**Creating a Django App**

Open Gitbash to source folder

To create virtual environment:

$ vagrant ssh

cd /vagrant/

python -m venv ~/env

source ~/env/bin/activate

In virtual environment type:

pip install -r requirements.txt

django-admin.py startproject profiles\_project .

python manage.py startapp profiles\_api

python manage.py runserver 0.0.0.0:8000

In local project Gitbash (not virtual environment):

git add .

git commit -am "Created django project and app"

$ git push origin

**Setting up the Database, Add a User Model Manager**

User Profile Model

Not using Django standard which doesn’t allow a login as an email

from django.db import models

from django.contrib.auth.models import AbstractBaseUser

from django.contrib.auth.models import PermissionsMixin

from django.contrib.auth.models import BaseUserManager

class UserProfileManager(BaseUserManager):

"""Manager for user profiles"""

def create\_user(self, email, name, password=None):

"""Create a new user profile"""

if not email:

raise ValueError('User must have an email address')

email = self.normalize\_email(email)

user = self.model(email=email, name=name)

user.set\_password(password)

user.save(using=self.\_db)

return user

def create\_superuser(self, email, name, password):

"""Create and save a new superuser with given details"""

user = self.create\_user(email, name, password)

user.is\_superuser=True

user.is\_staff = True

user.save(using=self.\_db)

return user

class UserProfile(AbstractBaseUser, PermissionsMixin):

"""Database model for users in the system"""

email = models.EmailField(max\_length=255, unique=True)

name = models.CharField(max\_length=255)

is\_active = models.BooleanField(default=True)

is\_staff = models.BooleanField(default=False)

objects = UserProfileManager()

USERNAME\_FIELD = 'email'

REQUIRED\_FIELDS = ['name']

def get\_full\_name(self):

"""Retrieve full name of user"""

return self.name

def get\_short\_name(self):

"""Retrieve short name of user"""

return self.name

def\_\_str\_\_(self):

"""Return string representation of our user"""

return self.email

**Set our custom user model**

In Oracle, go to ‘profile\_project’ folder and then ‘settings.py’ subfolder and add new line at bottom of standard script:

AUTH\_USER\_MODEL = 'profiles\_api.UserProfile'

**Create migrations and sync Database**

Open up Gitbash in local folder:

Type:

vagrant ssh

cd /vagrant

source ~/env/bin/activate

python manage.py makemigrations profiles\_api

then

python manage.py migrate

Then open a new Gitbash terminal and type:

git add .

git commit -am "Added custom user profile model, manager and migrations"

git push origin

**Set up Django admin**

**Creating a SuperUser**

Open Gitbash

Type:

vagrant ssh

cd /vagrant

source ~/env/bin/activate

python manage.py createsuperuser

enter email passwords etc