# Introduction

## Purpose:

Chat2DB is a multi-database client tool that has integrated the AIGC. It can convert natural language into SQL. It can also convert SQL into natural language and provide optimization suggestions for SQL to greatly enhance the efficiency of developers. Chat2DB supports various AI models and databases. With the help of AI, even non-SQL business operators in the future can use it to quickly query business data and generate reports.

## Framework:

The project consists of a front-end UI which is shown as web pages, as well as a back-end server. Users could install and run the project on Windows, Mac, Linux and web pages.

The project mainly uses **Electron+js+Java in order to support web and desktop applications. T**he primary programming languages used in the project are Java (62.0%), TypeScript (28.2%), HTML (5.3%), Less (3.8%), JavaScript (0.6%), and Shell (0.1%). Among all of them, there are 834 Java classes, which take up to 37399 lines of code.

# Deployment

## Run the Server

1. Install Maben
2. In the terminal enter the server package: cd chat2db-server
3. Use Maven to clean and install the project: mvn clean install
4. Enter the application directory: cd chat2db-server-start/target/
5. Run the application with APIkey argument: java -jar -Dloader.path=./lib -Dchatgpt.apiKey=xxxxx chat2db-server-start.jar
6. You can also run the server by launching the Spring boot Application, which is auto configured by IDE.

## Run the Client:

1. In the terminal enter the client package: cd chat2db-client
2. Install Node.js (including npm)
3. Use npm to install Yarn: npm install -g yarn
4. Use Yarn to resolve dependencies and download packages: yarn
5. Run the client: yarn run start:web

# Testcases:

Most test cases exist in the chat2db-server-start module and chat2db-server-start module. Among them, the Junit framework and Spring Boot test framework are used. As an abstract class, BaseTest.java has used the “@SpringBootTest” annotation from the Spring Boot test framework, and some other classes extend the BaseTest class and also use the “@Test” annotation from the Junit framework.

To test the whole project, since the project uses Maven for management, we can either use “mvn test” command in the terminal or run the “test” lifecycle in our IDE. In order to run a specific test case, we can either use the “mvn -Dtest=ClassName test” command in the terminal or simply run or debug the class in IDE.

# Testing