投票应用
综合表单，模板，视图，数据库等知识，创建一个投票应用。

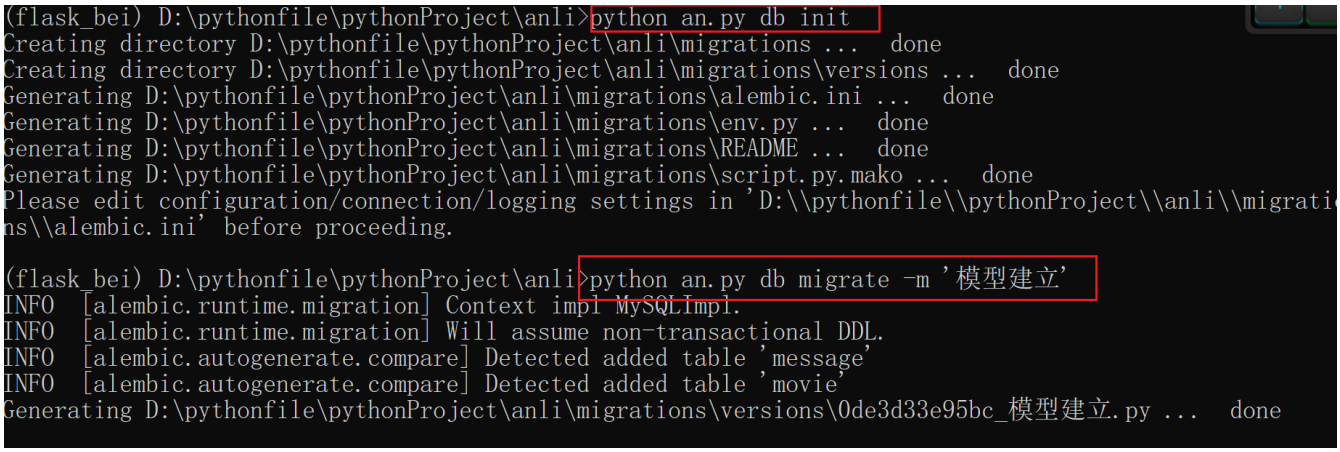
1.创建模型类

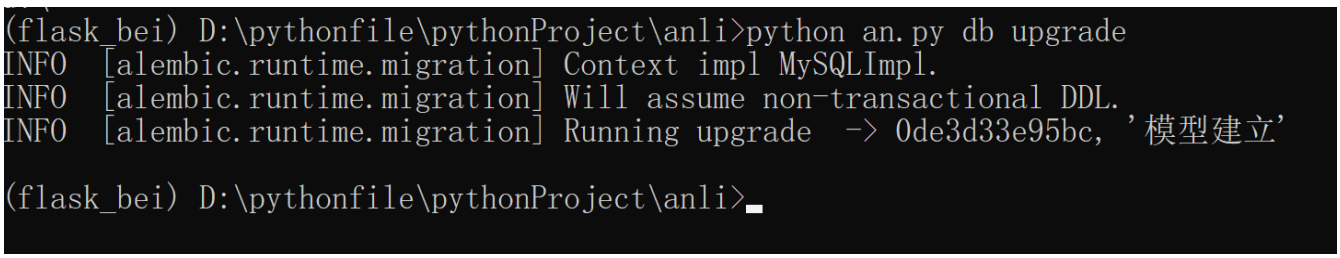
```
create database anli charset=utf8;
```

```
# -*- encoding: utf-8 -*-

# coding=utf-8
""" 电影投票应用 """
from datetime import datetime,timedelta
from flask import Flask,render_template,request,session,redirect,url_for
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate, MigrateCommand
from flask_script import Shell,Manager
from flask_wtf import FlaskForm
from wtforms import StringField,SubmitField
from wtforms.validators import DataRequired
import time
app = Flask(__name__)
app.config["DEBUG"] = True
app.config["SECRET_KEY"] ='SDAGSDGASDGASDGASDGDSAGDSA'

# 将flask对象交给扩展命令行工具管理
manager = Manager(app)

# 连接数据库
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql+pymysql://root:qwe123@127.0.0.1:3306/anli'
# 自动提交数据库中的改动
app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True
# 追踪对象
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True

# 创建数据库对象
db = SQLAlchemy(app)
```

```
# 第一个参数是Flask的实例，第二个参数是 Sqlalchemy数据库实例
Migrate(app, db)
# manager 是Flask-Script的实例，这条语句在flask-Script中添加一个db命令
manager.add_command('db', MigrateCommand)


class Form(FlaskForm):
    content = StringField(label='留言板',validators=[DataRequired()])
    submit = SubmitField(label='提交')


class Movie(db.Model):
    """ 创建电影模型类"""
    # id
    id = db.Column(db.Integer, primary_key=True)
    # 电影名称
    name = db.Column(db.String(40), unique=True)
    # 演员
    cast = db.Column(db.String(100))
    # 票数
    votes = db.Column(db.Integer, default=0)


class Message(db.Model):
    __tablename__ = 'message'
    id = db.Column(db.Integer,primary_key=True)
    content = db.Column(db.String(300))
    time = db.Column(db.DateTime, default=datetime.now, nullable=False)


if __name__ == '__main__':
    manager.run()
```

```
python an.py db init

python an.py db migrate -m '模型建立'

python an.py db upgrade
```

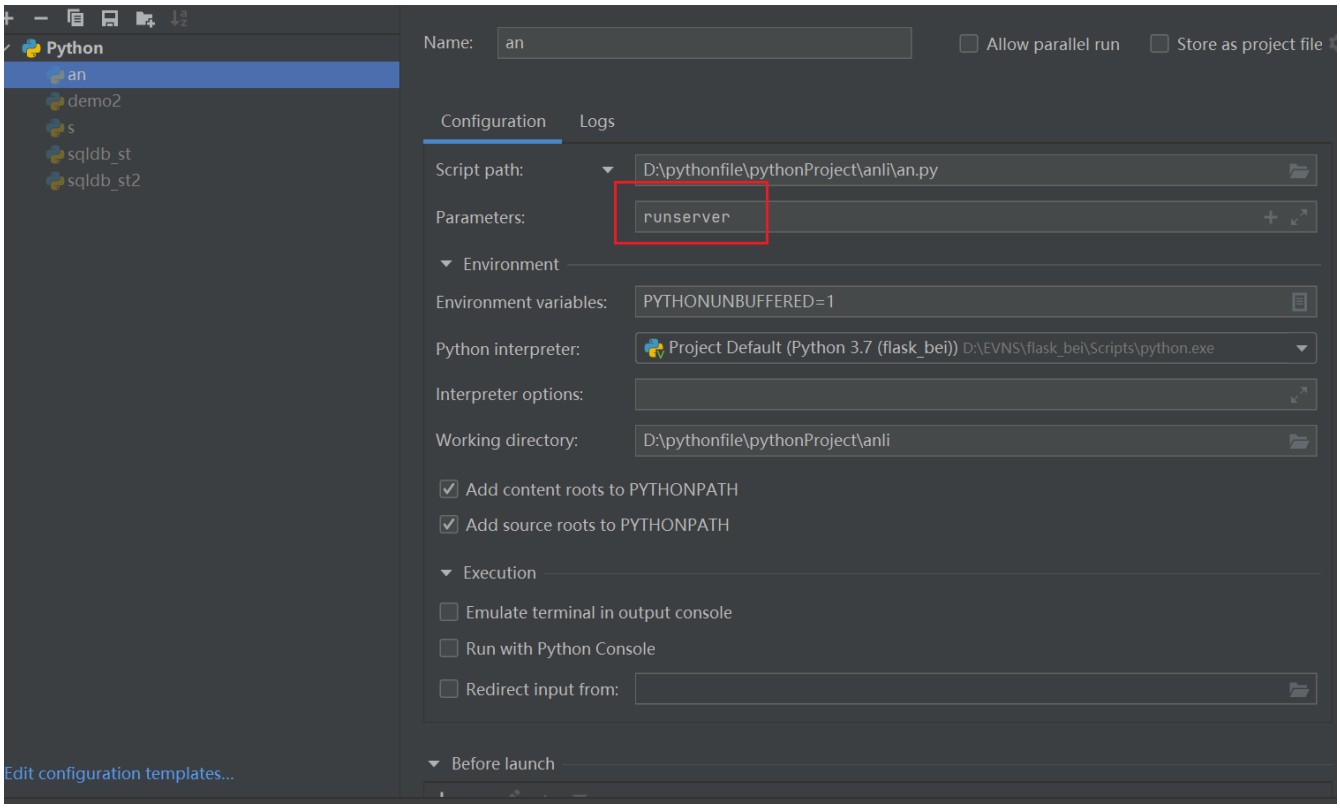




2.数据

```
insert into movie values(0,'前任3:再见前任','主演：韩庚 姚星彤 郑 恺 丽坤 李相烨',0);
insert into movie values(0,'霸王别姬','主演：张国荣 张丰毅 巩俐 英达 葛优',0);
insert into movie values(0,'芳华','黄轩领衔主演，苗苗、钟楚曦联合主演',0);
insert into movie values(0,'追龙','主演：甄子丹 刘德华 胡然 徐冬冬 伍允龙',0);
insert into movie values(0,'一代宗师','主演：梁朝伟 章子怡 张震宋 慧乔 赵本山',0);
```

3.视图




```
# -*- encoding: utf-8 -*-
""" 电影投票应用 """
from datetime import datetime,timedelta
from flask import Flask,render_template,request,session,redirect,url_for
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate, MigrateCommand
from flask_script import Shell,Manager
from flask_wtf import FlaskForm
from wtforms import StringField,SubmitField
from wtforms.validators import DataRequired
import time
app = Flask(__name__)
app.config["DEBUG"] = True
app.config["SECRET_KEY"] ='SDAGSDGASDGASDGASDGDSAGDSA'

# 将flask对象交给扩展命令行工具管理
manager = Manager(app)

# 连接数据库
app.config['SQLALCHEMY_DATABASE_URI']
= 'mysql+pymysql://root:qwe123@127.0.0.1:3306/anli'
# 自动提交数据库中的改动
app.config['SQLALCHEMY_COMMIT_ON_TEARDOWN'] = True
# 追踪对象
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = True
```

```
# 创建数据库对象
db = SQLAlchemy(app)
# 第一个参数是Flask的实例，第二个参数是 Sqlalchemy数据库实例
Migrate(app, db)
# manager 是Flask-Script的实例，这条语句在flask-Script中添加一个db命令
manager.add_command('db', MigrateCommand)


class Form(FlaskForm):
    content = StringField(label='留言板',validators=[DataRequired()])
    submit = SubmitField(label='提交')


class Movie(db.Model):
    """ 创建电影模型类"""
    # id
    id = db.Column(db.Integer, primary_key=True)
    # 电影名称
    name = db.Column(db.String(40), unique=True)
    # 演员
    cast = db.Column(db.String(100))
    # 票数
    votes = db.Column(db.Integer, default=0)


class Message(db.Model):
    __tablename__ = 'message'
    id = db.Column(db.Integer,primary_key=True)
    content = db.Column(db.String(300))
    time = db.Column(db.DateTime, default=datetime.now, nullable=False)


@app.route('/', methods=['GET', 'POST'])
def index():
    # 创建一个表单对象
    form = Form()
    # 将所有电影查询出来
    movie = Movie.query.all()
    # 将所有的留言按照id降序查询出来
    message = Message.query.order_by(db.desc(Message.id))
    # 每次都查询下session看是否有投票过，用来判断是否显示投票结果
    is_vote = session.get('is_vote')
    context = {
        'form': form,
        'movie': movie,
```

```
        'message': message,
        'is_vote': is_vote
    }
    if form.validate_on_submit():  # 是否提交了留言数据
        """验证通过"""
        content = form.content.data
        msg = Message(content=content)  # # 获取留言数据
        db.session.add(msg)
        db.session.commit()
    return render_template('index.html', **context)


@app.route('/votes/<int:id>')
def voted(id):
    """ 投票计数"""
    # 第一次请求没有session或者session保存的时间戳下于当前时间则票数加1
    is_vote = session.get('is_vote')
    if not (is_vote and float(is_vote) > time.time()):
        # 将对应电影查询出来
        m = Movie.query.get(id)
        m.votes += 1
        db.session.add(m)
        db.session.commit()
        # Flask 中session 是保存在cookie中，默认是关闭浏览器就超时，这里设置
session的超时时间
        session.permanent = True
        app.permanent_session_lifetime = timedelta(seconds=60*60*24)
        # 保存一个session 一天之后的时间戳
        session['is_vote'] = time.time() + 60*60*24
    # 投票成功，重定向回 index视图
    return redirect(url_for('index'))


if __name__ == '__main__':
    manager.run()
```

4.模板

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>投票</title>
</head>
<body>

        {% for m in movie %}
            <dt>{{ m.name }}:</dt>
            <dd>{{ m.cast }}</dd>
            <a href="{{ url_for('voted',id=m.id) }}">点击投票</a> <br>
<br>
        {% endfor %}
        <hr>

        <span>投票结果：</span>
        {% if is_vote %}
            {% for m in movie %}
                <p>{{ m.name }} 得票 : <span>{{ m.votes }}</span></p>
            {% endfor %}
        {% else %}
            <span>需要投票之后才能查看结果</span>
        {% endif %}
        <hr>

        <form action="/" method="post">
            {{ form.csrf_token }}
            {{ form.content.label }}
            {{ form.content }}
            {{ form.submit }}
        </form>
        <dt>留言：</dt>
        {% for msg in message %}
            <dt>网友{{ msg.time }} 留言:</dt>
            <dd>{{ msg.content }}</dd>
            <br>
        {% endfor %}

</body>
</html>
```

前任3:再见前任:
主演：韩庚 姚星彤 郑 恺 丽坤 李相烨
点击投票

霸王别姬:
主演：张国荣 张丰毅 巩俐 英达 葛优
点击投票

芳华:
黄轩领衔主演，苗苗、钟楚曦联合主演
点击投票

追龙:
主演：甄子丹 刘德华 胡然 徐冬冬 伍允龙
点击投票

一代宗师:
主演：梁朝伟 章子怡 张震宋 慧乔 赵本山
点击投票

投票结果： 需要投票之后才能查看结果

留言板 提交
留言：