

CAMPUS RECRUITMENT SYSTEM

By

ANUGRAH P

[Register no: 21352014]

Project Report

Submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Under the guidance of

DR. S.L. JAYALAKSHMI

[ASSISTANT PROFESSOR]



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING & TECHNOLOGY

PONDICHERRY UNIVERSITY PUDUCHERRY-605014

NOVEMBER 2022

BONAFIDE CERTIFICATE

This is to certify that this project work entitled "**Campus recruitment system**" is the Bonafide record of work done by Mr. ANUGRAH P(Reg. No: 21352071) in partial fulfilment for the degree of Master of Computer Application, Department of Computer Science, School of Engineering and Technology, Pondicherry University.

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

Signature of the guide

DR. S.L. JAYALAKSHMI
(Assistant Professor)
Department of Computer Science,
School of Engineering and
Technology, Pondicherry University,
Pondicherry - 605014

R.P SEENIVASAN
(Associate Professor),
Department of Computer Science,
School of Engineering and
Technology,
Pondicherry University,
Pondicherry - 605014

Submitted for the Viva-Voce Examination held on

30/11/2022

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

“No work can be completed successfully without the proper guidance & help of the trainer and other people” This project report is also a manifestation of the invaluable guidance, suggestions, support and encouragement extended by various people.

My sincere thanks to my internal guide **Dr. S.L. Jayalakshmi** for guiding me. Our special thanks to our university department head **Dr. S.L. Jayalakshmi** and our project coordinator **Mr. R.P. Seenivasan**, for providing us with this wonderful opportunity and for their valuable guidance and moral support. I am also grateful to **Dr. T. Chithralekha**, Professor of the Department of Computer Science and Dean. I am extremely thankful for sharing expertise, and encouragement extended to me. Their regular guidance gave us the direction to work during the project.

I thank all the members of the Department of Computer Science and my friends for the timely help, advice and suggestions, which contributed either directly or indirectly to the success of the project.

ABSTRACT

As there are hundreds of graduates passing every year from a college , it is a hassle for campus recruiters to select brilliant minds among them. The Campus Recruitment System is a tool for colleges to simplify the campus recruitment process. It also provides the list of suitable companies to the students, according to their educational qualification and their preferences. The system consists of student login, company login, placement officer login and admin login. The project is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details onto the system. The placement officer can add placement events and manage students. The admin can approve, connect with companies and can manage faulty accounts. The system also consists of a company login where various companies visiting the college can communicate with placement officers ,view a list of student profiles who are applied for the placement .

TABLE OF CONTENT

CHAPTER NO	TITLE PAGE	NO
	BONAFIDE CERTIFICATE	I
	ACKNOWLEDGEMENT	ii
	ABSTRACT	III
1	INTRODUCTION	6
	1.1. Background	6
	1.2. Significance	7
	1.3. Method used	7
	1.4. Limitation	7
2	PRELIMINARY	8
	2.1 Django framework	8
	2.1.2 View	8
	2.1.3 Template	8
	2.1.4 Python	9
3	PROJECT OVERVIEW	11
	3.1 An overview of the project	11
	3.2 Modules	12
	3.3 Data Flow Diagram (Overview)	13
	3.4. Data Flow Diagram(Diagrams)	14
	3.5 Sample Code	17
	3.6 Screenshots	21
4	CONCLUSION AND FUTURE SCOPE	

Chapter 1

INTRODUCTION

This project report will introduce how to build part of a campus recruitment system using the Django framework[1]. Django is an open source web application framework which is written in Python[2]. This campus recruitment system built using Django has four major components each of which has different functionality but similar architecture. In the project report I will demonstrate details of using Django to build one major component of this system: the group component, which is my major contribution to the whole system. Also the technique and process which is showed here can be applied to build the other three components in the campus recruitment system as well as other complex database-driven websites.

1.1. Background

The Campus Recruitment System is a tool for colleges to simplify the campus recruitment process. It also provides the list of suitable companies to the students, according to their educational qualification and their preferences. The system consists of student login, company login, placement officer login and admin login. The project is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details onto the system. The placement officer can add placement events and manage students. The admin can approve, connect with companies and can manage faulty accounts. The system also consists of a company login where

various companies visiting the college can communicate with placement officers ,view a list of student profiles who are applied for the placement .

1.2. Significance

The new Campus Recruitment system will reduce the workload of placement officers in a campus by automating most of his work .The system will also help students to get more accurate and efficient responses regarding campus placements. As the workload of placement officers reduces he/she can focus more on students and to help them get placed .

1.3. Method used

This system is built using the Django web application framework. Django was originally developed for the news-oriented site of the world company in Lawrence, Kansas[3]. It simplifies the development process of complex, data-base driven web applications like a news-oriented site. Its well-designed framework includes three major parts: model, view and template[4]. Our course management system consists of four components which are grades, marking, group and submission. Each component contains those three parts. When we develop the course management system, we first design the model of the relative component for data architecture, then the template for user interface, and at last we implement the view which includes all the functions.

1.4. Limitation

Complex websites such as the Campus Recruitment System usually take some time to test and validate. The system may have some potential bugs or flaws because of the development time constraint. However, because of the flexibility and powerful functionality of Django, these bugs or flaws can be fixed.

Chapter 2

PRELIMINARY

2.1 Django framework

Django is an open source web application framework written in Python. The primary goal of Django is to make the development of complex, data-based websites easier. Thus Django emphasises the reusability and pluggability of components to ensure rapid developments. Django consists of three major parts: model, view and template[4].

2.1.1 Model

Model[4] is a single, definitive data source which contains the essential field and behaviour of the data. Usually one model is one table in the database. Each attribute in the model represents a field of a table in the database. Django provides a set of automatically-generated database application programming interfaces (APIs) for the convenience of users.

2.1.2 View

View[4] is a short form of view file. It is a file containing Python function which takes web requests and returns web responses. A response can be HTML content or XML documents or a “404 error” and so on. The logic inside the view function can be arbitrary as long as it returns the desired response. To link the view function with a particular URL we need to use a structure called URLconf which maps URLs to view functions

2.1.3 Template

Django’s template[4] is a simple text file which can generate a text-based format like HTML and XML. The template contains variables and tags. Variables will be replaced by the result when the template is evaluated. Tags control the logic of the template. We also can modify the variables by using filters. For example, a lowercase filter can convert the variable from uppercase into lowercase.

2.2 Python

Python[2] is the language used to build the Django framework. It is a dynamic scripting language similar to Perl[5] and Ruby[6]. The principal author of Python is Guido van Rossum[7]. Python supports dynamic typing and has a garbage collector for automatic memory management. Another important feature of Python is dynamic name resolution which binds the names of functions and variables during execution[2].

Chapter 3

PROJECT OVERVIEW

3.1 An overview of the project

The system consists of student login, company login, placement officer login and admin login. The project is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details onto the system. The placement officer can add placement events and manage students. The admin can approve, connect with companies and can manage faulty accounts. The system also consists of a company login where various companies visiting the college can communicate with placement officers ,view a list of student profiles who are applied for the placement .

3.2 Modules

3.2.1 Admin

Manage Faculties : Admin can add, view placement officers

Approve or Reject Company : Admin can check the legitimacy of a Company and based on that he/she can approve or reject recruiters.

3.2.2 Placement officer

Post placement Events : Placement Officer can add, view placement events

View Students : Placement Officer can view list of students applied for the particular placement event

Publish Results : Placement officer will have to publish the result of placement events

3.2.3 Student

Register and update profile : Student will have to register add their marks, resume on the system

View and Apply for Placement Events

View Results

3.2.4 Company

View Placement Officer

View Students : Company can view student marks,profile, resumes etc

3.3 DATA FLOW DIAGRAM (DFD)

DFD are the most commonly used way of documenting the process of flow and required system. As their name suggests, they are a pictorial way of showing flow of data into, around the system. DFD was introduced by Demacro, Gane and Sarson. Data Flow Diagrams are constructed with four major components. They are:

Data Flow Diagram Symbols

3.3.1. Entities



External entities represent the sources of data that enter the system or the recipients of data that leave the system

3.3.2. Data Store



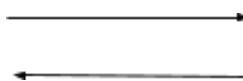
Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputs flow out of a data store and then through a process.

3.3.3. Process



Processes represent activity in which data is manipulated by being stored or retrieved or transformed in some way. A circle or oval represents it.

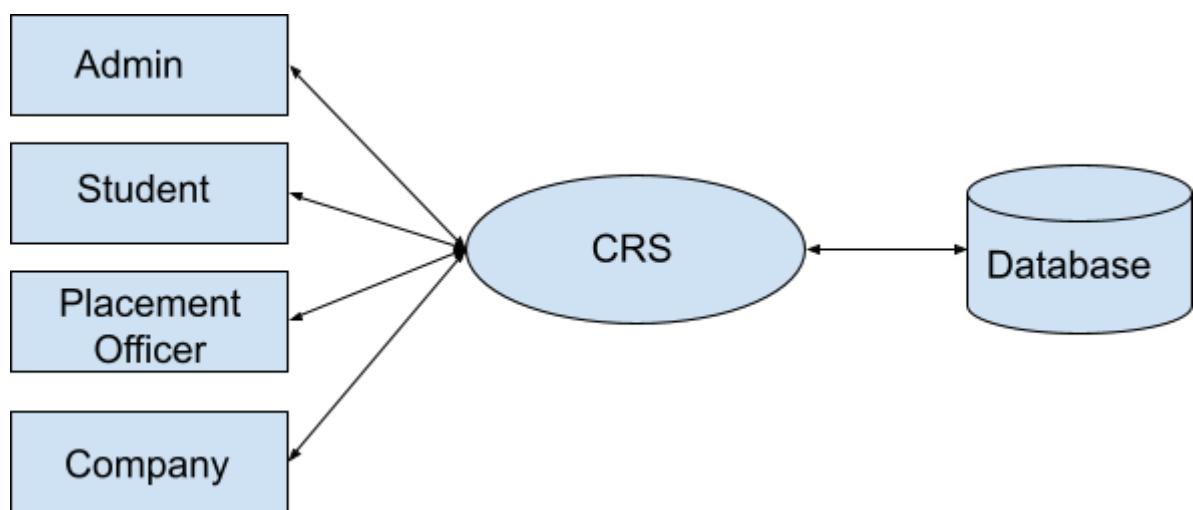
3.3.4. Data Flows



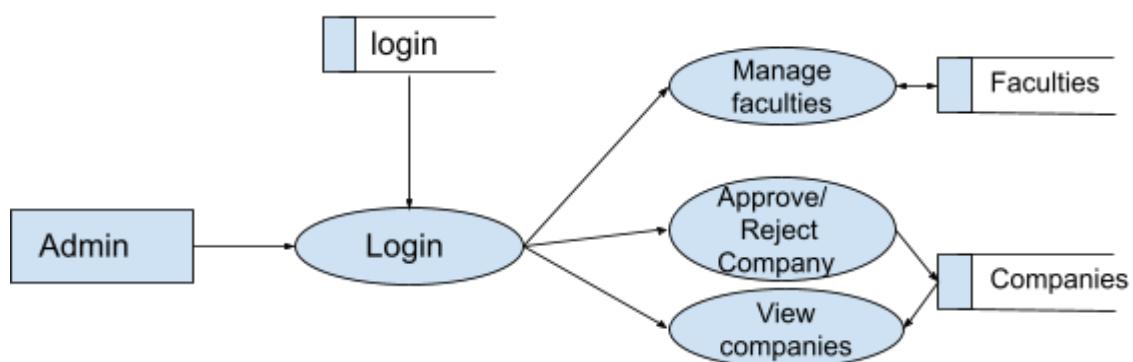
Data flow shows the flow of information from its source to its destination's line represents dataflow, with arrowheads showing the direction of flow

3.4. Data Flow Diagram

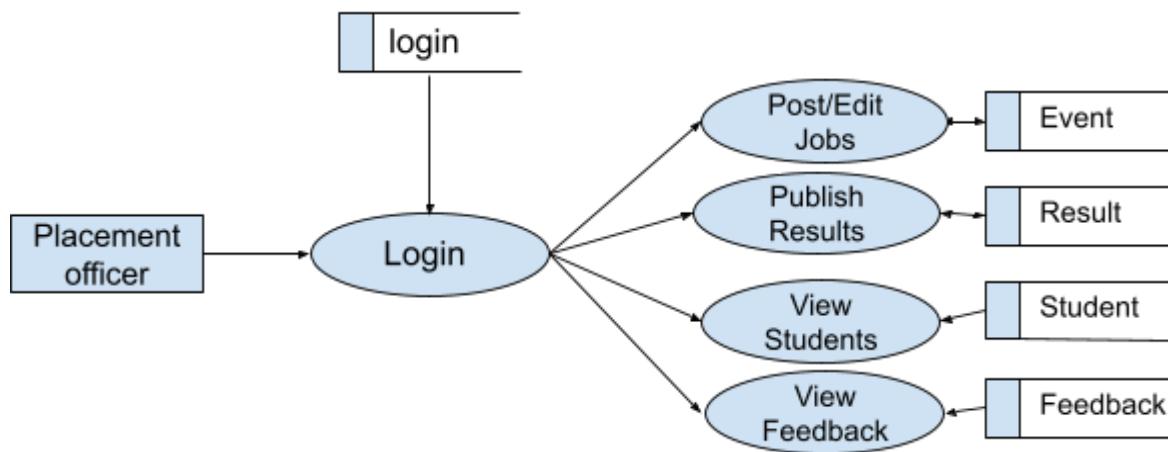
3.4.1 Level 0 :



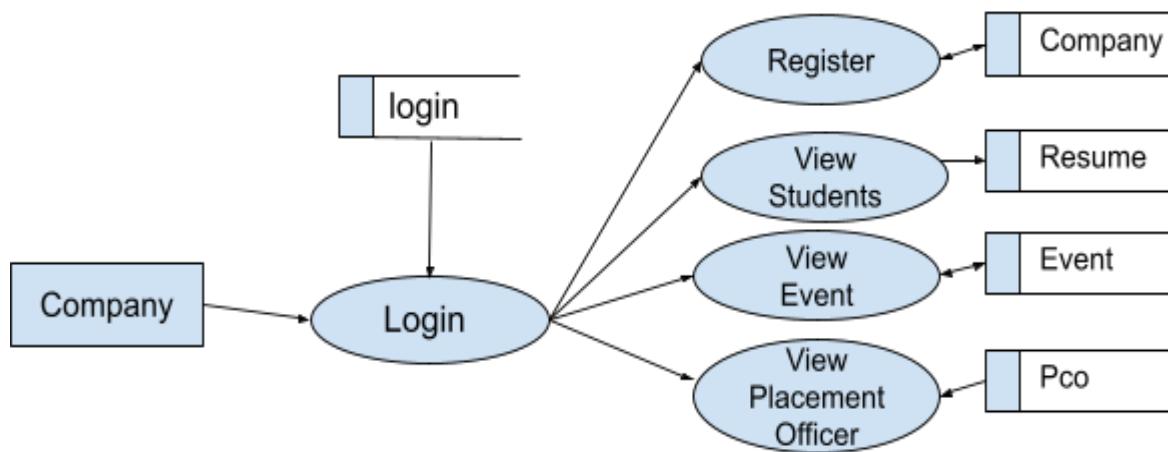
3.4.2. Admin



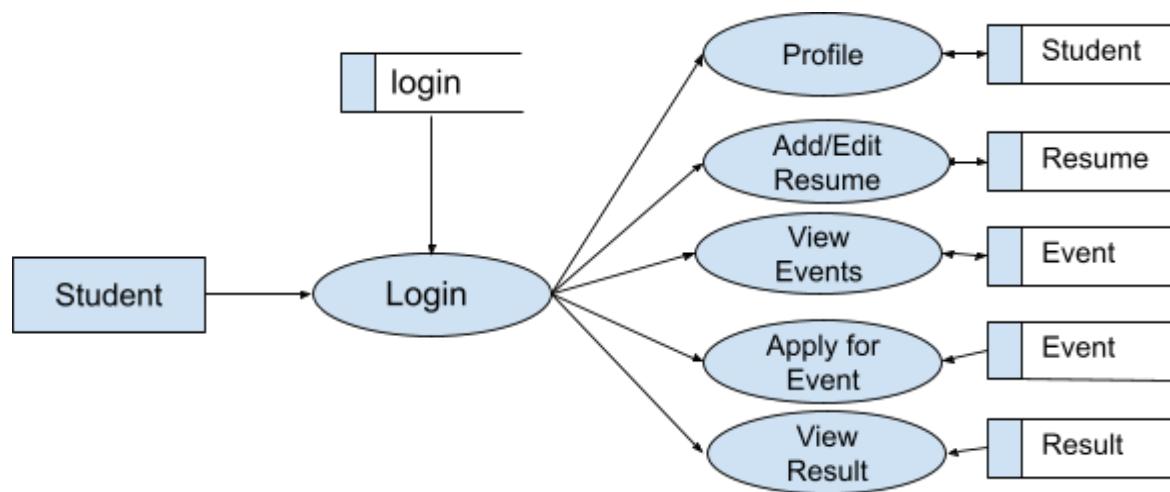
3.4.3. Placement Officer



3.4.4. Company



3.4.5. Student



3.5 Sample Code

Views.py

```
def register(request):
    new_var = request.POST or None
    form = regform(new_var)

    if request.method == 'POST':
        if form.is_valid():
            name = form.cleaned_data.get("name")
            mail = form.cleaned_data.get("email")
            passw = make_password(form.cleaned_data.get("Password"))
            comp_name = form.cleaned_data.get("comp_name")
            location = form.cleaned_data.get("location")
            description = form.cleaned_data.get("description")
            Phone = form.cleaned_data.get("Phone")
            url = form.cleaned_data.get("email")
            linkedin = form.cleaned_data.get("linkedin")
            if User.objects.filter(username=mail).count():
                messages.error(request, f'Account details allready
exists')
            else:
                user = User(username=mail,password=passw)
                user.save()
                id = User.objects.get(username=mail)
                comp =
company(name=name,comp_name=comp_name,location=location,description=des
cription,mobile=Phone,mail=mail,url=url,linkedin=linkedin,login_id=id)
                comp.save()
                g = Group.objects.get(name='pending')
                users = User.objects.get(username=mail)
                g.user_set.add(users)
                messages.success(request, f'Account Details Added ')
            return redirect("my-hom")
    return render(request, 'company/register.html',{'form':form})
@login_required
@allowed_users(allowed_roles=['comp'])
def CmpHome(request):
    return render(request, 'company/cmphome.html')
```

```

def viewStudentcom(request):
    user = company.objects.get(login_id = request.user)
    events = event.objects.filter(comp=user)
    value ={
        "value" : events
    }
    context = {

}

if request.method == "POST":

    event_id = request.POST['ename']

    ename = event.objects.get(id=event_id)
    con = eventcon.objects.filter(event_id = ename)

    data = con.values('student_id')

    stu = []

    for row in data:

        stu.append(student.objects.get(id = row['student_id']))
        print(student.objects.get(id = row['student_id']).id)

    context = {

        'myval' : stu
    }
    print(context['myval'])

return render(request, 'company/viewstu.html',context)

```

```

        return render(request, 'company/viewstu.html',value )

@login_required
@allowed_users(allowed_roles=['pending'])
def pendingCom(request):
    return render(request, 'company/pending.html')

@login_required
@allowed_users(allowed_roles=['comp'])
def comstats(request):
    return render(request, 'company/stats.html')

@login_required
@allowed_users(allowed_roles=['comp'])

def viewPcocom(request):

    pco = User.objects.filter(groups__name='pco')
    pco = pcoAdd.objects.all()
    context = {
        'pco': pco,
    }
    print(pco)
    print(context)
    return render(request, 'company/viewPcocom.html',context )

def home(request):
    events =event.objects.all()
    context = {
        'events': events,
    }
    return render(request, "faculty/home.html",context)

def stats(request):

    return render(request, 'faculty/stats.html')

*****Login*****
def login_page(request):

```

```

events =event.objects.all()
context = {
'events': events,
}
if request.method == 'POST':
    username = request.POST['name']
    password = request.POST['password']
    user = authenticate(request,
username=username,password=password)
    print(user)
    if user is not None:
        login(request, user)

    a=Group.objects.all()
    print(a)
    type = request.user.groups.all()
    for id in a:
        print(id)

    if type is not None:
        if(type[0] == a[0]):
            return redirect('pcHome')
        if(type[0] == a[1]):

            return redirect('cmphome')
        if(type[0] == a[2]):
            return redirect('stu_home')
        if(type[0] == a[3]):

            return redirect('crshome')
        if(type[0] == a[4]):
            return redirect('pendingCom')

    else:

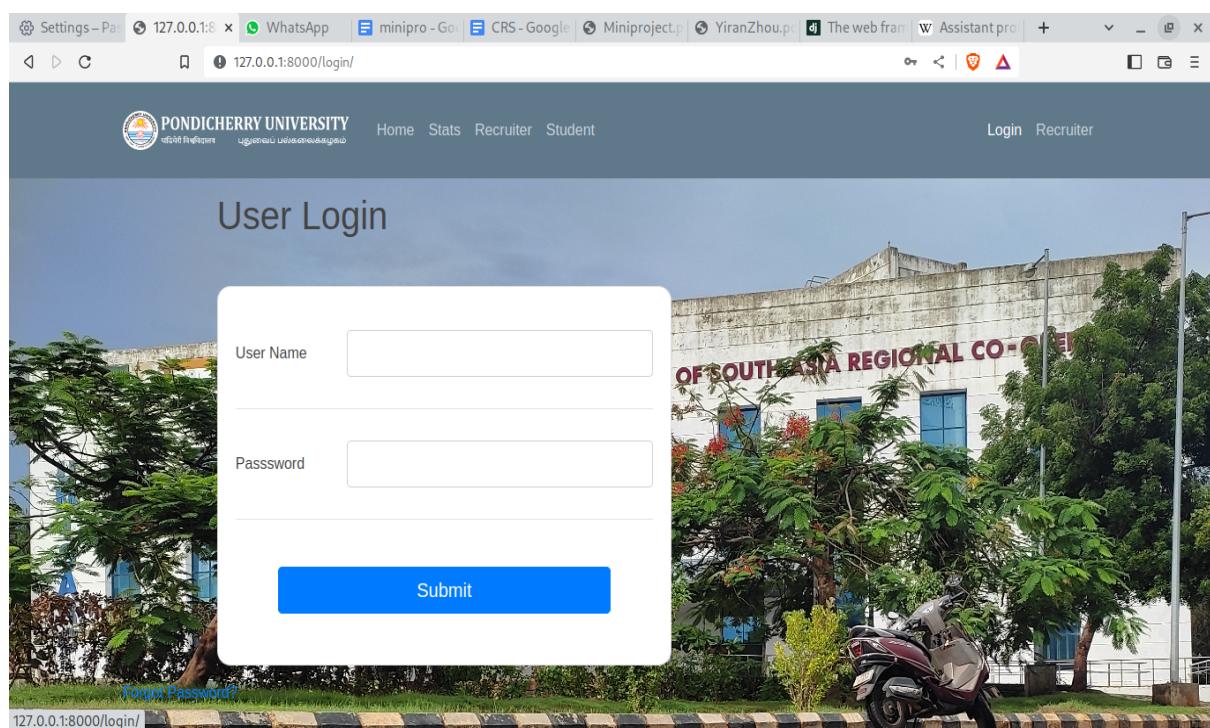
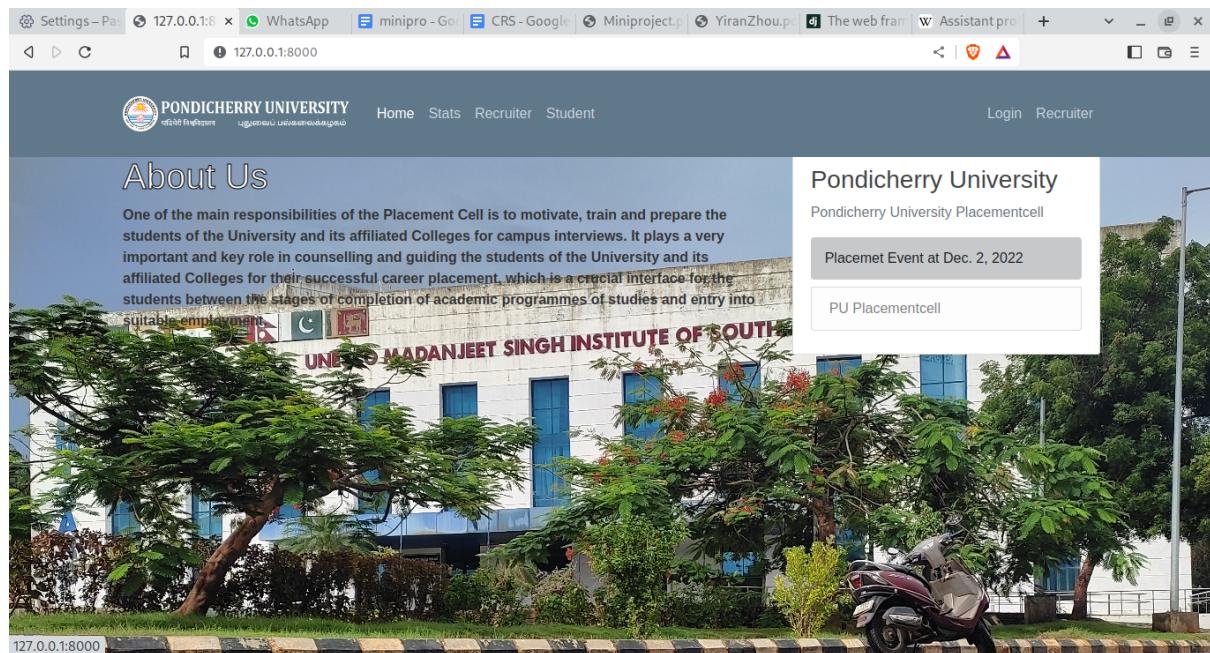
        return render(request, 'faculty/login.html')
else:
    messages.info(request,"User name or password is incorrect")

return render(request, 'faculty/login.html')

```

3.6 Screenshots

Home page & Login page



Company Registration & Statistics

Settings – Pa 127.0.0.1:8 WhatsApp minipro - Go CRS - Google Miniproject.p YiranZhou.p The web fram Assistant pro +

127.0.0.1:8000/comregister/

PONDICHERRY UNIVERSITY பாண்டிசெரி வினாக்கலாகம்

Home Stats Recruiter Student Login Recruiter

Recruter Registration

Name : Your Name

Company Name :

Company Location : eg : Bangalore,India

Job Description :

Email Id :

Phone Number :

Settings – Pa 127.0.0.1:8 WhatsApp minipro - Go CRS - Google Miniproject.p YiranZhou.p Statistics - Go Assistant pro +

127.0.0.1:8000/adminstats/

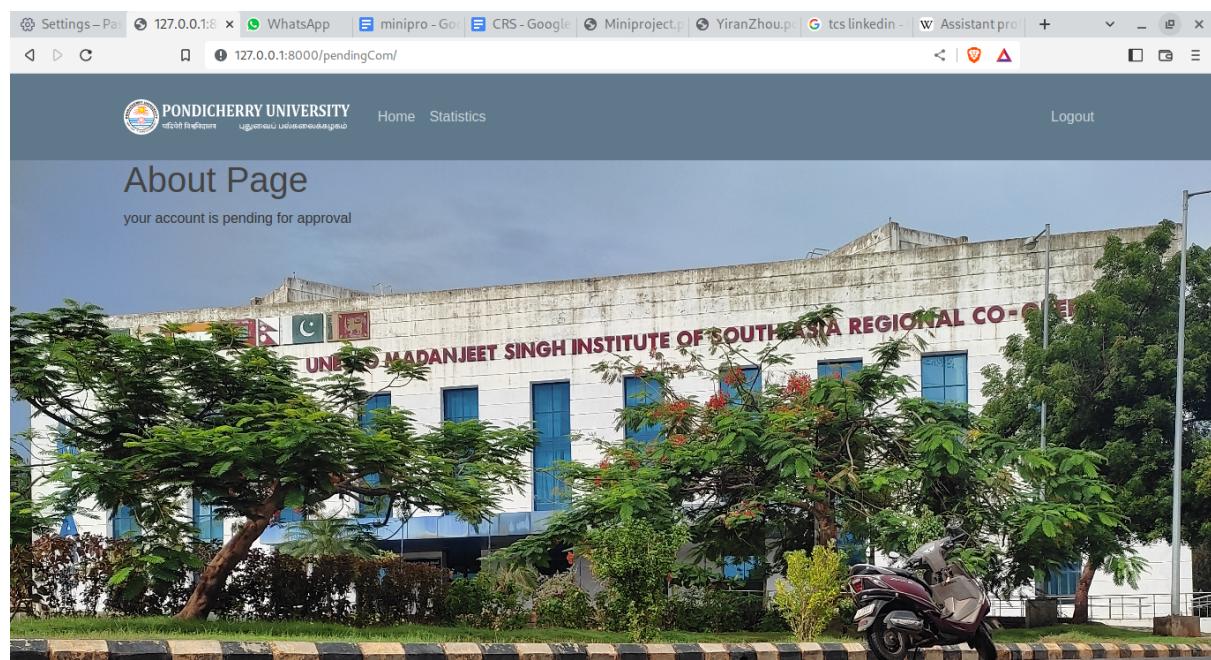
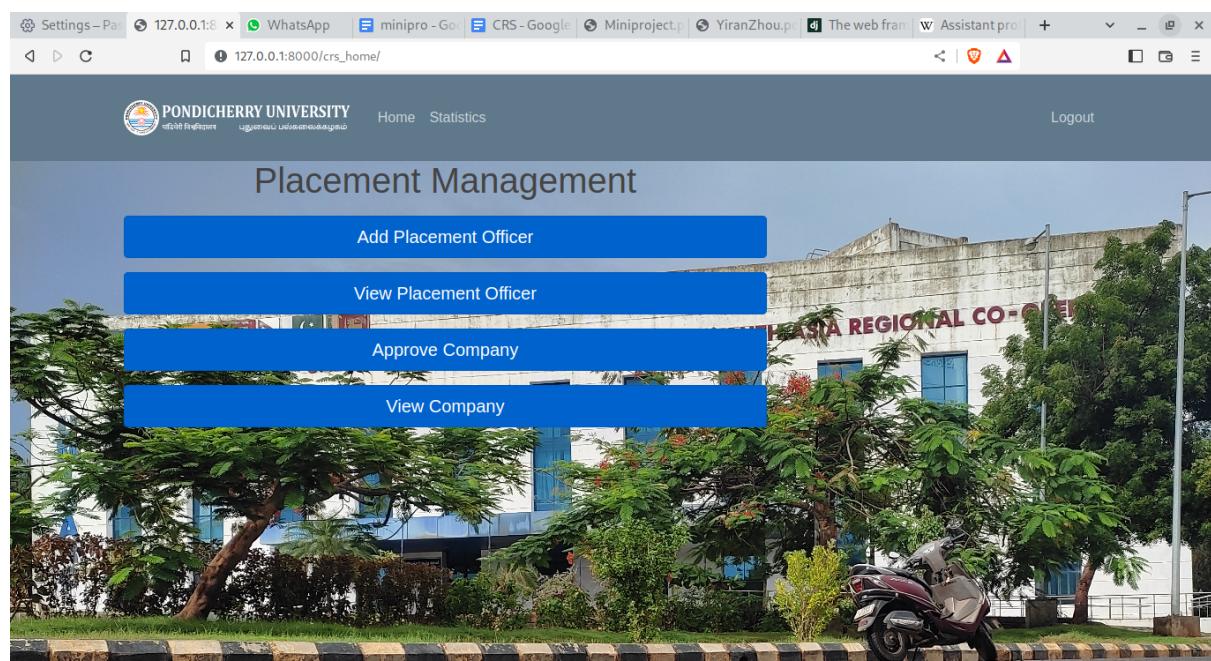
PONDICHERRY UNIVERSITY பாண்டிசெரி வினாக்கலாகம்

Home Statistics Logout

Statistics

Batch	No. of Companies	No. of Placements
2003-2004	9	24
2003-2004	9	24
2004-2005	28	68
2005-2006	55	357
2006-2007	56	806
2007-2008	63	1059
2008-2009	17	278

Admin Home & Company Home



New Tab 127.0.0.1:8000/

127.0.0.1:8000/approveCom/ 110% Logout

Most Visited Fedora Docs Fedora Magazine Fedora Project Telegram User Communities Red Hat Free Content Design/OS/KeyboardS... >>

PONDICHERRY UNIVERSITY பாடசாலை பல்கலைக்கழகம் Home Statistics

User Name	Company Name	Location	Description	Website	LinkedIn	Phone No	Mail id	Accept/Reject
Nawal	TCS	India	Developer	TCS	https://in.linkedin.com/company/tata-consultancy-services	8937372816	Nawal@mail.com	Approve Reject

New Tab 127.0.0.1:8000/ tata linkedin - Google Search

127.0.0.1:8000/viwecom/ 110% Logout

Most Visited Fedora Docs Fedora Magazine Fedora Project Telegram User Communities Red Hat Free Content Design/OS/KeyboardS... >>

PONDICHERRY UNIVERSITY பாடசாலை பல்கலைக்கழகம் Home Statistics

Active Companies

Company Name	Location	Description	Website	LinkedIn	Phone No	Mail id
TATA	India	Developer	TATA	https://www.linkedin.com/company/tata-group	8937372816	anugrahanu123@gmail.com

Chapter 3

CONCLUSION

The project titled as Campus Recruitment Management System was deeply studied and analysed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care of during the project time.

Campus Recruitment Management System is a platform that provides an interface between students and company. System provides the list of suitable companies to the students, according to their educational qualification and their preferences.