SecureChat - Quick Start Guide

🚀 Complete Setup (Windows)

One-Line Setup

```cmd
setup.bat
```

Manual Setup

1. **Install Dependencies**

```cmd
python -m venv .venv
.venv\Scripts\activate
pip install -r requirements.txt
```

2. **Configure Environment**

```cmd
copy .env.example .env
```

3. **Start MySQL** (Docker)

```cmd
docker run -d --name securechat-db -e MYSQL_ROOT_PASSWORD=rootpass -e MYSQL_DATABASE=securechat -e MYS
```

4. **Initialize Database**

```cmd
python -m app.storage.db
```

5. **Generate Certificates**

```cmd
```

```
python scripts\gen_ca.py
python scripts\gen_cert.py --cn server.local --out certs\server
python scripts\gen_cert.py --cn client.local --out certs\client
```

▶️ Run Application

Terminal 1 - Server:

```cmd
.venv\Scripts\activate
python -m app.server
```

Terminal 2 - Client:

```cmd
.venv\Scripts\activate
python -m app.client
```

🧪 Testing Commands

Run Attack Simulations

```cmd
python scripts\test_attacks.py
```

Verify Session Receipt

```cmd
python scripts\verify_receipt.py --receipt transcripts\client_receipt_[...].json --transcript transcripts\client_[...].txt --cert certs\c
```

Export Database

```cmd
python scripts\export_db.py
python scripts\export_db.py --info
```

Wireshark Filters

```
Capture Filter: tcp.port == 5000
Display Filter: tcp.stream eq 0
```

📋 Usage Flow

1. **Start Server** → Wait for "Listening on..."
2. **Start Client** → Certificate exchange happens automatically
3. **Choose (r)egister or (l)ogin**
4. **Enter credentials**
5. **Type messages** → Press Enter to send
6. **Type 'quit'** → Exit and generate receipt

🔍 Quick Checks

Verify Certificates Generated

```cmd
dir certs
```

Should show:

- ca_cert.pem
- ca_key.pem
- server_cert.pem
- server_key.pem
- client_cert.pem
- client_key.pem

Verify Database Connection

```cmd
python -c "import pymysql; pymysql.connect(host='localhost', user='scuser', password='scpass', database='securechat'); print('
```

Check Transcripts

```cmd
dir transcripts
```

🐛 Common Issues

Issue: MySQL Connection Failed

```cmd
docker ps
docker restart securechat-db
```

Issue: Import Error

```cmd
# Make sure you're in project root
cd securechat-skeleton
# Activate virtual environment
.venv\Scripts\activate
```

Issue: Certificate Not Found

```cmd
# Regenerate all certificates
python scripts\gen_ca.py
python scripts\gen_cert.py --cn server.local --out certs\server
python scripts\gen_cert.py --cn client.local --out certs\client
```

📊 File Structure Check

```
securechat-skeleton/
├── ✓ .env (from .env.example)
├── ✓ certs/ (6 .pem files)
├── ✓ transcripts/ (created during chat)
├── ✓ .venv/ (virtual environment)
└── ✓ app/, scripts/, requirements.txt
```

🎯 Submission Checklist

□ GitHub repository with 10+ commits

- □ README.md updated with your info
- □ Database exported (database_export.sql)
- □ Report document (RollNumber-FullName-Report-A02.docx)
- □ Test report (RollNumber-FullName-TestReport-A02.docx)
- □ Wireshark captures (screenshots)
- □ Session receipts (from transcripts/)
- □ Repository ZIP file

📞 Quick Commands Reference

Task	**Command**
Activate venv	`.venv\Scripts\activate`
Run server	`python -m app.server`
Run client	`python -m app.client`
Test attacks	`python scripts\test_attacks.py`
Verify receipt	`python scripts\verify_receipt.py --help`
Export DB	`python scripts\export_db.py`
DB info	`python scripts\export_db.py --info`
Init DB	`python -m app.storage.db`
Gen CA	`python scripts\gen_ca.py`
Gen cert	`python scripts\gen_cert.py --help`

🎬 Demo Script

1. Open 3 terminals
2. Terminal 1: Start Wireshark with filter `tcp.port == 5000`
3. Terminal 2: `python -m app.server`
4. Terminal 3: `python -m app.client`
5. Register user "alice" with password "secure123"
6. Send 3 messages
7. Type 'quit'
8. Run: `python scripts\verify_receipt.py [files]`
9. Show Wireshark capture (encrypted data)
10. Run: `python scripts\test_attacks.py`

Done! 🎉