* Assume you are a python backend developer, you received an assignment to create chat-bot to provide IT services, like tickets resolution, changing the assigned department and provide the data for given quires by the user. From now on I'll guide you to create a project.  
  Technoligies used: Django, Drf and google generative ai for converting NLP to SQL.
* create a python virtual environment using venv.
* use the env\_name as venv
* Add django, drf and google-generativeai and other dependencies.
* Install the libraries listed in requirements.txt file
* start the development server to test everything is working fine.
* Provide a gitignore file for the python application.
* please add as\_view argument for post call to the respective method in ChatbotViewSet
* please update Ticket status to Enum values
* please update the user\_id in UserQuery model to option field
* Can you run the migration commands to generate migration scripts and tables.
* Can you start the development sever
* I can't find the models in db
* remove all the methods from the ChatbotViewSet and add a query\_data method with post method
* add controller folder in chatbot app create file chatbot.py
* pass the list to generate\_content method with prompt and nlp\_query. prompt should be like """You are an expert in converting English questions to SQl query! The SQL database has the name STUDENT and has the following columns -NAME, CLASS, SECTION and marks In\nFor example, InExample 1 - How many entries of records are present?, the sQL command will be something like this SELECT COUNT(\*) FROM STUDENT; InExample 2 - Tell me all the students studying in Data Science class?, the SQL command will be something like this SELECT \* FROM STUDENT where CLASS="Data Science"; also the sql code should not have ··· in beginning or end""" replace the table schemas and example questions
* please include all the models that we have in chatbot application, excluding userquery
* load the dotenv and get the GOOGLE\_API\_KEY from .env
* create .env file and add the respective key init
* call the controller function in query\_data post view
* call the service method for converting nlp to sql in controller method
* create a repository method to execute the sql query on the db
* call the sql executor function and return the results to view function
* Start the development server
* can you update the table names to all lower case and remove the app name from prefix
* generate migration scripts and apply the migrations to update the table names
* @workspace update prompt the table names to match with table name in models.py
* please add created\_at and updated\_at in all the models
* run makemigrations and migrate scripts
* write a function to create a userquery record when user is requested something
* call the function in controller to audit the user requests