import os  
from app import db  
from datetime import datetime  
from werkzeug.security import generate\_password\_hash, check\_password\_hash  
from flask\_login import UserMixin  
from app import login  
from hashlib import md5  
from time import time  
import jwt  
from app import app  
  
  
class User(UserMixin, db.Model):  
 id = db.Column(db.Integer, primary\_key=True)  
 username = db.Column(db.String(64), index=True, unique=True)  
 email = db.Column(db.String(120), index=True, unique=True)  
 password\_hash = db.Column(db.String(128))  
 about\_me = db.Column(db.String(140))  
 last\_seen = db.Column(db.DateTime, default=datetime.utcnow)  
  
 def \_\_repr\_\_(self):  
 return '<User {}>'.format(self.username)  
  
 def set\_password(self, password):  
 self.password\_hash = generate\_password\_hash(password)  
  
 def check\_password(self, password):  
 return check\_password\_hash(self.password\_hash, password)  
  
 def avatar(self, size):  
 digest = md5(self.email.lower().encode('utf-8')).hexdigest()  
 return 'https://www.gravatar.com/avatar/{}?d=identicon&s={}'.format(  
 digest, size)  
  
 def get\_reset\_password\_token(self, expires\_in=600):  
 return jwt.encode(  
 {'reset\_password': self.id, 'exp': time() + expires\_in},  
 app.config['SECRET\_KEY'], algorithm='HS256')  
  
 @staticmethod  
 def verify\_reset\_password\_token(token):  
 try:  
 id = jwt.decode(token, app.config['SECRET\_KEY'],  
 algorithms=['HS256'])['reset\_password']  
 except:  
 return  
 return User.query.get(id)  
  
  
class Post(db.Model):  
 id = db.Column(db.Integer, primary\_key=True)  
 car\_brand = db.Column(db.String(140))  
 model = db.Column(db.String(140))  
 year = db.Column(db.String(140))  
 price = db.Column(db.String(140))  
 engine = db.Column(db.String(140))  
 transmission = db.Column(db.String(140))  
 power = db.Column(db.String(140))  
 location = db.Column(db.String(140))  
 color = db.Column(db.String(140))  
 mileage = db.Column(db.String(140))  
 description = db.Column(db.String(1000))  
  
 def \_\_repr\_\_(self):  
 return '<Post {} {} {} {} {} {} {} {} {} {} {}>'.format(self.car\_brand, self.model, self.year, self.price, \  
 self.engine, self.transmission, self.power, self.location, \  
 self.color, self.mileage, self.description)  
  
  
@login.user\_loader  
def load\_user(id):  
 return User.query.get(int(id))  
  
  
class Config(object):  
 MAIL\_SERVER = os.environ.get('MAIL\_SERVER')  
 MAIL\_PORT = int(os.environ.get('MAIL\_PORT') or 25)  
 MAIL\_USE\_TLS = os.environ.get('MAIL\_USE\_TLS') is not None  
 MAIL\_USERNAME = os.environ.get('MAIL\_USERNAME')  
 MAIL\_PASSWORD = os.environ.get('MAIL\_PASSWORD')  
 ADMINS = ['businesswizy@gmail.com']