        </div>
        <div class="Content-box">
            {% for pa in para %}
                <div class="dash_content" onclick="location.href='{% url
'disData' id=pa.id %}'">
                    <p>{{pa.title}}<br>
                    Subject: {{pa.subject}}</p>
                </div>
            {% endfor %}
        </div>
    {% endblock %}
```

View.py home:

```
def home(request):
    para=content.objects.order_by('?')[:3]
    return render(request, 'home.html', {'para' : para})
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
  {% load static %}

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Login Portal</title>
  <link

href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&family=Baloo+Bhaina+2&family=Bree+Serif&display=swap"
    rel="stylesheet">
  <link rel="stylesheet" href="{% static 'CSS/login.css' %}">
  <link rel="stylesheet" media="screen and (max-width: 600px)"
href="{% static 'CSS/login1.css' %}">
```

```

    <link rel="stylesheet" media="screen and (max-width: 300px)"
href="{% static 'CSS/login2.css' %}">
</head>

<body>
    <div id="LOGIN">
        <nav id="navbar">
            <div class="logo">
                
            </div>
            <ul style="margin-left: 30px;">
                <li class="item"><a href='/'>Home</a></li>
                <li class="item"><a href="#about">About US</a></li>
                <li class="item"><a href="#contact">Contact
Us</a></li>
            </ul>
        </nav>
        <div class="form-container">
            <div class="form-box">
                <div class="button-box">
                    <div id="btn"></div>
                    <button type="button" class="toogle-btn"
onclick="login()">Log In</button>
                    <button type="button" class="toogle-btn"
onclick="register()">Register</button>
                </div>
                <form id="login" action="/accounts/login"
method="post" class="input-group">
                    {% csrf_token %}
                    <input type="text" class="input-field"
name="username" placeholder="User Id" required>
                    <input type="password" class="input-field"
name="password" placeholder="Enter Password" required>
                    <input type="checkbox"
class="checkbox"><span>Remember Password</span>

```



```

        <button type="submit"
class="submit-btn">Login</button>
    </form>
    <form id="register" action="/accounts/register"
method="post" class="input-group">
        {% csrf_token %}
        <input type="text" class="input-field"
placeholder="User Id" name="username" required>
        <input type="email" class="input-field"
placeholder="Email Id" name="email" required>
        <input type="password" class="input-field"
placeholder="Enter Password" name="password1" required>
        <input type="password" class="input-field"
placeholder="Confirm Password" name="password2" required>
        <input type="checkbox" class="checkbox"><span>I
agree to Terms and Conditions.</span>
        <button type="submit" class="submit-btn">Register
Now</button>
    </form>
</div>
</div>
<script>
    var x = document.getElementById("login")
    var y = document.getElementById("register")
    var z = document.getElementById("btn")

    function register() {
        x.style.left = "-400px";
        y.style.left = "50px";
        z.style.left = "110px";
    }

    function login() {
        x.style.left = "50px";
        y.style.left = "450px";
        z.style.left = "0px";
    }

```

```
)  
  
    </script>  
</body>  
  
</html>
```

View.py register, login and logout:

```
def register(request):
    if request.method == 'POST':
        username=request.POST['username']
        email=request.POST['email']
        password1=request.POST['password1']
        password2=request.POST['password2']

        if password1 == password2:
            if User.objects.filter(username=username).exists():
                print("username taken")
            elif User.objects.filter(email=email).exists():
                print("email taken")
            else:
                user=User.objects.create_user(username=username,
password=password1, email=email, first_name='-', last_name='-')
                user.save()

        else:
            print("passwords dont match")

        return redirect('/')

    else:
        return render(request, 'register.html')

def login(request):
    if request.method == 'POST':
        username=request.POST['username']
        password=request.POST['password']

        user=auth.authenticate(username=username, password=password)
```

```
    if user is not None:
        auth.login(request, user)
        return redirect('/')

    else:
        messages.info(request, 'Invalid username or password')
        return redirect('login')

else:
    return render(request, 'login.html')

def logout(request):
    auth.logout(request)
    return redirect('/')
```

Enter_content.html

```
{% extends 'base.html' %}
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYXxFlc
+NcPbldKGj7Sk" crossorigin="anonymous">

    <title>Search List</title>
{% block content %}
<div class="container my-4" style="background-color: whitesmoke;
padding:20px">
    <h2 class="text-center">Add new entry</h2>

    <form action="{% url 'addDatatoBase' %}" method="post">
        {% csrf_token %}
        <div class="form-group">
            <label for="title">Title</label>
            <input type="text" class="form-control" name="title"
aria-describedby="emailHelp">
```

```

        </div>
        <div class="form-group">
            <label for="Subject">Subject</label>
            <input type="text" class="form-control" name="subject"
aria-describedby="emailHelp">
        </div>
        <div class="form-group">
            <label for="description">Description</label>
            <textarea class="form-control" style="overflow:scroll"
name="description" rows="3"></textarea>
        </div>
        <div class="form-group">
            <label for="description">Links:</label>
            <small id="emailHelp" class="form-text
text-muted">--name--:: http://example.com [new line]</small>
            <textarea class="form-control" style="overflow:scroll;
height:70px " name="links" rows="3"></textarea>
        </div>

        <button type="submit" id="add" class="btn btn-primary" >Add
New</button>
    </div>
{% endblock %}

```

Views.py addDataBase

```

def addDatatoBase(request):
    if request.method == 'POST':
        title=request.POST['title']
        subject=request.POST['subject']
        description=request.POST['description']
        Links=request.POST['links']
        author=request.user.username
        content_data=content(title=title, subject=subject,
description=description, author=author, Links=Links)
        content_data.save()

```

```
return redirect('/')
```

Views.py EditDatatoBase

```
def EditDatatoBase(request, id):  
    if request.method == 'POST':  
        content_data=content.objects.get(pk=id)  
        content_data.title=request.POST.get('title')  
        content_data.subject=request.POST.get('subject')  
        content_data.description=request.POST['description']  
        content_data.Links=request.POST['links']  
        content_data.save()  
  
    return redirect('/')
```

Display.html

```
{% extends 'base.html' %}
{% block content %}
{%for pa in para%}
    <div class="Content-box" style="color: whitesmoke; margin-top:
5%; height:fit-content; align-items:center; border: 1px solid white;
border-radius:30px">
        <h2 style="font-family: Arial, Helvetica, sans-serif;
text-align:center">
            {{pa.title}}
        </h2>
        <p style="font-family: Arial, Helvetica, sans-serif;
padding-bottom:10px; margin-left: 10px; margin-right: 10px;
margin-bottom:20px; border-bottom: 1px solid white;">Subject:
<b>{{pa.subject}}</b></p>
        <p style="font-family: Arial, Helvetica, sans-serif;
margin-left: 10px; margin-right: 10px;
margin-bottom:20px;">{{pa.description}}</p>
        <ol style="font-size:15px; margin-left: 20px;">
            {% for one in lin %}
            <li>
                <a href="{{one}}" target="_blank">{{one}}</a><br>
            </li>
            {% endfor %}
    </div>
{%endfor%}
```



```

    </p>
    <p style="font-family: Arial, Helvetica, sans-serif;
font-size:14px ; margin-left: 25px; margin-bottom:20px; bottom: 2%;
display:inline-block">Date created: {{pa.date_created}}</p>
    <p style="font-family: Arial, Helvetica, sans-serif;
font-size:14px ; margin-left: 200px; margin-bottom:20px; bottom: 2%;
display:inline-block">Rating: </p>
    <div class="back-rate" style="margin-left:10px; display:
inline-block;">
        {% if pa.rate >= 1 %}
        <span style="font-size: 20px; color:darkgoldenrod">&#9733
</span>
        {%else%}
        <span style="font-size: 20px;">&#9733</span>
        {%endif%}

        {% if pa.rate >= 2 %}
        <span style="font-size: 20px; color:darkgoldenrod">&#9733
</span>
        {%else%}
        <span style="font-size: 20px;">&#9733</span>
        {%endif%}

        {% if pa.rate >= 3 %}
        <span style="font-size: 20px; color:darkgoldenrod">&#9733
</span>
        {%else%}
        <span style="font-size: 20px;">&#9733</span>
        {%endif%}

        {% if pa.rate >= 4 %}
        <span style="font-size: 20px; color:darkgoldenrod">&#9733
</span>
        {%else%}
        <span style="font-size: 20px;">&#9733</span>
        {%endif%}

        {% if pa.rate >= 5 %}

```

```

        <span style="font-size: 20px; color:darkgoldenrod">&#9733
</span>
        {%else%}
        <span style="font-size: 20px;">&#9733</span>
        {%endif%}
    </div>
</div>

    <div class="rate">
        <input type="radio" id="star5" name="rate" value="5" />
        <label for="star5" title="text" style="color:
darkgoldenrod;">&#9733&#9733&#9733&#9733&#9733</label><br>
        <input type="radio" id="star4" name="rate" value="4" />
        <label for="star4" title="text" style="color:
darkgoldenrod;">&#9733&#9733&#9733&#9733</label><br>
        <input type="radio" id="star3" name="rate" value="3" />
        <label for="star3" title="text" style="color:
darkgoldenrod;">&#9733&#9733&#9733</label><br>
        <input type="radio" id="star2" name="rate" value="2" />
        <label for="star2" title="text" style="color:
darkgoldenrod;">&#9733&#9733</label><br>
        <input type="radio" id="star1" name="rate" value="1" />
        <label for="star1" title="text" style="color:
darkgoldenrod;">&#9733</label>
    </div>

{%endfor%}
{% endblock %}

```

Views.py display

```

    para=content.objects.filter(id=id)
    lin=re.findall("(?P<url>https?:\/\/[^\s]+)", para[0].Links)
    nam=re.findall("(:.:.)",para[0].Links)
    return render(request, 'display.html', {'para' : para, 'lin':lin,
nam':nam})

```

Dashboard.html

```
{% extends 'base.html' %}

{% block content %}
<p style="color:whitesmoke; display: inline-block; margin-top: 8%;
margin-left: 30%;">YOUR CONTENT:</p>
<button onclick="location.href='/addData'" style="color: white;
display: inline-block; margin-left:22%; background-color:tomato;
border-radius:10px; border-color:tomato; width:100px;
text-align:center; padding:5px">+ Add new</button><br>
<div class="Content-box" style="margin-left:23%; width: 800px;
height:auto">
    {% for pa in para %}
        <div class="dash_content" onclick="location.href='{% url
'disData' id=pa.id %}'">
            <p>{{pa.title}}<br>
            Subject: {{pa.subject}}</p>
        </div>
        <div style="margin-bottom: 20px;">
            <button onclick="location.href='{% url 'deleteData' id=pa.id
%}'" style="color: white; display: inline-block; margin-left:22%;
background-color:tomato; border-radius:10px; border-color:tomato;
width:100px; text-align:center; padding:5px">Delete</button>
            <button onclick="location.href='{% url 'editData' id=pa.id
%}'" style="color: white; display: inline-block; margin-left:22%;
background-color:rgba(57, 102, 57, 0.788); border-radius:10px;
border-color:rgba(57, 102, 57, 0.788); width:100px;
text-align:center; padding:5px">Edit</button>
        </div>
    {% endfor %}
</div>
{% endblock %}
```

View.py

```
def dashboard(request):
    para=content.objects.filter(author=request.user.username)
    return render(request, 'dashboard.html', {'para' : para})
def deleteData(request, id):
```

```
    para=content.objects.filter(id=id)
    para.delete()
    return redirect('/')

def editData(request, id):
    para=content.objects.filter(id=id)
    return render(request, 'EditData.html', {'para' : para})
```

Search.html

Views.py search

```
def search(request):
    search_key=request.POST['search_key']
    try:
        para=content.objects.filter(title__icontains=search_key)
    except content.DoesNotExist:
        para=None
    return render(request, 'search.html', {'para' : para})
```

ContactUs.html

```
style>
body {
  background: linear-gradient(rgba(0,0,0,0.5), rgba(0,0,0,0.5)), url("../image/About2.jpg")no-repeat center center/cover;
  /* opacity: 0.3; */
}

.card:hover {
  background-color: #14181c;
  box-shadow: 7px 5px 15px #232728;
  height: 600px;
  width: 300px;
}
/style>

body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark ">...
</nav>
<h1 class="text-center my-4 text-decoration-underline fw-bold fst-italic" style="color:darkgray ">Our Team</h1>
<h2 id="Front1" class="text-center my-4 text-decoration-underline fw-bold fst-italic" style="color:#232728 ">Front-End Team</h2>
<div class="d-flex me-3" id="front">
  <div class="card"
    style="width: 18rem; margin: 23px; background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
    
    <div class="card-body">
```

```
style="width: 18rem; margin: 23px; background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">

<div class="card-body">
  <h5 class="card-title">Ashwani Kumar</h5>
  <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
  <a href="mailto: 20bcs023@iiitdwd.ac.in" class="btn btn-primary">Email</a>
</div>
</div>
<div class="card" style="width: 18rem; margin: 23px; background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
  
  <div class="card-body">
    <h5 class="card-title">Hemang Ranga</h5>
    <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
    <a href="mailto: 20bcs057@iiitdwd.ac.in" class="btn btn-primary">Email</a>
  </div>
</div>
<div class="card" style="width: 18rem; margin: 23px; background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
  
  <div class="card-body">
    <h5 class="card-title">Rakshitha</h5>
    <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
    <a href="mailto: 20bcs0107@iiitdwd.ac.in" class="btn btn-primary">Email</a>
  </div>
</div>
<div class="card" style="width: 18rem; margin: 23px; background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
  
  <div class="card-body">
    <h5 class="card-title">Yashu Mittal</h5>
```

```

    <h5 class="card-title">Yashu Mittal</h5>
    <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
    <a href="mailto: 20bcs139@iiitdwd.ac.in" class="btn btn-primary">Email</a>
  </div>
</div>
</div>
<h2 id="back1" class="text-center my-4 text-decoration-underline fw-bold fst-italic" style="color:darkgray ">Back-End Team</h2>
<div class="d-flex me-3" id="back">
  <div class="card" style="width: 18rem;margin: 23px;background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
    
    <div class="card-body">
      <h5 class="card-title">Ashis J Kalathil</h5>
      <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
      <a href="mailto: 20bcs022@iiitdwd.ac.in" class="btn btn-primary">Email</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;margin: 23px;background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
    
    <div class="card-body">
      <h5 class="card-title">Sunil Patidar</h5>
      <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
      <a href="mailto: 20bcs0131@iiitdwd.ac.in" class="btn btn-primary">Email</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;margin: 23px;background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
    
    <div class="card-body">
      <h5 class="card-title">Veekshith Ravula Reddy</h5>
      <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>

```

```

      <h5 class="card-title">Veekshith Ravula Reddy</h5>
      <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
      <a href="mailto: 20bcs111@iiitdwd.ac.in" class="btn btn-primary">Email</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;margin: 23px;background-color: #b4aec7;; transition: all 1s ease-in-out 0.5;">
    
    <div class="card-body">
      <h5 class="card-title">Yugal Deep</h5>
      <p class="card-text">CSE Second Year <br>IIIT Dharwad <br> 2020-24 </p>
      <a href="mailto: 20bcs140@iiitdwd.ac.in" class="btn btn-primary">Email</a>
    </div>
  </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
  crossorigin="anonymous"></script>
</body>
</html>

```

Chapter - 7

Testing

TEST PLAN

A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from test engineers.

Test plan document formats can be as varied as the products and organizations to which they apply. There are three major elements that should be described in the test plan: Test Coverage, Test Methods, and Test Responsibilities.

Test coverage

Test coverage in the test plan states what requirements will be verified during what stages of the product life. Test Coverage is derived from design specifications and other requirements, such as safety standards or regulatory codes, where each requirement or specification of the design ideally will have one or more corresponding means of verification. Test coverage for different product life stages may overlap, but will not necessarily be exactly the same for all stages. For example, some requirements may be verified during the Design Verification test, but not repeated during the Acceptance test. Test coverage also feeds back into the design process, since the product may have to be designed to allow test access.

Test methods

Test methods in the test plan state how test coverage will be implemented. Test methods may be determined by standards, regulatory agencies, or contractual agreements, or may have to be created new. Test methods also specify test equipment to be used in the performance of the tests and establish pass/fail criteria. Test methods used to verify hardware design requirements can range from very simple steps, such as visual inspection, to elaborate test procedures that are documented separately.

Test responsibilities

Test responsibilities include what organizations will perform the test methods and at each stage of the product life. This allows test organizations to plan, acquire or develop test equipment and other resources necessary to implement the test methods for which they are responsible. Test responsibilities also include, what data will be collected, and how that data will be stored and reported (often referred to as "deliverables"). One outcome of a successful test plan should be a record or report of the verification of all design specifications and requirements as agreed upon by all parties.

TEST FIXTURE

Test fixture refers to the fixed state used as a baseline for running tests in software testing. The purpose of a test fixture is to ensure that there is a well known and fixed environment in which tests are run so that results are repeatable. Some people call this the test context.

Example of fixtures:

- Loading a database with a specific, known set of data
- Erasing a hard disk and installing a known clear operating system installation
- Copying a specific known set of files

Four phases of a test:

- Set up -Setting up the test fixture.
- Exercise- -Interact with the system under test.
- Verify- -Determine whether the expected outcome has been obtained.
- Tear down- -Tear down the test fixture to return to the original state.

Use of fixtures

Some advantages of fixture include separation of the test initialization from the testing, reusing a known state for more than one test, and the special assumption by the testing framework that the fixture set up works.

TEST CASES

1. GUI Test Cases :

- Total no of features that need to be checked
- Look & Feel Look for Default values if at all any (date & Time, if at all any required)
- Look for spell check

2. Positive Test Cases:

- The positive flow of the functionality must be considered
- Valid inputs must be used for testing
- Must have a positive perception to verify whether the requirements are justified.

3. Negative Test Cases:

- Must have a negative perception.
- Invalid inputs must be used for tests.

Chapter - 8

Future Enhancement And Conclusion

FUTURE ENHANCEMENT

Although it is already an advanced version of all the previous available web applications as it also has limitations, some of them are as :

- We can provide login facilities separately to students, experts, faculties, admin, etc.
- We can provide a forum to ask their query and get solutions related to them.
- We can enable chatting with their friends, professors, mentors, etc. and can talk about their personal problems.
- We can provide email facilities to users.
- We can provide Video lectures on various subjects
- We can provide PPT's of various subjects and topics

CONCLUSION

Tutor's Corner has been developed such that it can be accessed from any type of system. The system will be capable of providing required study content and all the activities needed by a student pursuing engineering within a given time frame with no errors and the system should be available and operational all the time. The system is developed with an aim of usability so that it is an easy to use system that requires the least amount of user input possible. Using this system, a user should only have general computer knowledge. An easy, well-structured module will show the correct path to reach the destination. Users will be authenticated to ensure that no unauthorized users gain access to private information.

Chapter - 9

References

1. <https://www.youtube.com/watch?v=hKB-YGF14SY>
2. https://www.youtube.com/watch?v=L5W_rGMs_now
3. https://www.youtube.com/watch?v=6mbwJ2xhgzM&list=PLu0W_9lI9agiCUZYRsvtGTXdxkzPyltg
4. <https://learndjango.com/>
5. <https://www.youtube.com/watch?app=desktop&v=OTmQOjsl0eg>
6. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>
7. <https://www.w3schools.com/html/default.asp>
8. <https://www.tutorialspoint.com/django/index.htm>
9. <https://youtu.be/OTmQOjsl0eg>
10. <https://www.youtube.com/watch?v=rHux0gMZ3Eg>