Using AI Tools (such as ChatGPT, Github Copilot,….) implement the following projects:

1. Balance Query Service via SIP voice call

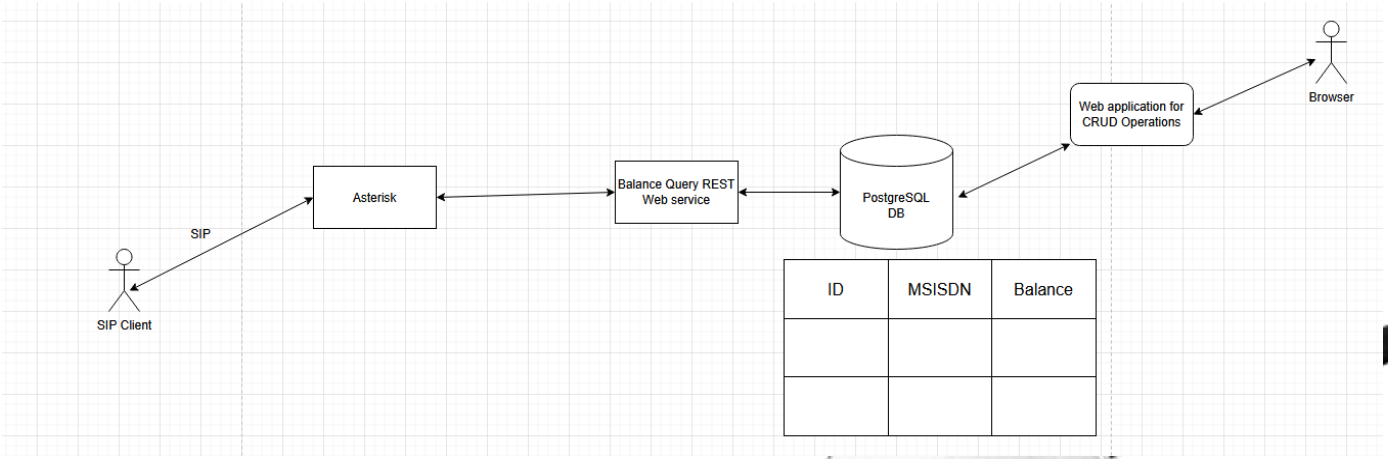
Balance Query Services through voice call, the user will call specific extension from his SIP Client on his/her mobile/PC and he will listen to recorded welcome message and ask the user to enter his phone number by DTMF after user enter his phone number/MSISDN, he will listen to his current balance value and play end voice message to user then end the call.

The Asterisk java script will call REST web service endpoint to query the balance of MSISDN which entered by user via DTMF

There will be web application using Servlet to provide CRUD Operations for Database table

You will use the following Technologies:

- Java
- JDBC
- REST Web Service
- Asterisk/AGI/…
- PostgreSQL
- SIP client
- Servlet
- HTML, CSS, JS



2. Prepaid Charging for Voice Call

We will emulate voice call and its real-time charging. We will have two application:

- First application(Mobile Phone) will act as mobile phone which will capture the voice from microphone and send it via UDP protocol to second application (MSC). This first application(Mobile Phone) when started will start send UDP traffic and will print every one minute on screen/stdout the elapsed minutes from start the mobile application
- Second Application (MSC) which will receive traffic from UDP protocol and play the voice data at PC Speaker. And the MSC will deduct from User Balance every one minute 5 L.E per minute. And at the end of the voice call the application will generate CDR with call details.

The first application will start as following command line:

Java mobile <MSISDN>

Such as :

```
Java mobile 01223456789
```

And its output will be such as:

```
Starting voice call as MSISDN 01223456789
Capturing Voice from Microphone and send via UDP…..
1 minutes elapsed
2 minutes elapsed
3 minutes elapsed
4 minutes elapsed
5 minutes elapsed
```

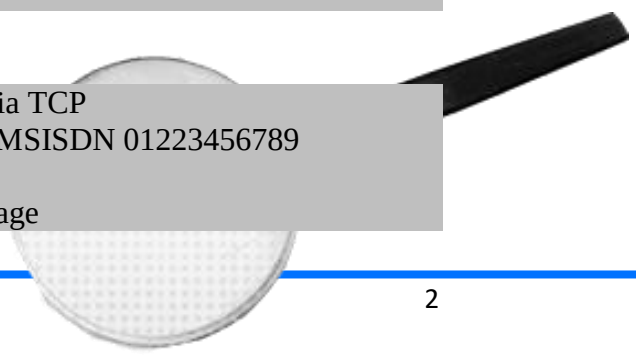
The Second application will start as following command line:

Such as :

```
Java msc
```

And its output will be such as:

```
Waiting for voice call Signaling start message via TCP
Accept Voice call start signaling message from MSISDN 01223456789
Capturing UDP traffic and play via speaker …..
Call End after receiving end call signaling message
```



Generating CDR line: 01223456789, 2025-03-01T15:26:47.398253 , 2025-03-01T15:24:47.398253 ,4,Normal call Clearing,20,1001

- when the first application (Mobile Phone) start will get the MSISDN via main(Arguments) and send to second application via TCP Protocol via 'Start Call' Signaling message to identify itself to MSC application and when the first application (Mobile Phone) shutdown (addShutdownHook/….), you send to second application (MSC) that call us end 'End Call' to stop charging
- The Second application (MSC) will start receive UDP Traffic and play it via Speaker after received via TCP Connection 'Start Call' signaling message
- The Second application (MSC) will deduct balance from this MSISDN which received at 'Start Call' signaling message every one minute fixed amount 5 L.E per minute.
- The Second application (MSC) will stop charging and stop play audio traffic and stop received from UDP after receiving 'End Call' signaling message
- The Second application (MSC) will calculate the call Cost and append CDR to file /tmp/calls.cdr
- Generate CDR file at end of call contain following info:
  - MSISDN
  - Start Time
  - End Time
  - Duration
  - Call result(normal call clearing, user not found)
  - Call cost
  - Balance after call end

Bonus Requirements:

- Encrypt traffic at TCP and UDP
- Compress UDP Traffic
- Make MSC application to support concurrent/many and calls/Mobiles at same time
- Save the voice data of call to file with this format /tmp/voice_call_msisdn_01223355_date_2025_03_01_Time_10_20_30.wav/mp3
- Make the CDR file to rotate every one hour such as: /tmp/calls_CDR_2025_03_02_10.cdr