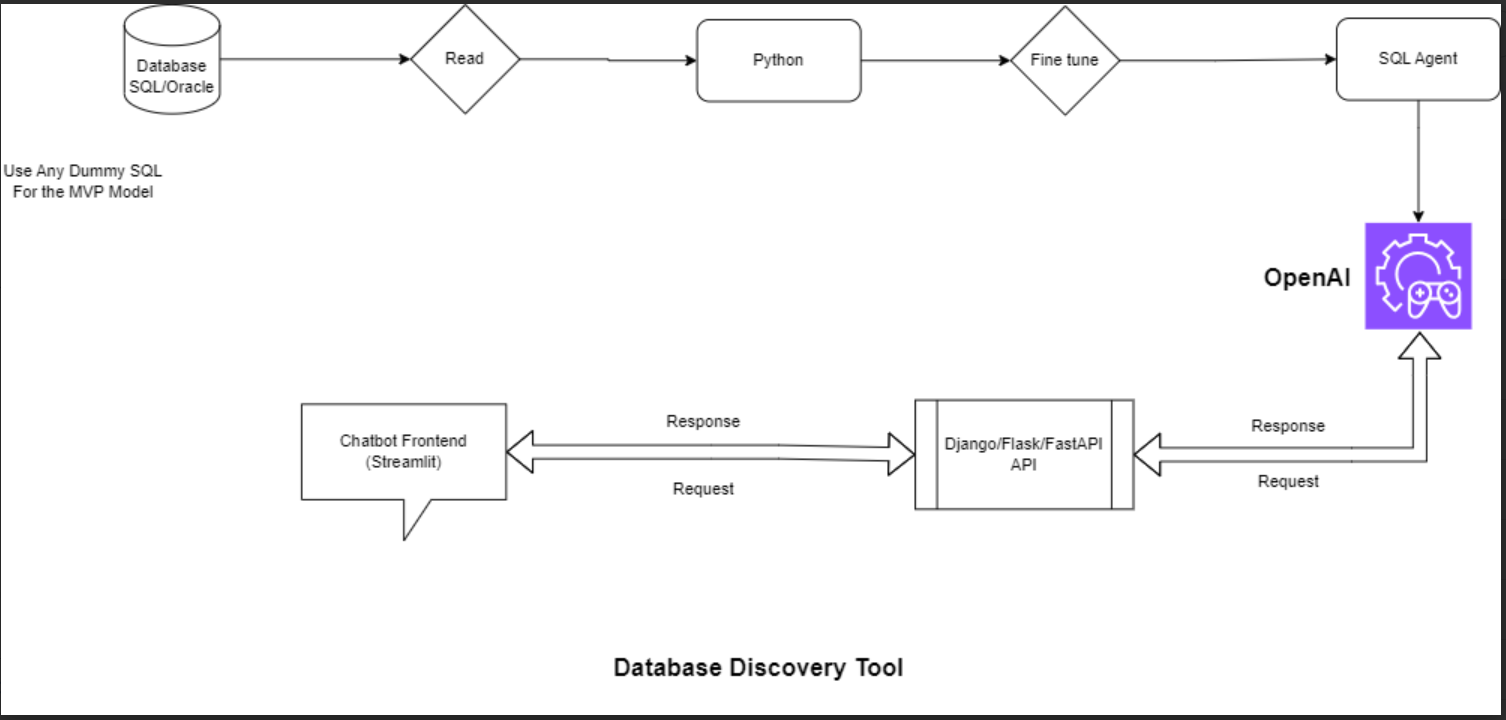
2024

**Methodology for Database Discovery Tool**



**Figure: Workflow**

# **Step by Step Execution**

***Step 1*. Database Support**

In this step we communicate with different types of databases, like SQL and Oracle. This means it can connect and retrieve information from a variety of database systems using Python, providing users with more flexibility and compatibility across various database environments.

***Step 2*. Data Extraction**

In this step we are using python for our Extract, Transform, Load (ETL) processes this involves efficiently reading and extracting data from the connected databases. Python handled the data-related tasks, ensuring a robust and effective extraction process and save the result in csv files which in turn are converted to .db files for sqlite.

***Step 3*. Fine-Tuning**

In this step fine-tuning mechanisms to optimize the performance and accuracy of data extraction processes. This Ensures the ETL tool finds data accurately and quickly.

***Step 4*. Integration with OpenAI**

In this step we have utilized SQL Agent for communication with OpenAI, By communicating with OpenAI, the SQL agent get the ability to understand and respond in a more intelligent and context-aware manner.

***Step 5*. API Integration**

In this step we made Django API endpoints for requesting and receiving data. This means that external systems or applications can interact with the SQL Agent through OpenAI by sending requests and receiving responses through these APIs.

***Step 6*. Streamlit Frontend**

In this step we made a streamlit frontend to chat with the SQL Agent. The user can ask question about the database and receive responses in form of insights.