**Python Dajngo Wednesday**

**EP.1  
ลง Python, Virtualenv, สร้าง Project**

ลง Virtualenv pip install virtualenv

สร้างโฟล์เดอร์ mkdir myclub

เข้าไปในโฟลเดอร์ cd myclub

สร้าง venv python -m venv virt

เข้าไปใน venv virt\Scripts\activate

source virt/bin/activate (mac)

ลง Django pip install django

สร้าง Project django-admin startproject myclub\_website

เข้าโฟลเดอร์ mywebsite cd myclub\_website

รันเซิฟเวอร์ python manage.py runserver

เข้าหน้าเวปโดยใช้ <http://localhost:8000/>

สร้าง App python manage.py startapp events

ลงทะเบียน App >>>> settings

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.sessions',

    'events',

]

ตั้งค่า urls ของแอปให้ urls ของ Project รู้จัก

from django.contrib import admin

from django.urls import path , include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('events.urls'))

]

สร้าง urls.py ของ app evnets

from django.urls import path

from events import views

urlpatterns = [

    path('', views.home, name="home"),

]

views.py

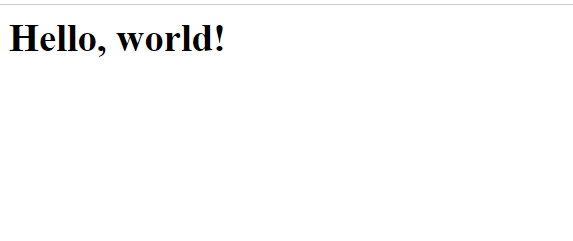
from django.shortcuts import render

def home(request):

    return render(request, 'home.html', {})

สร้างโฟล์เดอร์ templates/home.html

    <h1>Hello, world!</h1>



**EP.2  
Add A Calendar To Your Website**

View.py

from django.shortcuts import render

import calendar

from calendar import HTMLCalendar

from datetime import datetime

def home(request, year=datetime.now().year,

month=datetime.now().strftime('%B')):

    name = "John"

    month = month.capitalize()

    age = 47

    # convert month from name to number

    month\_number = list(calendar.month\_name).index(month)

    month\_number = int(month\_number)

    # create a calendar

    cal = HTMLCalendar().formatmonth(year, month\_number)

    now = datetime.now()

    current\_year = now.year

    time = now.strftime("%I:%M:S %p")

    return render(request, 'home.html', {

        "name":name,

        "year":year,

        "month":month,

        "month\_number":month\_number,

        "cal":cal,

        "current\_year":current\_year,

        "time":time

        })

urls.py

urlpatterns = [

    # int:number

    # str:string

    # path:whole urls/

    # slug:hyphen-and\_underscores\_stuff

    # UUID: universally unique identifier

    path('', views.home, name="home"),

    path('<int:year>/<str:month>/', views.home, name="home"),

]

home.html

<center>

    <h1>Hello {{ name }}!  {{ month }}  {{ year }}</h1>

    As of {{ time }}

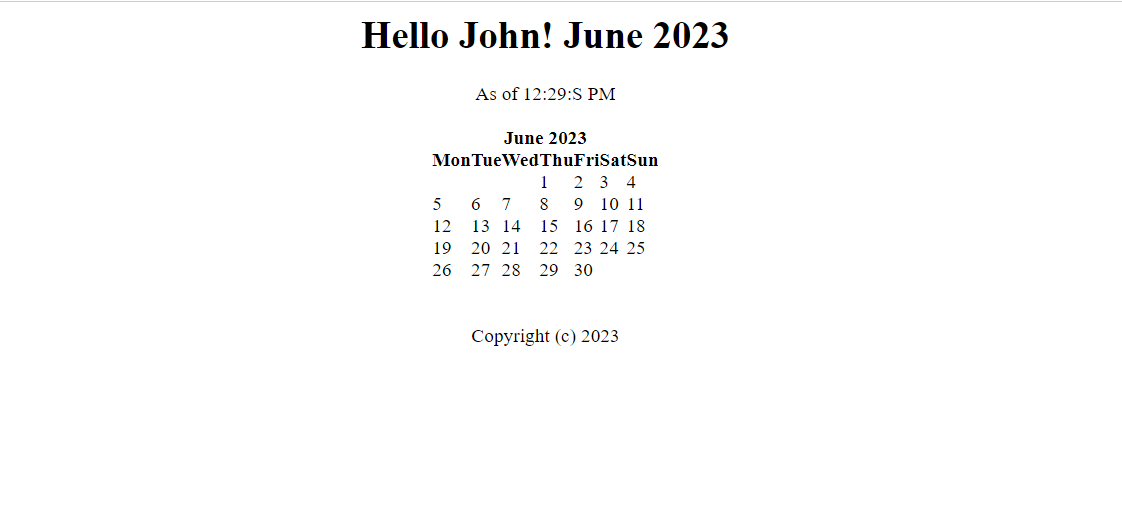
    <br><br>

    {{ cal|safe }}

    <br><br>

    Copyright (c) {{ current\_year}}

 </center>



**EP.3  
How To Use Templates and Custom HTML**

ที่ templates /events/home.html

Views.py

return render(request, 'events/home.html', {

ที่ templates /events/base.html

HEADER STUFF

<br><br>

{% block content %}

{% endblock content %}

<br><br>

FOOTER STUFF

ที่ templates /events/home.html

{% extends 'events/base.html' %}

{% block content %}

    <center>

        <h1>Hello World !!! {{name}}</h1>

        As of {{ time }}

        <br><br>

        {{ cal|safe }}

        <br><br>

        Copyright (c) {{ current\_year}}

    </center>

{% endblock content %}

ไปที่ bootstrap copy template มาวางที่ bast.html

ใส่ templates พร้อม navbar จาก bootstrap

events/navbar.html

<nav class="navbar navbar-dark bg-dark">

    <div class="container-fluid">

      <a class="navbar-brand" href="{% url 'home' %}">My Club</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">

        <ul class="navbar-nav me-auto mb-2 mb-lg-0">

          <li class="nav-item">

            <a class="nav-link" href="{% url 'home' '2023' 'March' %}">March</a>

          </li>

        </ul>

        <form class="d-flex" role="search">

          <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

          <button class="btn btn-outline-success" type="submit">Search</button>

        </form>

      </div>

    </div>

  </nav>

events/base.html

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Bootstrap demo</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

  </head>

  <body>

    {% include 'events/navbar.html' %}

    <br>

<div class="container">

    {% block content %}

            <h1>Hello, world!</h1>

    {% endblock content %}

</div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>

  </body>

</html>



**EP.4 Use Multiple Database Tables With django**

สร้างชื่อผู้ใช้ admin และ password

python manage.py migrate ,python manage.py createsuperuser

model.py

from django.db import models

class Venue(models.Model):

    name = models.CharField('Venue Name',max\_length=120)

    address = models.CharField(max\_length=300)

    zip\_code = models.CharField('Zip Code', max\_length=15)

    phone = models.CharField('Contact Phone', max\_length=25)

    web = models.URLField('Website Address')

    email\_address = models.EmailField('Email Address')

    def \_\_str\_\_(self):

        return self.name

class MyClubUser(models.Model):

    first\_name = models.CharField(max\_length=30)

    last\_name = models.CharField(max\_length=30)

    email = models.EmailField('User Email')

    def \_\_str\_\_(self):

        return self.first\_name + '  ' + self.last\_name

class Event(models.Model):

    name = models.CharField('Event Name',max\_length=120)

    event\_date = models.DateTimeField('Event Date')

*# venue = models.CharField(max\_length=120)*

    venue = models.ForeignKey(Venue, blank=True, null=True,

on\_delete=models.CASCADE)

    manager = models.CharField(max\_length=120)

    description = models.TextField(blank=True)

    attendees = models.ManyToManyField(MyClubUser, blank=True)

    def \_\_str\_\_(self):

        return self.name

admin.py

from django.contrib import admin

from .models import Venue

from .models import MyClubUser

from .models import Event

admin.site.register(Venue)

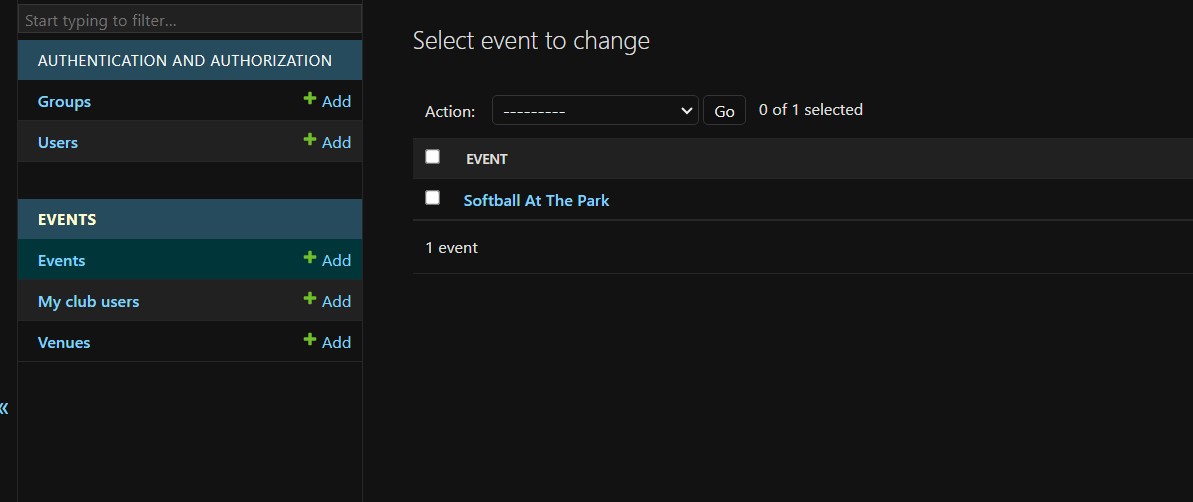
admin.site.register(MyClubUser)

admin.site.register(Event)

python manage.py makemigrations

python manage.py migrate

pythom mange.py runserver



**EP.5**

**Fetch Data From a Database And Output To A Webpage**

urls.py

urlpatterns = [

    path('', views.home, name="home"),

    path('<int:year>/<str:month>/', views.home, name="home"),

    path('events', views.all\_events, name='list-events'),

]

views.py

from .models import Event

def all\_events(request):

    event\_lsit = Event.objects.all()

    return render(request, 'events/event\_list.html', {

        'event\_list':event\_lsit})

Template/events/event\_list.html >>>> bootstrap/card

{% extends 'events/base.html' %}

{% block content %}

        <h1>Events !!!</h1>

       {% for event in event\_list %}

       <br>

       <div class="card">

        <div class="card-header">

          {{event}}

        </div>

        <div class="card-body">

          <h5 class="card-title">Venue :{{event.venue}}</h5>

          <p class="card-text">

            <ul>

                <li>Date : {{event.evnt\_date}}</li>

                <li>Venue Website : {{event.venue.web}}</li>

                <li>Manager : {{event.manager}}</li>

                <li>Discription : {{event.description}}</li>

                <li>Attendees : <br> {% for user in

event.attendees.all %}</li>

                {{user}}

                <br>

                {% endfor %}

            </ul>

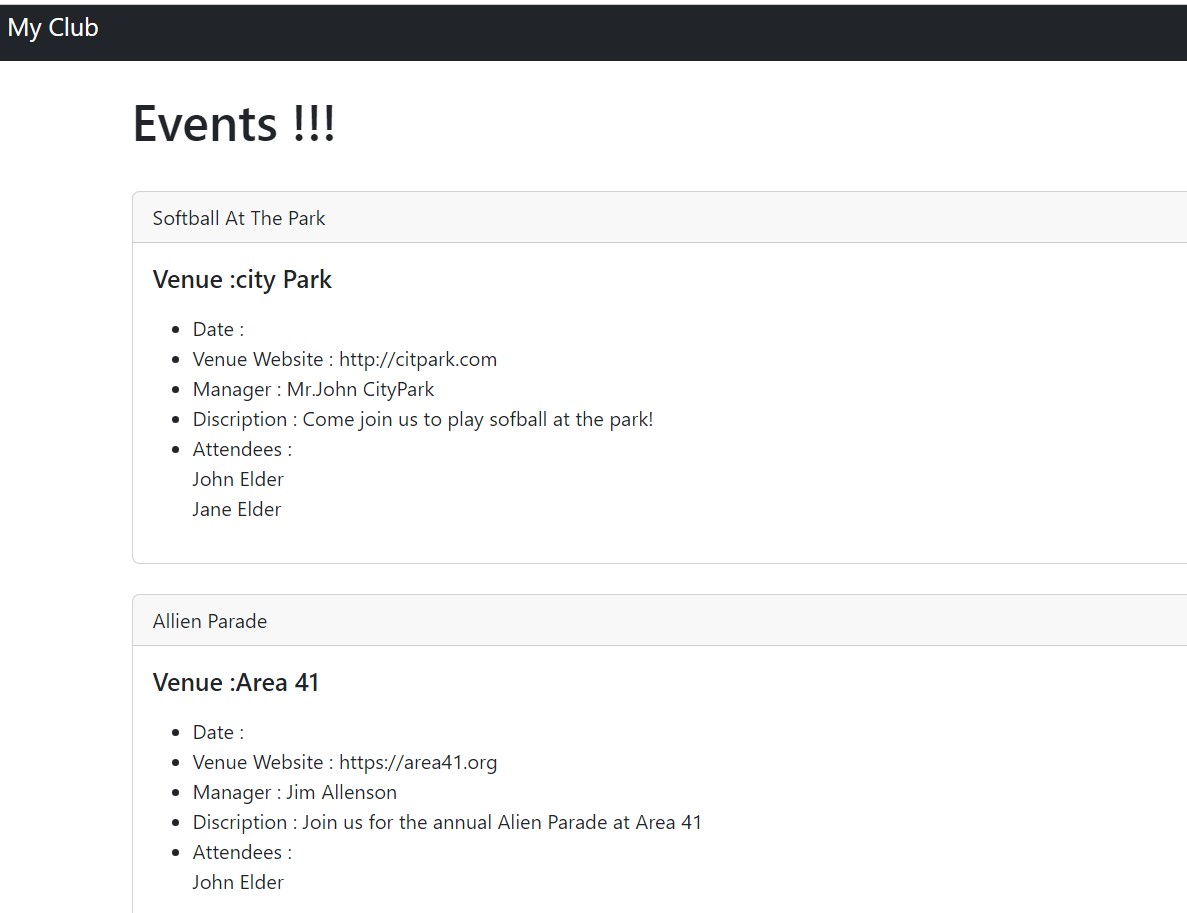
          </p>

        </div>

      </div>

       {% endfor %}

{% endblock content %}



**EP.6 How To Modify The Django Admin Area**

admin.py

*# admin.site.register(Venue)*

admin.site.register(MyClubUser)

*# admin.site.register(Event)*

@admin.register(Venue)

class VenueAdmin(admin.ModelAdmin):

    list\_display = ('name','address','phone')

    ordering = ('name',)

    search\_fields = ('name','address')

@admin.register(Event)

class EventAdmin(admin.ModelAdmin):

    fields = (('name', 'venue'),'event\_date','description','manager')

    list\_display = ('name','event\_date', 'venue')

    list\_filter = ('event\_date','venue')

    ordering = ('event\_date',)

model.py

from django.contrib.auth.models import User

class Event(models.Model):

    name = models.CharField('Event Name',max\_length=120)

    event\_date = models.DateTimeField('Event Date')

*# venue = models.CharField(max\_length=120)*

    venue = models.ForeignKey(Venue, blank=True, null=True, on\_delete=models.CASCADE)

    manager = models.ForeignKey(User, blank=True, null=True, on\_delete=models.SET\_NULL)

    description = models.TextField(blank=True)

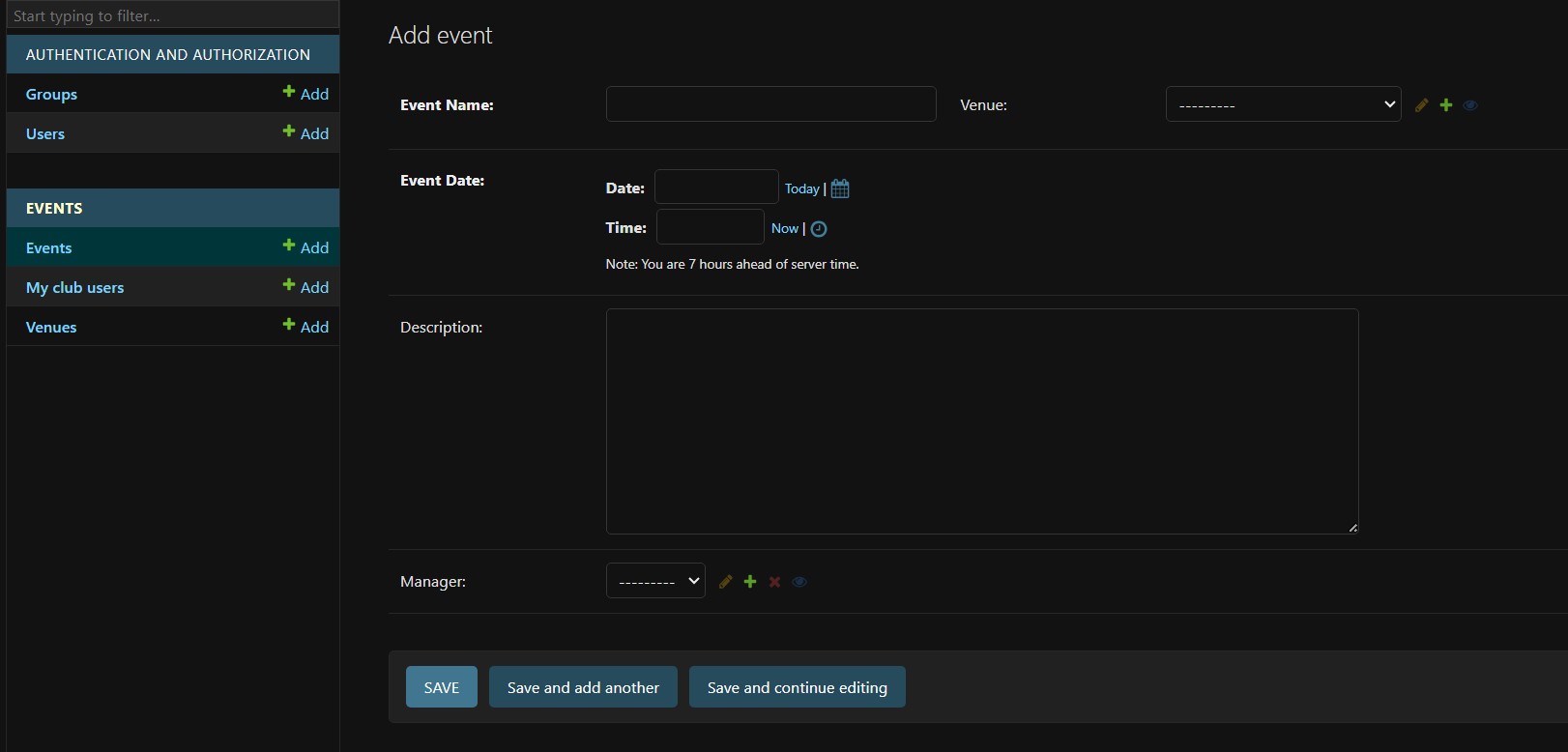
    attendees = models.ManyToManyField(MyClubUser)

    def \_\_str\_\_(self):

        return self.name

python manage.py makemigrations

python manage.py migrate



**EP.7**

**How To Add Database Forms To A Web Page**

form.py

from django import forms

from django.forms import ModelForm

from .models import Venue

*#Create a venue form*

class VenueForm(ModelForm):

    class Meta:

        model = Venue

        fields = ('name','address','zip\_code','phone','web','email\_address')

        labels = {

            'name': '',

            'address': '',

            'zip\_code': '',

            'phone': '',

            'web': '',

            'email\_address': '',

        }

        widgets = {

            'name': forms.TextInput(attrs={'class':'form-control','placeholder':'Venue Name'}),

            'address': forms.TextInput(attrs={'class':'form-control','placeholder':'Address'}),

            'zip\_code': forms.TextInput(attrs={'class':'form-control','placeholder':'Zip Code'}),

            'phone': forms.TextInput(attrs={'class':'form-control','placeholder':'Phone'}),

            'web': forms.TextInput(attrs={'class':'form-control','placeholder':'Web Site'}),

            'email\_address': forms.EmailInput(attrs={'class':'form-control','placeholder':'Email Address'}),

        }

Template/events/add\_venue.html

    <center>

        <h1>Add Venue To Database</h1>

        <br><br>

        {% if submitted %}

            Your Venue Was Submitted succefully

        {% else %}

            <form action="", method="POST">

                {% csrf\_token %}

                {{form.as\_p}}

       <input type="submit" value="submit" class="btn btn-secondary">

            </form>

        {% endif %}

    </center>

urls.py

 path('add\_venue', views.add\_venue, name="add-venue"),

views.py

from django.http import HttpResponseRedirect

from .form import VenueForm

def add\_venue(request):

    submitted = False

    if request.method == "POST":

        form = VenueForm(request.POST)

        if form.is\_valid():

            form.save()

            return HttpResponseRedirect('/add\_venue?submitted=True')

    else:

        form = VenueForm

        if 'submitted' in request.GET:

            submitted = True

    return render(request, 'events/add\_venue.html', {'form':form,'submitted':submitted})

navbar.html

<a class="nav-link" href="{% url 'add-venue' %}">Add Venue</a>

**EP.8**

**How To Query The Database For Venues**

Templates/events/venue.html

{% extends 'events/base.html' %}

{% block content %}

        <h1>Venues....</h1>

        <br>

        <table class="table table-striped table-hover table-bordered">

            {% for venue in venue\_list %}

                <tr>

                    <td><a href="{% url 'show-venue' venue.id %}">{{venue}}</a></td>

                </tr>

            {% endfor %}

        </table>

{% endblock content %}

urls.py

urlpatterns = [

    path('', views.home, name="home"),

    path('<int:year>/<str:month>/', views.home, name="home"),

    path('events', views.all\_events, name='list-events'),

    path('add\_venue', views.add\_venue, name="add-venue"),

    path('list\_venues', views.list\_venues, name="list-venues"),

    path('show\_venue/<venue\_id>', views.show\_venue, name="show-venue"),

]

views.py

from datetime import datetime

from django.http import HttpResponseRedirect

from .models import Event

from .models import Venue

from .form import VenueForm

def show\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

    return render(request, 'events/show\_venue.html', {

        'venue':venue})

def list\_venues(request):

    venue\_list = Venue.objects.all()

    return render(request, 'events/venue.html', {

        'venue\_list':venue\_list})

Templates/events/show\_venue.html

{% extends 'events/base.html' %}

{% block content %}

        <div class="card">

            <div class="card-header">Venue</div>

            <div class="card-body">

              <h5 class="card-title">{{venue}}</h5>

              <p class="card-text">

                <ul>

                    <li>{{venue.address}}</li>

                    <li>{{venue.zip\_code}}</li>

                    <li>{{venue.phone}}</li>

                    <li>{{venue.email\_address}}</li>

                    <li><a

href="{{venue.web}}">{{venue.web}}</a></li>

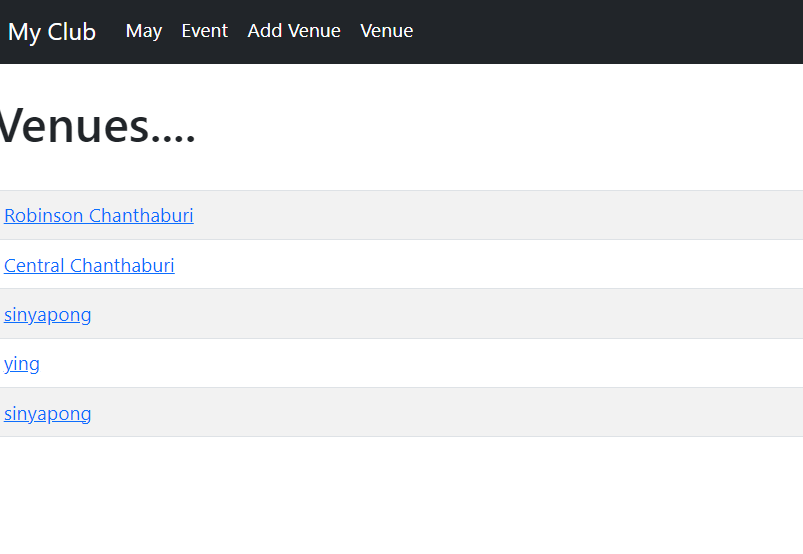
                </ul>

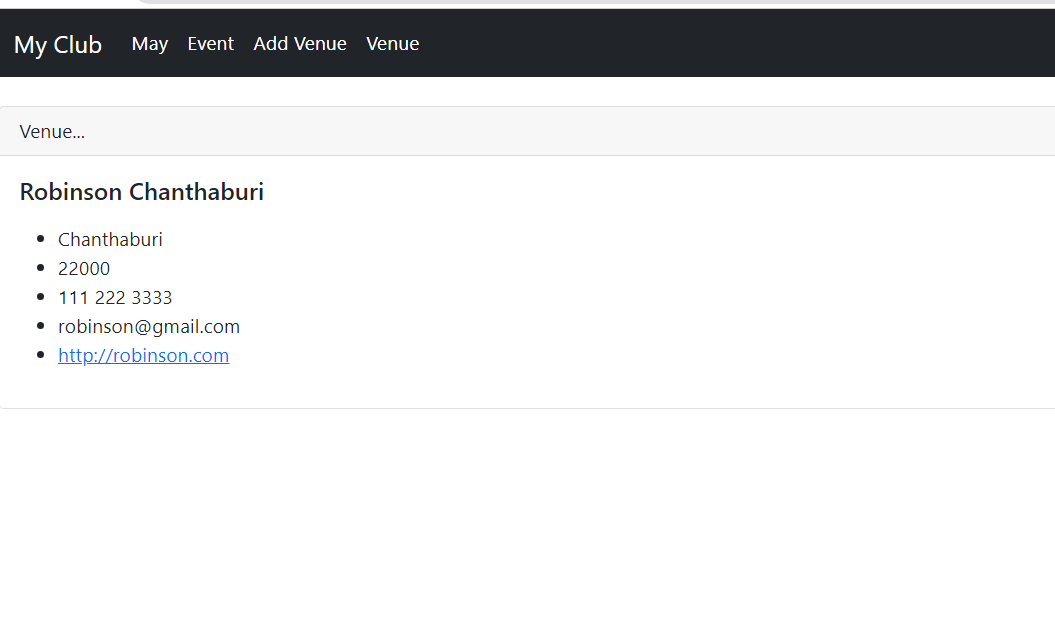
              </p>

            </div>

          </div>

{% endblock content %}





**EP.9**

**Create A Search Bar**

Templates/evnets/search\_venue.html

{% extends 'events/base.html' %}

{% block content %}

    <center>

    {% if searched %}

        <h1>You Searched For{{searched}}</h1>

        {% for venue in venues %}

            <a href="{% url 'show-venue' venue.id %}">{{venue}}</a><br>

        {% endfor %}

    {% else %}

        <h1>Hey! You Forgot To Search For A Venue...</h1>

    {% endif %}

    </center>

{% endblock content %}

Urls.py

 path('search\_venues', views.search\_venues, name="search-venues"),

views.py

def search\_venues(request):

    if request.method == "POST":

        searched = request.POST['searched']

        venues = Venue.objects.filter(name\_\_contains=searched)

        return render(request, 'events/search\_venues.html', {

            'searched':searched,

            'venues':venues})

nevbar.html

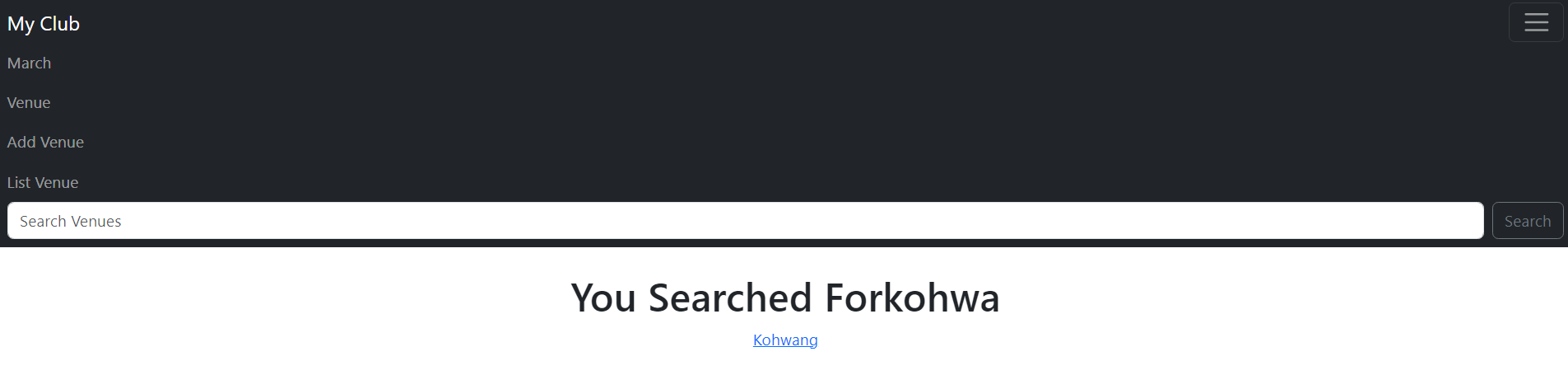
<form class="d-flex" method="POST" role="search" action="{% url 'search-venues' %}">

          {% csrf\_token %}

          <input class="form-control me-2" type="search" placeholder="Search Venues" aria-label="Search" name="searched">

          <button class="btn btn-outline-secondary" type="submit">Search</button>

        </form>



**EP.10**

**Update And Edit Venues**

Templates/evnets/update\_venue.html

{% extends 'events/base.html' %}

{% block content %}

    <h1>Update Venues....</h1>

    <br>

    <form action="" method="POST">

       {%csrf\_token%}

        {{form.as\_p}}

        <br>

        <input type="submit" value="Update" class="btn btn-secondary">

{% endblock content %}

Urls.py

path('update\_venue/<venue\_id>', views.update\_venue, name="update-venue"),

views.py

def update\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

    form = VenueForm(request.POST or None, instance=venue)

    if form.is\_valid():

        form.save()

        return redirect('list-venues')

    return render(request, 'events/update\_venues.html', {

        'venue':venue, 'form':form})

Templates/evnets/venue.html

{% extends 'events/base.html' %}

{% block content %}

        <h1>Venues....</h1>

        <br>

        <table class="table table-striped table-hover table-bordered">

            {% for venue in venue\_list %}

                <tr>

                    <td><a href="{% url 'show-venue' venue.id %}">{{venue}}</a></td>

                    <td align="right">

                        <a href="{% url 'update-venue' venue.id %}" class="btn btn-outline-secondary">Update</a>

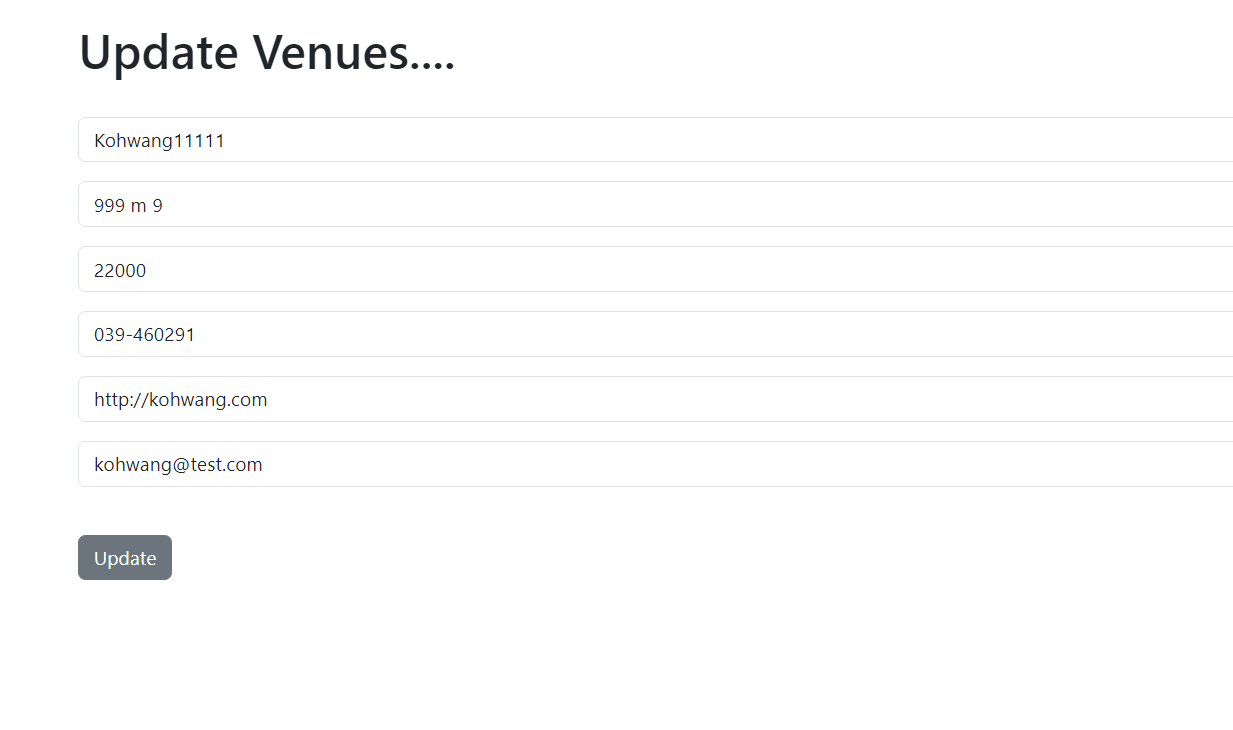
                    </td>

                </tr>

            {% endfor %}

        </table>

{% endblock content %}



**EP.11**

**create An Add Event Page**

Templates/evnets/add\_event.html

{% extends 'events/base.html' %}

{% block content %}

    <h1>Add Event</h1>

    <br><br>

    {% if submitted %}

    Your Event Was Submitted succefully

    {% else %}

    <form action="", method="POST">

        {% csrf\_token %}

        {{form.as\_p}}

        <input type="submit" value="submit" class="btn btn-secondary">

    </form>

    {% endif %}

    <br><br>

{% endblock content %}

urls.py

path('add\_event', views.add\_event, name="add-event"),

nevbar.html

<li class="nav-item">

            <a class="nav-link" href="{% url 'add-event' %}">Add Event</a>

          </li>

views.py

def add\_event(request):

    submitted = False

    if request.method == "POST":

        form = EventForm(request.POST)

        if form.is\_valid():

            form.save()

            return HttpResponseRedirect('/add\_event?submitted=True')

    else:

        form = EventForm

        if 'submitted' in request.GET:

            submitted = True

    return render(request, 'events/add\_event.html', {'form':form,'submitted':submitted})

Form.py

from .models import Venue, Event

class EventForm(ModelForm):

    class Meta:

        model = Event

        fields = ('name','event\_date','venue','manager','attendees','description')

        labels = {

            'name': '',

            'event\_date': 'YYYY-MM-DD HH:MM:SS',

            'venue': 'Venue',

            'manager': 'Manager',

            'attendees': 'Attendees',

            'description': '',

        }

        widgets = {

            'name': forms.TextInput(attrs={'class':'form-control','placeholder':'Event Name'}),

            'event\_date': forms.TextInput(attrs={'class':'form-control','placeholder':'Event\_date'}),

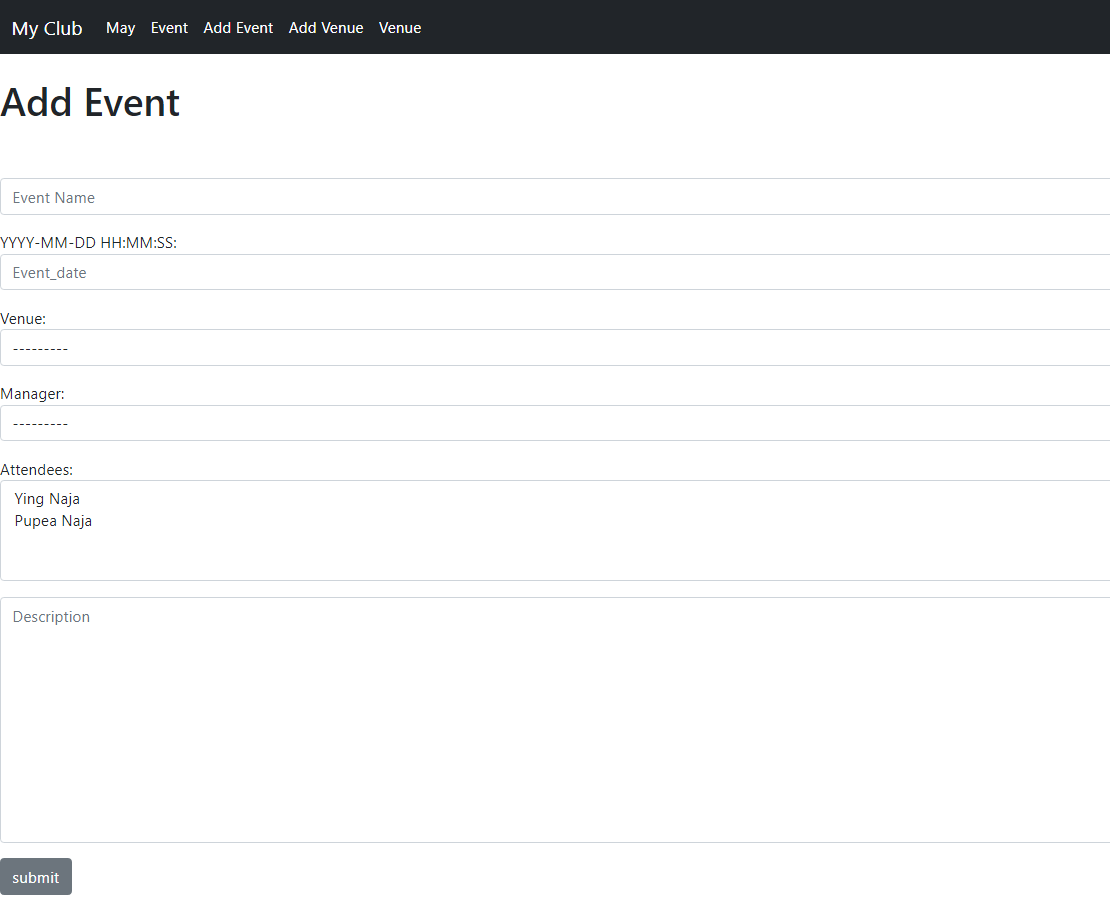
            'venue': forms.Select(attrs={'class':'form-select','placeholder':'Venue'}),

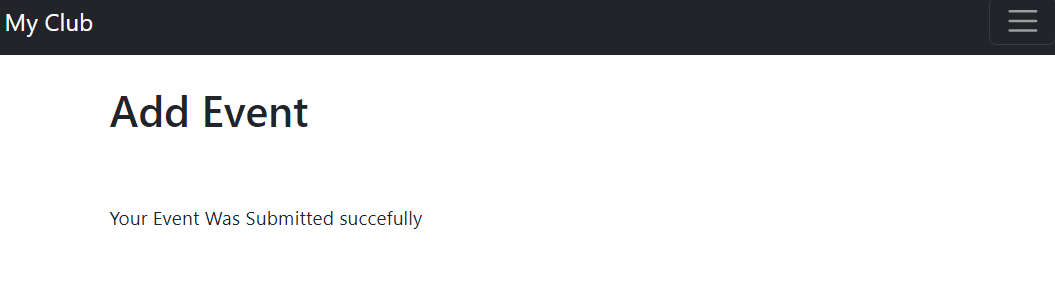
            'manager': forms.Select(attrs={'class':'form-select','placeholder':'Manager'}),

            'attendees': forms.SelectMultiple(attrs={'class':'form-control','placeholder':'Attendees'}),

            'description': forms.Textarea(attrs={'class':'form-control','placeholder':'Description'}),

        }





**EP.12**

**create An Update Events Page**

Templates/evnets/update\_event.html

{% extends 'events/base.html' %}

{% block content %}

    <h1>Update Event....</h1>

    <br>

    <form action="" method="POST">

       {%csrf\_token%}

        {{form.as\_p}}

        <br>

        <input type="submit" value="Update" class="btn btn-secondary">

{% endblock content %}

Urls.py

path('update\_event/<event\_id>', views.update\_event, name="update-event"),

Views.py

def update\_event(request, event\_id):

    event = Event.objects.get(pk=event\_id)

    form = EventForm(request.POST or None, instance=event)

    if form.is\_valid():

        form.save()

        return redirect('list-events')

    return render(request, 'events/update\_event.html', {

        'event':event, 'form':form})

Templates/evnets /event\_list.html

            <ul>

                <li>Date : {{event.event\_date}}</li>

                <li>Venue Website : {{event.venue.web}}</li>

                <li>Manager : {{event.manager}}</li>

                <li>Discription : {{event.description}}</li>

                <li>Attendees : <br>

                    {% for user in event.attendees.all %}</li>

                     {{user}}<br>

                    {% endfor %}

            </ul>

          </p>

        </div>

        <div class="card-footer text-muted">

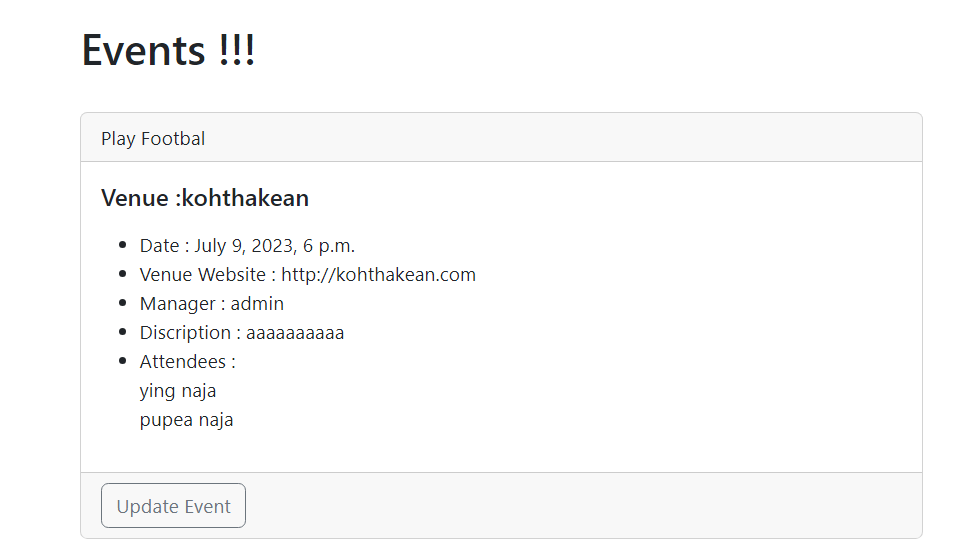
          <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

        </div>

      </div>

       {% endfor %}

{% endblock content %}



**EP.13**

**Delete Items From the Database**

Urls.py

path('delete\_event/<event\_id>', views.delete\_event, name="delete-event"),

    path('delete\_venue/<venue\_id>', views.delete\_venue, name="delete-venue"),

event\_list.html

<div class="card-footer text-muted">

          <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

          <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

        </div>

Venue.html

<td align="right">

                        <a href="{% url 'update-venue' venue.id %}" class="btn btn-outline-secondary">Update</a>

                        <a href="{% url 'delete-venue' venue.id %}" class="btn btn-outline-danger">Delete</a>

                    </td>

views.py

def delete\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

    venue.delete()

    return redirect('list-venues')

def delete\_event(request, event\_id):

    event = Event.objects.get(pk=event\_id)

    event.delete()

    return redirect('list-events')

**EP.14**

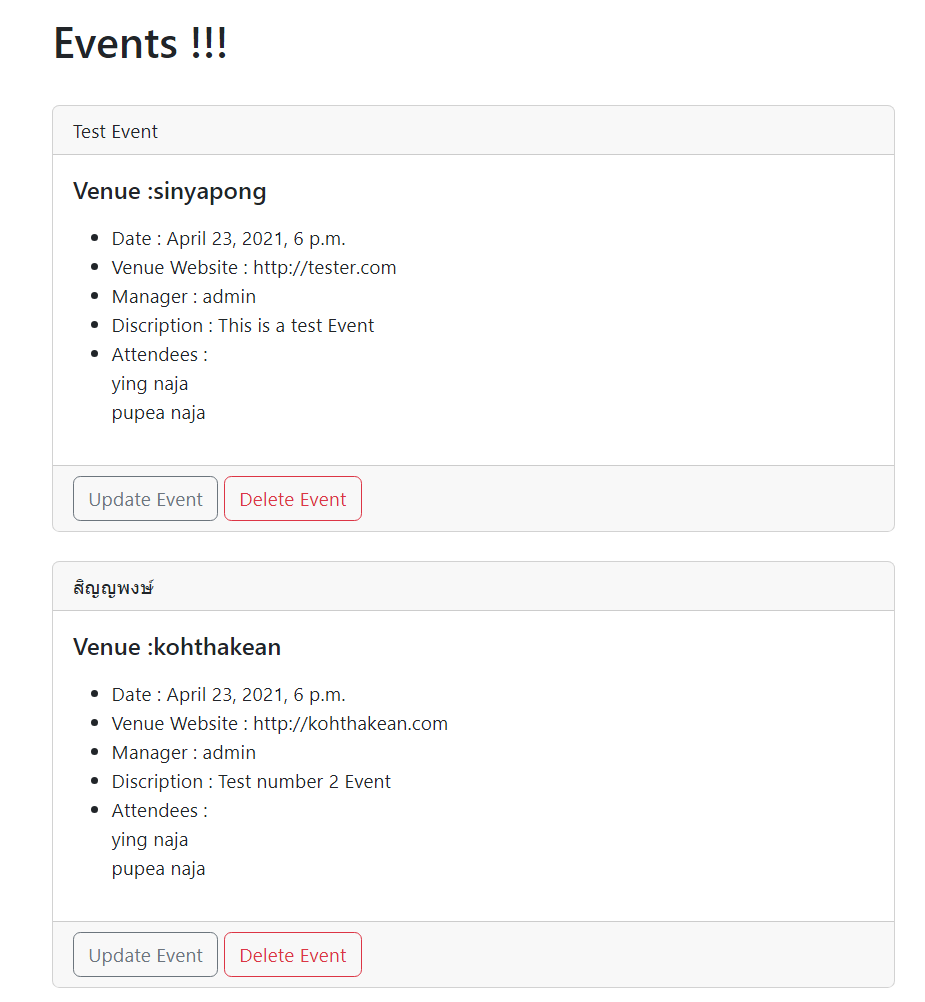
**How To Order Items From the Database**

views.py

def all\_events(request):

    event\_list = Event.objects.all().order\_by('event\_date')

    return render(request, 'events/event\_list.html', {'event\_list':event\_list})



**EP.15**

**How To Generate Text Files Dynamically With Djagno**

ursl.py

path('venue\_text',views.venue\_text, name="venue-text"),

views

from django.http import HttpResponse

def venue\_text(request):

    response = HttpResponse(content\_type='text/plain')

    response['Content-Disposition'] = 'attachment; filename=venues.text'

    # Designnate The Model

    venues = Venue.objects.all()

    lines = []

    for venue in venues:

        lines.append(f'{venue.name}\n{venue.address}\n{venue.zip\_code}\n{venue.phone}\n{venue.web}\n{venue.email\_address}\n\n')

    """ lines = ["This is line 1\n",

            "Sinyapong Sukito\n"] """

    response.writelines(lines)

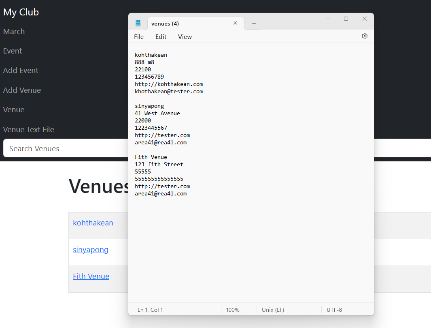
    return response

navber.html

<li class="nav-item">

            <a class="nav-link" href="{% url 'venue-text' %}">Venue Text File</a>

          </li>



**EP.16**

**How To Generate CSV Files Dynamically With Djagno**

ursl.py

path('venue\_csv',views.venue\_csv, name="venue-csv"),

views

import csv

def venue\_csv(request):

    response = HttpResponse(content\_type='text/csv')

    response['Content-Disposition'] = 'attachment; filename=venues.csv'

    # Create CSV

    writer = csv.writer(response)

    # Designnate The Model

    venues = Venue.objects.all()

    #Add column headings to the csv file

    writer.writerow(['Venue Name','Address','Zip code','Phone','Web Address','Email Address'])

    for venue in venues:

        writer.writerow([venue.name,venue.address,venue.zip\_code,venue.phone,venue.web,venue.email\_address])

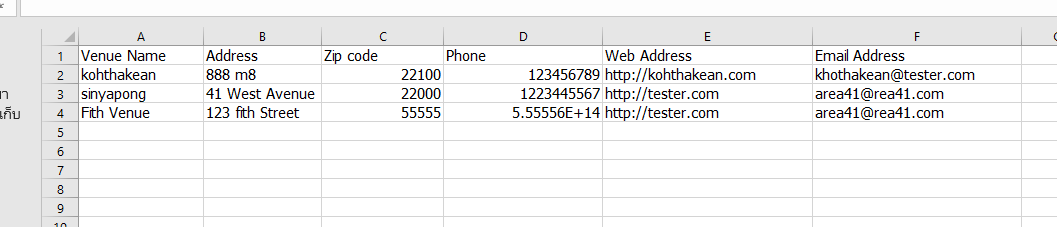
    return response

navbar.html

<li class="nav-item">

            <a class="nav-link" href="{% url 'venue-csv' %}">Venue CSV File</a>

          </li>



**EP.17**

**How To Generate PDF Files Dynamically With Djagno**

Pip install reportlab

views

from django.http import FileResponse

import io

from reportlab.pdfgen import canvas

from reportlab.lib.units import inch

from reportlab.lib.pagesizes import A4

def venue\_pdf(request):

    # Create Bytestream buffer

    buf = io.BytesIO()

    # Create a cavvas

    c = canvas.Canvas(buf,pagesize=letter, bottomup=0)

    # Create a text object

    textob = c.beginText()

    textob.setTextOrigin(inch, inch)

    textob.setFont("Helvetica",14)

    # Add some line of text

    """ lines = ["This is line 1",] """

    # Designnate The Model

    venues = Venue.objects.all()

    lines = []

    for venue in venues:

        lines.append(venue.name)

        lines.append(venue.address)

        lines.append(venue.zip\_code)

        lines.append(venue.phone)

        lines.append(venue.web)

        lines.append(venue.email\_address)

        lines.append(" ")

    # Loop

    for line in lines:

        textob.textLine(line)

    # Finish Up

    c.drawText(textob)

    c.showPage()

    c.save()

    buf.seek(0)

    return FileResponse(buf, as\_attachment=True,filename="venue.pdf")

ursl.py

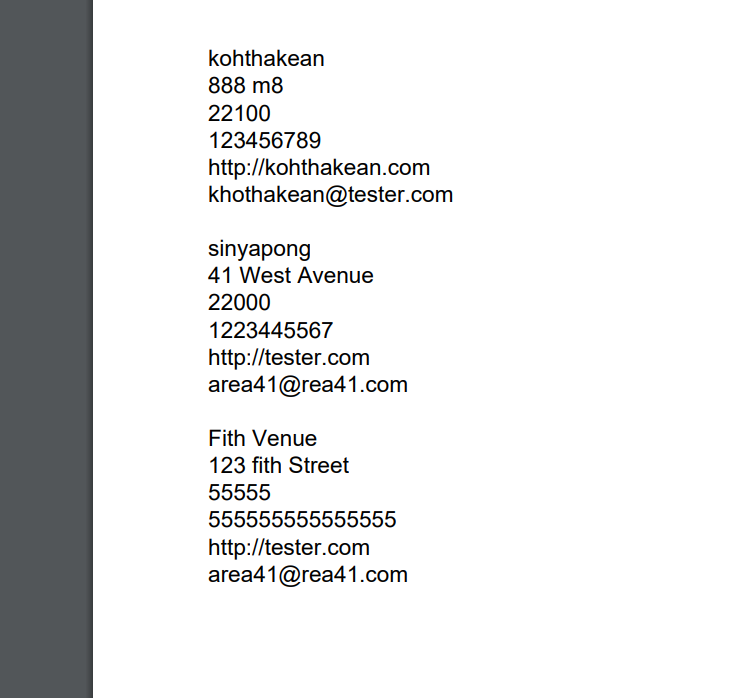
path('venue\_pdf',views.venue\_pdf, name="venue-pdf"),

navbar.html

<li class="nav-item">

            <a class="nav-link" href="{% url 'venue-pdf' %}">Venue PDF File</a>

          </li>



**EP.18**

**pagination For Django**

venue.html

{% extends 'events/base.html' %}

{% block content %}

        <h1>Venues....</h1>

        <br>

        <table class="table table-striped table-hover table-bordered">

            {% for venue in venues %}

                <tr>

                    <td><a href="{% url 'show-venue' venue.id %}">{{venue}}</a></td>

                    <td align="right">

                        <a href="{% url 'update-venue' venue.id %}" class="btn btn-outline-secondary">Update</a>

                        <a href="{% url 'delete-venue' venue.id %}" class="btn btn-outline-danger">Delete</a>

                    </td>

                </tr>

            {% endfor %}

        </table>

        <br><br>

        <!--

        {{venues}}<br>

        Has Previous: {{venues.has\_previous}}<br>

        Has Next: {{venues.has\_next}}<br>

        Current Page: {{venues.number}}<br>

        Number of Pages: {{venues.paginator.num\_pages}}<br>

        <hr><br><br>

        -->

        {% if venues.has\_previous %}

            <a href="?page=1">&laquo First</a>

            <a href="?page={{venues.previous\_page\_number}}">Previous</a>

        {% endif %}

        Page {{venues.number}} of {{venues.paginator.num\_pages}}

        {% if venues.has\_next %}

            <a href="?page={{venues.next\_page\_number}}">Next</a>

            <a href="?page={{venues.paginator.num\_pages}}">Last &raquo</a>

        {% endif %}

{% endblock content %}

views.py

from django.core.paginator import Paginator

def list\_venues(request):

    # venue\_list = Venue.objects.all().order\_by('?')

    venue\_list = Venue.objects.all()

    # Set up Pagination

    p = Paginator(Venue.objects.all(), 1)

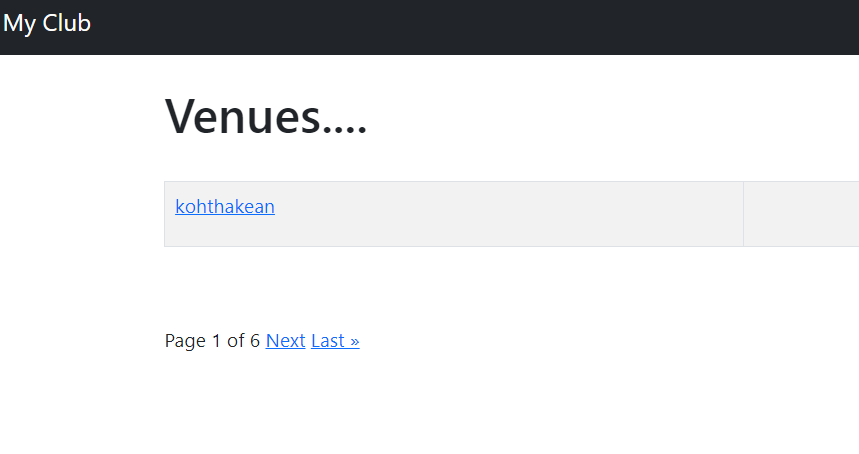
    page = request.GET.get('page')

    venues = p.get\_page(page)

    return render(request, 'events/venue.html', {

        'venue\_list': venue\_list,

        'venues': venues})



**EP.19**

**Special Hack To Style Pagination With Bootstrap**

venue.html

{% extends 'events/base.html' %}

{% block content %}

        <h1>Venues....</h1>

        <br>

        <table class="table table-striped table-hover table-bordered">

            {% for venue in venues %}

                <tr>

                    <td><a href="{% url 'show-venue' venue.id %}">{{venue}}</a></td>

                    <td align="right">

                        <a href="{% url 'update-venue' venue.id %}" class="btn btn-outline-secondary">Update</a>

                        <a href="{% url 'delete-venue' venue.id %}" class="btn btn-outline-danger">Delete</a>

                    </td>

                </tr>

            {% endfor %}

        </table>

        <br><br>

        <nav aria-label="Page navigation example">

            <ul class="pagination">

                {% if venues.has\_previous %}

                    <li class="page-item"><a class="page-link" href="?page=1">&laquo First</a></li>

                    <li class="page-item"><a class="page-link" href="?page={{venues.previous\_page\_number}}">Previous</a></li>

                {% endif %}

                <!--

                <li class="page-item"><a class="page-link disabled" href="#">Page {{venues.number}} of {{venues.paginator.num\_pages}}</a></li>

                -->

                {% for i in nums %}

                    <li class="page-item"><a class="page-link" href="?page={{forloop.counter}}">{{forloop.counter}}</a></li>

                {% endfor %}

                {% if venues.has\_next %}

                    <li class="page-item"><a class="page-link" href="?page={{venues.next\_page\_number}}">Next</a></li>

                    <li class="page-item"><a class="page-link" href="?page={{venues.paginator.num\_pages}}">Last &raquo</a></li>

                {% endif %}

            </ul>

        </nav>

        <br><br>

{% endblock content %}

Views.py

def list\_venues(request):

    # venue\_list = Venue.objects.all().order\_by('?')

    venue\_list = Venue.objects.all()

    # Set up Pagination

    p = Paginator(Venue.objects.all(), 2)

    page = request.GET.get('page')

    venues = p.get\_page(page)

    nums = 'a' \* venues.paginator.num\_pages

    return render(request, 'events/venue.html', {

        'venue\_list': venue\_list,

        'venues': venues,

        'nums': nums})



**EP.20**

**Customize Django Admin Titles - Django Wednesdays #20**

Myclub\_website/urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('events.urls')),

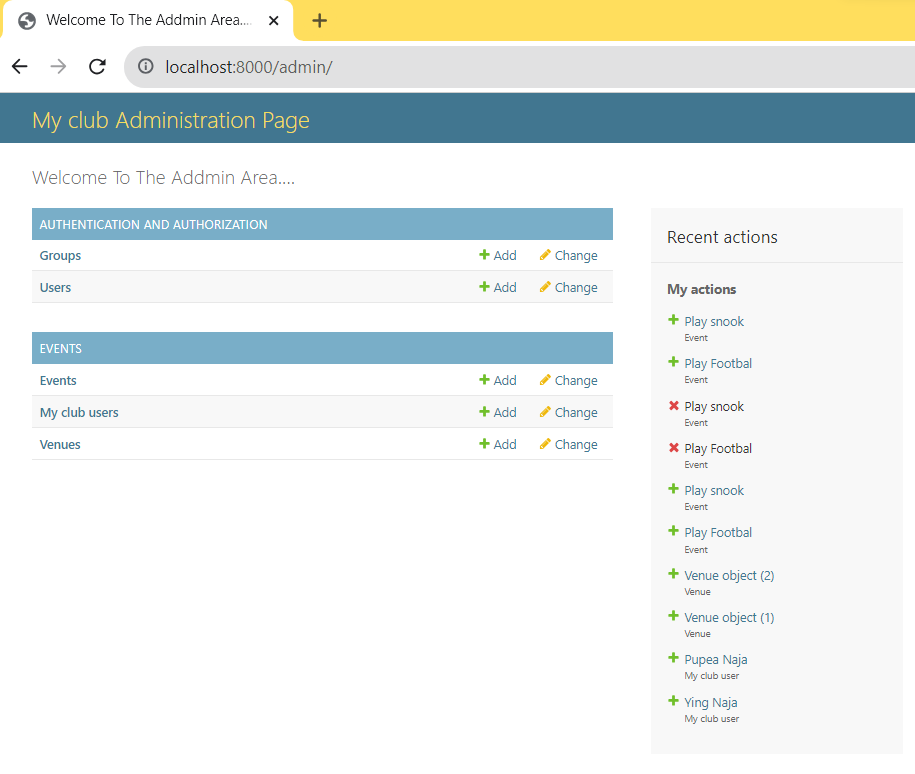
]

# Configure Admin Titles

admin.site.site\_header="My club Administration Page"

admin.site.site\_title="Browser Title"

admin.site.index\_title="Welcome To The Addmin Area...."



**EP.21**

# Login With User Authentication

Myclub\_website/urls.py

Python manage.py startapp members

Settings.py >>>>>> ‘members’,

Myclub\_website/Urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('events.urls')),

    path('members/', include('djange.contrip.auth.urls')),

    path('members/', include('members.urls')),

]

# Configure Admin Titles

admin.site.site\_header="My club Administration Page"

admin.site.site\_title="Browser Title"

admin.site.index\_title="Welcome To The Addmin Area...."

member/urls.py

from django.urls import path

from . import views

urlpatterns = [

    path('login\_user', views.login\_user, name="login"),

]

Member/Views.py

from django.shortcuts import render, redirect

from django.contrib.auth import authenticate, login, logout

from django.contrib import messages

def login\_user(request):

    if request.method == "POST":

        username = request.POST['username']

        password = request.POST['password']

        user = authenticate(request, username=username, password=password)

        if user is not None:

            login(request, user)

            return redirect('home')

        else:

            messages.success(request, ("There Was An Error In Your Login"))

            return redirect('login')

    else:

        return render(request, 'authenticate/login.html',{})

Base.html

<body>

    {% include 'events/navbar.html'%}    <br>

    <div class="container">

        {% if messages %}

            {% for message in messages %}

              <div class="alert alert-warning alert-dismissible fade show" role="alert">

                <strong>{{ message }}</strong>

                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

              </div>

            {% endfor %}

        {% endif %}

        {% block content %}

        {% endblock content%}

    </div>

members/templates/authenticate/login.html

{% extends 'events/base.html' %}

{% block content %}

    <h1>Login</h1>

    <br>

        <form action="", method="POST">

            {% csrf\_token %}

            <form>

                <div class="mb-3">

                  <label for="exampleInputEmail1" class="form-label">Username</label>

                  <input type="text" class="form-control" name="username" aria-describedby="emailHelp">

                </div>

                <div class="mb-3">

                  <label for="exampleInputPassword1" class="form-label">Password</label>

                  <input type="password" class="form-control" name="password">

                </div>

                <button type="submit" class="btn btn-secondary">Submit</button>

              </form>

        </form>

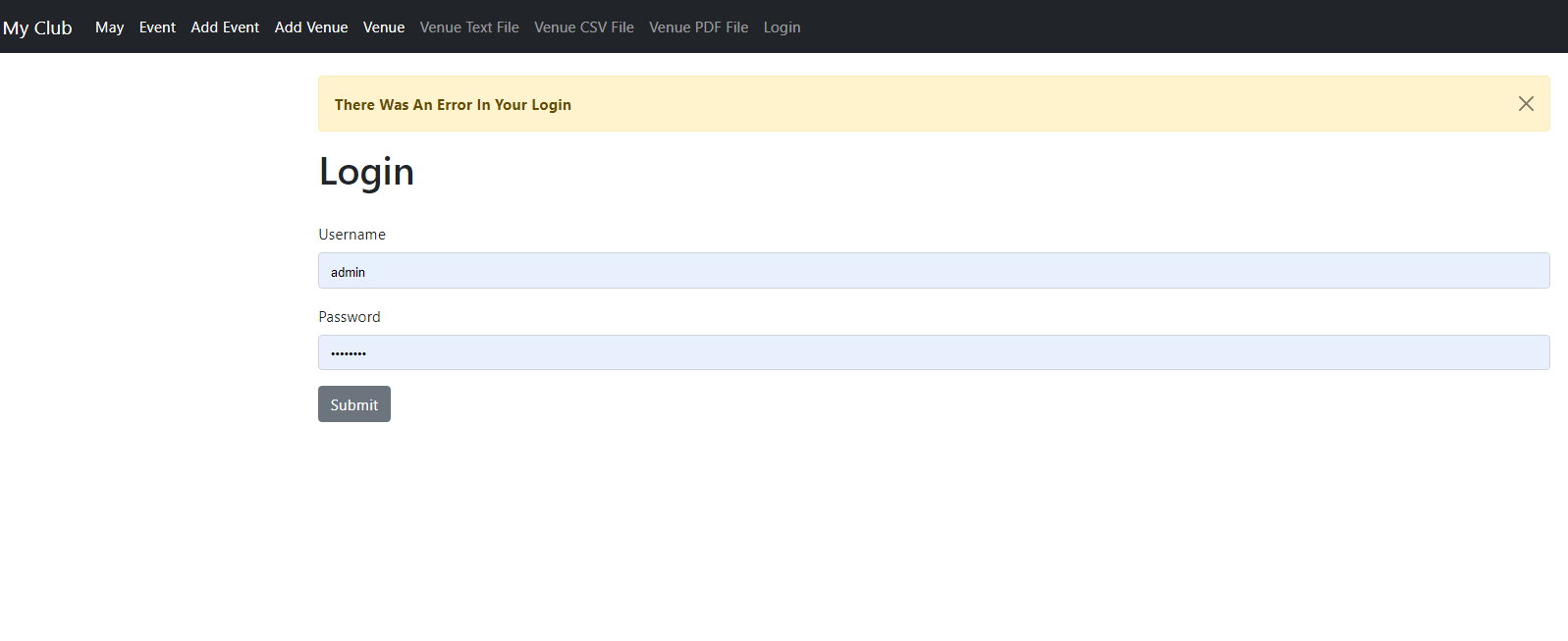
{% endblock content %}

Navbar.html

<li class="nav-item">

            <a class="nav-link" href="{% url 'login' %}">Login</a>

          </li>



**EP.22**

# Log Out With User Authentication

Navbar.html

{% if user.is\_authenticated %}

            <li class="nav-item">

              <a class="nav-link" href="{% url 'logout' %}">Logout</a>

            </li>

          {% else %}

            <li class="nav-item">

              <a class="nav-link" href="{% url 'login' %}">Login</a>

            </li>

          {% endif %}

Members/Urls.py

path('logout\_user', views.logout\_user, name="logout"),

members/views.py

def logout\_user(request):

    messages.success(request, ("You Were Logged Out!"))

    logout(request)

    return redirect('home')



**EP.23** Navbar Dropdown Menus

Navbar.html

<li class="nav-item">

    <a class="nav-link active" aria-current="page" href="{% url 'home' '2023' 'May' %}">

Calendar</a>

</li>

<li class="nav-item dropdown">

     <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"

data-bs-toggle="dropdown" aria-expanded="false"> Events</a>

     <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

       <li><a class="dropdown-item" href="{% url 'list-events' %}">All Event</a></li>

       <li><a class="dropdown-item" href="{% url 'add-event' %}">Add Event</a></li>

      </ul>

</li>

    <li class="nav-item dropdown">

     <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"

data-bs-toggle="dropdown" aria-expanded="false"> Venues</a>

    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

          <li><a class="dropdown-item" href="{% url 'list-venues' %}">All Venue</a></li>

          <li><a class="dropdown-item" href="{% url 'add-venue' %}">Add Venue</a></li>

        </ul>

     </li>

     <li class="nav-item dropdown">

       <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button"

data-bs-toggle="dropdown" aria-expanded="false"> Download</a>

      <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

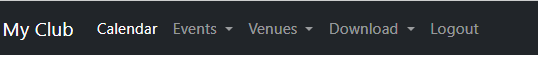
       <li><a class="dropdown-item" href="{% url 'venue-text' %}">Venue Text File</a></li>

       <li><a class="dropdown-item" href="{% url 'venue-csv' %}">Venue CSV File</a></li>

       <li><a class="dropdown-item" href="{% url 'venue-pdf' %}">Venue PDF File</a></li>

      </ul>

    </li>



**EP.24 How To Register Users**

Navbar.html

Member/views.py

from django.contrib.auth.forms import UserCreationForm

def register\_user(request):

    return render(request, 'authenticate/register\_user.html')

member/urls.py

path('register\_user', views.register\_user, name="register\_user"),

navbar.html

{% else %}

            <li class="nav-item">

              <a class="nav-link" href="{% url 'register\_user' %}">Register</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{% url 'login' %}">Login</a>

            </li>

        {% endif %}

Member/Views.py

def register\_user(request):

    if request.method == "POST":

        form = UserCreationForm(request.POST)

        if form.is\_valid():

            form.save()

            username = form.cleaned\_data['username']

            password = form.cleaned\_data['password1']

            user = authenticate(username=username, password=password)

            login(request, user)

            messages.success(request, ("Registration Successfully"))

            return redirect('home')

    else:

        form = UserCreationForm()

    return render(request, 'authenticate/register\_user.html', {'form':form})

member/register\_user.html

{% extends 'events/base.html' %}

{% block content %}

{% if form.errors %}

    <div class="alert alert-warning alert-dismissible fade show" role="alert">

        <strong>There was an error with your form...</strong>

        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

    </div>

{% endif %}

    <h1>Register</h1>

    <br>

        <form action="{% url 'register\_user' %}", method="POST">

            {% csrf\_token %}

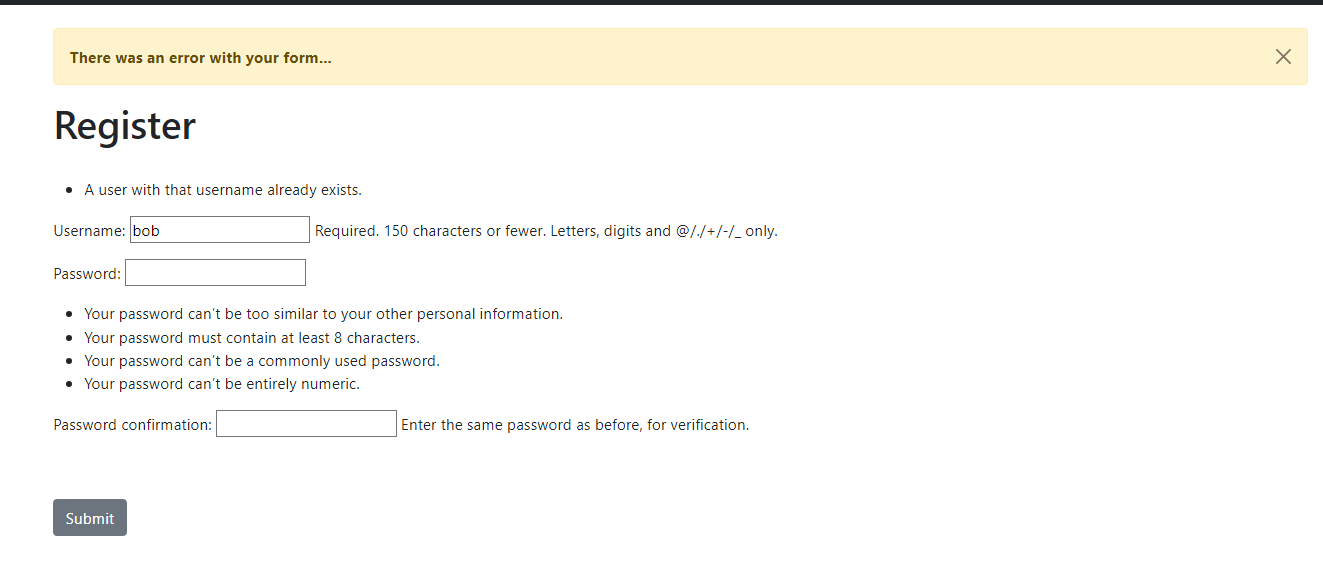
            {{ form.as\_p }}

            <br><br>

            <button type="submit" class="btn btn-secondary">Submit</button>

        </form>

{% endblock content %}



**EP.25 Add Extra Registration Fields**

Members/Forms.py

from django.contrib.auth.forms import UserCreationForm

from django.contrib.auth.models import User

from django import forms

class RegisterUserForm(UserCreationForm):

    email = forms.EmailField()

    first\_name = forms.CharField(max\_length=50)

    last\_name = forms.CharField(max\_length=50)

    class Meta:

        model = User

        fields = ('username','first\_name','last\_name', 'email', 'password1','password2')

members/views.py

from .forms import RegisterUserForm

def register\_user(request):

    if request.method == "POST":

        form = RegisterUserForm(request.POST)

        if form.is\_valid():

            form.save()

            username = form.cleaned\_data['username']

            password = form.cleaned\_data['password1']

            user = authenticate(username=username, password=password)

            login(request, user)

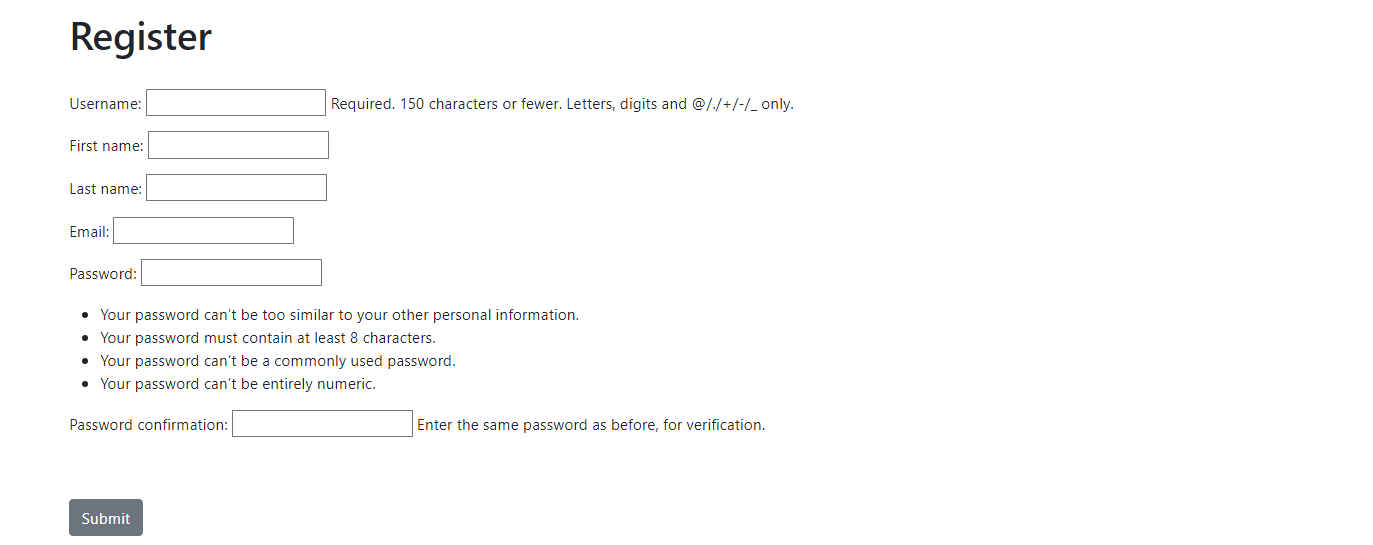
            messages.success(request, ("Registration Successfully"))

            return redirect('home')

    else:

        form = RegisterUserForm()

    return render(request, 'authenticate/register\_user.html', {'form':form})



# EP.26 Style Django Registration Forms

Members/Forms.py

class RegisterUserForm(UserCreationForm):

    email = forms.EmailField(widget=forms.EmailInput(attrs={'class':'form-control'}))

    first\_name = forms.CharField(max\_length=50, widget=forms.TextInput(attrs={'class':'form-control'}))

    last\_name = forms.CharField(max\_length=50, widget=forms.TextInput(attrs={'class':'form-control'}))

    class Meta:

        model = User

        fields = ('username','first\_name','last\_name', 'email', 'password1','password2')

    def \_\_init\_\_(self, \*args, \*\*kwargs):

        super(RegisterUserForm, self).\_\_init\_\_(\*args, \*\*kwargs)

        self.fields['username'].widget.attrs['class'] = 'form-control'

        self.fields['password1'].widget.attrs['class'] = 'form-control'

        self.fields['password2'].widget.attrs['class'] = 'form-control'

members/register.html

{% endif %}

    <div class="shadow p-4 mb-5 bg-body rounded">

Event/base.html

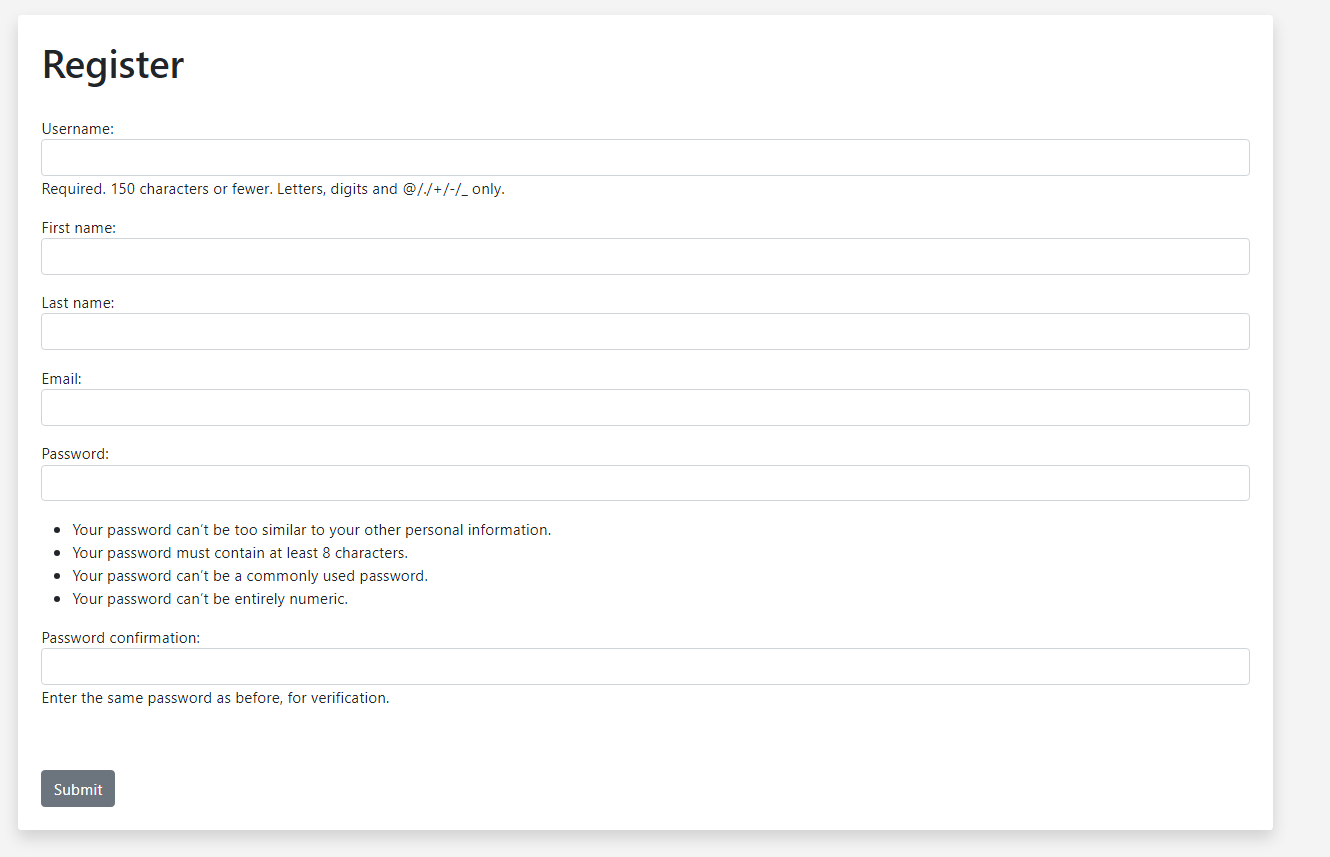
<title>My club</title>

    <style>

      body {background: #f4f4f4;}

    </style>

  </head>



# EP.27 Lock Down Your Django App

event/add\_event.html (add\_venue.html)

{% block content %}

{% if user.is\_authenticated %}

    <h1>Add Event</h1>

………………………………………………………………………………………………

{% else %}

        <h1>Sorry! You Myst Be Logged In To Wies This Page....</h1>

        <a href="{% url 'login' %}">Login Here</a>

{% endif %}

event/event\_list.html(venue.html) update\_event ,updatevenue, navebar

{% if user.is\_authenticated %}

              <div class="card-footer text-muted">

                <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

                <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

              </div>

{% endif %}

# EP.28 Determine Correct User To Edit Events

update\_event.html

{% if user.is\_authenticated %}

        {% if user == event.manager %}

        <h1>Update Event....</h1>

        <br>

        <form action="" method="POST">

        {%csrf\_token%}

                {{form.as\_p}}

                <br>

                <input type="submit" value="Update" class="btn btn-secondary">

        {% else %}

                <h2>Access Denied</h2>

                <p>Only the event's Manager can update this event...</p>

                Logged In User : {{user}}

                <br>

                Event Manager : {{event.manager}}

        {% endif %}

{% else %}

        <h1>Sorry! You Myst Be Logged In To Wies This Page....</h1>

        <a href="{% url 'login' %}">Login Here</a>

{% endif %}

Event\_list.html

{% if user.is\_authenticated %}

              {% if user == event.manager %}

                <div class="card-footer text-muted">

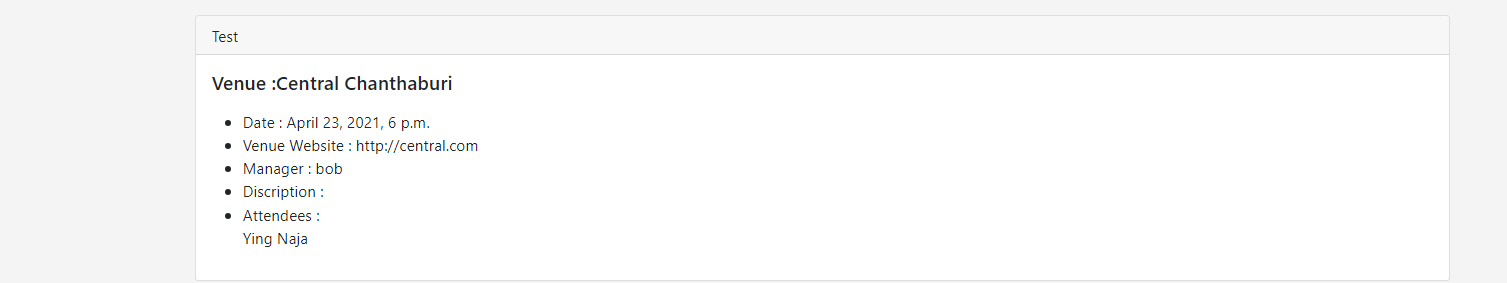
                  <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

                  <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

                </div>

              {% endif %}

            {% endif %}



# EP.29 Add Venue Owner

Event/Model.py

class Venue(models.Model):

    name = models.CharField('Venue Name',max\_length=120)

    address = models.CharField(max\_length=300)

    zip\_code = models.CharField('Zip Code', max\_length=15)

    phone = models.CharField('Contact Phone', max\_length=25)

    web = models.URLField('Website Address')

    email\_address = models.EmailField('Email Address')

    owner = models.IntegerField("Venue Owner", blank=False, default=1)

    def \_\_str\_\_(self):

        return self.name

views.py

def add\_venue(request):

    submitted = False

    if request.method == "POST":

        form = VenueForm(request.POST)

        if form.is\_valid():

            venue = form.save(commit=False)

            venue.owner = request.user.id

            venue.save()

            # form.save()

            return HttpResponseRedirect('/add\_venue?submitted=True')

    else:

        form = VenueForm

        if 'submitted' in request.GET:

            submitted = True

    return render(request, 'events/add\_venue.html', {'form':form,'submitted':submitted})

# EP.30 Use Different Forms For Different Users

Event/form.py

เปลี่ยน class EventForm เป็น EventFormAdmin

# User Event Form

class EventForm(ModelForm):

    class Meta:

        model = Event

        fields = ('name','event\_date','venue','attendees','description')

        labels = {

            'name': '',

            'event\_date': 'YYYY-MM-DD HH:MM:SS',

            'venue': 'Venue',

            'attendees': 'Attendees',

            'description': '',

        }

        widgets = {

            'name': forms.TextInput(attrs={'class':'form-control','placeholder':'Event Name'}),

            'event\_date': forms.TextInput(attrs={'class':'form-control','placeholder':'Event\_date'}),

            'venue': forms.Select(attrs={'class':'form-select','placeholder':'Venue'}),

            'attendees': forms.SelectMultiple(attrs={'class':'form-control','placeholder':'Attendees'}),

            'description': forms.Textarea(attrs={'class':'form-control','placeholder':'Description'}),

        }

Views.py

def add\_event(request):

    submitted = False

    if request.method == "POST":

        if request.user.is\_superuser:

            form = EventFormAdmin(request.POST)

            if form.is\_valid():

                form.save()

                return HttpResponseRedirect('/add\_event?submitted=True')

        else:

            form = EventForm(request.POST)

            if form.is\_valid():

                # form.save()

                event = form.save(commit=False)

                event.manager = request.user

                event.save()

                return HttpResponseRedirect('/add\_event?submitted=True')

    else:

        # just going to the page not submitting

        if request.user.is\_superuser:

            form = EventFormAdmin

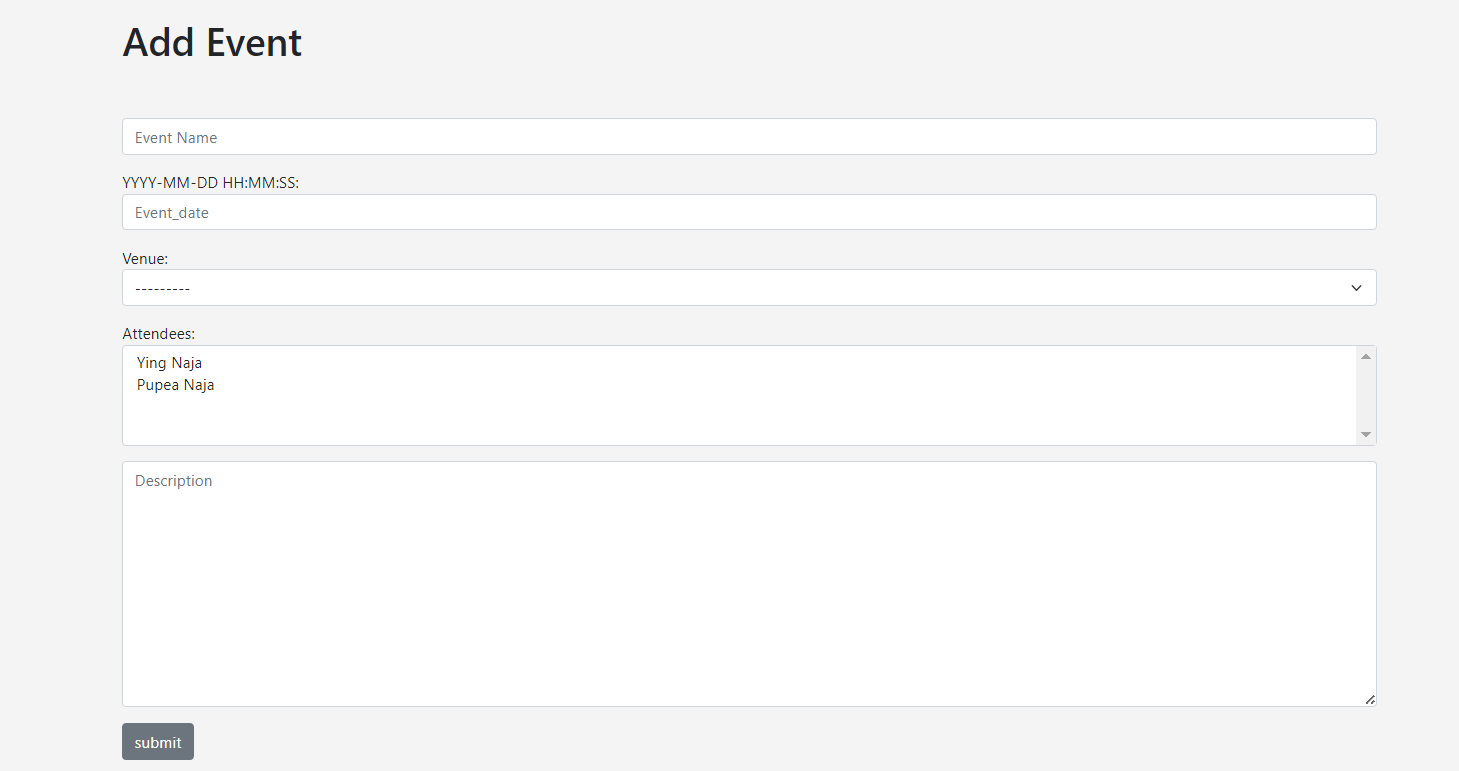
        else:

            form = EventForm

        if 'submitted' in request.GET:

            submitted = True

    return render(request, 'events/add\_event.html', {'form':form,'submitted':submitted})



# EP.31 Only Let Manger Delete Event

Event/views.py

def delete\_event(request, event\_id):

    event = Event.objects.get(pk=event\_id)

    if request.user == event.manager:

        event.delete()

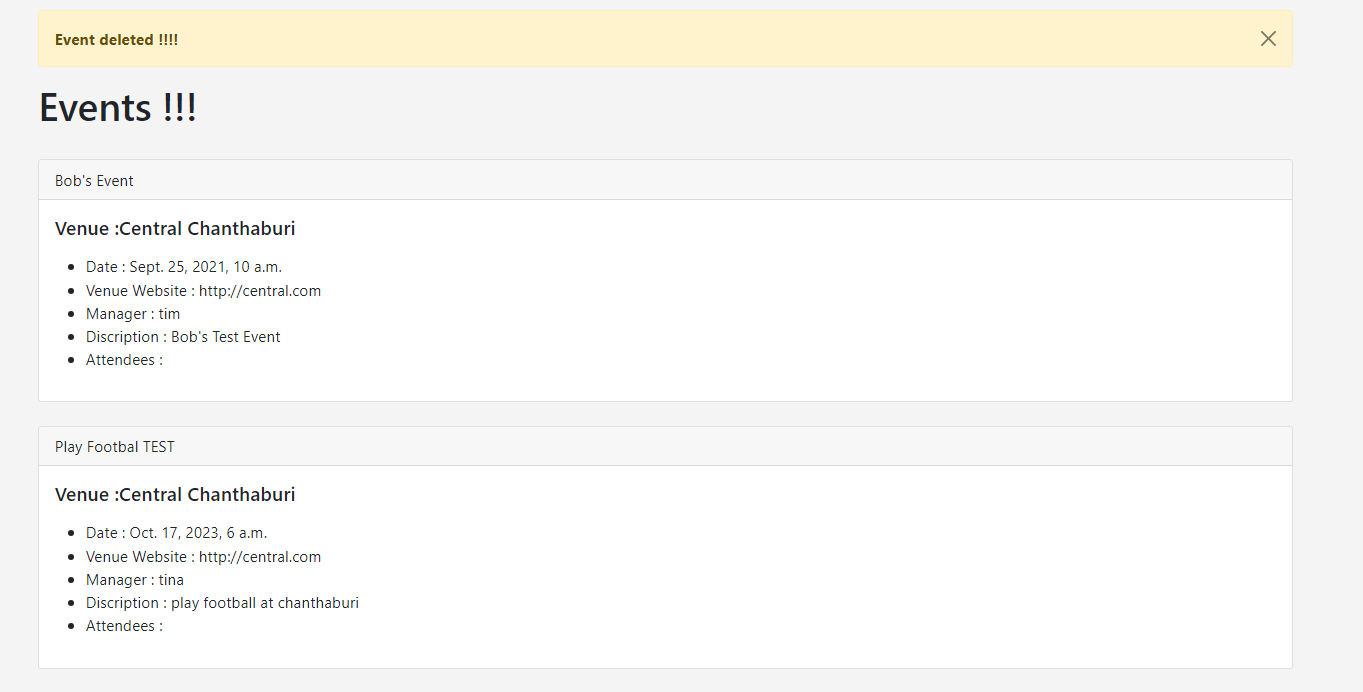
        messages.success(request, ("Event deleted !!!!"))

        return redirect('list-events')

    else:

        messages.success(request, ("You aren't Authorized To Delete This Event"))

        return redirect('list-events')



# EP.32 Access Venue Owner

Event/show\_venue.html

<p class="card-text">

                    <ul>

                        <li>{{ venue.address }}</li>

                        <li>{{ venue.zip\_code }}</li>

                        <li>{{ venue.phone }}</li>

                        <li>{{ venue.email\_address }}</li>

                        <li><a href="{{venue.web}}">{{ venue.web }}</a></li>

                        <li>Owner: {{ venue\_owner }} </li>

                        <li>Owner Email: {{ venue\_owner.email }} </li>

                    </ul>

Views.py

from django.contrib.auth.models import User

def show\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

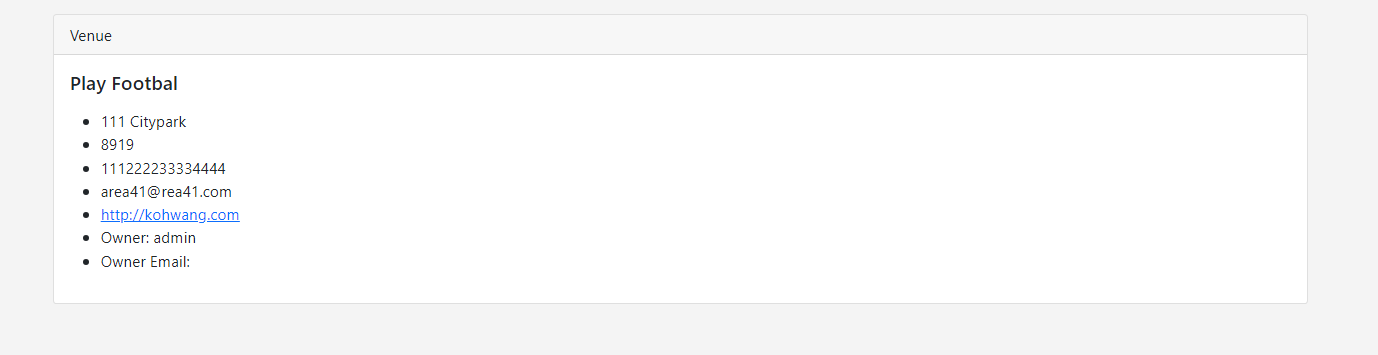
    venue\_owner = User.objects.get(pk=venue.owner)

    return render(request, 'events/show\_venue.html', {

        'venue':venue,

        'venue\_owner':venue\_owner

        })



# EP.33 Add My\_Events Page

Event/my\_event.html

{% extends 'events/base.html'%}

    {% block content %}

        <h2>My Event</h2>

    {% endblock content %}

Urls.py

path('my\_events',views.my\_events, name="my\_events"),

views.py

def my\_events(request):

    if request.user.is\_authenticated:

        me = request.user.id

        events = Event.objects.filter(attendees=me)

        return render(request, 'events/my\_events.html',{

            'events':events,

            'me':me})

    else:

        messages.success(request, ("You aren't Authorized To view this page"))

        return redirect('home')

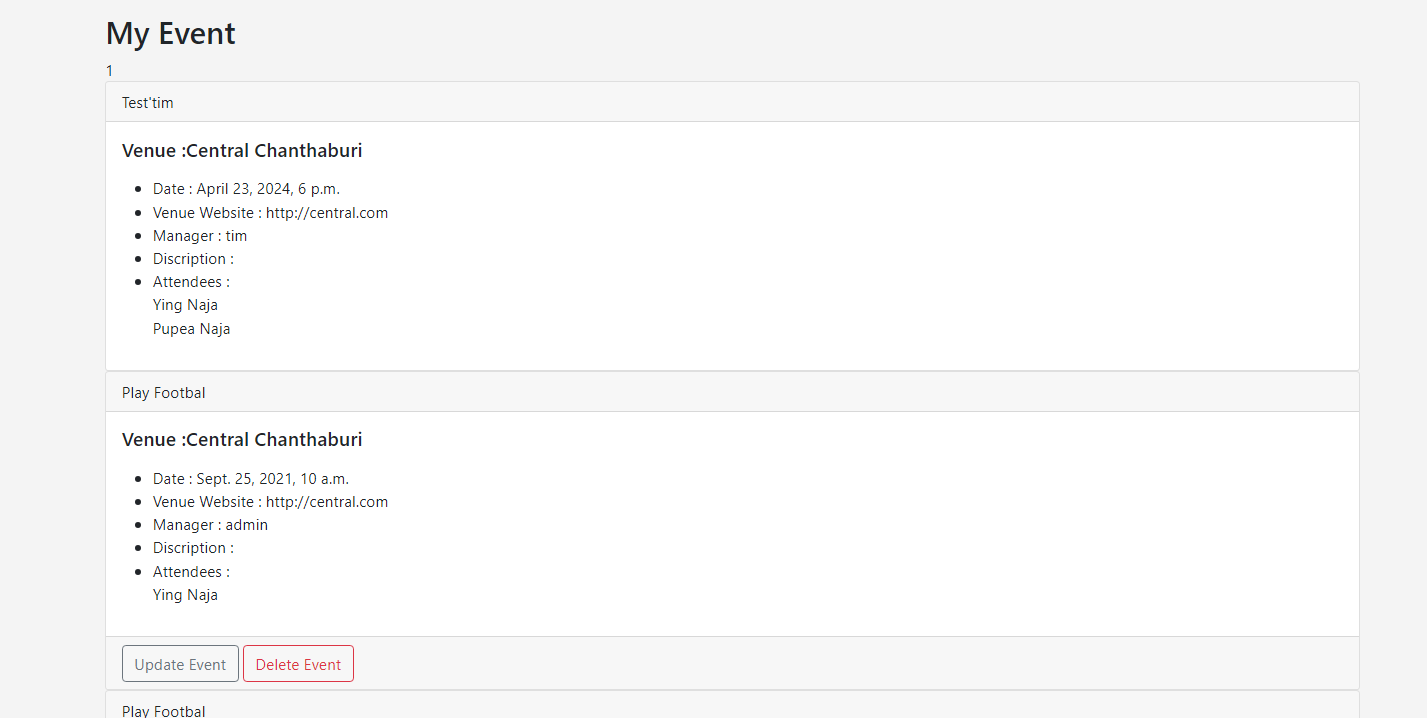
navbar.html

{% if user.is\_authenticated %}

              <li><a class="dropdown-item" href="{% url 'add-event' %}">Add Event</a></li>

              <li><a class="dropdown-item" href="{% url 'my\_events' %}">My Events</a></li>

              {% endif %}



# EP.34 Search Events

# Urls.py

path('search\_events',views.search\_events, name="search\_events"),

# views.py

def search\_events(request):

    if request.method == "POST":

        searched = request.POST['searched']

        events = Event.objects.filter(description\_\_contains=searched)

        return render(request, 'events/search\_events.html',{

            'searched':searched,

            'events':events})

    else:

        return render(request,'events/search\_events.html',{})

Event/search\_event.html

{% extends 'events/base.html'%}

    {% block content %}

    <div class="card">

        <div class="card-header">

            {% if searched %}

                You Searched For {{ searched }}

            {% else %}

                Search Event.....

            {% endif %}

   <br><br>

        <form method="POST" role="search" action="{% url 'search\_events' %}">

            {% csrf\_token %}

            <input class="form-control me-2" type="search" placeholder="Search Events" aria-label="Search" name="searched">

            <br>

            <button class="btn btn-outline-success" type="submit">Search</button>

        </form>

    </div></div><br>

                {% for event in events %}

                <div class="card">

                    <div class="card-header">

                      {{event}}

                    </div>

                    <div class="card-body">

                      <h5 class="card-title">Venue :{{event.venue}}</h5>

                      <p class="card-text">

                        <ul>

                            <li>Date : {{event.event\_date}}</li>

                            <li>Venue Website : {{event.venue.web}}</li>

                            <li>Manager : {{event.manager}}</li>

                            <li>Discription : {{event.description}}</li>

                           <li>Attendees : <br> {% for user in event.attendees.all %}</li>

                                                    {{user}}<br>

                                                {% endfor %}

                        </ul>

                      </p>

                    </div>

                        {% if user.is\_authenticated %}

                          {% if user == event.manager %}

                            <div class="card-footer text-muted">

                              <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

                              <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

                            </div>

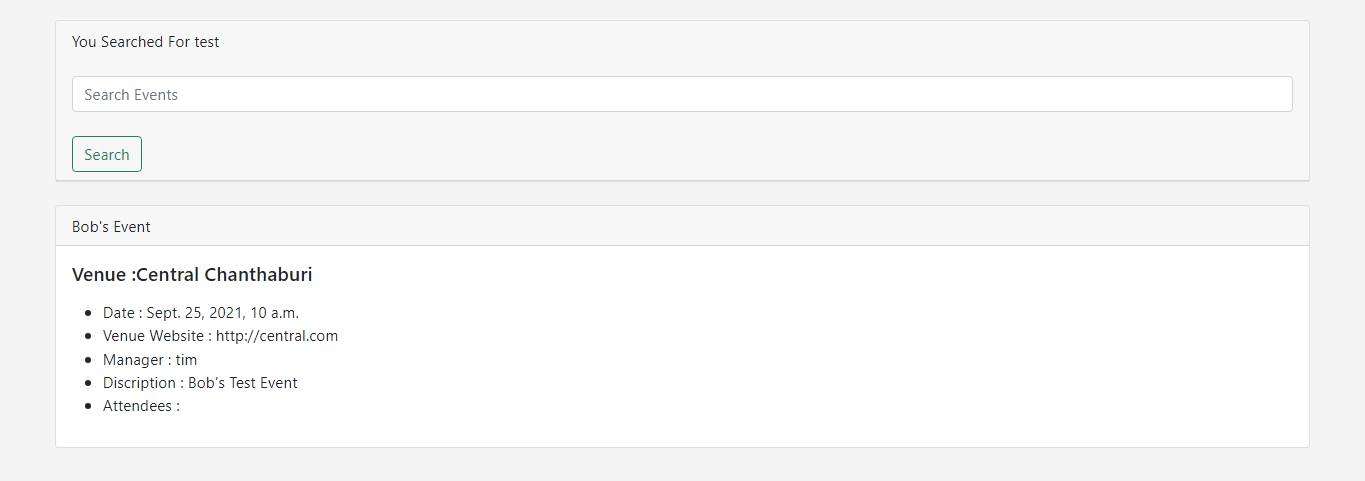
                          {% endif %}

                        {% endif %}

                  </div>

                {% endfor %}

    {% endblock content %}



# EP.35 Style Venue Search Results

# search\_venues.html

{% if searched %}

                <h1>You Searched For {{ searched }}</h1>

                {% for venue in venues %}

                <div class="card">

                    <div class="card-header">Venue</div>

                    <div class="card-body">

                        <h5>{{ venue }}</h5>

                        <p class="card-text">

                            <ul>

                                {% if venue.address %}

                                    <li>{{ venue.address }}</li>

                                {% endif %}

                                {% if venue.zip\_code %}

                                    <li>{{ venue.zip\_code }}</li>

                                {% endif %}

                                {% if venue.phone %}

                                    <li>{{ venue.phone }}</li>

                                {% endif %}

                                {% if venue.email\_address %}

                                    <li>{{ venue.email\_address }}</li>

                                {% endif %}

                                {% if venue.web %}

                                    <li><a href="{{venue.web}}">{{ venue.web }}</a></li>

                                {% endif %}

                                {% if venue.owner %}

                                    <li>Owner ID: {{ venue.owner }} </li>

                                {% endif %}

                            </ul>

                        </p>

                    </div>

                </div>

                {% endfor %}

            {% else %}

                <h1>Hey! You Forgot To Search For A Venue....</h1>

            {%  endif %}

# EP.36 Query Filter Events By Date

**Views.py**

 cal = HTMLCalendar().formatmonth(year, month\_number)

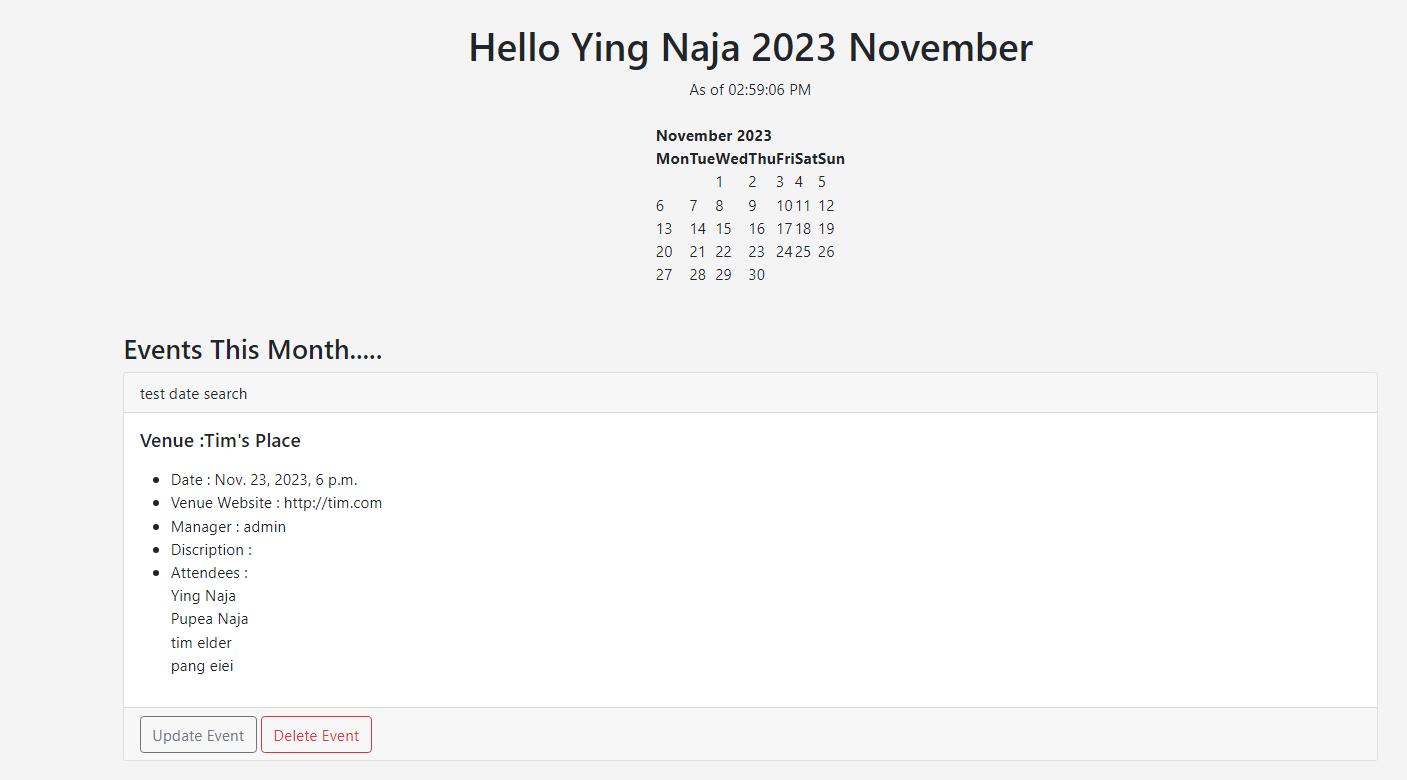
    now = datetime.now()

    current\_year = now.year

    time = now.strftime("%I:%M:%S %p")

    # Query the Events model for Dates

    event\_list = Event.objects.filter(event\_date\_\_year = year, event\_date\_\_month = month\_number)



Home.html

<h3>Events This Month.....</h3>

        {% if event\_list %}

            {% for event in event\_list %}

                <div class="card">

                    <div class="card-header">

                      {{event}}

                    </div>

                    <div class="card-body">

                      <h5 class="card-title">Venue :{{event.venue}}</h5>

                      <p class="card-text">

                        <ul>

                            <li>Date : {{event.event\_date}}</li>

                            <li>Venue Website : {{event.venue.web}}</li>

                            <li>Manager : {{event.manager}}</li>

                            <li>Discription : {{event.description}}</li>

                            <li>Attendees : <br> {% for user in event.attendees.all %}</li>

                                                    {{user}}<br>

                    {% endfor %}

                        </ul>

                      </p>

                    </div>

                        {% if user.is\_authenticated %}

                          {% if user == event.manager %}

                            <div class="card-footer text-muted">

                              <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

                              <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

                            </div>

                          {% endif %}

                        {% endif %}

                  </div>

                  <br>

            {% endfor %}

        {% else %}

            Sorry...There are no events this month....

        {% endif %}

# EP.37 Django 4.0 Release Notes!

# EP.38 Upload Images To Django

**settings.py**

import os

STATIC\_URL = '/static/'

MEDIA\_ROOT = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

STATICFILES\_DIRS = (

    os.path.join(BASE\_DIR, 'static'),

)

Urls.py

from django.conf import settings

from django.conf.urls.static import static

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('events.urls')),

    path('members/', include('django.contrib.auth.urls')),

    path('members/', include('members.urls')),

]+ static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

Model.py

class Venue(models.Model):

    name = models.CharField('Venue Name',max\_length=120)

    address = models.CharField(max\_length=300)

    zip\_code = models.CharField('Zip Code', max\_length=15)

    phone = models.CharField('Contact Phone', max\_length=25)

    web = models.URLField('Website Address')

    email\_address = models.EmailField('Email Address')

    owner = models.IntegerField("Venue Owner", blank=False, default=1)

    venue\_image = models.ImageField(upload\_to="images/", null=True, blank=True)

add\_venue.html

 {% else %}

            <form action="", method="POST" enctype="multipart/form-data">

                {% csrf\_token %}

                {{form.as\_p}}

       <input type="submit" value="submit" class="btn btn-secondary">

            </form>

Form.py

class VenueForm(ModelForm):

    class Meta:

        model = Venue

        fields = ('name','address','zip\_code','phone','web','email\_address','venue\_image')

        labels = {

            'name': '',

            'address': '',

            'zip\_code': '',

            'phone': '',

            'web': '',

            'email\_address': '',

            'venue\_image': '',

        }

Views.py

def add\_venue(request):

    submitted = False

    if request.method == "POST":

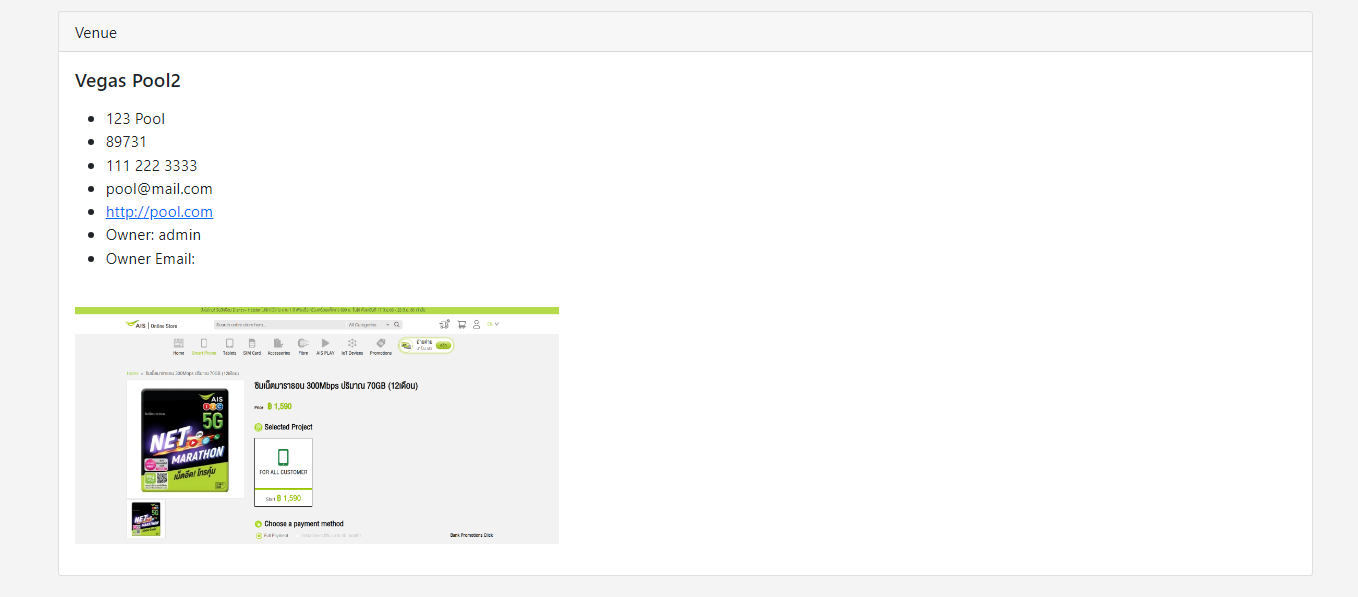
        form = VenueForm(request.POST, request.FILES)

show\_venue.html

{% if venue.venue\_image %}

                        <img src="{{ venue.venue\_image.url }}" width=500>

                    {% endif %}



# EP.39 Edit Uploaded Images

**views.py**

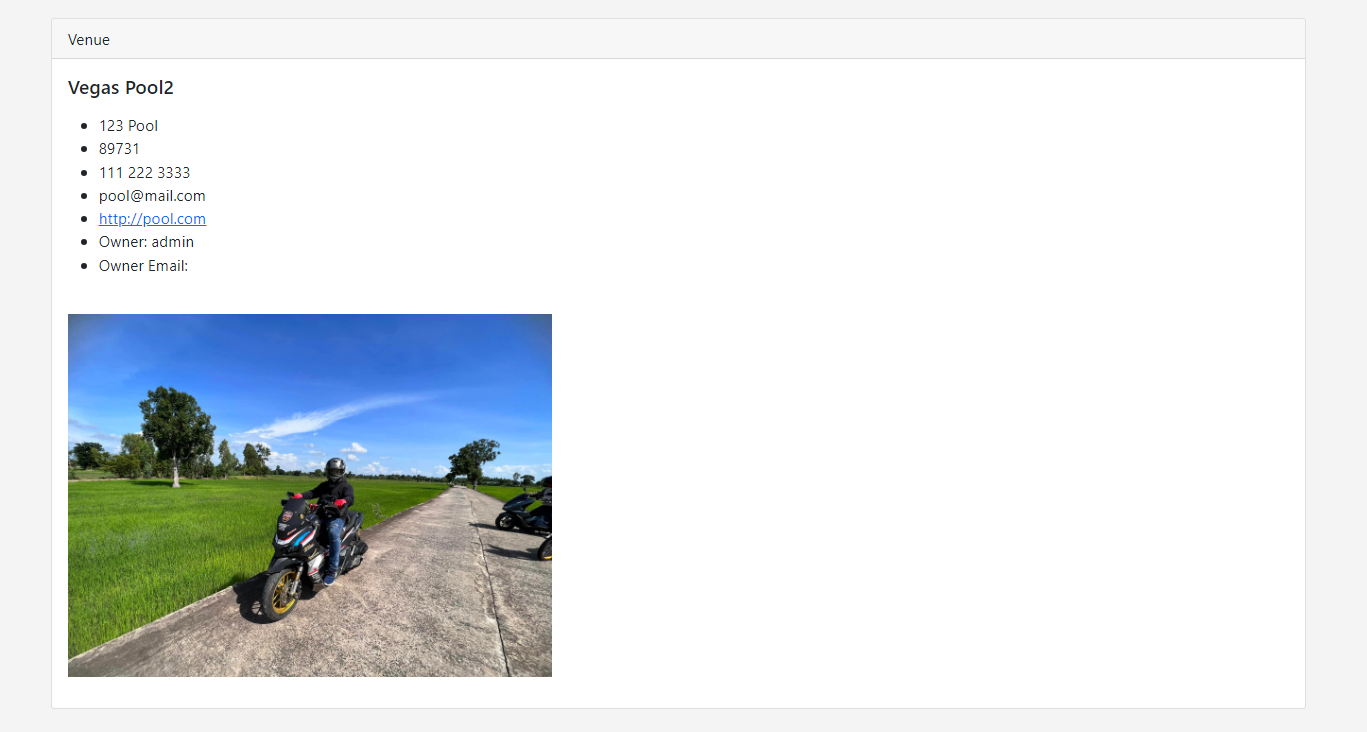
def update\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

    form = VenueForm(request.POST or None, request.FILES or None, instance=venue)

update\_vanue.html

<form action="" method="POST" enctype="multipart/form-data">



# EP.40 How To Add Calculated Fields To Models

**models.py**

class Event(models.Model):

    name = models.CharField('Event Name', max\_length=120)

    event\_date = models.DateTimeField('Event Date')

    # venue = models.CharField(max\_length=120)

    venue = models.ForeignKey(Venue, blank=True, null=True, on\_delete=models.CASCADE)

    # manager = models.CharField(max\_length=120)

    manager = models.ForeignKey(User, blank=True, null=True, on\_delete=models.SET\_NULL)

    description = models.TextField(blank=True)

    attendees = models.ManyToManyField(MyClubUser, blank=True)

    def \_\_str\_\_(self):

        return self.name

    @property

    def Days\_till(self):

        today = date.today()

        days\_till = self.event\_date.date() - today

        days\_till\_stripped = str(days\_till).split(",", 1)[0]

        return days\_till\_stripped

    @property

    def Is\_Past(self):

        today = date.today()

        if self.event\_date.date() < today:

            thing = "Past"

        else:

            thing = "Future"

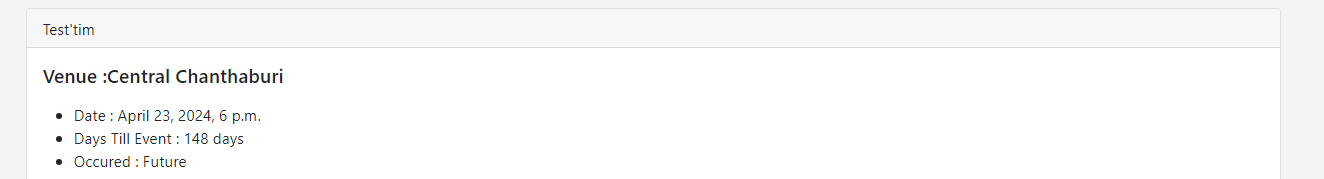
        return thing

event\_list.html

<li>Date : {{event.event\_date}}</li>

                <li>Days Till Event : {{event.Days\_till}}</li>

                <li>Occured : {{event.Is\_Past}}</li>

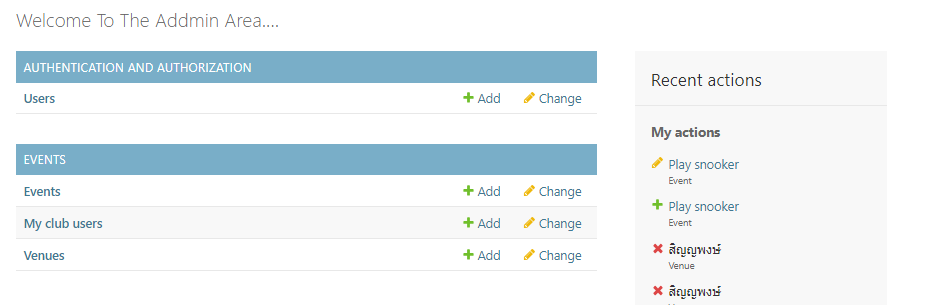


# EP.41 Remove Groups From Admin Site

**Models.py**

from django.contrib.auth.models import Group

admin.site.unregister(Group)



# EP.42 Event Approval

**Models**.py

class Event(models.Model):

    name = models.CharField('Event Name', max\_length=120)

    event\_date = models.DateTimeField('Event Date')

    # venue = models.CharField(max\_length=120)

    venue = models.ForeignKey(Venue, blank=True, null=True, on\_delete=models.CASCADE)

    # manager = models.CharField(max\_length=120)

    manager = models.ForeignKey(User, blank=True, null=True, on\_delete=models.SET\_NULL)

    description = models.TextField(blank=True)

    attendees = models.ManyToManyField(MyClubUser, blank=True)

    aprroved = models.BooleanField('Aprroved',default=False)

python mange.py makemigrations

python mange.py migrate

admin.py

@admin.register(Event)

class EventAdmin(admin.ModelAdmin):

    fields = (('name', 'venue'),'event\_date','description','manager','aprroved')

    list\_display = ('name','event\_date', 'venue')

    list\_filter = ('event\_date','venue')

    ordering = ('event\_date',)

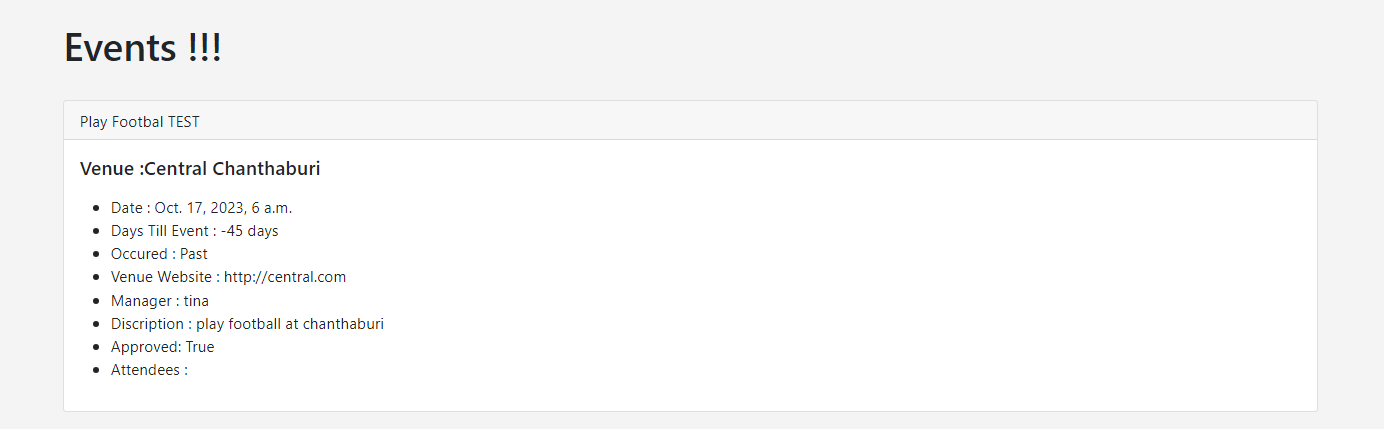
event\_list.html

{% for event in event\_list %}

      {% if event.aprroved %}

<li>Discription : {{event.description}}</li>

       <li>Approved: {{ event.aprroved }}</li>



# EP.43 Build An Event Approval Page With Checkboxes

Admin\_approval.html

{% extends 'events/base.html' %}

{% block content %}

    <h1>Admin Event Approval Portal</h1>

    <br>

    <table class="table table-striped table-hover table-bordered">

        <thead>

          <tr>

            <th scope="col">Event Name</th>

            <th scope="col">Approved</th>

          </tr>

        </thead>

        <tbody>

            {% for event in event\_list %}

            <tr>

                <td>{{event}}</td>

                <td>

                    {% if event.aprroved %}

                        <div class="form-check">

                         <input class="form-check-input" type="checkbox" checked="checked"

                                value={{event.id}} name="boxes">

                        </div>

                    {% else %}

                        <div class="form-check">

                            <input class="form-check-input" type="checkbox"

                                value={{event.id}} name="boxes">

                        </div>

                    {% endif %}

                </td>

            </tr>

            {% endfor %}

        </tbody>

    </table>

    <br>

    <button class="btn btn-secondary" type="submit">Update Approval</button>

{% endblock content %}

Urls.py

path('admin\_approval',views.admin\_approval, name="admin\_approval"),

views.py

def admin\_approval(request):

    event\_list = Event.objects.all().order\_by('-event\_date')

    if request.user.is\_superuser:

        if request.method == "POST":

            id\_list = request.POST.getlist('boxes')

            # Uncheck all events

            event\_list.update(aprroved=False)

            # Update database

            for x in id\_list:

                Event.objects.filter(pk=int(x)).update(aprroved=True)

            messages.success(request, "Event List Approval Updated!!!")

            return redirect('list-events')

        else:

            return render(request, 'events/admin\_approval.html',{

                'event\_list':event\_list,})

    else:

        messages.success(request, "You aren't authorized to view this page!")

        return redirect('home')

    return render(request, 'events/admin\_approval.html')

navbar.html

{% if user.is\_authenticated %}

      {% if user.is\_superuser %}

    <li class="nav-item">

<a class="nav-link" href="{% url 'admin\_approval' %}">Admin</a>

</li>

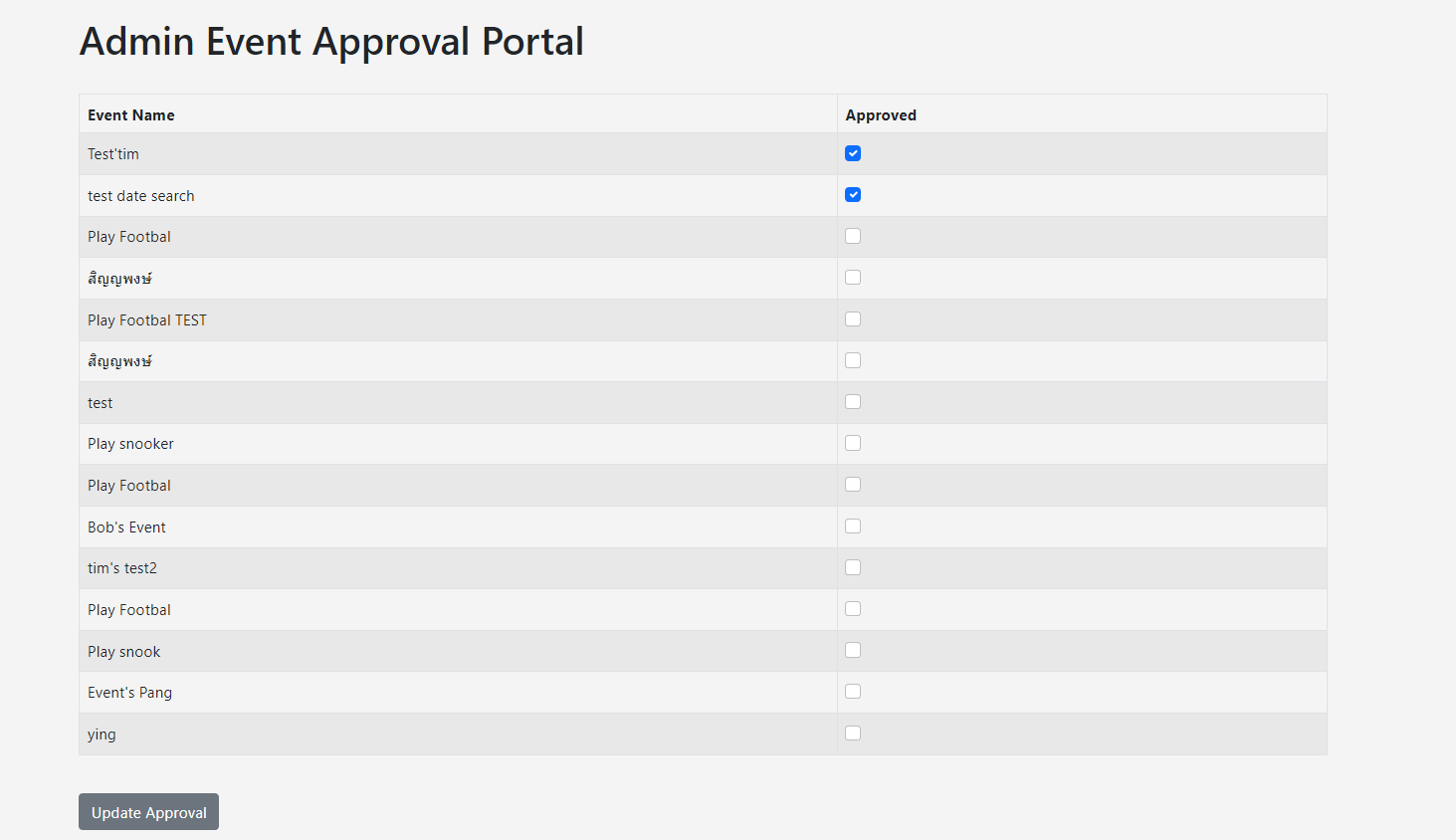
       {% endif %}

            <li class="nav-item">

              <a class="nav-link" href="{% url 'logout' %}">Logout</a>

            </li>

          {% else %}



# EP.44 Count The Number Of Events and Venues

viesw.py

def admin\_approval(request):

    event\_count = Event.objects.all().count()

    venue\_count = Venue.objects.all().count()

    user\_count = User.objects.all().count()

    event\_list = Event.objects.all().order\_by('-event\_date')

    if request.user.is\_superuser:

        if request.method == "POST":

            id\_list = request.POST.getlist('boxes')

            # Uncheck all events

            event\_list.update(aprroved=False)

            # Update database

            for x in id\_list:

                Event.objects.filter(pk=int(x)).update(aprroved=True)

            messages.success(request, "Event List Approval Updated!!!")

            return redirect('list-events')

        else:

            return render(request, 'events/admin\_approval.html',{

                'event\_list':event\_list,

                'event\_count':event\_count,

                'venue\_count':venue\_count,

                'user\_count':user\_count

                })

    else:

        messages.success(request, "You aren't authorized to view this page!")

        return redirect('home')

    return render(request, 'events/admin\_approval.html')

admin\_approval.html

<form action="" method="POST">

        {% csrf\_token %}

        <div class="card">

            <div class="card-header">

              Event - Venue - User Count List

            </div>

            <div class="card-body">

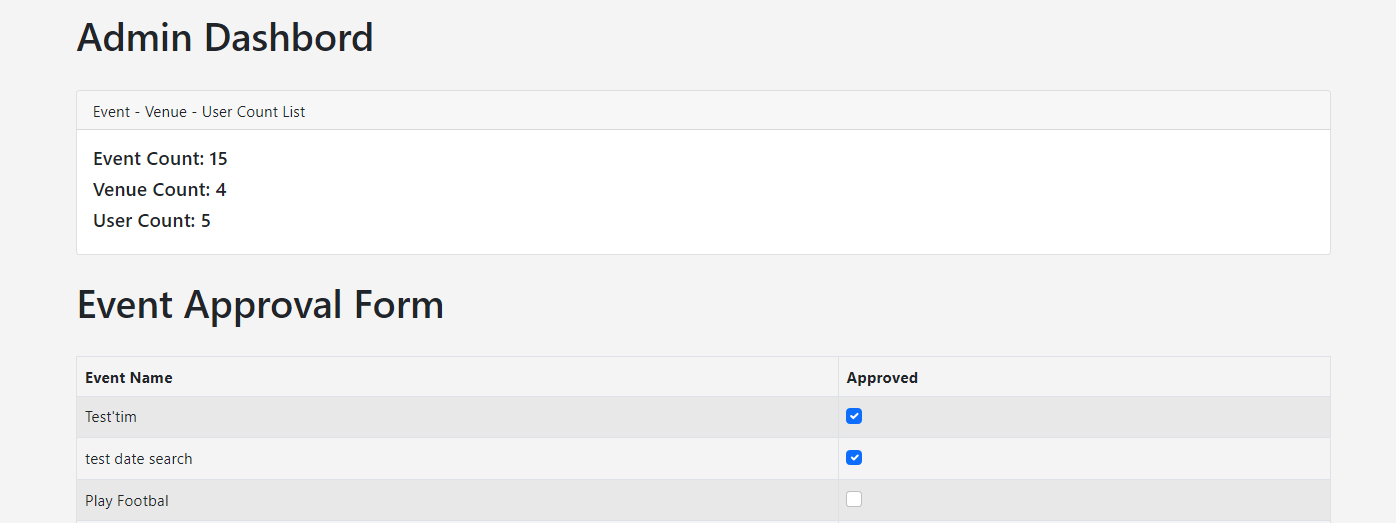
              <h5 class="card-title">Event Count: {{ event\_count }}</h5>

              <h5 class="card-title">Venue Count: {{ venue\_count }}</h5>

              <h5 class="card-title">User Count: {{ user\_count }}</h5>

            </div>

          </div>



# EP.45 Show All Events By Venue

viesw.py

def admin\_approval(request):

    # Get the venues

    venue\_list = Venue.objects.all()

    # Show Count

    event\_count = Event.objects.all().count()

    venue\_count = Venue.objects.all().count()

    user\_count = User.objects.all().count()

return render(request, 'events/admin\_approval.html',{

                'event\_list':event\_list,

                'event\_count':event\_count,

                'venue\_count':venue\_count,

                'user\_count':user\_count,

                'venue\_list':venue\_list,

                })

# Admin\_approval.html

<table class="table table-striped table-hover table-bordered">

        <thead>

          <tr>

            <th scope="col">Vanue Name</th>

            <th scope="col">Events</th>

          </tr>

        </thead>

        <tbody>

            {% for venue in venue\_list %}

            <tr>

                <td><a href="{% url 'show-venue' venue.id %}">{{ venue }}</a></td>

                <td><a class="btn btn-sm btn-secondary"

                    href="{% url 'venue-event' venue.id %}">List Events</a></td>

            </tr>

            {% endfor %}

        </tbody>

    </table>

# Views.py

def venue\_events(request, venue\_id):

    venue = Venue.objects.get(id=venue\_id)

    events = venue.event\_set.all()

    if events:

        return render(request, 'events/venue\_event.html',{

            'events':events,

        })

    else:

        messages.success(request, "That Venue Has No Events!")

        return redirect('admin\_approval')

# urls.py

path('venue\_events/<venue\_id>',views.venue\_events, name="venue-event"),

# venue\_event.html

{% extends 'events/base.html' %}

{% block content %}

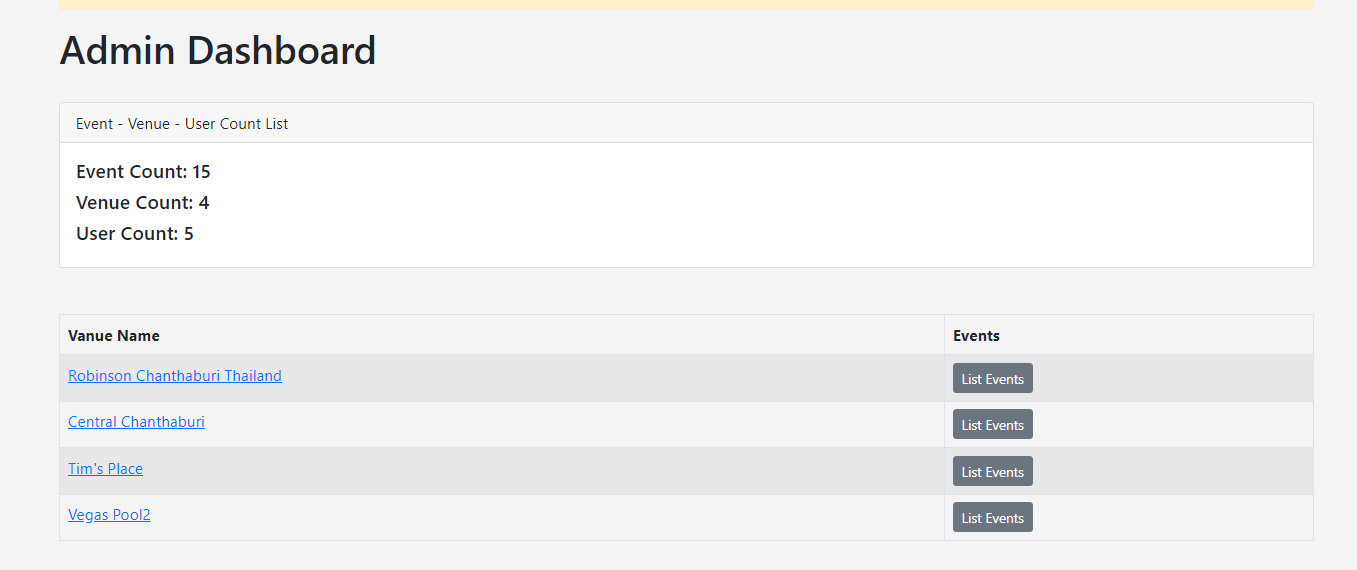
    <h1>List Events</h1>

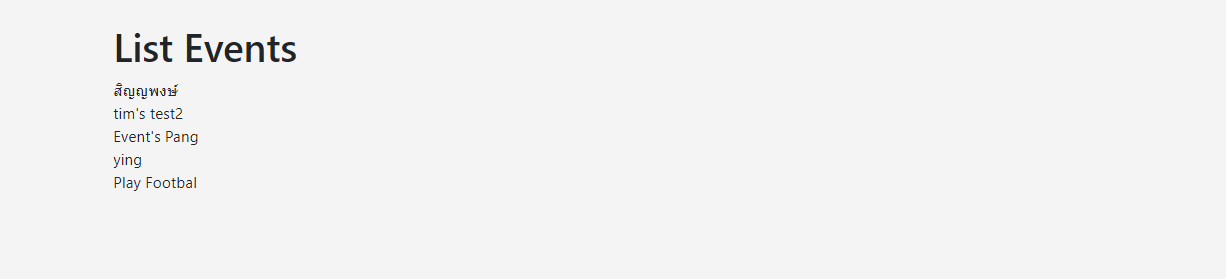
    {% for event in events %}

        {{event}}<br>

    {% endfor %}

{% endblock content %}





Urls.py

path('show\_event/<event\_id>',views.show\_event, name="show-even")

veiws.py

def show\_event(request, event\_id):

    event = Event.objects.get(pk=event\_id)

    return render(request, 'events/show\_event.html',{

            'event':event,})

Show\_event.html

<h1>Show Event</h1>

      {% if event.aprroved %}

       <br>

      <div class="card">

        <div class="card-header">

          {{event}}

        </div>

        <div class="card-body">

          <h5 class="card-title">Venue :{{event.venue}}</h5>

          <p class="card-text">

            <ul>

                <li>Date : {{event.event\_date}}</li>

                <li>Days Till Event : {{event.Days\_till}}</li>

                <li>Occured : {{event.Is\_Past}}</li>

                <li>Venue Website : {{event.venue.web}}</li>

                <li>Manager : {{event.manager}}</li>

                <li>Discription : {{event.description}}</li>

                <li>Approved: {{ event.aprroved }}</li>

                <li>Attendees : <br> {% for user in event.attendees.all %}</li>

                                        {{user}}<br>

                                    {% endfor %}

            </ul>

          </p>

        </div>

            {% if user.is\_authenticated %}

              {% if user == event.manager %}

                <div class="card-footer text-muted">

                  <a href="{% url 'update-event' event.id %}" class="btn btn-outline-secondary">Update Event</a>

                  <a href="{% url 'delete-event' event.id %}" class="btn btn-outline-danger">Delete Event</a>

                </div>

              {% endif %}

            {% endif %}

      </div>

      {% endif %}

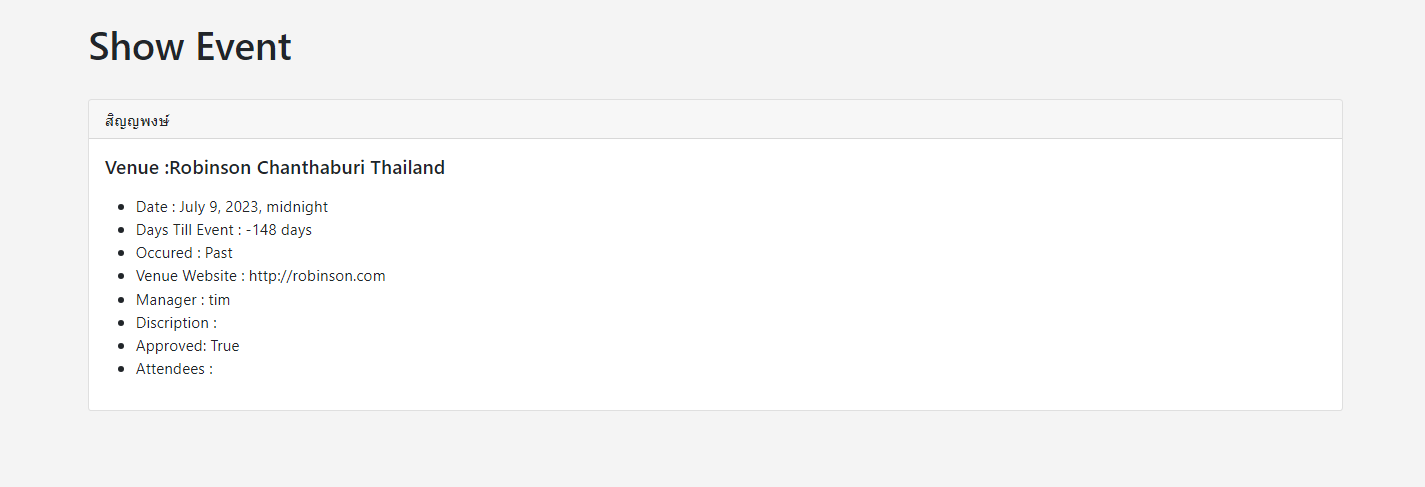
Venue\_event.html

<h1>List Events</h1>

    {% for event in events %}

        <a href="{% url 'show-event' event.id %}">{{event}}</a> <br>

    {% endfor %}



# EP.47 List Events On Venue Pages

viesw.py

def show\_venue(request, venue\_id):

    venue = Venue.objects.get(pk=venue\_id)

    venue\_owner = User.objects.get(pk=venue.owner)

    # Grab the events from that venue

    events = venue.event\_set.all()

    return render(request, 'events/show\_venue.html', {

        'venue':venue,

        'venue\_owner':venue\_owner,

        'events':events,

        })

Show\_venue.html

<h1>Event At This Venue....</h1>

                {% if events %}

                {% for event in events %}

                    <a href="{% url 'show-event' event.id %}">{{event}}</a> <br>

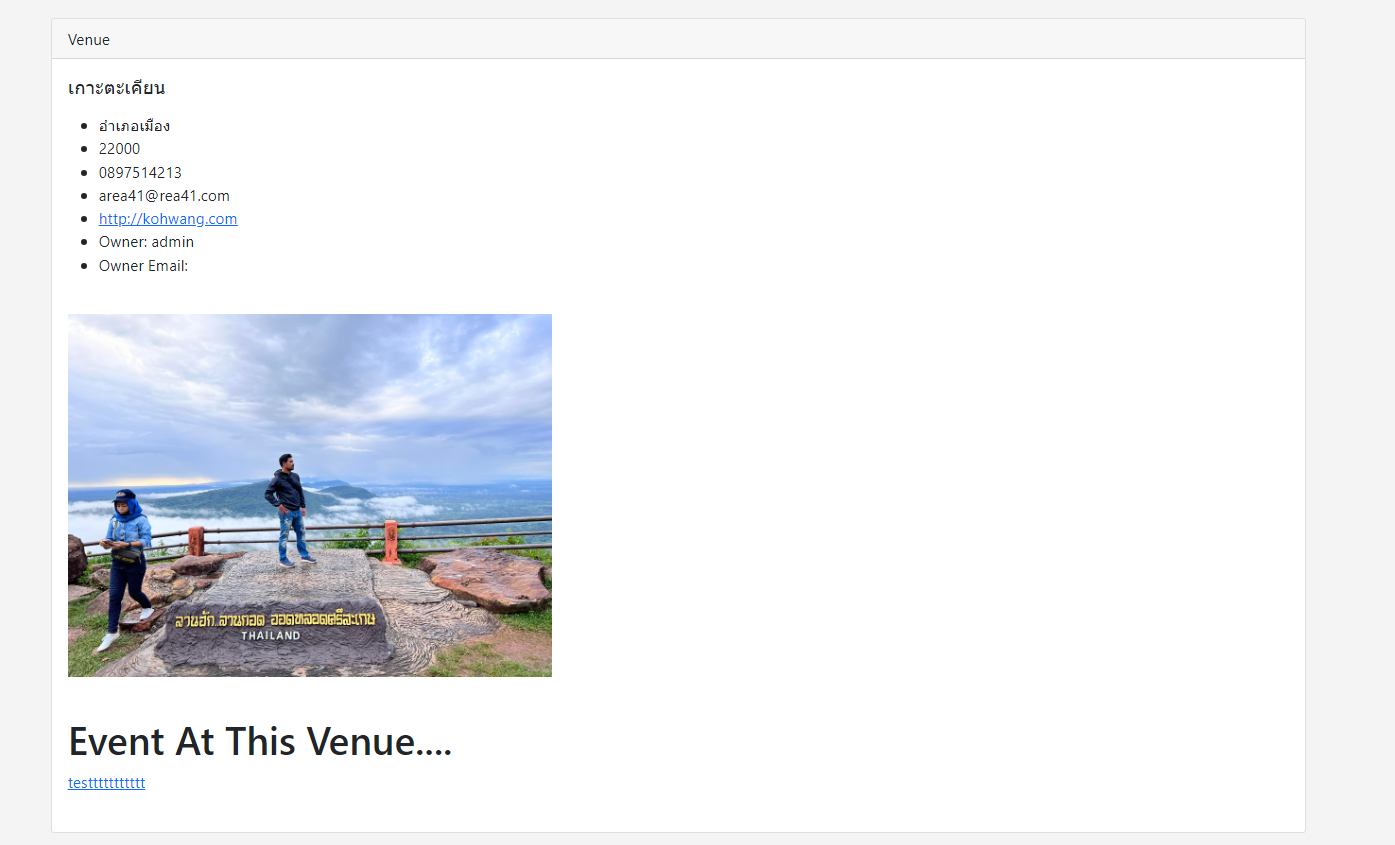
                {% endfor %}

                <br>

                {% else %}

                   <h4>No Event Scheduled At Thie Venue</h4>

                {% endif %}



# EP.46 Build A Website With Django 4.0!