from datetime import datetime

from hired import db, login\_manager

from flask\_login import UserMixin

@login\_manager.user\_loader

def load\_user(user\_id):

if Recruiter.query.get(int(user\_id)):

return Recruiter.query.get(int(user\_id))

else:

return User.query.get(int(user\_id))

#metadata = db.MetaData()

#Job\_User = db.Table('Job\_User', metadata,

# db.Column('job\_id', db.Integer, db.ForeignKey('job.id'), nullable=True),

# db.Column('user\_id', db.Integer, db.ForeignKey('user.id'), nullable=True)

# )

class Job\_User(db.Model):

id = db.Column(db.Integer, primary\_key=True)

job\_id = db.Column(db.Integer, db.ForeignKey('job.id'), nullable=True)

user\_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=True)

def \_\_repr\_\_(self):

return f"Job\_User('{self.job\_id}', '{self.user\_id}')"

class User(db.Model, UserMixin):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(50), unique=True, nullable=False)

email = db.Column(db.String(120), unique=True, nullable=False)

image\_file = db.Column(db.String(20), nullable=False, default='default.jpg')

password = db.Column(db.String(60), nullable=False)

degree\_type = db.Column(db.String(7), nullable=True)

experience = db.Column(db.Integer, nullable=True)

major = db.Column(db.String(30), nullable=True)

skill\_1 = db.Column(db.String(30), nullable=True)

skill\_1\_level = db.Column(db.Integer, nullable=True)

dbms = db.Column(db.String(10), nullable=True)

jobs = db.relationship('Job\_User', backref='job', lazy=True)

def \_\_repr\_\_(self):

return f"User('{self.username}', '{self.email}', '{self.image\_file}', '{self.degree\_type}')"

class Job(db.Model):

id = db.Column(db.Integer, primary\_key=True)

job\_id = db.Column(db.Integer, unique=True, nullable=False)

date\_posted = db.Column(db.DateTime, nullable=False, default=datetime.utcnow)

skill = db.Column(db.String(20), nullable=False)

location = db.Column(db.String(20), nullable=False)

recruiter\_id = db.Column(db.Integer, db.ForeignKey('recruiter.id'), nullable=False)

users = db.relationship('Job\_User', backref='user', lazy=True)

def \_\_repr\_\_(self):

return f"Job('{self.company}', '{self.skill}', '{self.location}')"

class Recruiter(db.Model, UserMixin):

id = db.Column(db.Integer, primary\_key=True)

username = db.Column(db.String(50), unique=True, nullable=False)

email = db.Column(db.String(120), unique=True, nullable=False)

image\_file = db.Column(db.String(20), nullable=False, default='default.jpg')

password = db.Column(db.String(60), nullable=False)

location = db.Column(db.String(20), nullable=True)

jobs = db.relationship('Job', backref='company', lazy=True)

def \_\_repr\_\_(self):

return f"Recruiter('{self.username}')"