Jacaranda Health Software Developer Interview Assignment

The data shared in the excel sheet contains the data we use and work with on a daily basis. For the privacy of our users, we have edited the content of the messages and replaced them with dummy text. Included, are two worksheets:

1) **Messages:** Incoming and Outgoing messages.
    a) *body_text:* The content of the message.
    b) *Id:* Id of each message.
    c) *Incoming:* Field showing if the message is incoming or outgoing.
    d) *user_id:* FIeld Indicating the id of the sender.
    e) tickect_id: Field indicating the ticket id.
    f) created_at: FIeld indicating when the ticket was created.
2) **Subjects:** Contain a ticket-subject mapper
    a) *ticket_id*: Field indicating the ticket id
    b) subject: Subject of the ticket

Task

1) Using the messages and subject worksheets, get all the unique tickets together with details associated with that ticket. Create an output that captures the:
    a) ticket_id
    b) subject
    c) Phone number. Phone number is in the Subject field. Not all the subjects have a phone number.
    d) *intents*. Intents are indicated at the beginning of the body text. The text after ***Detected intent.*** For example, in the body text below, the intent is "***qsurvey_positive***"
        Body text: *Detected intent:* ***qsurvey_positive*** *(confidence: 0.340005435464587) At longed but fame her*
    NB. *Not all messages have an intent*
    e) All incoming messages associated with that ticket.
    f) All outgoing messages associated with that ticket
    NB. Incoming messages and Outgoing Messages should contain:
        i) Message content. Message content without the Detected intent and *confidence* section
        ii) Id
        iii) User_id
        iv) Date time

Sample output


```
[{
        "ticket_id" : 5054,
        "subject" : "No Ideal I Was Breastfeeding",
        "phone": "254716000000",
        "intents": ["pain", "headache"],
        "incoming_messages": [{
                        "id": 43060552854,
                        "message":"Was artless, been nor come bower the it",
                        "created":"2019-02-27T05:02:16.000Z",
                        "updated":"2019-02-27T05:02:16.000Z",
                        "user_id": 43019547057
                }],
        "outgoing_messages": [{
                        "id": 43060552854,
                        "description":"And had in was fabled seraphs nor it beyond",
                        "created":"2019-02-27T05:02:16.000Z",
                        "updated":"2019-02-27T05:02:16.000Z",
                        "user_id": 43019547057
                 },
                {
                        "id": 4304352854,
                        "description":"And had in was fabled seraphs nor it beyond",
                        "created":"2020-02-27T05:02:16.000Z",
                        "updated":"2020-02-27T05:02:16.000Z",
                        "user_id": 4301954357
                }]

}]
```


2) Using Flask, Build API endpoints that:
    a) Save the ticket outputs to an SQL database
    b) Retrieve the output from the database (batch and a single record by id)
    c) Updates the record in the database
3) Bonus Points
    a) Create a User interface to display the data
    b) Write tests for your application

Please share a github/gitlab repository containing your code for the above repository.