گزارش پروژه مهندسی اینترنت

اعضای گروه: مهراوه احمدی، مهدیه اثنی عشری، مهسا رضوی

فاز دوم:

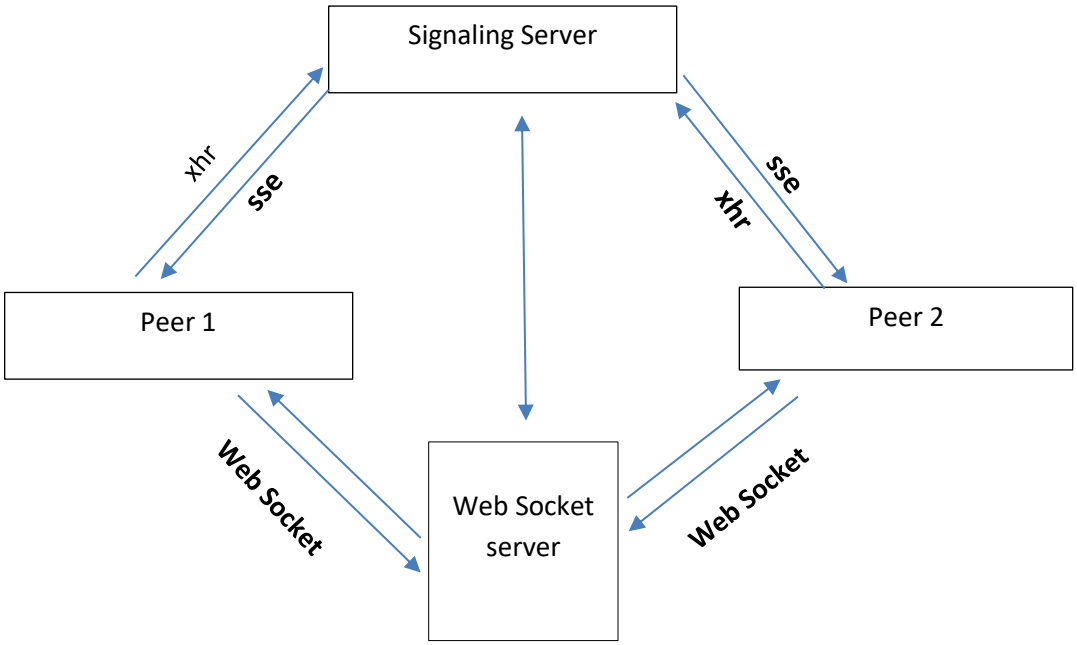
الف) توضیح پیاده‌سازی:

در این فاز امکان چت متنی کاربر با مخاطبانش پیاده‌سازی شده است. این امکان از طریق ارتباط وب سوکت برقرار می‌شود.

برای این منظور ۲ app فلسک پیاده‌سازی شده است، یکی برای سرور سیگنالینگ و دیگری برای سرور وب سوکت.

در فاز قبل برای هر کاربر امکان login و sign up و همچنین مشاهده لیست مخاطبان و ایجاد room فراهم شده بود. در این فاز پس از ایجاد room پیام‌های متنی بین دو کاربر از طریق ارتباط وب سوکت منتقل میشوند.

اطلاعات کاربران از جمله session کاربر، بین دو سرور سیگنالینگ و سرور وب سوکت share می‌شود تا امکان چت متنی در room ایجاد شده بین کاربر و مخاطبش را فراهم کند.



ب) توضیح کد:

- ورود به صفحه چت:

در فاز قبل با تابع join2 ، برای کاربران صفحه‌ی html ای برای شروع چت نشان داده میشد. در این فاز با این تابع یک صفحه جدید برای شروع چت برای آنها باز میشود.

```python
@app.route("/join2/<string:user>/<string:room>/", methods=['GET'])
def join2(user, room):
    b = request.user_agent.browser
    if b == 'chrome':
        b = 'google-chrome'
    webbrowser.get(b).open_new_tab('http://0.0.0.0:5000/'+ user + '/' + room + '/')
    return render_template('pv.html')
```

با این تابع ابتدا بروزر کاربر تشخیص داده می‌شود سپس در همان بروزر tab جدیدی برای آغاز چت باز می‌شود. همچنین session جاری و room id برای سرور چت فرستاده می‌شود . با url داده شده، به تابع index در سرور چت می‌رویم:

```python
@main.route('/<string:user>/<string:room_id>/', methods=['GET', 'POST'])
def index(user, room_id):
    session['name'] = user
    session['room'] = room_id
    return redirect(url_for('.chat'))
```

در تابع index با توجه به room id و session ای که در url دریافت کرده است به صفحه چت مربوط به room میرود.

```python
@main.route('/chat/')
def chat():
    print(session['name'])
return render_template('chat.html', name=session['name'], room=session['room'])
```

سپس با تابع chat صفحه chat.html باز می‌شود.

```javascript
$(document).ready(function(){
            socket = io.connect('http://' + document.domain + ':' + location.port + '/chat');
            socket.on('connect', function() {
                socket.emit('joined', {});
            });
```

در script بالا کانکشن برقرار می‌شود و join فراخوانی می‌شود.

```python
@socketio.on('joined', namespace='/chat')
def joined(message):
    room = session.get('room')
    join_room(room)
emit('status', {'msg': session.get('name') + ':'+'has entered the room.'}, room=room)
```

در تابع join ، با emit شدن ایونت status پیام join شدن کاربر به چت روم نشان داده میشود.

با اسکریپت زیر بعد از نوشتن پیام و فشردن کلید enter تابع text فراخوانی می‌شود:

```
$('#text').keypress(function(e) {
                    var code = e.keyCode || e.which;
                    if (code == 13) {
                        text = $('#text').val();
                        $('#text').val('');
                        socket.emit('text', {msg: text});
                    }
                });
```

تابع text در events.py :

```
@socketio.on('text', namespace='/chat')
def text(message):
    room = session.get('room')
    length = random.randint(10,25)
    pm_id  = randomword(length)
    pm = SkypePmModel(pm_id=pm_id, sender=session.get('name'), text= message['msg'])
    pm.save()
    emit('message', {'msg': session.get('name') + ':' + message['msg']}, room=room)
```

در این تابع پیام به کلاینت مقابل رسانده می‌شود و با socket.on پیام خوانده می‌شود. همچنین پیام‌ها در دیتابیس ذخیره می‌شوند.

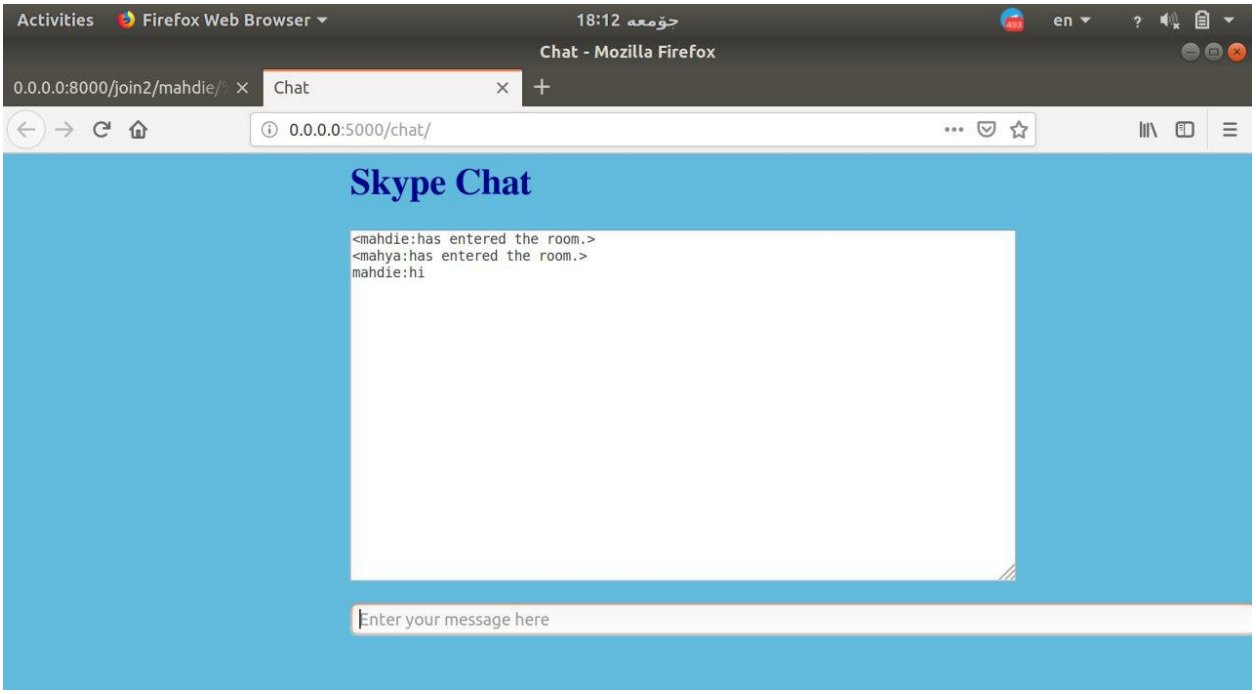
```
socket.on('message', function(data) {
    $('#chat').val($('#chat').val() + data.msg + '\n');
    $('#chat').scrollTop($('#chat')[0].scrollHeight);
});
```

نمونه ای از پیام‌های ذخیره شده در دیتابیس:

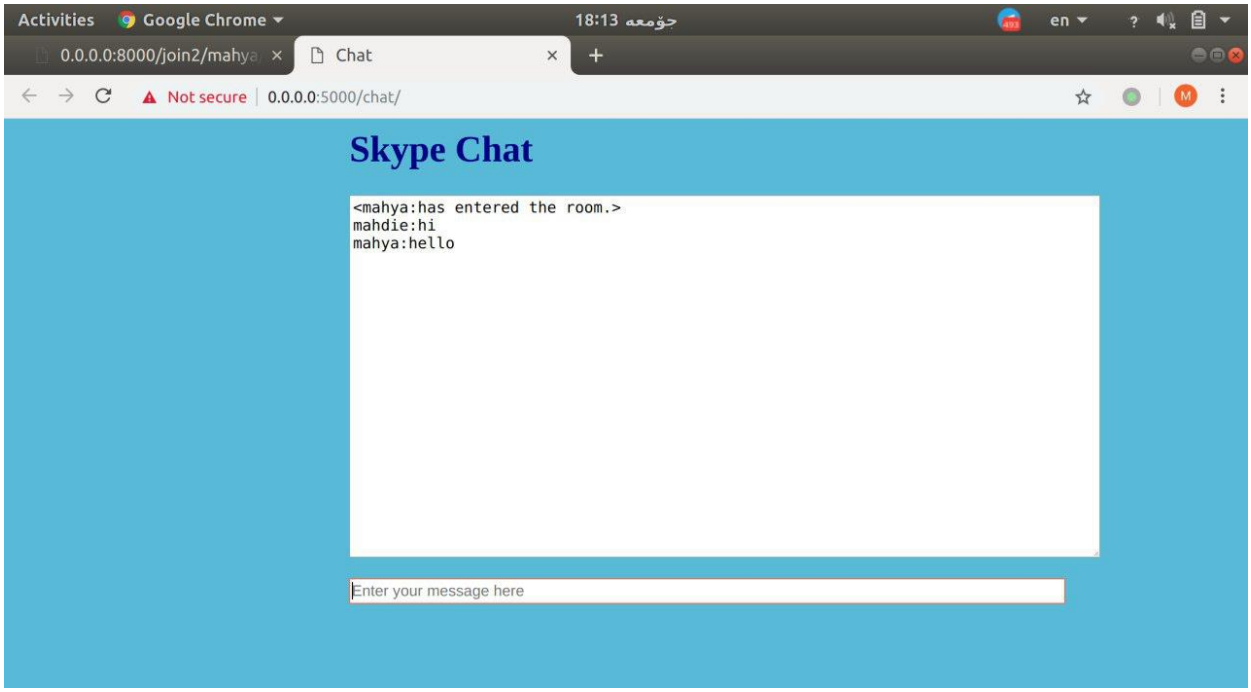
```
{
  "_id" : ObjectId("5c263c05c8dbc913e844b84b"),
  "pm_id" : "vtmfczrzsiipehon",
  "sender" : "mahdie",
  "text" : "hii"
}
{
  "_id" : ObjectId("5c263c0ac8dbc913e844b84c"),
  "pm_id" : "rjtckwgwhc",
  "sender" : "mahya",
  "text" : "hello"
}
{
  "_id" : ObjectId("5c263c0cc8dbc913e844b84d"),
  "pm_id" : "vrxoksbcuxrmiozk",
  "sender" : "mahya",
  "text" : "dear"
}
{
  "_id" : ObjectId("5c263c17c8dbc913e844b84e"),
  "pm_id" : "rmyhwagzturwazyehqlo",
  "sender" : "mahdie",
  "text" : "☺"
}
```

ج) تصاویر خروجی:

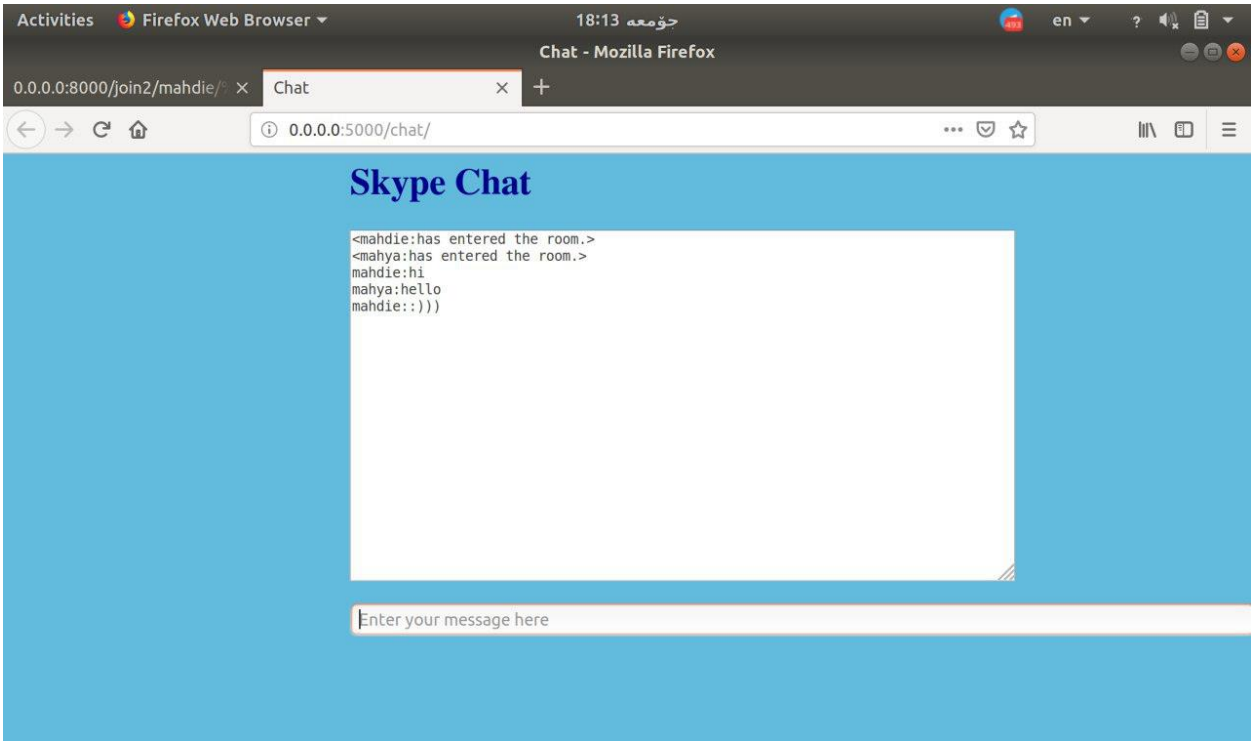
صفحه چت به هنگام ورود دو کاربر به room



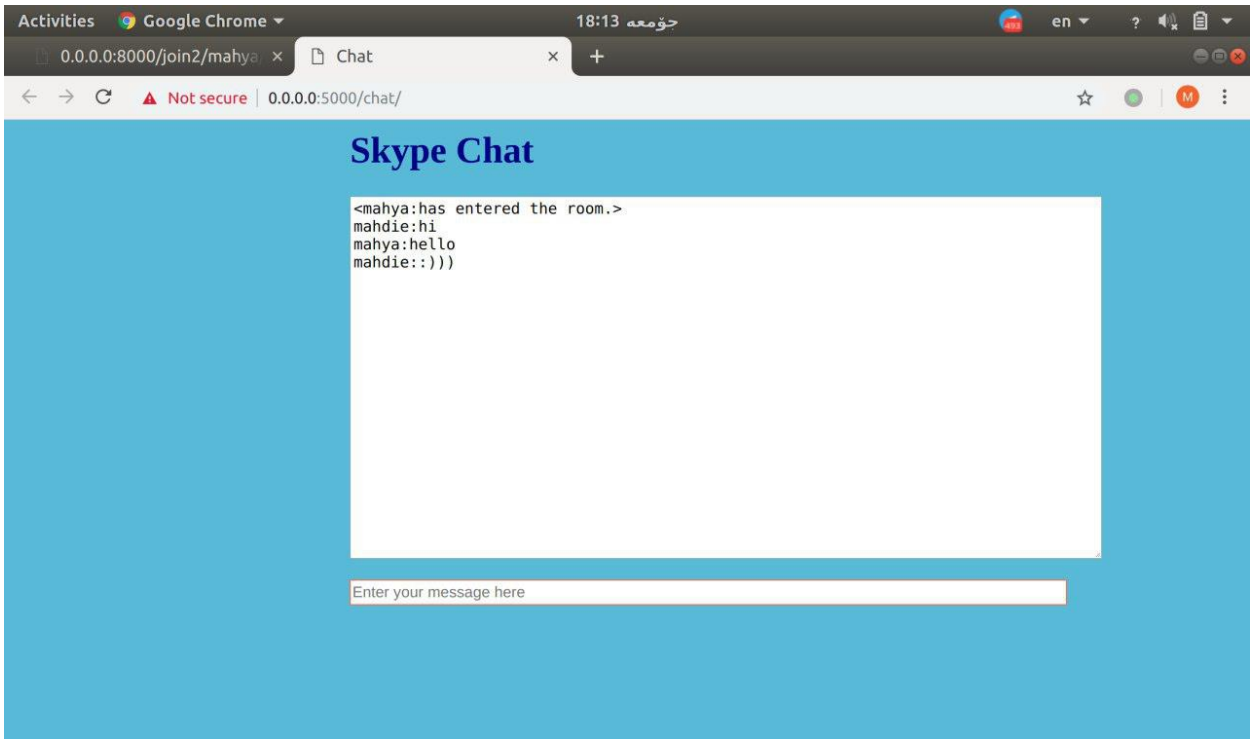
امکان چت متنی:



صفحه room برای کاربر ۱:



صفحه room برای کاربر ۲:



د) نیازمندی‌ها:

- نصب دیتابیس redis

- نصب دیتابیس mongo

- نصب ماژول‌هایی که در فایل requirements آمده است.

ه) طریقه‌ی اجرا:

ابتدا سرور redis را با کامند redis-server اجرا می‌کنیم.

سرور سیگنالینگ با دستور زیر run می‌شود. این سرور روی پورت 8000 run می‌شود.

```
gunicorn deploy:app --worker-class gevent --bind 0.0.0.0:8000
```

در پروژه فولدری به نام Chat است که برای اجرای سرور چت، باید در آن فولدر کامند زیر را اجرا کنیم. سرور چت روی پورت 5000 run می‌شود.

```
FLASK_APP=chat.py flask run
```

و) لینک github پروژه :

https://github.com/mehraveh/skype_chat