- Steps taken to configure Hasura GraphQL engine
    - I configured the Hasura GraphQL engine through a Docker instance. After installing the instance, I copied the Postgres database from my local machine to the Docker instance so that there was connectivity between the GraphQL engine and Postgres. I also had to update the docker-compose.yaml file to enable the GraphQL engine to point to the newly created database.

- Challenges
    - After connecting the database with the GraphQL engine, I had to explicitly click "Track" tables to ensure that my data was being read by the API. Before doing this, I kept running into the "root_query" not found error.
    - I ran into No-Null violation error when I initially ran the mutation query to insert a new album and artist. After doing some research on the Internet, I found out that the issue is likely with the database's misconfiguration where it is not auto-creating an id for the new artist's entry.
        - I initially added the ID for the artist and album manually within the GraphQL query. However, I kept getting errors.
        - After further research, I found out that I had to reconfigure my database columns to generate a new incremental id (primary_key) if it is not provided.
        - I first created a sequence:
            - CREATE SEQUENCE artist_artist_id_seq OWNED BY artist.artist_id;

        - I then found out what the maximum ID is for the artist
            - SELECT MAX(artist_id) FROM artist;
            - I then inserted the next sequence number to restart the table entry from:
            - ALTER SEQUENCE artist_artist_id_seq RESTART WITH 285;
            - I ran into a similar problem with the album table, and I repeated the same steps to get the desired answer.