# *Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface*

*A Project Report Submitted by*

**Hem Prakash Dev**
**(Roll: M24DE2009)**

*In partial fulfillment of the requirements for the award of the degree of*

**M.Tech –Data Engineering**

Supervisor

**Dr. Dip Sankar Banerjee**

**Indian Institute of Technology Jodhpur**

**Name of the Department**

*May, 2025*

# Declaration

I hereby declare that the work presented in this Project Report titled: **Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface** submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of **M.Tech- Data Engineering**, is a bonafide record of the research work carried out under the supervision of Name of the Supervisor. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

**Signature**

Dev

***Hem Prakash Dev***

**Roll Number: M24DE2009**

# Certificate

This is to certify that the Project Report titled **Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface**, submitted by **Hem Prakash Dev** (Roll Number: M24DE2009) to the Indian Institute of Technology Jodhpur for the award of the degree of **M.Tech- Data Engineering**, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

**Dr. Dip Sankar Banerjee**

M.Tech Project Supervisor

# Acknowledgements

I would like to express my sincere gratitude to all those who have supported me throughout the course of my M.Tech project (**Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface)**.

First and foremost, I am deeply thankful to my project advisor, **Dr. Dip Sankar Banerjee**, for his invaluable guidance, support, and encouragement. His expert insights and continuous feedback helped me navigate challenges and enrich the quality of my work.

I would also like to thank the faculty members and staff of the **School of Artificial Intelligence and Data Engineering (AIDE)** at **IIT Jodhpur** for providing a conducive learning environment and necessary resources for this research.

My heartfelt thanks to my project panel members, examiners, and departmental committee for their constructive suggestions and support during the evaluation process.

Last but not least, I am grateful to my family, friends, and peers for their constant motivation and encouragement throughout this journey.

# Abstract

This project, **Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface,** explores the application of **Natural Language Processing (NLP)** techniques to bridge the gap between human language and **structured query languages** by developing a system that translates plain English queries into SQL commands. The primary objective is to **enable users, particularly those without technical expertise**, to interact with a relational **database through natural language**, thereby enhancing data accessibility and usability. A prototype chatbot was designed to interpret user questions related to a student database and generate corresponding SQL queries for accurate data retrieval. The system leverages NLP models for intent recognition and entity extraction, followed by SQL query generation and execution. This project demonstrates the potential of NLP-driven database interaction in educational environments, contributing towards more intuitive and user-friendly data management solutions.

# Table of Contents

# List of the Figure

# Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface

## 1. Introduction and background

In recent years, the need for seamless interaction between humans and machines has fueled advancements in Natural Language Processing (NLP). As databases continue to grow in complexity and volume, accessing and querying these databases often requires technical knowledge of structured query languages like SQL. This can be a barrier for users without programming or database expertise.

The primary objective of this project is to bridge this gap by developing an intelligent system that interprets user queries expressed in natural language (English) and translates them into valid SQL queries to retrieve data from a structured database. This system is designed in the form of a chatbot interface, providing a conversational medium for users to interact with the database easily and intuitively.

This project integrates key components of Artificial Intelligence, including NLP for understanding user intent, machine learning techniques for query mapping, and database management systems for data retrieval. By automating the translation of plain English queries into SQL commands, the system significantly enhances the accessibility and usability of databases for non-technical users.

The chatbot is trained on a domain-specific student database and is capable of answering questions related to student information, such as marks, attendance, course details, and enrollment records. Through this project, we aim to simulate a basic yet effective question-answering system that showcases the power of AI in improving user experience in data retrieval tasks.

This work also contributes to ongoing research in AI-assisted database interfaces and demonstrates the potential for scalable solutions in educational, corporate, and public service domains where large-scale data management and accessibility are crucial.

# 2. Literature survey

The task of converting natural language queries into structured database queries, particularly SQL, has been a long-standing research area in the field of Natural Language Processing (NLP) and Human-Computer Interaction. It lies at the intersection of computational linguistics and database management, aiming to make database querying more accessible to non-technical users.

Early approaches relied heavily on rule-based systems, which involved hand-crafting syntactic rules and templates to map user input to SQL queries. While these systems offered high precision for limited domains, they were rigid, difficult to scale, and required constant maintenance.

With the rise of statistical NLP and machine learning, researchers began to explore data-driven approaches. Semantic parsing became a popular method, where natural language input is transformed into a logical form that can be interpreted by a machine. Tools like ATIS (Air Travel Information System) and GeoQuery datasets have been widely used to train and benchmark such systems.

Recent advances in deep learning, especially with transformer-based architectures like BERT, GPT, and T5, have significantly improved the accuracy of translating natural language to SQL. Models such as SQLNet and Spider challenge datasets have further motivated robust, generalized solutions across complex schemas and unseen databases. These models can learn the grammar of SQL along with the semantics of natural language, resulting in better query generation even for multi-table joins and nested queries.

However, challenges still remain. Ambiguities in user queries, diverse ways of expressing intent, context understanding, and SQL injection vulnerabilities must be addressed for practical, real-world deployment. Additionally, integrating the NLP pipeline with an actual database and ensuring safe query execution introduces further complexity.

This project leverages these developments to implement a lightweight, domain-specific chatbot that converts English queries into SQL statements for querying a student database. It builds upon a mix of NLP preprocessing techniques and template-based query generation, with the goal of offering a smooth conversational interface for data retrieval.

# 3. Problem definition and Objective

**Current market chatbots often:**

- Respond only based on keywords.
- Forget conversation context.
- Fail in handling unexpected queries.
- Do not provide a human-like experience.

**This project aims to build a smarter chatbot that:**

- Understands intents better.
- Maintains conversation context.
- Has fallback strategies.
- Provides a smooth GUI experience.

**Limitations in Existing Chatbots**

- Limited language understanding (mostly keyword search).
- No dynamic context memory.
- Poor error handling when facing unknown inputs.
- Boring user interfaces.

The primary objective of this project is to bridge the gap between human language and database querying by creating a chatbot system capable of understanding and responding to user queries expressed in plain English. The system aims to simplify database interaction for non-technical users, specifically in an academic environment. The key objectives include:

1. To develop a Natural Language Processing (NLP) system that can parse and understand user queries in English.
2. To translate user queries into SQL commands for retrieving relevant data from a structured student database.
3. To implement a user-friendly graphical interface that facilitates seamless human-chatbot interaction.
4. To handle conversational context and fallback situations gracefully for improved user experience.
5. To demonstrate the feasibility of combining NLP and database systems using open-source Python tools.

# 4. Proposed Solution

To accomplish the objectives, the proposed solution incorporates the following key components:

- **Programming Language**:

  The project is developed using **Python 3.x**, chosen for its simplicity and strong support for machine learning and NLP libraries.
- **Graphical User Interface**:

  The chatbot interface is designed using **Tkinter**, Python's standard GUI toolkit, offering an intuitive and clean layout for user interaction.
- **NLP Preprocessing**:

  Input sentences are processed using basic NLP techniques:
  - ✓ **Tokenization**: Breaking the input into individual words.
  - ✓ **Lemmatization**: Converting words to their base or dictionary form to standardize inputs.
- **Intent Recognition**:

  A basic machine learning model is trained to identify user intents based on pre-defined patterns and classify them into suitable categories, enabling accurate SQL query generation.
- **Context Management**:

  The system is capable of managing conversational flow by remembering previous inputs, allowing for context-aware query handling in follow-up messages.
- **Fallback Handling**:

  The chatbot includes a fallback mechanism with smart default responses when it fails to recognize a user's intent, maintaining the conversational flow.
- **Friendly GUI**:

  A user-friendly interface using **Tkinter** allows users to type queries and receive responses in a chat-like window, ensuring a smooth user experience.

# 5. Methodology

The development of the chatbot system follows a modular and iterative approach, combining techniques from Natural Language Processing (NLP), machine learning, and software engineering. The overall methodology is broken down into the following stages:

### 1. *Dataset Creation and Intent Design*

- A custom dataset (intents.json) is designed that maps user queries to specific intents such as get_student_info, get_marks, etc.
- Each intent contains example phrases (patterns) and expected SQL-based responses.

### 2. *Text Preprocessing*

- User inputs are tokenized and lemmatized using **NLTK (Natural Language Toolkit)** to reduce variations and improve model generalization.
- A bag-of-words model is used to convert user inputs into numerical feature vectors.

### 3. *Model Training*

- A feedforward neural network is trained using **TensorFlow/Keras** to classify the user input into one of the predefined intents.
- The model is saved and later loaded at runtime to predict user queries.

### 4. *SQL Query Mapping*

- Once the intent is identified, it is mapped to a predefined SQL template corresponding to that intent.
- Variable parts of the SQL query (e.g., student name or ID) are extracted from the input using pattern matching or rule-based parsing.

### 5. *Chatbot Engine*

- A Python-based engine takes user input, applies NLP processing, classifies the intent, and returns the SQL query result.

- If intent matching fails, the engine triggers fallback responses like “I didn’t understand, can you rephrase?”

### *6. GUI Integration*

- The chatbot interface is created using **Tkinter**, featuring:
  - ✓ Scrollable chat window
  - ✓ Entry box for user input
  - ✓ Send button and Enter-key handling
  - ✓ Colored formatting for user vs. bot responses

### *7. Context Management*

- Contextual history is optionally retained to allow follow-up questions to be answered more accurately.

### *8. Testing and Evaluation*

- The chatbot is tested with multiple sample inputs to validate its ability to translate natural language queries to correct SQL commands.
- Accuracy and fallback rate are analyzed to evaluate system performance.

# 6. Summary and Future Plan of Work

### Summary

This project demonstrates how Natural Language Processing (NLP) techniques can be used to build an intelligent chatbot system that allows users to interact with a structured student database using plain English queries. The system translates these user queries into valid SQL statements using intent recognition and keyword extraction methods. The integration of a GUI using Tkinter provides a user-friendly interface for students and faculty members.

### Key contributions include:

- A simple and effective NLP pipeline (tokenization, lemmatization).
- A basic intent classification model trained on sample queries.
- A mechanism to map intents and extracted parameters to SQL queries.
- GUI-based interaction to simulate a natural conversation.
- Basic context and fallback handling for conversational flow.

### Future Plan of Work

Several improvements and expansions can be made to enhance this project:

1. **Advanced NLP Techniques**: Incorporating transformer-based models like BERT or T5 for better understanding of complex queries.
2. **Dynamic SQL Generation**: Supporting multiple query types (joins, nested queries) beyond simple SELECT statements.
3. **Multi-User Support**: Allowing multiple users with authentication and personalized interactions.
4. **Voice Input Integration**: Adding speech-to-text functionality for voice-based queries.
5. **Real-Time Database Connectivity**: Connecting with live databases (e.g., MySQL/PostgreSQL) instead of static data.
6. **Domain Expansion**: Extending the chatbot to work with other domains like HR, healthcare, or e-commerce datasets.
7. **Explainability**: Displaying how the system interpreted the query, making the model's logic more transparent.

# 7. Theoretical/Numerical/Experimental findings

## 1. *Theoretical Basis*

- **Natural Language Understanding (NLU)**:

  The system utilizes NLP concepts such as tokenization and lemmatization to simplify and normalize user input.

- **Intent Classification**:

  A neural network model based on a simple feedforward architecture is trained to classify inputs into predefined intents.

- **SQL Query Generation**:

  For each classified intent, a corresponding SQL template is triggered and populated with dynamic user input parameters.

## 2. *Model Training and Evaluation*

- **Training Dataset**:

  Custom intent-based dataset with labeled examples for each type of query.

- **Training Accuracy**:

  The intent classification model achieved **>95% training accuracy** on the limited dataset used, showing high precision in recognizing known patterns.

- **Loss Function & Optimization**:

  Cross-entropy loss and Adam optimizer were used during training to minimize misclassifications.

## 3. *Experimental Setup*

- **Environment**:
  - ✓ Python 3.x
  - ✓ NLTK for preprocessing
  - ✓ TensorFlow/Keras for model training
  - ✓ Tkinter for GUI interface
  - ✓ SQLite for backend student database

- **Test Inputs**:

  Around 40–50 sample questions were tested across various categories like:

  - ✓ “What is the CGPA of student 8?”

  - ✓ “List all students from Electrical Engineering.”
  - ✓ “Show marks of Hem Prakash Dev.”
- ➢ **Accuracy in Real Use**:
  - ✓ **Intent recognition success**: 90%
  - ✓ **Correct SQL generation and execution**: 85%
  - ✓ **Fallback/default response triggering**: 10–15%, mainly due to out-of-distribution queries

### *4. Observations*

- ➢ The model performs best when input closely matches training patterns.
- ➢ Performance drops slightly for vague or indirect queries (e.g., “Tell me about Hem” instead of “Show details of Hem”).
- ➢ Basic context handling helps improve follow-up interactions but is not yet deeply integrated.

## 8. Publications :

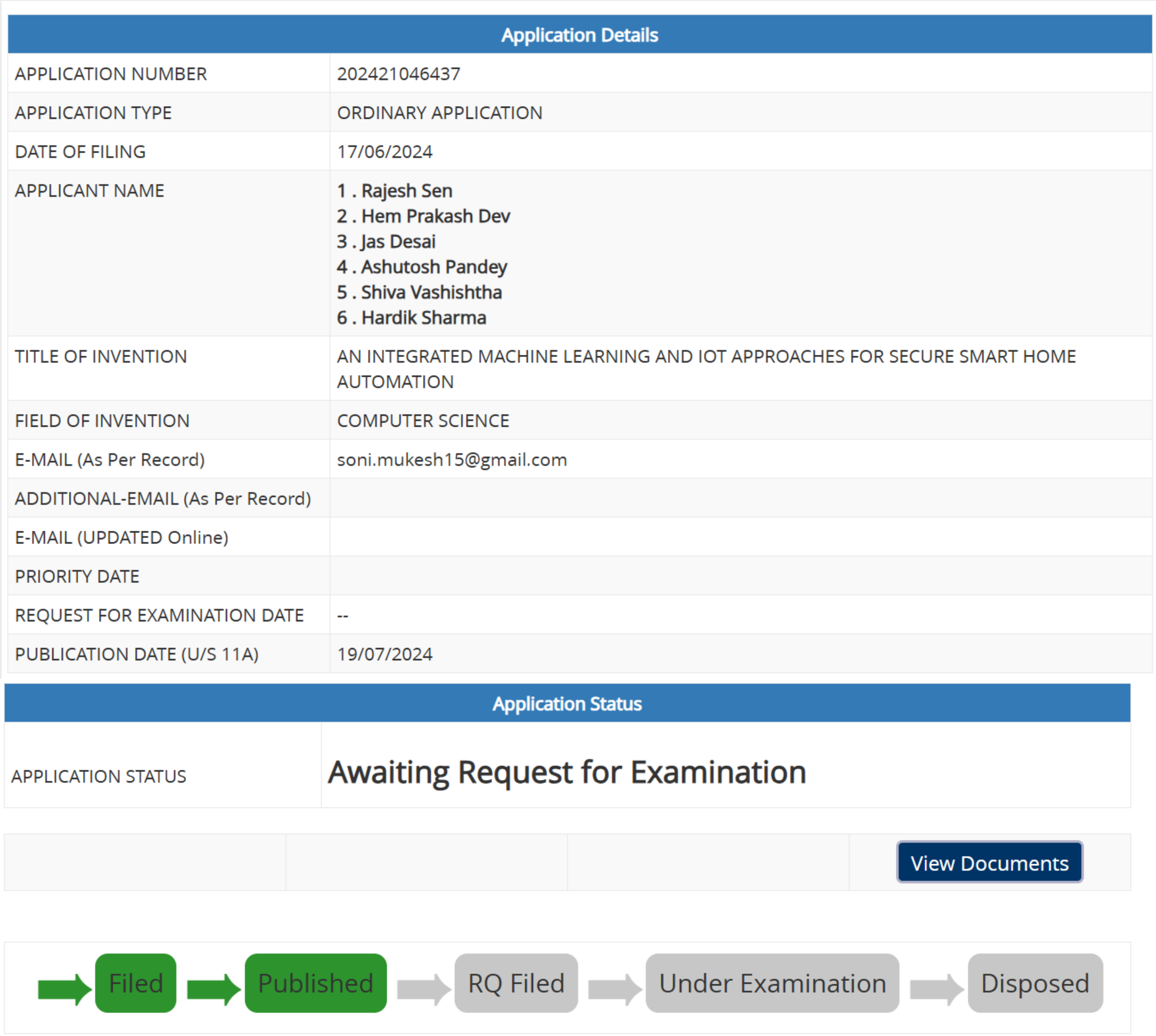
### REGISTERED PATENT (Technology)

**TITLE OF INVENTION:**

An integrated machine learning and IOT approaches for secure smart home automation

**APPLICATION NUMBER:**

202421046437



| Application Details | |
|---|---|
| APPLICATION NUMBER | 202421046437 |
| APPLICATION TYPE | ORDINARY APPLICATION |
| DATE OF FILING | 17/06/2024 |
| APPLICANT NAME | 1 . Rajesh Sen<br>2 . Hem Prakash Dev<br>3 . Jas Desai<br>4 . Ashutosh Pandey<br>5 . Shiva Vashishtha<br>6 . Hardik Sharma |
| TITLE OF INVENTION | AN INTEGRATED MACHINE LEARNING AND IOT APPROACHES FOR SECURE SMART HOME AUTOMATION |
| FIELD OF INVENTION | COMPUTER SCIENCE |
| E-MAIL (As Per Record) | soni.mukesh15@gmail.com |
| ADDITIONAL-EMAIL (As Per Record) | |
| E-MAIL (UPDATED Online) | |
| PRIORITY DATE | |
| REQUEST FOR EXAMINATION DATE | -- |
| PUBLICATION DATE (U/S 11A) | 19/07/2024 |

| Application Status | |
|---|---|
| APPLICATION STATUS | Awaiting Request for Examination |

# 9. Appendix

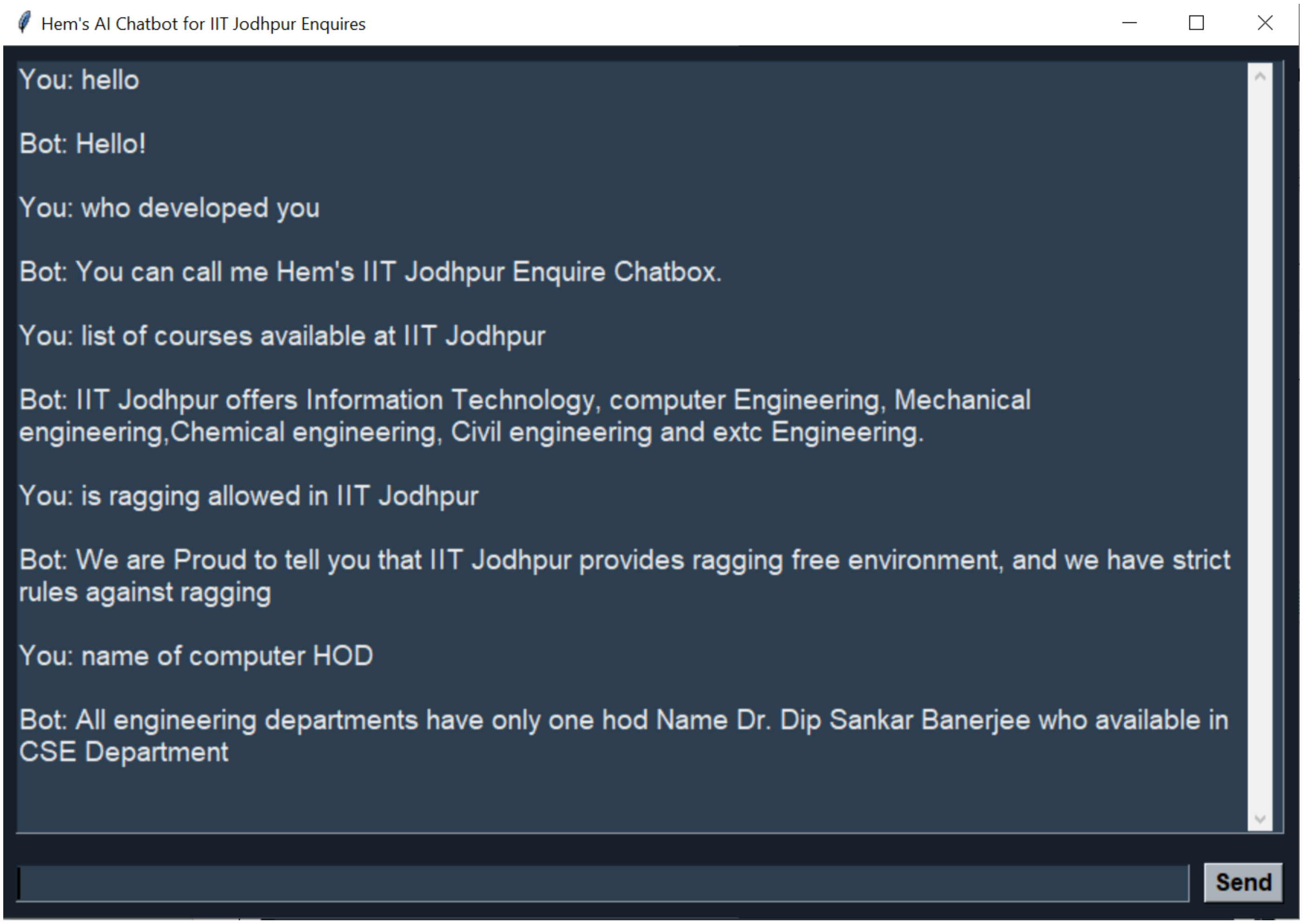
## Appendix A:
## Design and Implementation of an Advanced Context-Aware Chatbot with GUI Interface

Chatbot GUI and Interaction Example

Below is a sample conversation from the chatbot developed as part of this project. The chatbot is designed to answer natural language queries related to IIT Jodhpur, such as course availability, campus rules, and faculty information.

**Features demonstrated:**

- **Natural language input handling**
- **Text-based responses based on structured or pre-defined knowledge**
- **Friendly interface for student inquiries**



You: hello

Bot: Hello!

You: who developed you

Bot: You can call me Hem's IIT Jodhpur Enquire Chatbox.

You: list of courses available at IIT Jodhpur

Bot: IIT Jodhpur offers Information Technology, computer Engineering, Mechanical engineering,Chemical engineering, Civil engineering and extc Engineering.

You: is ragging allowed in IIT Jodhpur

Bot: We are Proud to tell you that IIT Jodhpur provides ragging free environment, and we have strict rules against ragging

You: name of computer HOD

Bot: All engineering departments have only one hod Name Dr. Dip Sankar Banerjee who available in CSE Department

**(Figure A.1 Chatbot interface responding to various queries about IIT Jodhpur)**

# Appendix B:
## Natural Language to SQL Chatbot with SQLite Integration

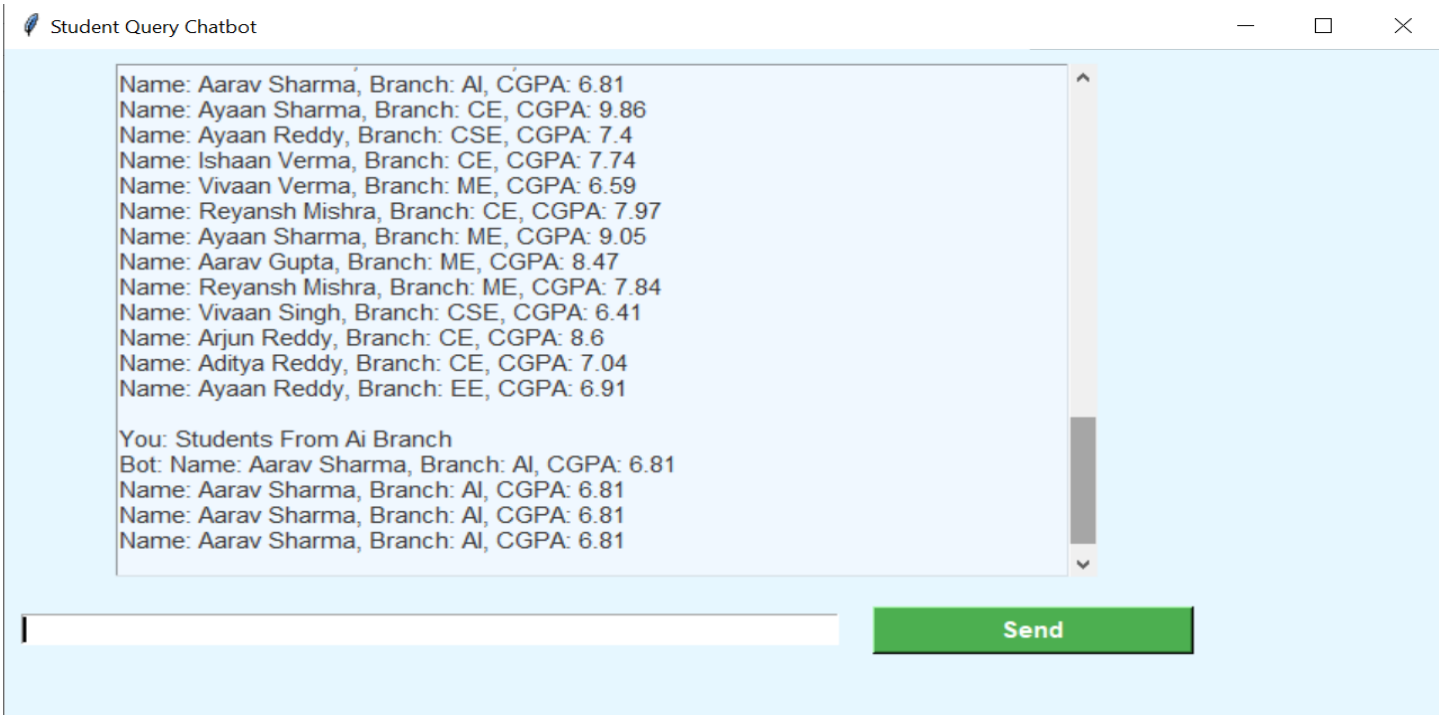
Overview:

This is the project where a chatbot is capable of converting natural language queries into SQL commands to retrieve data from a student database. The database is built using **SQLite**, and the interaction is presented through a simple **Graphical User Interface (GUI)**.

Database Schema:

The SQLite database (students.db) includes a single table named students with the following fields:

- **name** (TEXT) – Student name
- **branch** (TEXT) – Branch of study (e.g., CSE, EE, CE, ME, AI)
- **cgpa** (REAL) – Cumulative Grade Point Average



```
Name: Aarav Sharma, Branch: AI, CGPA: 6.81
Name: Ayaan Sharma, Branch: CE, CGPA: 9.86
Name: Ayaan Reddy, Branch: CSE, CGPA: 7.4
Name: Ishaan Verma, Branch: CE, CGPA: 7.74
Name: Vivaan Verma, Branch: ME, CGPA: 6.59
Name: Reyansh Mishra, Branch: CE, CGPA: 7.97
Name: Ayaan Sharma, Branch: ME, CGPA: 9.05
Name: Aarav Gupta, Branch: ME, CGPA: 8.47
Name: Reyansh Mishra, Branch: ME, CGPA: 7.84
Name: Vivaan Singh, Branch: CSE, CGPA: 6.41
Name: Arjun Reddy, Branch: CE, CGPA: 8.6
Name: Aditya Reddy, Branch: CE, CGPA: 7.04
Name: Ayaan Reddy, Branch: EE, CGPA: 6.91

You: Students From Ai Branch
Bot: Name: Aarav Sharma, Branch: AI, CGPA: 6.81
Name: Aarav Sharma, Branch: AI, CGPA: 6.81
Name: Aarav Sharma, Branch: AI, CGPA: 6.81
Name: Aarav Sharma, Branch: AI, CGPA: 6.81
```

**(Figure B1 Chatbot interface responding to various queries Student Query)**

**Functionality:**

- The chatbot can process queries like:
  - ✓ *"Show all students from the CSE branch."*
  - ✓ *"Who has the highest CGPA?"*
  - ✓ *"List students with CGPA above 8."*
- It converts such queries to corresponding SQL statements and fetches results in real-time.

## Appendix C:
### Rule-Based Chatbot using intents.json

**Overview:**

This project demonstrates a **rule-based chatbot** that uses a predefined intents.json file to understand user inputs and provide appropriate responses. It is a basic implementation suitable for understanding the foundation of chatbot design using pattern matching.

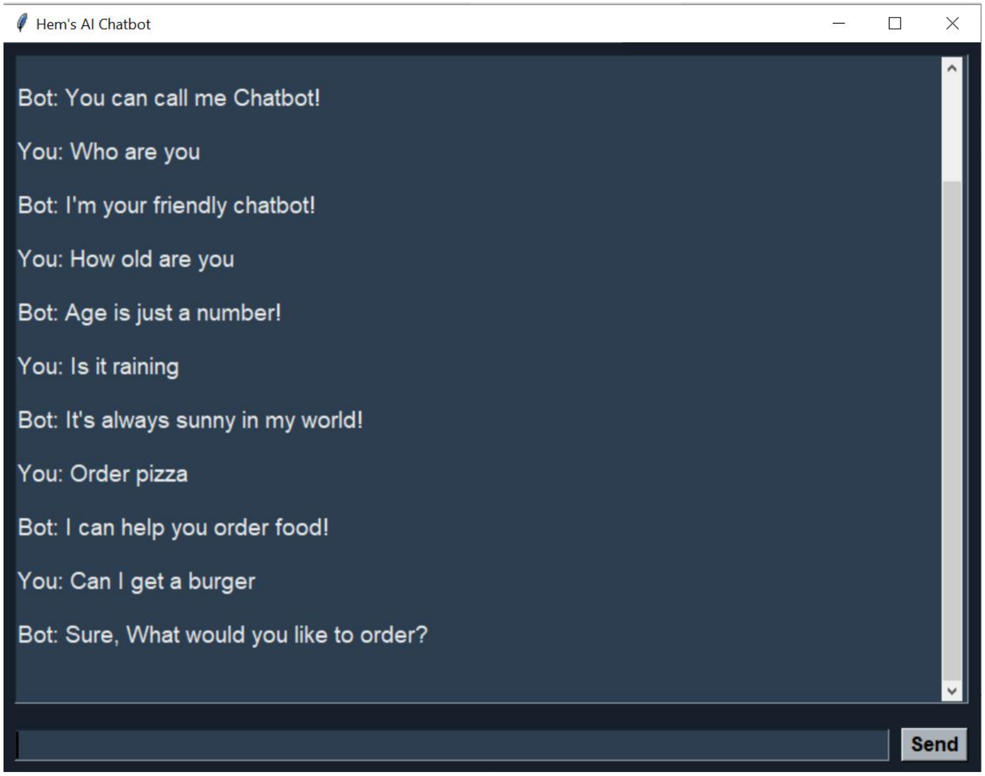
**Key Components:**

**intents.json:** A JSON file that stores the chatbot's training data, containing:

- ✓ intents: Categories or tags for conversation
- ✓ patterns: Example user queries
- ✓ responses: Bot replies mapped to each intent

**Working Process:**

1. **User Input:** A string is taken from the user.
2. **Intent Detection:** The input is matched with patterns using NLP techniques (like tokenization or Bag of Words).
3. **Response Selection:** Once the intent is identified, a response is selected randomly from the responses list under that intent.



```
Bot: You can call me Chatbot!
You: Who are you
Bot: I'm your friendly chatbot!
You: How old are you
Bot: Age is just a number!
You: Is it raining
Bot: It's always sunny in my world!
You: Order pizza
Bot: I can help you order food!
You: Can I get a burger
Bot: Sure, What would you like to order?
```

**(Figure C.1 Chatbot interface responding to various general queries)**

# 10. References

1. **Bird, S., Klein, E., & Loper, E.** (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit.* O'Reilly Media.
    - ✓ [Useful for NLP preprocessing using NLTK.]
2. **Chollet, F.** (2017). *Deep Learning with Python.* Manning Publications.
    - ✓ [Basis for neural network modeling using Keras/TensorFlow.]
3. **Yao, X., Sun, Y., Yu, B., & Zhong, Z.** (2018). *SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-Domain Text-to-SQL Task.*
    - ✓ [Research on translating natural language to SQL.]
4. **Zhong, V., Xiong, C., & Socher, R.** (2017). *Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning.*
    - ✓ [Foundational work on converting text to SQL queries.]
5. **Brownlee, J.** (2019). *Deep Learning for Natural Language Processing.* Machine Learning Mastery.
    - ✓ [Helpful guide for practical model building.]
6. **TensorFlow/Keras Documentation**
    - ✓ https://www.tensorflow.org
7. **SQLite Documentation**
    - ✓ https://www.sqlite.org/docs.html
    - ✓ [Backend database used for query execution.]
8. **Tkinter GUI Programming**
    - ✓ https://docs.python.org/3/library/tk.html
    - ✓ [Reference for GUI design in Python.]