

| | | | | | | | |
|-------------------|--------------------|--------------------|--------------|--------------|------------|-------------|----------------------|
| 과제 유형 | ■ 기업연계기반 | | | | | | |
| 과제명 | 고려대학습Chatting room | | | | | | |
| 팀명 | digighetto | | | | | | |
| 수강 교과목명 | 컵스톤디자인2 | | | 교과목 번호 | | DCCS452 | |
| 교과목 담당교수 | 소 속 | 컴퓨터정보학과 | 성 명 | 서민석 | | | |
| | E - mail | mins@korea.ac.kr | 교내전화 | 044-860-1379 | | | |
| 지도교수 | 소 속 | 컴퓨터 융합 소프트웨어학과 | 성 명 | 박대희 | | | |
| | E - mail | dhpark@korea.ac.kr | 교내전화 | 01022991344 | | | |
| 산업체 참여 인력(PM) 해당시 | 소 속 | | 성 명 | | | | |
| | E - mail | | 핸드폰 | | | | |
| 구분 | 성명 | 학과 | 학년 | 학번 | 휴대전화 | E - mail | |
| 참여 학생 | 팀장 | YUTIANCHEN | 컴퓨터융합소프트웨어학과 | 4 | 2020270674 | 01056159792 | 3281873153@qq.com |
| | 팀원 | LIPEIQI | 컴퓨터융합소프트웨어학과 | 4 | 2020270675 | 01028626155 | 2436842879@qq.com |
| | | ZOUHUAN | 컴퓨터융합소프트웨어학과 | 4 | 2021270647 | 01098963338 | zh1141533194@163.com |
| | | | | | | | |
| | | | | | | | |

*복수전공의 경우 본소속학과(복수전공)으로 표기

위와 같이 규정에 의해 과제를 완료하였음을 결과보고서로 제출합니다.

2024. 11. 18.

교과목 담당교수: 박대희 (인 또는 서명)

지도교수: 박대희 (인 또는 서명)

대표학생: YUTIANCHEN (인 또는 서명)

고려대학교 세종 SW중심대학사업단 귀하

| 작품과제명 | 과제명 기재 |
|--------------------------------|--|
| <p>과제 개요</p> | <p>○ 과제 선정 배경</p> <p>-우리는 학습 생활에서 종종 프로그래밍과 수학과 같은 어려운 문제에 직면하곤 합니다. 선생님이 한 번 설명해 주셔도 완전히 이해하지 못할 때가 있습니다. 수업 후 교수님께 질문할 수 있지만, 성격이 내성적인 학생들은 교수님께 다시 설명을 요청하기가 어려울 수 있습니다. 그래서 우리 팀은 익명 계정을 등록할 수 있는 채팅방을 만들었습니다. 누구나 그곳에서 익명으로 질문을 올릴 수 있어, 이러한 문제를 해결하는 데 도움이 되기를 바랍니다.</p> |
| <p>과제 내용</p> | <p>○ 과제 구성</p> <p>-1.레이아웃 설계: HTML과 CSS를 사용하여 채팅창, 입력창, 전송 버튼, 온라인 사용자 목록 등 채팅방의 구조를 정의합니다</p> <p>-2.실시간 통신 처리: 실시간 메시지 전송을 위해 WebSocket 프로토콜을 사용합니다. Node.js에서는 Socket.IO 라이브러리가 자주 사용됩니다.</p> <p>-3.서버 선택: VPS 같은 클라우드 서비스를 사용해 웹사이트를 배포합니다.</p> |
| <p>결과물의 활용방안 및 기대효과</p> | <p>사용자는 웹사이트에 익명으로 등록하고 로그인할 수 있으며, 텍스트, 이미지 또는 음성 형식으로 질문하거나 답변할 수 있습니다.</p> |

| 수행 방법 | 구분 | 성명 | 과제 참여 내용(역할) |
|-------|----|-------------|---------------------|
| | 팀장 | YUTIAN CHEN | 웹사이트 작성및클라우드 서비스 제공 |
| | 팀원 | LIPEIQI | 실시간 통신 처리 |
| | 팀원 | ZOUHU AN | MYSQL |
| | 팀원 | | |
| | 팀원 | | |
| | 팀원 | | |

결과물





```

# app.py - C:\Users\32818\Desktop\20241119\app.py (1.1.0)
File Edit Format Run Options Window Help
def allowed_file(filename): #检查上传内容是否为图片类型
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
@app.route('/index')
@app.route('/')
def index():
    # 用户名和密码
    return render_template('index.html', username=session.get('username'))
@app.route('/login', methods=[GET, POST])
def login():
    if request.method == POST:
        email=request.form.get('email')
        username = request.form.get('username')
        password = request.form.get('password')
        insert_code=INSERT INTO logins(user_email,username,password,profile_img)
        VALUES ('%s','%s','%s','%s')%(email,username,password,profile_img)
        cursor.execute(insert_code)
        print(cursor.lastrowid)
        # 插入成功后返回消息
        return jsonify({'status': 'error', 'message': 'required code'})
    else:
        cursor.execute(insert_code)
        return render_template('register.html')
@app.route('/register', methods=[GET, POST])
def register():
    if request.method == POST:
        data = request.json
        username = data.get('username')
        password = data.get('password')
        find_code=select * from logins where username=%s %(username)
        cursor.execute(find_code)
        if password==user[0][password]:
            session['username'] = username
            return jsonify({'status': 'success', 'message': 'Login successful'})
    else:
        return render_template('register.html')

```

```

# app.py - C:\Users\32818\Desktop\20241119\app.py (1.1.0)
File Edit Format Run Options Window Help
# 退出程序
@app.route('/logout')
def logout():
    session.pop('username', None)
    return redirect(url_for('login'))
@app.route('/profile', methods=[GET, POST])
def edit_profile():
    # 当前用户编辑个人资料页面
    return render_template('users.html')
# 上传图片
@app.route('/update-profile', methods=[POST])
def update_profile():
    # 上传图片数据
    username = request.form.get('username')
    email = request.form.get('email')
    new_password = request.form.get('password')
    random_password = request.form.get('confirm_password')
    find_code=select * from logins where username=%s %(username)
    update_code = UPDATE logins SET username=%s, password=%s WHERE user_email=%s
    current_user = conn_mysql(find_code)
    if not current_user:
        return jsonify({'status': 'can not find user'}, 404)
    print(current_user[0])
    # 更新用户名和密码
    current_user[0]['username'] = username
    current_user[0]['email'] = email
    # 新增密码了和密码, 设置两次输入是同一款
    if new_password:
        if new_password != confirm_password:
            flash('两次输入的密码不一致!', 'error')
            return redirect(url_for('login'))
    # 提交数据修改成功
    conn_mysql(update_code)
    flash('个人信息修改成功!', 'success')

```



