**CPSC449**

**Project 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Team Members:**

**Bony Roy ( Dev 1- owns the Sharding the tracks database)**

**Suramya Singh ( Dev 2)**

**Brandon Tomich ( OPS)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

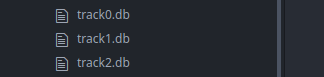
**Bony Roy: (Developer 1)**

I have worked on sharding of tracks database using GUID. Separating tracks from other 3 microservices(user, description, playlist). Also, change other part of the codes in other microservices as required. GUID is the key for Tracks database tracks table. 3 database created-

Database0= track0.db

Database1= track1.db

Database2= track2.db



Data is inserted into Tracks table in separate databases using json script. Function written to generate database number based on GUID.

**Commands to run my part of code: (please makesure .env file is present. Sometime it is removed when uploading)**

export FLASK\_APP=Tracks.py APP\_CONFIG=api.cfg

flask init

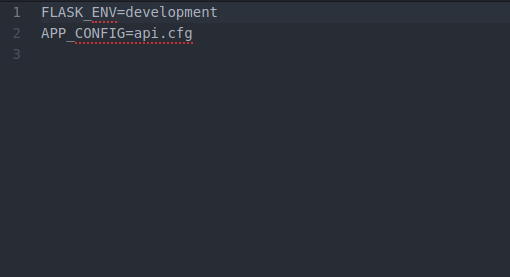
flask popdb

export FLASK\_APP=user.py APP\_CONFIG=api.cfg

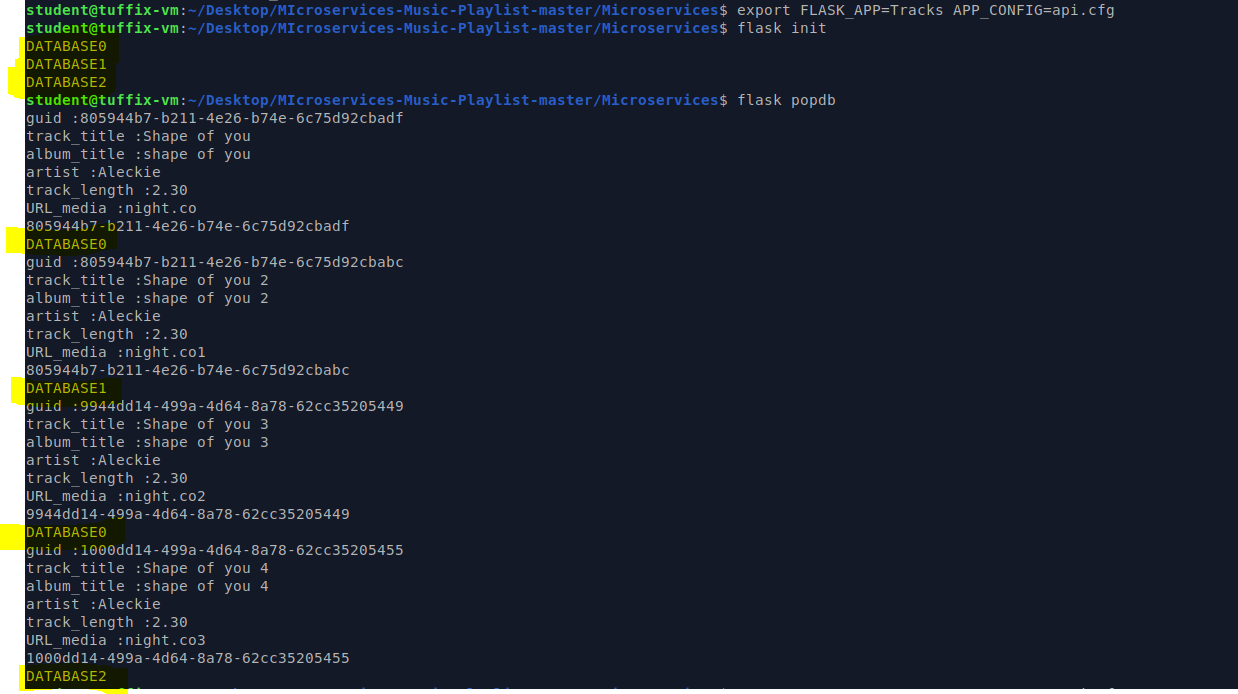
flask init

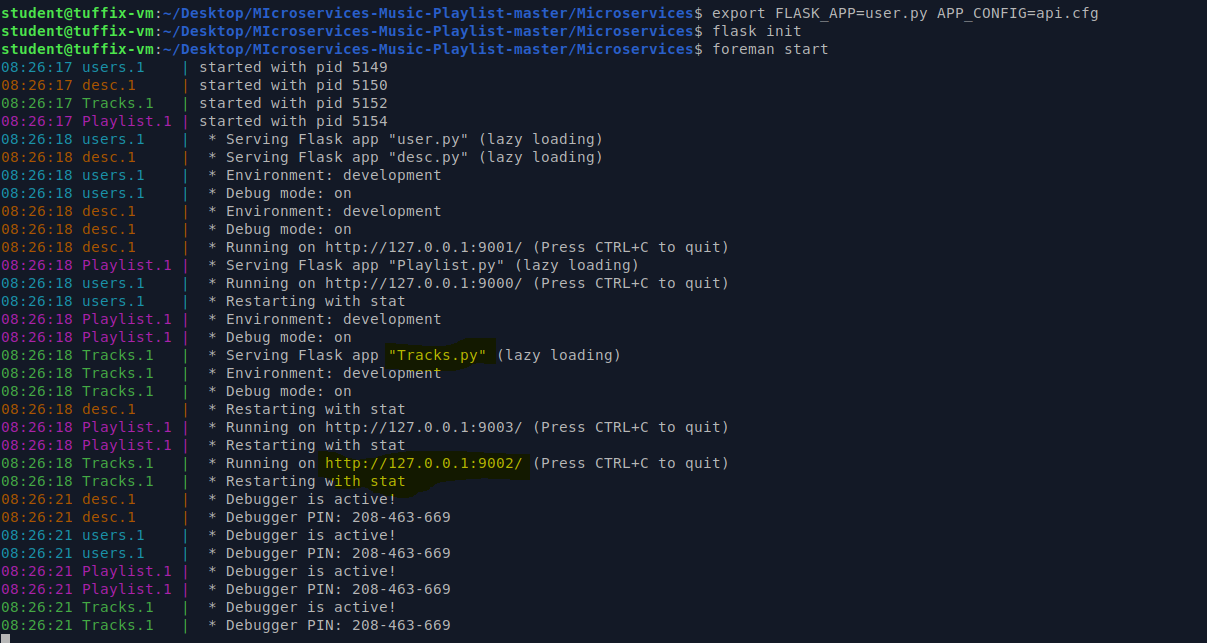
foreman start

**.end File if deleted from folder: (Not needed if you provide** APP\_CONFIG=api.cfg while running but best practice is to provide**)**

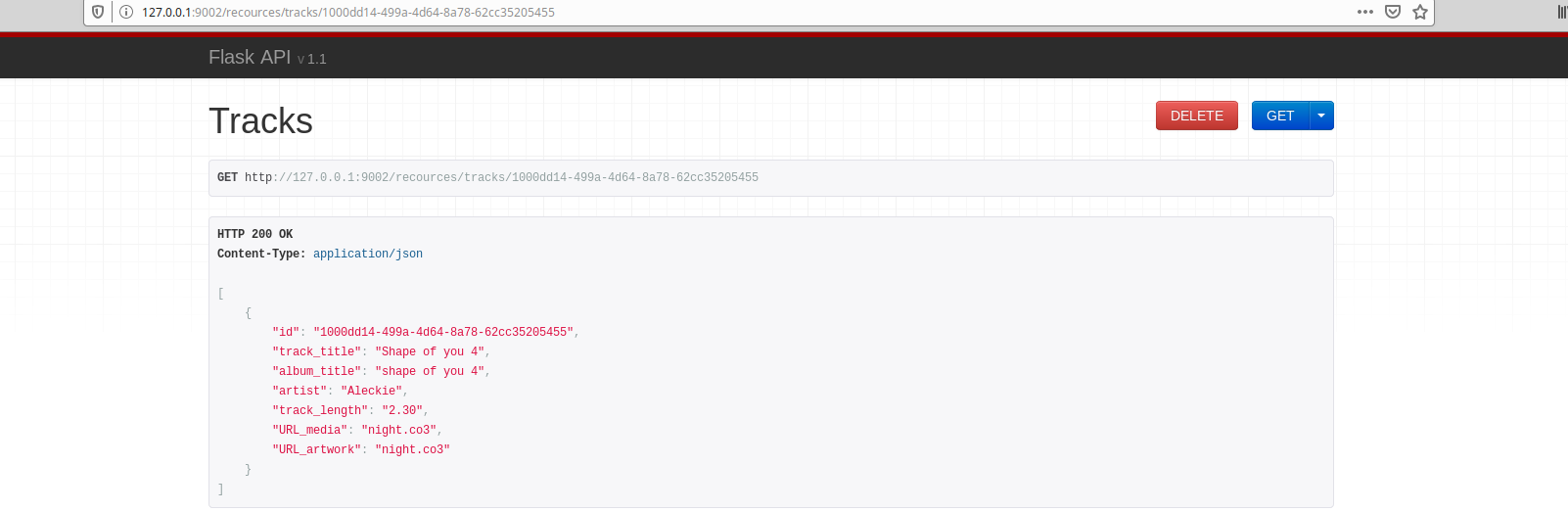


**Screenshots of sharded databases created.**





**Now copy any GUID created and check in API:**



**Similarly we can delete any Tracks or Add any new track. You do not need to provide any id GUID will be created.  
--------------------------------------------------END OF Dev 1 Role---------------------------------------------------------**