MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Final Year Internship Report

on

Web Development Internship

Submitted By:

Manoj Kumar

0901CS181059

Faculty Mentor:

Dr.Ranjeet Kumar Singh

Assistant Professor, Department Of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

Web Development Internship

A final year internship report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Manoj Kumar

0901CS181059

Internship Faculty Mentor:

Mr. Pushkar Goyal, Team Head,Praedico Global Research

Submitted to: **Dr Ranjeet Kumar Singh**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

s is certified that **Manoj Kumar** (0901CS181059) has submitted the Internship report titled **Web elopment** of the work he has done under the mentorship of **Prof Ranjeet Kumar Singh** in partial fulfilment ne requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from adhav Institute of Technology and Science, Gwalior.

Prof Ranjeet Kumar singh
Faculty Mentor
Professor
Computer Science and Engineering

Dr. Manish Dixit
Professor and Head,
Computer Science and Engineering
Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

age

Address: First Floor, Garima Arcade, Shinde ki Chhawani, Gwalior
Email: mail@praedicoglobalresearch.com
Website: www.praedicoglobalreserach.com

Ref.: PGR-2022/I-540 **Date:** 11th – May - 2022

CERTIFICATE OF INTERNSHIP

This certificate is awarded to

Mr./Miss. MANOJ KUMAR

In appreciation for your accomplishments in the company as an intern
(Position titled-***"Web Developer – PYTHON DJANGO"***)
at **Praedico Global Research Pvt. Ltd.**,
from **Jan 10th, 2022 to May 10th, 2022**.

We take this opportunity to wish you a long, happy and successful career.

Worlds Finest Robotic Stock Researchers

Authorized Signatory
Praedico Global Research Pvt. Ltd.

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this Internship report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in CSE at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof Ranjeet Kumar singh**, **Professor** , Department of CSE.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Manoj Kumar
0901CS181059
IV Year,
Computer Science and Engineering

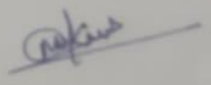
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

ıe full semester internship has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute Technology and Science** to allow me to continue my disciplinary/interdisciplinary internship as a curriculum quirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum)18), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, r. **R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

would sincerely like to thank my department, **Department of Computer Science and Engineering, for lowing** me to explore this internship. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of omputer Science and Engineering, for his continued support during the course of this engagement, which eased ıe process and formalities involved.

am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof Ranjeet Kumar Singh**, rofessor, Department of Computer Science and Engineering, for his continued support and close mentoring ıroughout the internship. I am also very thankful to the faculty and staff of the department



Manoj Kumar
0901CS181066
IV Year,
Computer Science and Engineering

Abstract

In web development I have learned full stack developing course where we have created blogsite. In this website project anyone can read and write blogs and admin can manage all the activities and user can register in website and read and write according to us. Many students have already found how much fun it is to learn programming with Python. It's now time to advance Python to the next level. This course will cover Django, an open source Python web framework that helps you save time and enjoy web development. It's Computer Science Course who wish to make his career in IT companies then he can opt for this course. how to create beautiful web applications with minimal effort. The Model-View-Controller (MVC) architectural paradigm is used by Django. Its purpose is to make building complex, database-driven websites easier. Django stresses component reusability and "pluggability," quick development, and the DRY (Don't Repeat Yourself) concept. Even settings, files, and data models are written in Python. Setup and configuration, template language, and other topics will be covered throughout the event.

Chapter-1

Introduction:

Blogging has become such a mania that a new blog is being created every hour,every minute,every second .A blog is the best voice among the online crowd.People who want to gain and share knowledge or information can use this website.Users can share their views on different aspects, current trends,current topics through blog posts.

It is helpful for all those who want to post their blogs ,who want to read new blogs on a daily basis.Blogs on different categories will be available to read.Users can later update or delete their posts if they feel any need to do so.

If users are facing any query they can fill in their contact details and send their queries to the admin. And there is an admin who can review posts,can add,delete and update different categories,subcategories.It can see the required details of all the registered users.

Objectives

To provide a platform where users can create and see different blog articles about the organization.
Users can submit their queries related to the website by filling their details in the contact form.
Blogs created can be deleted or updated by the users.
Admin can see all the blogs posted on the website.
Admin can delete inappropriate blogs.
Admin can add categories and subcategories under which blogs will be posted.

Scope

Build a online blogging website
Data retrieval and search functionality ,django mailing system for forgetting password.
Created database tables in django admin.

1.1 Modules

- **User**

On the landing page users will see the two active buttons- admin login and user login so general users will click on the user login to enter the user homepage.

A. Guest User

- **Home Page**

Navigation Bar—links

Categories

Guest Users can see the different categories available to post and read blogs.
Using the SubcategoryShow button on this page user can see different subcategories under aparticular category.

SubCategories

Guest Users can see the different Subcategories available to post and read blogs.

Blogs
Users can see all the blogs which have been posted on the website.

Contact
Using this contact form users can submit their which they are facing regarding the website or anything related to the website.

About
Using this you can read information related to the website.

Login
Clicking on the login button on the navigation bar, a login form will open using which a registered user can log in to the website.
As soon as the user logs in a session will be created which will be alive until he logs out.
Register
Clicking on the register button on the navigation bar, a registration form will open using which new user can register filling the required credentials

B. Authenticated User
Home Page
Navigation Bar – links

Categories
Authenticated Users can see the different categories available to post and read blogs.
Using the SubcategoryShow button on this page user can see different subcategories under a particular category.

SubCategories
Guest Users can see the different Subcategories available to post and read blogs.
Blogs
ALL BLOGS - User can see all the which have been posted on the website.
MY BLOGS - User can see his own blog posts.
Update - User can update his blog using this form which will open on next
page.
Delete - User can delete his blog post if wish to do so.
ADD BLOG - Clicking on this link a form will open using which a user can select
category then subcategory and title ,image icon,content for the blog post and click on the submit to upload the blog.

Contact
Using this contact form users can submit their which they are facing regarding the website or anything related to the website.

.About
Using this you can read information related to the website.

Logout
Clicking on the logout button user can log out of the website and his session will be expired at that moment itself.

Admin
1. Before login
Home Page
Navigation Bar

Login
Using the Login Button on the website admin can login to the website.
As soon as the admin logs in a session will be created which will be alive until he logs out.

2. After Login
Home Page

Navigation Bar

Category
1. Category add link will open new page where admin add new categories.
Update - On this page, admin can use update button which will open new page admin can update information regarding category.
Delete - Using this button admin can delete category.

SubCategory
1. SubCategory add link will open new page where admin add new categories. Update - On this page, admin can use update button which will open new page
admin can update information regarding Subcategory.
Delete - Using this button admin can delete Subcategory.

Users
Users option on the navigation bar will show the list of all the registered users to admin.

Blogs
Blogs option will show all the blogs which have been posted on the website to the admin.
Admin can delete inappropriate blogs if he feels the need to do so.

Logout
Clicking on the logout button admin can log out of the website and his session will expired at that moment itself.

2. **FIGURES**

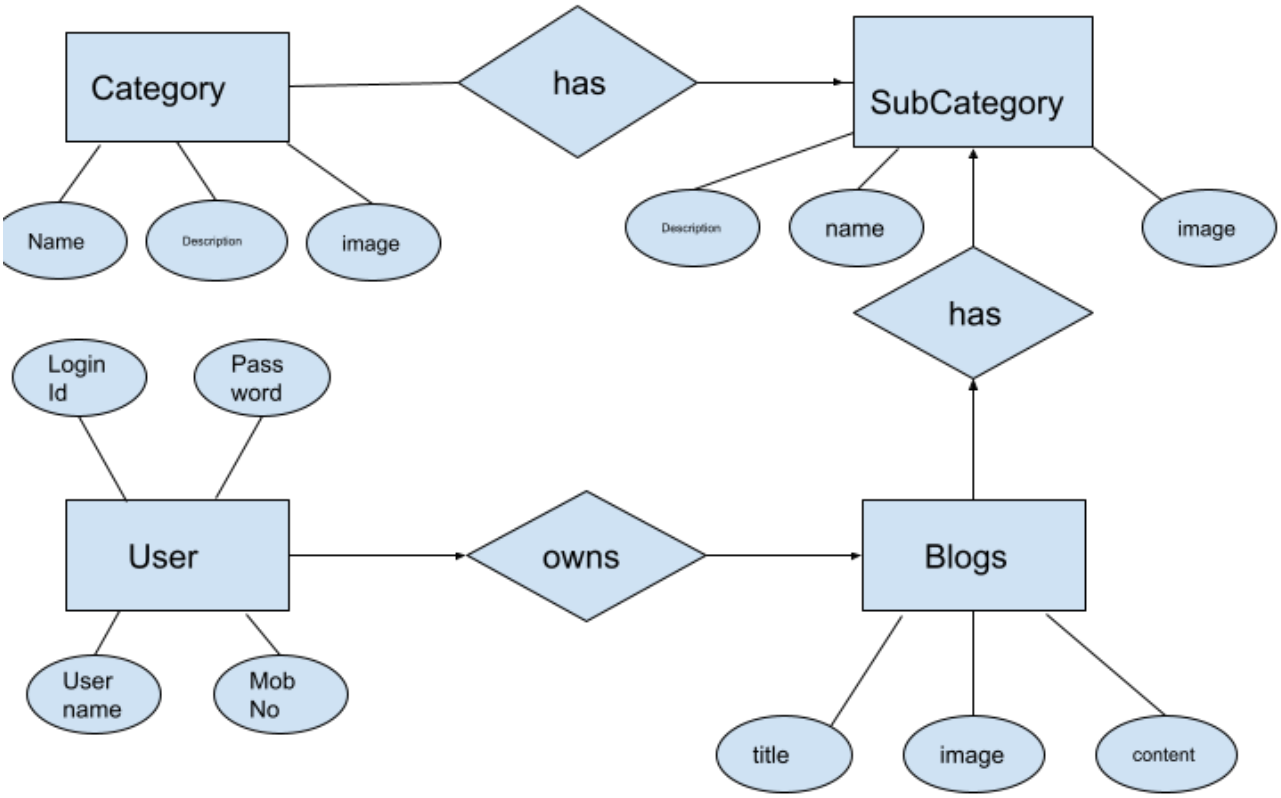


Fig-2(ER Diagram)

In this web application, Users (login id ,password,username,mobno) owns blogs (title,image,content),user is a foreign key for the blog table.Subcategory (name,title,description) contains blogs which is a foreign key for blog table.Category (name,title,description) has subcategories as a foreign key

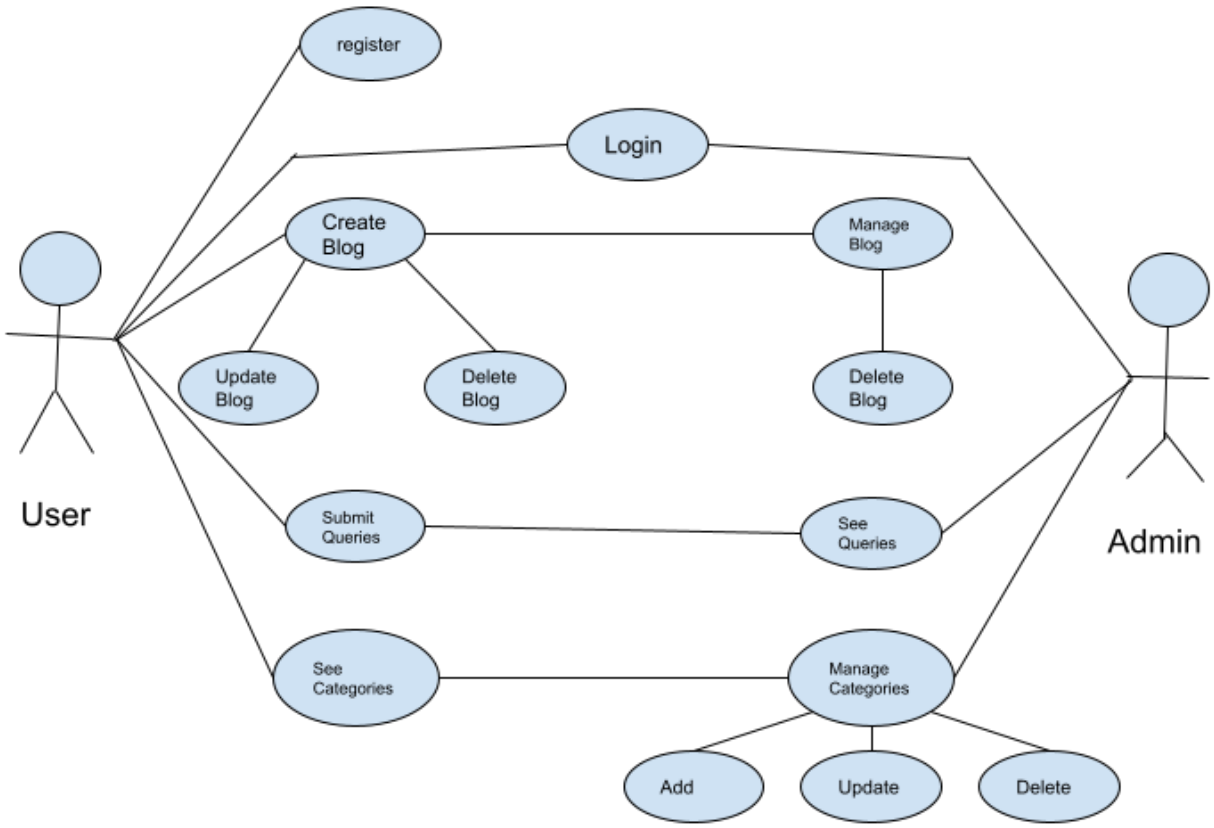


Fig-2(use of case Diagram)

User will register and login on website.He can create,update,delete bogs,can submit queries,can see categories and subcategories.Admin can login,see queries sent by users, add,update, delete categories and subcategories.

Chapter-2

1. **Requirement Analysis**

Phase 1: Requirement collection and analysis: As a part of the standard protocol, we will create a static prototype of the website which will be non – working. This prototype will give you an idea of how the actual website will look like.

Phase 2: Design: In this phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture. We will appoint a designer to provide a user friendly, eye catchy design to your project. To ensure a top-quality design you can give any reference website or template. This will help us to visualize the requirement and help us to provide the website of your choice.

Phase 3: Coding: Once the system design phase is over, the next phase is coding. In this phase, developers start to build the entire system by writing code using the chosen programming language.

Phase 4: Testing: Once the software is complete, it will be deployed in the testing environment. The testing team starts testing the functionality of the entire system. This will be done to verify that the entire application works according to your requirement. During this phase, QA and testing team may find some bugs/defects which they will communicate to our developers. The development team will fix the bug and send it back to QA for a re-test. This process will continue until the software is bug-free, stable, and working according to your business needs

Phase 5: Installation/Deployment: Once the software testing phase is over and no bugs or errors left in the system then the final deployment process will start.

Chapter -3

4 System Requirement Specification

4.1 System Configuration

A system configuration in systems engineering defines the computers, processes, and devices that compose the system and its boundary. More general the system configuration is the specific definition of the elements that define and/or prescribe what a system is composed of.

4.2 Software Requirement

- Technology : python django
- IDE: pycharm /atom
- Client side technologies: HTML , CSS, JAVASCRIPT ,BOOTSTRAP
- Server side techhnologies: python
- Data base server : sqlite
- Operating systems: Microsoft windows/ linux

4.3 Hardware Requirements

Intel Core i5 processor

8GB RAM

1 TB hard disk

Chapter -4

Technology used

HTML

CSS

BOOTSTRAP

PYTHON

DJANGO

VISUAL STUDIO CODE EDITOR

DB SQLITE

Short Intro about technologies used:

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

CASCADING STYLE SHEET (CSS)

Cascading Style Sheets (CSS) are a collection of rules we use to define and modify web pages CSS are similar styles in Word. CSS allow Web designers to have much more control over their pages look and layout. For instance, you could create a style that defines the body text to be Verdana, 10 point. Later on, you may easily change the body text to Times New Roman, 12 point by just changing the rule in the CSS. Instead of having to change the font on each page of your website, all you need to do is redefine the style on the style sheet, and it will instantly change on all of the pages that the style sheet has been applied to. With HTML styles, the font change would be applied to each instance of that font and have to be changed in each spot

CSS can control the placement of text and objects on your pages as well as the look of those HTML information creates the objects (or gives objects meaning), but styles describe how the objects should appear. The HTML gives your page structure, while the CSS creates the "presentation". An extemal CSS is really just a text file with a css extension. These files can be created with Dreamweaver, a CSS editor, or even Notepad.

The best practice is to design your web page on paper first so you know where you will want to use styles on your page. Then you can create the styles and apply them to your page

LANGUAGE Used: Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming Python is often described as a "batteries included" language due to its comprehensive standard library

Bootstrap

Bootstrap is one of the most popular front-end frameworks out there. It contains some amazing CSS classes for UI development.

Bootstrap has pre-defined CSS files and JavaScript code, which you can link with HTML files. Those CSS files contain classes that can be directly used on HTML elements. We used it in our Django static files DataFlair tutorial.

Django

Django is basically a high-level Python web application framework that enables the rapid development of web applications. It achieves so with pragmatic, much cleaner design and is also easy to use (in comparison of other frameworks) thus is very popular among web developers.
It is a backend framework used to resolve problems of connectivity with databases, other server problems, **SEO solutions,** etc so that a web developer need not write the same code for the similar modules (like database connection, admin interface) for each website.
All the functionality comes in the Django framework in the form of web applications. You just have to import those applications according to your need and thus you can concentrate more on the unique application of your website rather than dealing with all these backend problems.

CRUD

CRUD stands for Create, Read, Update & Delete. These are the four basic operations which are executed on Database Models. We are developing a web app which is capable of performing these operations.
Since we are developing a library app, let's take an example of the same. In a library, books are objects. The books have attributes like name, author, etc. We need an application which can perform CRUD operations on book object. The CRUD operations are defined as follows:

1. **Read Operation**

The ability of the application to read data from the database.

2. **Create Operation**

The ability of the application to store data in the database.

3. **Update Operation**

The ability of the application to edit the stored value in the database.

4. **Delete Operation**

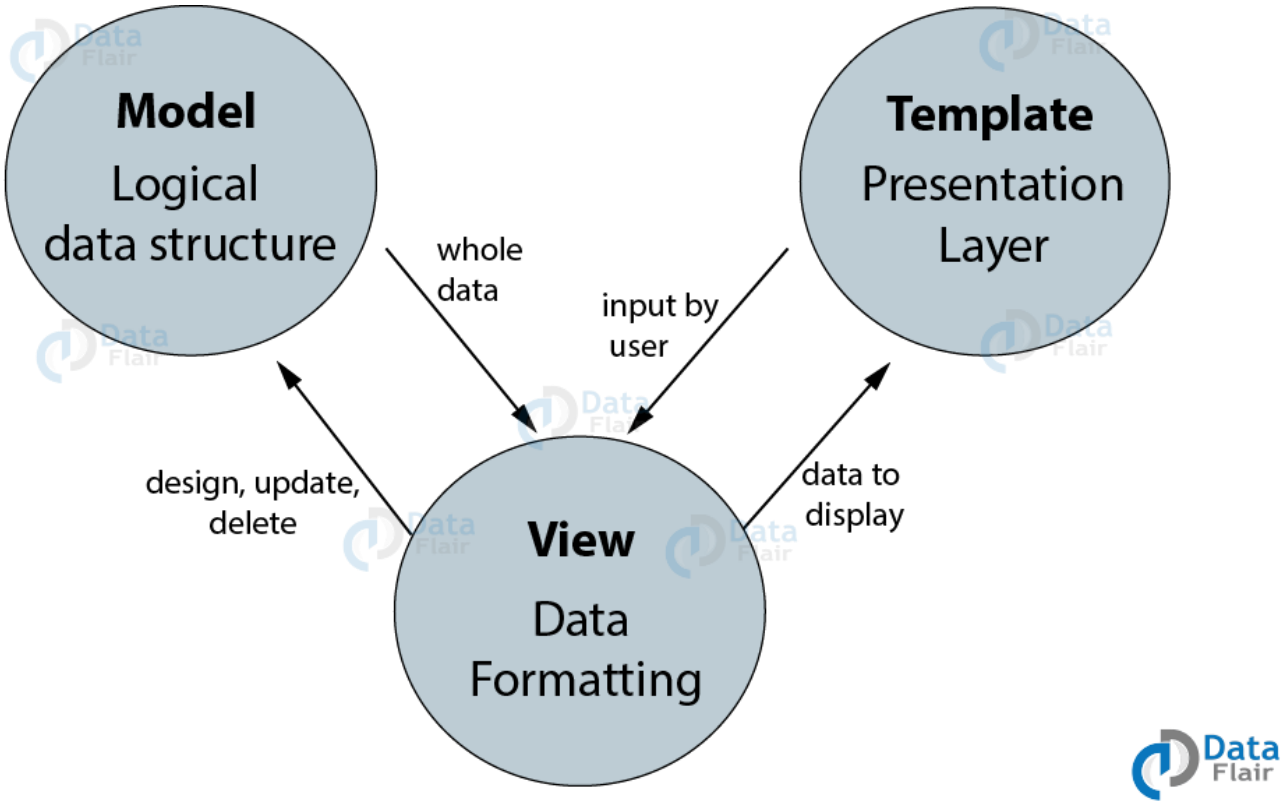
The ability of the application to delete the value in the database.

We are going to develop the operations in the same order.
Majority of applications on the internet are CRUD applications. **For example –** Facebook uses CRUD operations to save your data on their database. You can change your profile picture that means

perform the update operation. Of course, you can see the data in-app or browser which is read operation.
Also, you can delete your Facebook account which is delete operation. Summarising it, almost all the applications you use are CRUD applications.
For developers, making a CRUD application is one of the very first steps. If you can make a CRUD app from technology then you can start deploying projects.

MTV Architecture Components (Model, Template, and View)

This is a variation of the MVC pattern as you can see in the acronym itself the Template keyword replaces the Controller. Although, the Template is not exactly functioning as the controller and has some different properties than the controller.
The definitions of Model still remain the same that is, the Model contains the **logical file structure of the project** and is the middleware & data handler between database and view. The Model provides a definition of how the data formats as coming from the view so, it stores in the database and vice-versa, i.e., the retrieving information from the database transfers to the view in the displayable format.



fig

Chapter-5

5.Enhancements

Proposed Enhancement:

1. More functionality can be added
2. Search Optimization can be enhanced
3. Security can be improved
4. User Interface can be improved.

Chapter-6

Coding & Output

usermodule: views.py

```
from django.shortcuts import render, redirect, HttpResponse
from userapp.models import Register, Blog
from django.contrib import messages
from adminapp.models import category, subcategory
from datetime import date


# Create your views here.

# def home(request):
#     return render(request,'userapp/userbluhome.html')

def home(request):
    rec = request.session.get('sessid', False)
    context = {'rec': rec}
    return render(request, 'userapp/userbluhome.html', context)


def reguser(request):
    if request.method == "POST":
        unm = request.POST.get('unm')
        eid = request.POST.get('eid')
        mno = request.POST.get('mno')
        pwd1 = request.POST.get('pwdru1')
        pwd2 = request.POST.get('pwdru2')
        regdate = date.today()

        if pwd1 == pwd2:
            request.session['sessid']: eid
            user_data = Register(unm=unm, mno=mno, eid=eid,
pwd=pwd1, regdate=regdate)
            user_data.save()
            messages.success(request, 'you are successfully
registered')
            return redirect('registered')

    return render(request, 'userapp/userbluhome.html')


def registered(request):
    return render(request, 'userapp/indexalu.html')


def login(request):
    leid = request.POST.get('leid')
    pwd1 = request.POST.get('pwdlu')

    res = Register.objects.filter(eid=leid, pwd=pwd1).exists()
```

```
        if res:
            request.session['sessid'] = leid
            messages.success(request, 'you are logged in successfully')
            return redirect('/userapp/login')
        else:
            return render(request, 'userapp/userbluhome.html')


    def logout(request):
        del request.session['sessid']
        messages.success(request, 'You are logged out successfully')
        return redirect('/userapp')


    def contactblu(request):
        return render(request, 'userapp/contactblu.html')


    def addblogcat(request):
        data = category.objects.all()

        c_id = request.GET.get('name')
        print(c_id)

        context = {'data': data, 'c_id': c_id}

        return render(request, 'userapp/addblogcat.html', context)


    def addblog(request, c_id):
        rec = request.session.get('sessid', False)
        unm = Register.objects.get(eid=rec)

        data1 = category.objects.get(id=c_id)

        data2 = subcategory.objects.filter(cname=c_id)

        if request.method == "POST":
            scname = request.POST.get('scname')
            print('subcategory', scname)
            print('category...', name)
            title = request.POST['btitle']
            content = request.POST['blgcontent']
            post_date = date.today()

            blg_data = Blog(title=title, content=content, slug='',
    unm=unm, subcategory=scname, timestamp=post_date)
            blg_data.save()

        context = {'data1': data1, 'data2': data2}
        return render(request, 'userapp/addblog.html', context)


    def showbloghome(request):
```

```
    blog_data = Blog.objects.all()
    rec = request.session.get('sessid', False)
    context = {'blog_data': blog_data, 'rec': rec}

    return render(request, 'userapp/showbloghome.html', context)


def showblog(request, slug):
    blg_data = Blog.objects.filter(slug=slug).first()
    current_user = blg_data.unm
    rec = request.session.get('sessid', False)
    loggedin_user = Register.objects.get(unm=current_user)

    context = {'blg_data': blg_data, 'rec': rec, 'l_user':
loggedin_user}

    return render(request, 'userapp/showblog.html', context)


def myblogs(request):
    rec = request.session.get('sessid', False)
    unm = Register.objects.get(eid=rec)
    user_blog = Blog.objects.filter(unm=unm)

    context = {'user_blog': user_blog}

    return render(request, 'userapp/myblogs.html', context)


def updateblog(request, blg_id):
    data = Blog.objects.get(id=blg_id)
    context = {'data': data}

    if request.method == "POST":
        title = request.POST.get('btitle')
        icon = request.POST.get('bgicon')
        content = request.POST.get('blgcontent')

        Blog.objects.filter(id=blg_id).update(title=title,
bl_image=icon, content=content)
        messages.success(request, 'Your post has been updated
successfully')

        return redirect('/userapp/showbloghome/')

    return render(request, 'userapp/updateblog.html', context)


def deleteblog(request, bl_id):
    Blog.objects.filter(id=bl_id).delete()
    messages.error(request, 'Your post has been deleted')
    return redirect('/userapp/showbloghome')


from django.core.mail import EmailMessage
```

```
from django.conf import settings
from django.template.loader import render_to_string


def success(request, email, pwed, unm):
    context = {}

    context = {'email': email}

    template = render_to_string('userapp/success.html')

    email = EmailMessage(

        'thanks',
        f'Hello "{unm}" Your password is {pwed} Please use this
password to login ',
        settings.EMAIL_HOST_USER,
        [email]

    )

    email.fail_silently = False
    email.send()

    return HttpResponse('error please try again')


def forget_password(request):
    rec = request.session.get('sessid', False)

    if request.method == "POST":
        email1 = request.POST.get('fmail')
        if not email1:
            return HttpResponse('please ennter your email id')
        email = Register.objects.get(eid=email1)
        m_email = email.eid
        pwed = email.pwd
        unm = email.unm
        success(request, m_email, pwed, unm)

    return render(request, 'userapp/success.html', {'rec': rec})


def search(request):
    rec = request.session.get('sessid', False)
    data = Blog.objects.filter()

    query = request.GET.get('query')

    blogtitle = Blog.objects.filter(title__icontains=query)
    blogcontent = Blog.objects.filter(content__icontains=query)
    allposts = blogtitle.union(blogcontent)

    context = {'rec': rec, 'allposts': allposts}
```

```
    return render(request, 'userapp/search.html', context)
```

user module : urls.py

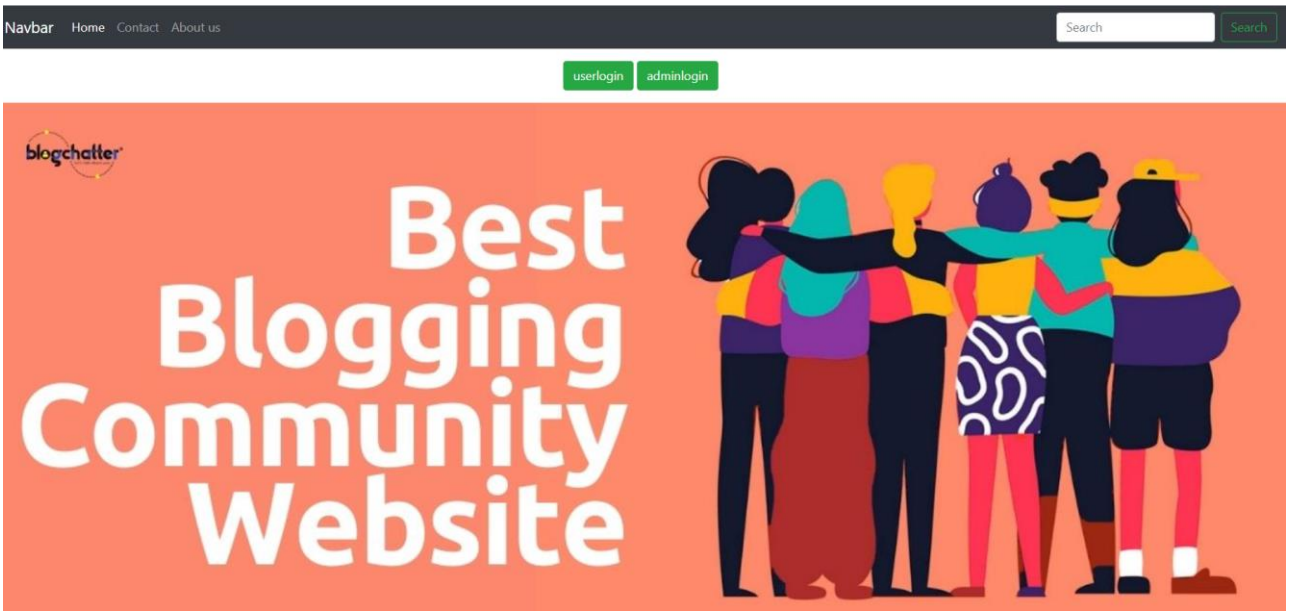
```
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('reguser', views.reguser, name='reguser'),
    path('registered', views.registered, name='registered'),
    path('login', views.login, name='login'),
    path('logout', views.logout, name='logout'),
    path('contactblu/', views.contactblu, name='contactblu'),
    path('addblogcat/', views.addblogcat, name='addblogcat'),
    path('addblog/<str:c_id>', views.addblog, name='addblog'),
    # path('showcatalu',views.showcatalu,name='showcatalu'),
    path('showbloghome/', views.showbloghome, name='showbloghome'),
    path('showblog/<str:slug>', views.showblog, name='showblog'),
    path('myblogs', views.myblogs, name='myblogs'),
    path('updateblog/<str:blg_id>/', views.updateblog, name='updateblog'),
    path('deleteblog/<str:bl_id>/', views.deleteblog, name='deleteblog'),

    path('success/', views.forget_password, name='success'),
    path('search/', views.search, name='search'),

]
```

Admin module : models.py

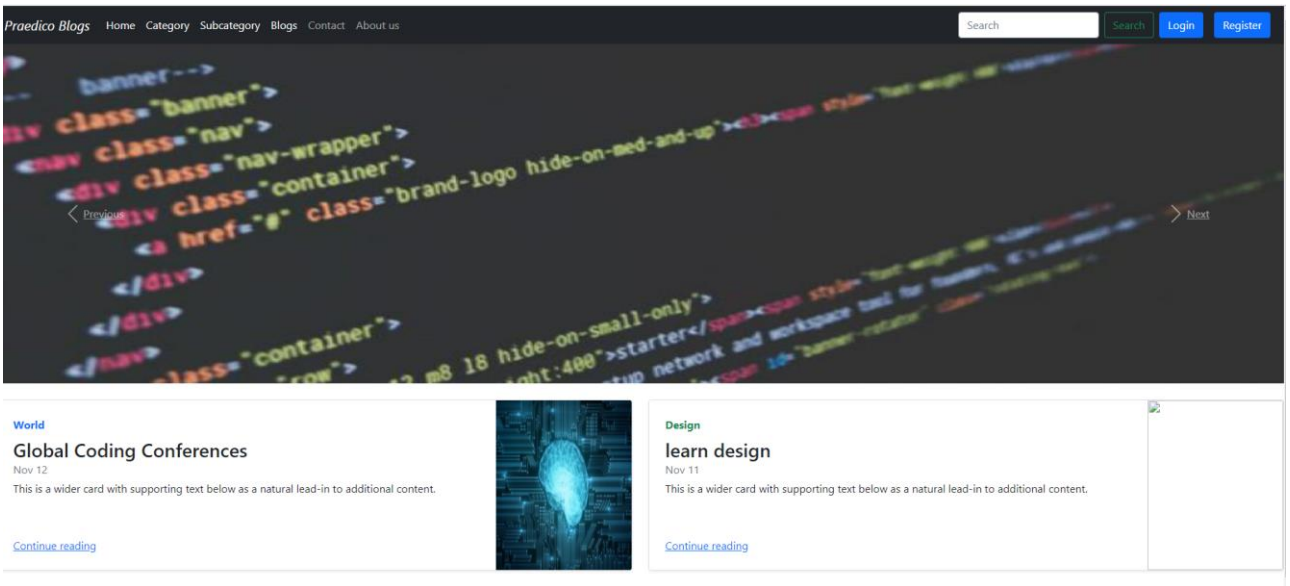
```
from django.db import models


# Create your models here.
class category(models.Model):
    catname = models.CharField(max_length=30)
    caticon = models.ImageField()
    catdesc = models.CharField(max_length=2000)
    regdate = models.DateField(auto_now=False, auto_now_add=True)

    def __str__(self):
        return self.catname


class subcategory(models.Model):
    scatname = models.CharField(max_length=30)
    scicon = models.ImageField()
    scdesc = models.CharField(max_length=2000)
    regdate = models.DateField()
    cname = models.ForeignKey(category, on_delete=models.CASCADE)
```

Front End

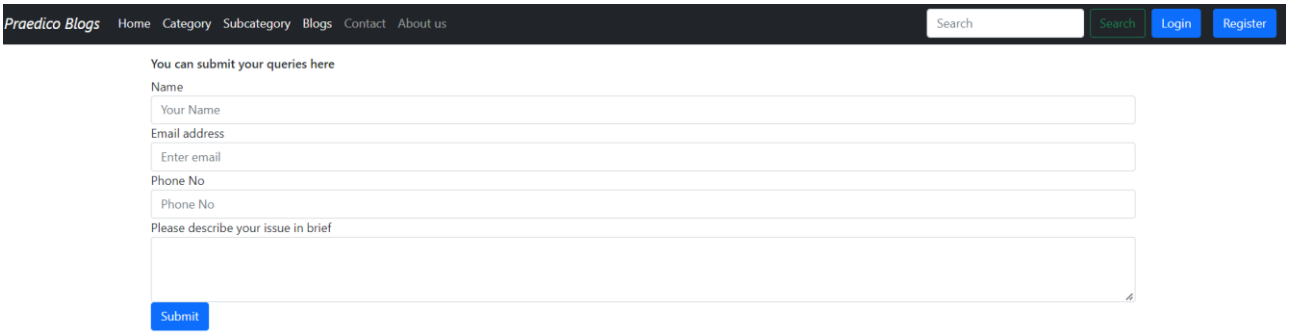
Landing Page



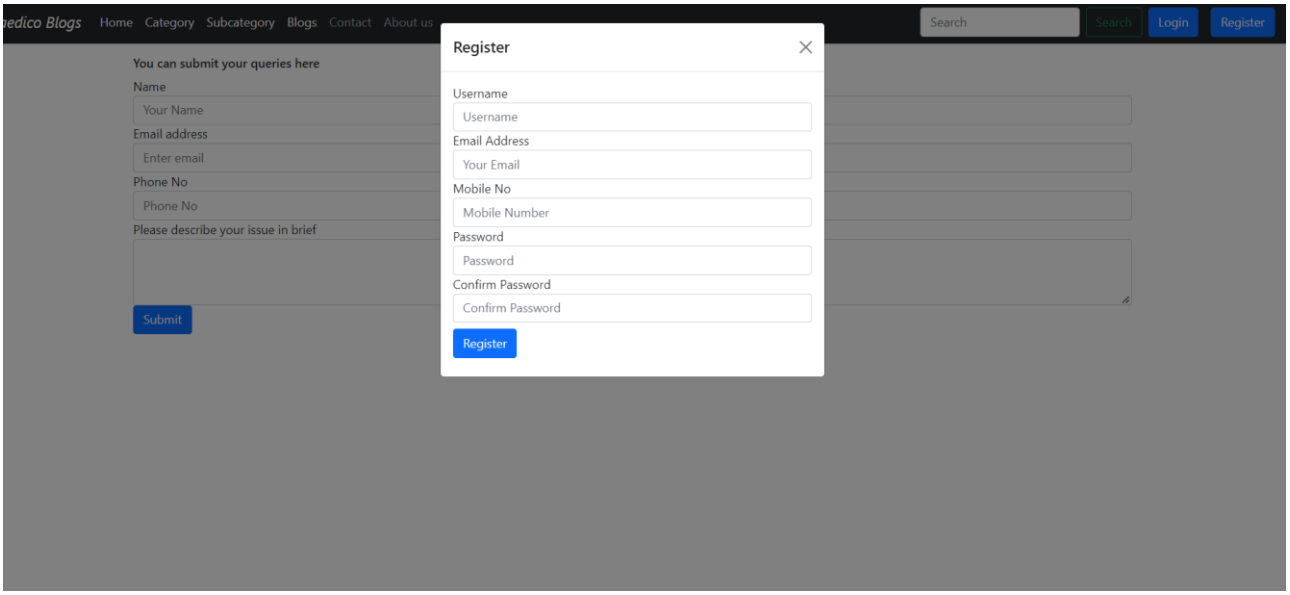
User Login page



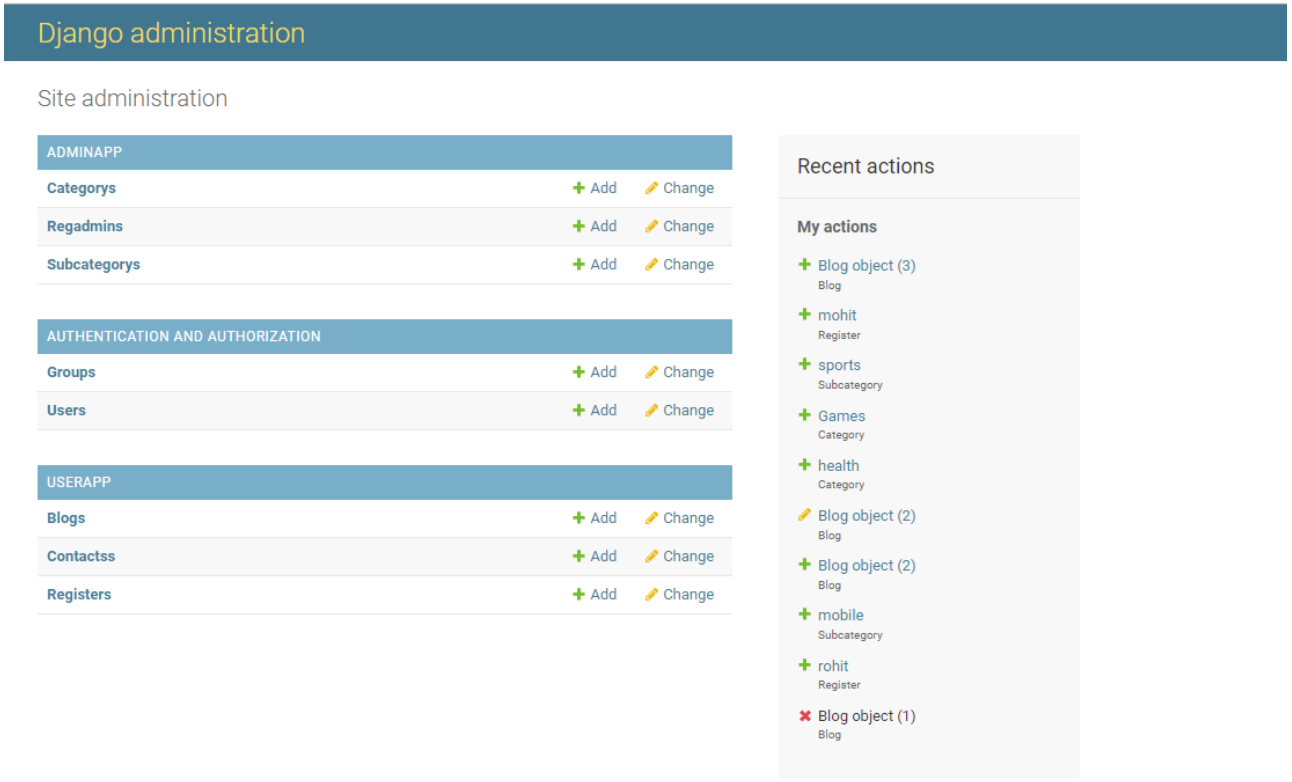
User Contact page



User Reg. Page

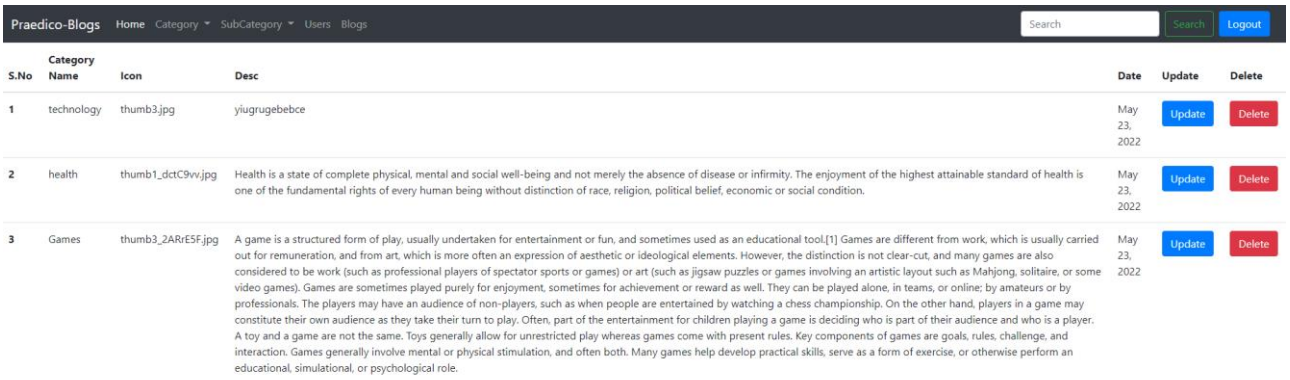


Admin panel

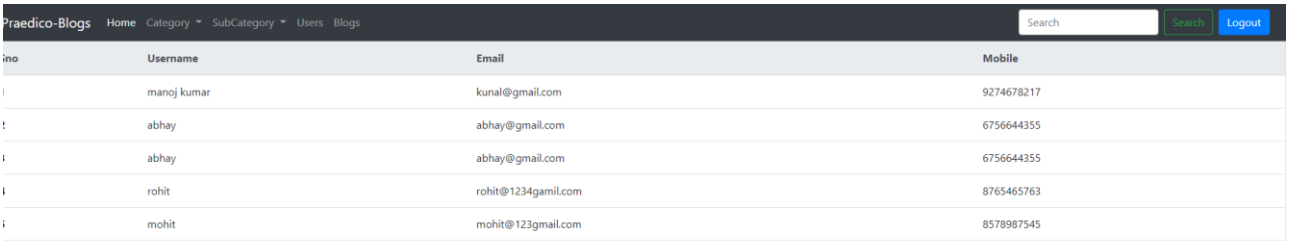


Admin login page

Admin Crud Operation



User show page



Conclusion:

Users can visit website, read posts,submit their queries,can see about website,search for blogs using special keywords,log in or register on website.
After login and register users can add,see, delete and update their own blog posts, and can also fill the contact form to submit their queries.
Admin after login on the website can add, update, delete categories and subcategories, can delete inappropriate blog posts, can see details of all the registered users.

FPR

FORMAT

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj Kumar		Department	CSE	
Industry/Organization	Praedico Global Research		Date/Duration	DD/MM/YR -DD/MM/YR 10-01-22 to 31-01-22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation		✓			
Performance/Quality of work		✓			
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work		✓			
Comment on nature of work done/Area/Topic	HTML CSS JS done				
OVERALL GRADE (Any one)	POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT				
Name of Industry Mentor	Pushkar Agagal				
Signature of Industry Mentor	Pushkar				

Receiving Date		**Name of Faculty Mentor**	Ranjeet Kumar Singh	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	MANOJ KUMAR		Department	CSE	
Industry/Organization	Paaedica Global Research		Date/Duration	01/02/22 to 15/02/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work			✓		
Comment on nature of work done/Area/Topic	HTML, SQL queries, Python				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT**				
Name of Industry Mentor	Pushkar Goyal				
Signature of Industry Mentor	Pushkar				

Receiving Date		**Name of Faculty Mentor**	Ranjeet Kumar Singh	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj Kumar		Department	CSE	
Industry/Organization	Praedico global Research		Date/Duration	16/02/22 to 1/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work			✓		
Comment on nature of work done/Area/Topic	bootstrap, oops Python				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT**				
Name of Industry Mentor	Kushkar Goyal				
Signature of Industry Mentor	Kushkar				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Date:

FORMAT

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj kumar		Department	CSE	
Industry/Organization	Praedico global Research		Date/Duration	02/03/22 to 16/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work			✓		
Behaviour/Discipline/Team work					✓
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Django, Data connectivity				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** ✓				
Name of Industry Mentor	Pushkar Goyal				
Signature of Industry Mentor	Pushkar				

Receiving Date		Name of Faculty Mentor		Sign	

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj kumar		Department	CSE	
Industry/Organization	Praedico global Research		Date/Duration	16/03/22 to 01/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work			✓		
Behaviour/Discipline/Team work					✓
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Blogging project				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT**				
Name of Industry Mentor	Rishkar Goyal				
Signature of Industry Mentor	Rishkar				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Date:

FORMAT

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj Kumar		Department	CSE	
Industry/Organization	Praedico Global Research		Date/Duration	02/04/022 to 15/04/022	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work					✓
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	Blogging project				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** ✓				
Name of Industry Mentor	Pushkar Goyal				
Signature of Industry Mentor	Pushkar				

Receiving Date		**Name of Faculty Mentor**		**Sign**	

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Date:

FORMAT

FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Manoj Kumar		Department	016/04/022 to 01/05/022	
Industry/Organization			Date/Duration		
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Blogging Project				
OVERALL GRADE (Any one)	**POOR/AVERAGE/GOOD/VERY GOOD/EXCELLENT** ✓				
Name of Industry Mentor	Pushkar Goyal				
Signature of Industry Mentor	Pushkar				

Receiving Date		Name of Faculty Mentor		Sign	