

# Final Project Proposal – Part V

Assigned Date: 12/12/2023

Please finalize the report for your final project by incorporating the following components and consolidating them with your proposal:

- Please include screenshots of the user interface displaying the query results from your database application.
- Please provide an explanation of your application code and how it utilizes SQL to perform queries or updates.

(1)

## 1.進行CRUD



## 2登入介面



## 3.註冊更新資料庫



## 4.使用者介面



(2)

透過html、css、javascript模擬一間串流公司後台數據庫，並使用python的 flask 和ajax來連接資料庫和提升使用者體驗，且運用jquery和bootstrap等來美化設計  
部分程式碼:

### 1.flask實作

```
1. # SQLite configurations
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///site.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

class Account(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    fullname = db.Column(db.String(255), nullable=False)
    username = db.Column(db.String(100), unique=True, nullable=False)
    password = db.Column(db.String(255), nullable=False)
    email = db.Column(db.String(255), nullable=False)

# Create tables
with app.app_context():
    db.create_all()

# http://localhost:5000/pythonlogin/ - this will be the login page
@app.route('/pythonlogin/', methods=['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']

        # Check if account exists using SQLite
        account = Account.query.filter_by(username=username, password=password).first()

        if account:
            # Create session data
            session['loggedin'] = True
            session['id'] = account.id
            session['username'] = account.username
            # Redirect to home page
            return redirect(url_for('home'))
        else:
            msg = 'Incorrect username/password!'
```

```

        return render_template('index.html', msg=msg)

# http://localhost:5000/register
@app.route('/register', methods=['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and
'password' in request.form and 'email' in request.form:
        fullname = request.form['fullname']
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']

        # Check if account exists using SQLite
        existing_account =
Account.query.filter_by(username=username).first()

        if existing_account:
            msg = 'Account already exists!'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address!'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers!'
        elif not username or not password or not email:
            msg = 'Please fill out the form!'
        else:
            # Create a new account
            new_account = Account(fullname=fullname, username=username,
password=password, email=email)
            db.session.add(new_account)
            db.session.commit()

            msg = 'You have successfully registered!'

    elif request.method == 'POST':
        msg = 'Please fill out the form!'

    return render_template('register.html', msg=msg)

```

## 2.ajax應用(新增資料)

```

@app.route("/ajax_add", methods=["POST", "GET"])
def ajax_add():
    msg = ''
    if request.method == 'POST':
        txtname = request.form['txtname']
        txtdepartment = request.form['txtdepartment']
        txtphone = request.form['txtphone']
        print(txtname)
        if txtname == '':
            msg = 'Please Input name'
        elif txtdepartment == '':
            msg = 'Please Input Department'
        elif txtphone == '':
            msg = 'Please Input Phone'
        else:
            cur = get_db().cursor()
            cur.execute("INSERT INTO tblemployee (name,department,phone)
VALUES (?, ?, ?)",
                        (txtname, txtdepartment, txtphone))
            get_db().commit()
            msg = 'New record created successfully'
    return jsonify(msg)

```