CS/VT 4990: Advanced Web Development with Django

Assignment: Business CRM

Objectives

Assume you work for ACME, Inc. and are tasked to build an in-house CRM tool for the sales team to track their sales pipeline.

A CRM stands for Customer Relationship Management. CRM software is used by many businesses in many industries to manage customer data and sales workflow. Popular CRMs include Salesforce.com, SugarCRM, and High Rise.

As a final capstone project for the course, we’ll build a business CRM. Up until now, most of our assignments have had up to two or three models and four or five views. This business CRM will have many more models and views.

In this project we’ll be deploying the site to a real server backed by a real database.

Requirements

- A Business CRM has the following models (as developed in class):

```python
class Stage(models.Model):
    name = models.CharField(max_length = 200)
    order = models.IntegerField(help_text = 'The order this is displayed on the screen')
    description = models.TextField(blank = True, null = True)
    value = models.IntegerField(help_text = 'On a scale of 0 to 100 of the stage of the pipeline')

class Company(models.Model):
    name = models.CharField(max_length = 200)
    website = models.URLField(max_length = 200, blank = True, null = True)
    address1 = models.CharField(max_length = 200, blank = True, null = True)
    address2 = models.CharField(max_length = 200, blank = True, null = True)
    city = models.CharField(max_length = 200, blank = True, null = True)
    state = models.CharField(max_length = 200, blank = True, null = True)
    zipcode = models.CharField(max_length = 200, blank = True, null = True)
    country = models.CharField(max_length = 200, blank = True, null = True)
    phone = models.CharField(max_length = 200, blank = True, null = True)

class Contact(models.Model):
    company = models.ForeignKey(Company, blank = True, null = True)
    first_name = models.CharField(max_length = 200)
    last_name = models.CharField(max_length = 200)
    address1 = models.CharField(max_length = 200, blank = True, null = True)
    address2 = models.CharField(max_length = 200, blank = True, null = True)
    city = models.CharField(max_length = 200, blank = True, null = True)
    state = models.CharField(max_length = 200, blank = True, null = True)
    zipcode = models.CharField(max_length = 200, blank = True, null = True)
    country = models.CharField(max_length = 200, blank = True, null = True)
    phone = models.CharField(max_length = 200, blank = True, null = True)
    email = models.EmailField(max_length = 200, blank = True, null = True)

class Campaign(models.Model):
    name = models.CharField(max_length = 200)
    description = models.TextField(blank = True, null = True)

```
user = models.ForeignKey(User)
create_date = models.DateTimeField(auto_now_add=True)
class Reminder(models.Model):
opportunity = models.ForeignKey(Opportunity)
date = models.DateField()
note = models.CharField(max_length = 200)
completed = models.BooleanField(default=False)
class Report(models.Model):
name = models.CharField(max_length = 200)
link = models.URLField()
class CallLog(models.Model):
opportunity = models.ForeignKey(Opportunity)
date = models.DateTimeField(auto_now_add=True)
note = models.TextField()
user = models.ForeignKey(User)
class OpportunityStage(models.Model):
opportunity = models.ForeignKey(Opportunity)
stage = models.ForeignKey(Stage)
timestamp = models.DateTimeField(auto_now_add=True)
user = models.ForeignKey(User)

- Views to List, See the details of, Create, Update, and Delete the following:
  - Log A Call
  - Manage Stage (examples of stages are Lead, Proposal Sent, Proposal Accepted, Implementation, Won, Lost, Contact Later)
  - Company
  - Contacts
  - Reminders
  - Opportunities
  - Campaigns
- Record the user and timestamp when an opportunity moves from one stage to another.
- Show a log of the stages for an opportunity
- On the home page, show a dashboard highlighting upcoming reminders, top opportunities, etc.
- Include a search for all opportunities, contacts, and notes
- Host the site on a real server (such as digitalocean.com). Purchase a domain name to point your site to.
  Deploy the site using nginx on the front end to serve static files, mod_wsgi for the Django application, and MySQL or PostgreSQL for the back-end database.

**Extra Credit Ideas**

- Require a login on all views that relate to holding customer data

**Hints**