# Complete Setup Guide: Quantum Drug Discovery Flask App

## 📋 Prerequisites

- Python 3.8 or higher
- Visual Studio Code
- Git (optional but recommended)

## 🛠️ Step-by-Step Setup in Visual Studio Code

### Step 1: Create Project Structure

1. Open Visual Studio Code
2. Create a new folder for your project: `quantum-drug-discovery`
3. Open the folder in VS Code (`File` → `Open Folder`)

### Step 2: Create File Structure

Create the following files and folders in your project directory:

```
quantum-drug-discovery/
├── app.py              # Main Flask application
├── templates/
│   └── index.html      # Frontend template
├── static/
│   ├── css/
│   ├── js/
│   └── images/
├── requirements.txt    # Python dependencies
└── README.md           # Project documentation
```

### Step 3: Set Up Python Environment

#### Option A: Using VS Code Terminal

1. Open Terminal in VS Code: `Ctrl + `` (backtick)
2. Create virtual environment:

bash

```
python -m venv quantum_env
```

3. Activate virtual environment:

bash

```
# Windows
quantum_env\Scripts\activate

# macOS/Linux
source quantum_env/bin/activate
```

### Option B: Using VS Code Python Extension

1. Install Python extension for VS Code
2. Press `Ctrl+Shift+P` → Type "Python: Create Environment"
3. Select "Venv" → Choose Python interpreter

## Step 4: Create Requirements File

Create `requirements.txt` with the following content:

txt

```
Flask==2.3.3
pennylane==0.32.0
numpy==1.24.3
pandas==2.0.3
matplotlib==3.7.2
seaborn==0.12.2
scikit-learn==1.3.0
Werkzeug==2.3.7
```

## Step 5: Install Dependencies

In the VS Code terminal, run:

bash

```
pip install -r requirements.txt
```

## Step 6: Create the Flask Application

1. Copy the Flask app code into `app.py`
2. Create `templates` folder

3. Copy the HTML template into `templates/index.html`

## Step 7: Configure VS Code for Flask Development

### Install Recommended Extensions:

1. **Python** - Microsoft (Essential)
2. **Flask Snippets** - cstrap
3. **HTML CSS Support** - ecmel
4. **Auto Rename Tag** - Jun Han
5. **Prettier** - Prettier (for code formatting)

### Configure VS Code Settings:

Create `.vscode/settings.json`:

json

```
{
  "python.defaultInterpreterPath": "./quantum_env/Scripts/python.exe",
  "python.terminal.activateEnvironment": true,
  "flask.app": "app.py",
  "python.linting.enabled": true,
  "python.linting.pylintEnabled": true,
  "emmet.includeLanguages": {
    "html": "html"
  }
}
```

### Create Launch Configuration:

Create `.vscode/launch.json`:

json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python: Flask",
      "type": "python",
      "request": "launch",
      "program": "${workspaceFolder}/app.py",
```

```
        "console": "integratedTerminal",
        "justMyCode": true,
        "env": {
          "FLASK_APP": "app.py",
          "FLASK_ENV": "development"
        }
      }
    ]
}
```

### Step 8: Project Structure Verification

Your final structure should look like:

```
quantum-drug-discovery/
├── .vscode/
│   ├── settings.json
│   └── launch.json
├── quantum_env/        # Virtual environment
├── templates/
│   └── index.html
├── static/           # For future CSS/JS files
├── app.py
├── requirements.txt
└── README.md
```

# 🚀 Running the Application

### Method 1: Using VS Code Debugger

1. Press `F5` or go to `Run and Debug` panel
2. Select "Python: Flask" configuration
3. The app will start in debug mode

### Method 2: Using Terminal

1. Ensure virtual environment is activated
2. Run in terminal:

bash

```
python app.py
```

### Method 3: Using Flask Command

bash

```
export FLASK_APP=app.py
export FLASK_ENV=development
flask run
```

## 🌐 Accessing the Application

1. Open your web browser
2. Navigate to: `http://localhost:5000` or `http://127.0.0.1:5000`
3. You should see the Quantum Drug Discovery interface

## 🔧 Troubleshooting Common Issues

### Issue 1: Module Not Found

**Problem**: `ModuleNotFoundError: No module named 'pennylane'` **Solution**:

bash

```
pip install pennylane numpy matplotlib pandas scikit-learn seaborn flask
```

### Issue 2: Virtual Environment Not Activated

**Problem**: Packages installed globally instead of in virtual environment **Solution**:

- Check bottom-left of VS Code status bar for Python interpreter
- Should show path to `quantum_env`
- If not, press `Ctrl+Shift+P` → "Python: Select Interpreter" → Choose `quantum_env`

### Issue 3: Port Already in Use

**Problem**: `Address already in use` **Solution**:

- Change port in `app.py`: `app.run(debug=True, port=5001)`
- Or kill process using port 5000

### Issue 4: Template Not Found

**Problem**: `TemplateNotFound: index.html` **Solution**:

- Ensure `templates/` folder exists in same directory as `app.py`
- Check file name is exactly `index.html`

# 🎯 Testing the Application

### 1. Train the Model

- Enter training parameters (samples: 500, epochs: 60)
- Click "Start Quantum Training"
- Wait for progress bar to complete

### 2. Test Predictions

- Use sample molecules or enter custom values:
  - **Good Drug**: MW=300, LogP=2.5, PSA=60, H-donors=2
  - **Poor Drug**: MW=600, LogP=6.0, PSA=150, H-donors=8
- Click "Predict Drug Effectiveness"

### 3. View Results

- Training progress plots
- Confusion matrix
- Model accuracy metrics

# 🔬 Understanding the Science

### Quantum Advantage:

- **Classical Computers**: Struggle with molecular quantum states
- **Quantum Computers**: Naturally represent quantum systems
- **Speed Up**: Years → Months for drug discovery

### Machine Learning Integration:

- Quantum Neural Networks (QNNs)
- Variational quantum circuits
- Quantum feature encoding
- Classical optimization

### Real-World Applications:

- **Pfizer**: Using quantum ML for molecular analysis
- **IBM Quantum Network**: Pharmaceutical partnerships
- **Cost Savings**: Billions in R&D expenses

# 📈 Next Steps

### Enhance the Application:

1. **Add More Features**: ADMET properties, toxicity prediction
2. **Real Data**: Integrate with ChEMBL or PubChem databases
3. **Better UI**: Add animations, 3D molecular visualizations
4. **Export Results**: PDF reports, CSV downloads
5. **User Authentication**: Save experiments, history tracking

### Deploy to Production:

1. **Cloud Deployment**: Heroku, AWS, or Azure
2. **Docker**: Containerize the application
3. **Database**: PostgreSQL for storing results
4. **Scalability**: Redis for caching, load balancing

# 🎊 Congratulations!

You now have a fully functional quantum machine learning web application for drug discovery running in VS Code!

The application demonstrates how quantum computing can revolutionize pharmaceutical research by leveraging the quantum nature of molecules for more efficient drug candidate screening.