



**TEAM PLUSONE PRESENTS :**

# **Alumni Tracking System**

DR111 – Govt. of GOA, SIH 2020.

Day 1



## KEY DISCUSSION POINTS

- The Problem Statement
- Importance of Choosing the Right Technology
- User Case Diagrams
- Code Snippets
- Project Demo

# THE PROBLEM STATEMENT

The Directorate of Higher Education has 7 Government Colleges and 26 aided colleges under it. Altogether, more than 10000

students that pass out from them every year either choose to opt for further studies, work or have their own startups.

Currently, there is no mechanism for the colleges and the Directorate to keep a track of the students passing out.

Hence, the problem is to develop a web based application for colleges and the Directorate to keep a track of the Alumni.



## CHOOSING THE CORRECT TECH - STACK

# Django - The Webframework for Perfectionists with Deadlines

# WHY IS CHOOSING A TECH FRAMEWORK IMPORTANT?



STRUCTURED  
TEMPLATES



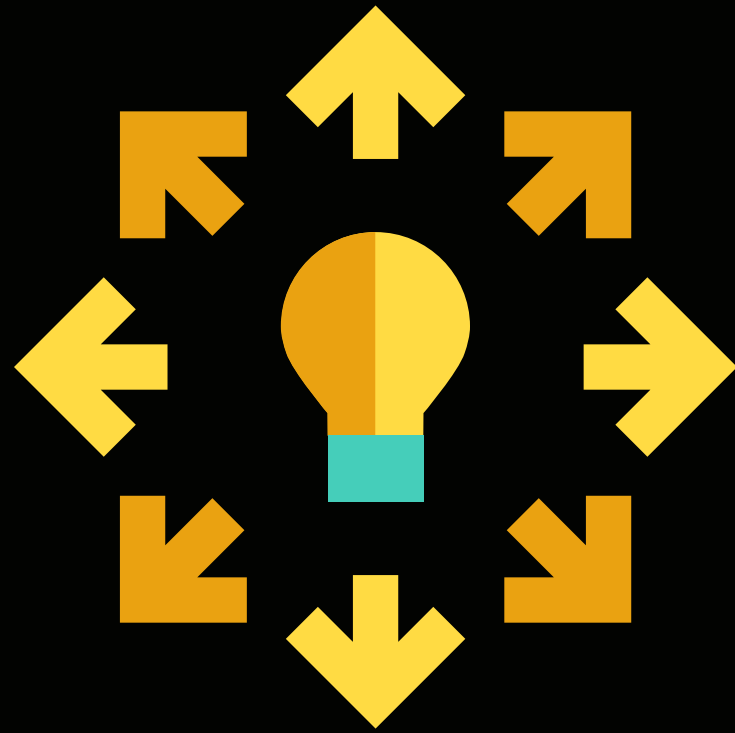
RELIABLE AND  
ACTIVE  
COMMUNITY



WELL  
DEFINED  
SECURITY  
FEATURES



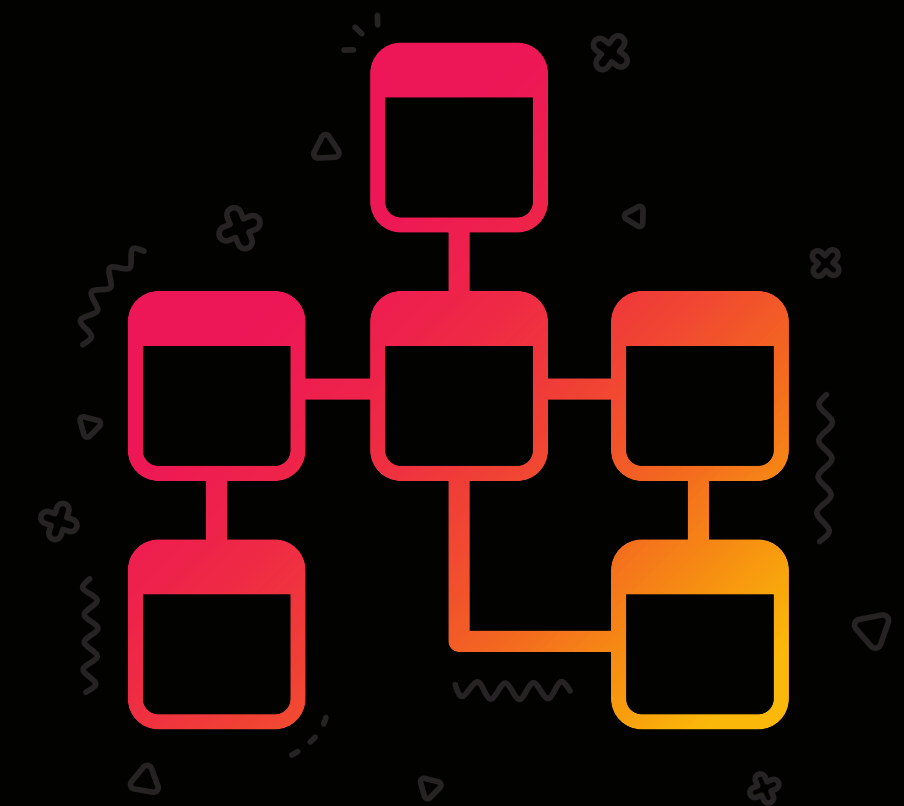
# WHAT DOES DJANGO PROVIDE?



**FAST & SCALABLE**



**EASY TO USE  
ADMIN PANEL**



**EASY TO DESIGN  
DATABASE  
SCHEMAS**



**TEMPLATE LANGUAGE FOR  
SEAMLESS CLIENT SIDE  
INTERACTION**

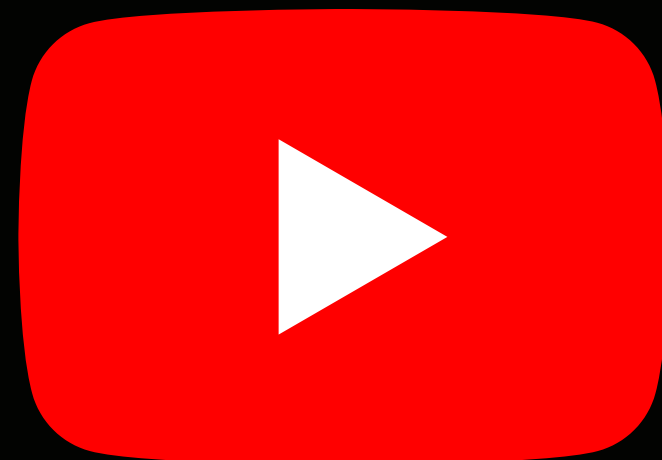
# POPULAR APPS USING DJANGO



Instagram



Dropbox



YouTube



Mozilla



Spotify

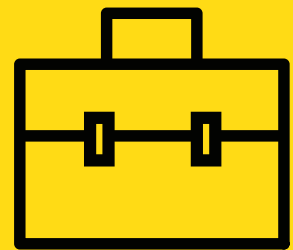


# DATABASE

## WHY POSTGRESQL?



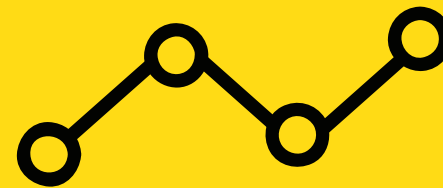
PostgreSQL



Safe and  
Secure with  
Client-Server  
Model



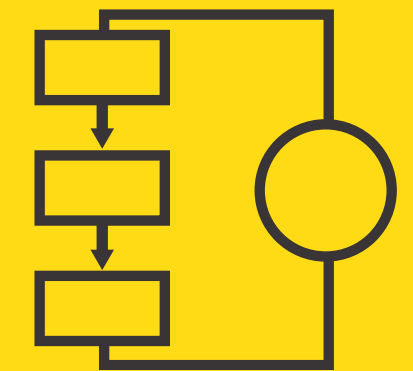
Very Popular in  
FOSS  
developer  
community



Fast and  
Robust

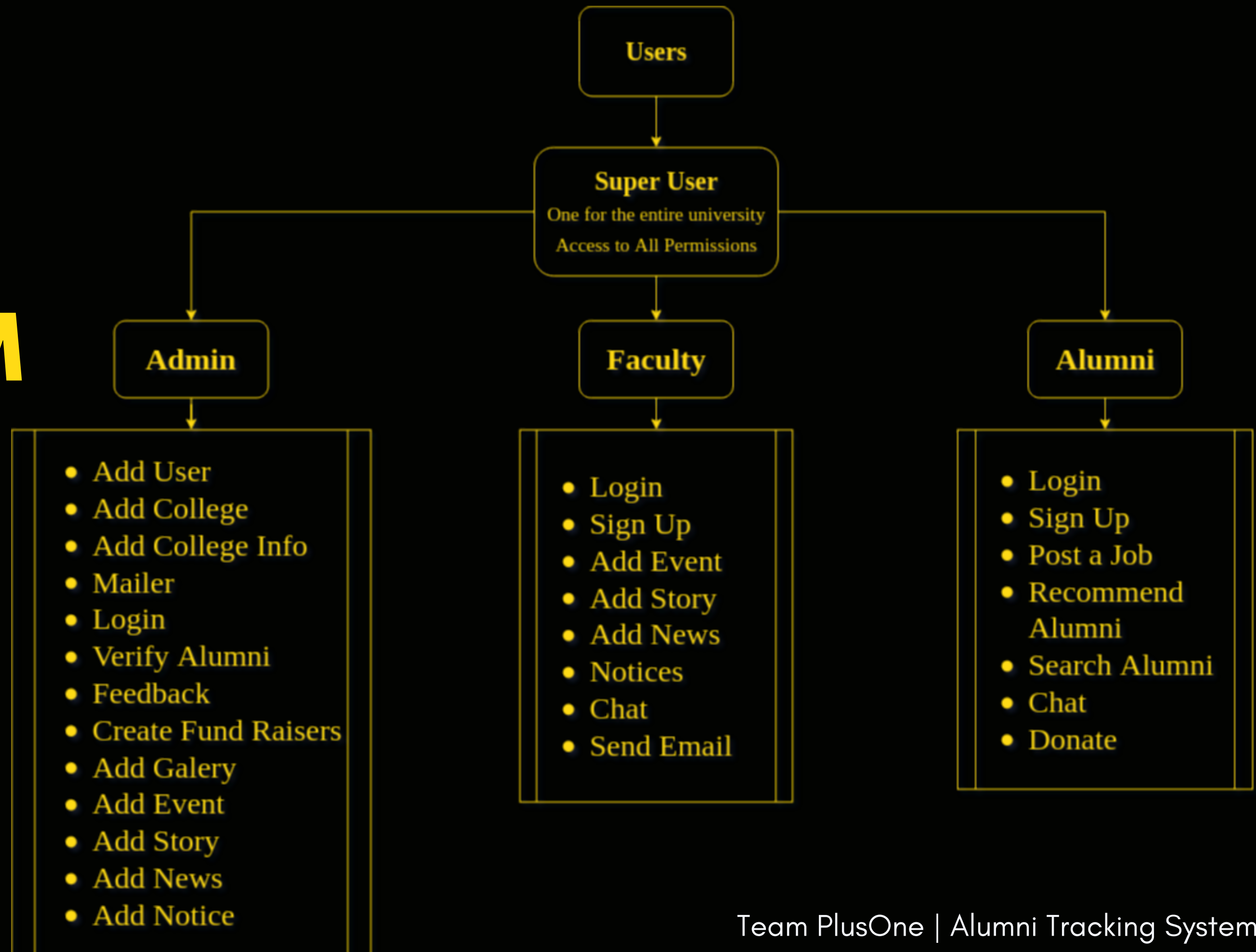


Cross Platform  
Integrability



Supports  
multiple types  
of data  
encoding

# THE USER CLASS DIAGRAM



```

from django.db import models
from django.contrib.auth.models import AbstractUser
from django.conf import settings
from college.models import College, Department, Course, Specialization
from django.db.models.signals import pre_save, post_delete
from django.utils.text import slugify

def upload_user_image_location(instance, filename):
    file_path = 'user/{user_id}/{filename}'.format(
        user_id=str(instance.id),
        filename=filename
    )
    return file_path

class Position(models.Model):
    position_name = models.CharField(max_length=200)

    def __str__(self):
        return self.position_name

class User(AbstractUser):
    is_verified = models.BooleanField(default=False)
    is_alumni = models.BooleanField(default=False)
    is_faculty = models.BooleanField(default=False)
    profile_photo = models.ImageField(upload_to=upload_user_image_location)
    profile_complete = models.BooleanField(default=0)
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    full_name = models.SlugField(editable=False)
    college = models.ForeignKey(College, on_delete=models.SET_NULL, null=True, blank=True)
    course = models.ForeignKey(Course, on_delete=models.SET_NULL, null=True, blank=True)
    department = models.ForeignKey(Department, on_delete=models.SET_NULL, null=True, blank=True)
    specialization = models.ForeignKey(Specialization, on_delete=models.SET_NULL, null=True, blank=True)
    dob = models.DateField(null=True, blank=True)
    system_date_joined = models.DateTimeField(verbose_name="Date Joined", auto_now=True)
    system_last_login = models.DateTimeField(verbose_name="Last Login", auto_now=True)
    email = models.EmailField(null=True, unique=True)
    facebook_profile = models.URLField(max_length=1000, null=True, blank=True)
    twitter_profile = models.URLField(max_length=1000, null=True, blank=True)
    linkedin_profile = models.URLField(max_length=1000, null=True, blank=True)
    location = models.CharField(max_length=200, null=True, blank=True)
    position = models.ForeignKey(Position, on_delete=models.SET_NULL, null=True, blank=True)
    phone = models.CharField(max_length=20, null=True, blank=True)
    about_me = models.TextField(null=True, blank=True)

```



```
class Alumni(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    year_of_passing = models.IntegerField(null=True)
    unique_id = models.CharField(unique=True, max_length=200)
    profile_verified = models.BooleanField(default=0)
    company = models.CharField(max_length=200, null=True, blank=True)
    title = models.CharField(max_length=200, null=True, blank=True)
    address = models.CharField(max_length=200, null=True, blank=True)
    resume = models.URLField(max_length=1000, null=True, blank=True)

    def __str__(self):
        return (
            self.user.full_name + " " + self.unique_id
        )

class Faculty(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    college_joined_year = models.IntegerField(null=True)
    research_interest = models.CharField(max_length=300, null=True)
    unique_id = models.CharField(unique=True, max_length=200)
    is_admin = models.BooleanField(default=False)
    is_active = models.BooleanField(default=True)
    is_staff = models.BooleanField(default=True)
    is_superuser = models.BooleanField(default=False)

    def __str__(self):
        return (
            self.user.full_name
            + ", "
            + self.user.department.name
            + ", "
            + self.user.college.name
        )

def pre_save_User(sender, instance, *args, **kwargs):
    if not instance.full_name:
        instance.full_name = slugify(instance.first_name + instance.last_name)

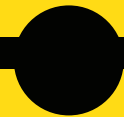
pre_save.connect(pre_save_User, sender=User)
```

# CRITICAL POINTS

- **FAST AND SCALABLE**
- **EASY TO USE**
- **UNIQUE ALUMNI RECOMMENDATION SYSTEM**
- **BENEFACTION SYSTEM**
- **SIMPLE AND SECURE**
- **BUILT IN MAILING PROTOCOL**

# THE ROLL OUT

## PROPOSED TIMELINE



### DAY 1

Database Schema  
Admin Page  
Home Page



### DAY 2

Integrated Mailer  
Fund Raiser  
Real Time Chat



### DAY 3

U I changes and  
integration of other  
suggestions.

# THANK YOU !!

## Team Members:

- Mohit Agarwal (Leader)
- Rajas Ajay Kulkarni
- Himanshu Shekhar Jha
- Chirag Agarwal
- Simran Singh
- Rahul Lodha